

Robot-Assisted Full Automation Interface: Touch-Response On Zebrafish Larvae

Zur Erlangung des akademischen Grades eines

DOKTORS DER INGENIEURWISSENSCHAFTEN (Dr.-Ing.)

von der KIT-Fakultät für
Maschinenbau
des Karlsruher Instituts für Technologie (KIT)

angenommene

DISSERTATION

von

M.Sc. Yanke Wang
geb. in Henan, China

Tag der mündlichen Prüfung:

Hauptreferent:

Korreferent:

Korreferent:

28.07.2023

apl. Prof. Dr. Markus Reischl

Prof. Dr. Veit Hagenmeyer

Prof. Dr. Nicholas S. Foulkes

I declare that I have developed and written the enclosed paper completely by myself, and have not used sources or means without declaration in the text.

Karlsruhe, 24.05.2023

.....
(Yanke Wang)

Kurzfassung

Automatisierte Labore zeichnen sich durch ihre Komplexität und Bedeutung für die Lebenswissenschaften aus. Sie folgen strengen Protokollen, und viele Parameter können die Ausgaben beeinflussen. Traditionell werden chemische oder biologische Experimente zu Großteilen manuell durchgeführt, was die Arbeit im Labor zeitaufwändig, mühsam, und ineffizient macht. Allerdings ist die menschliche Teilnahme in manchen speziellen und komplexen Prozessen nicht erwünscht, um subjektive Ergebnisse zu verhindern. Robotik und künstliche Intelligenz können in solchen Prozessen angewandt werden, um manuelle Arbeitsabläufe zu minimieren und Durchsatz und die Zuverlässigkeit der experimentellen Ergebnisse zu optimieren.

Die vorliegende Arbeit zielt darauf ab, ein Konzept zur Umsetzung einer vollautomatischen Schnittstelle zu geben. Dazu zählt, Daten automatisch zu erfassen, zu bewerten und zu analysieren unter zu Hilfenahme von Methoden der automatisierten Bildverarbeitung, des maschinellen Lernens und der Statistik. Diese Themenbereiche werden mit Blick auf die Umsetzung in automatisierten Laboren untersucht. Am Beispiel der Wirkstoffanalyse durch Touch-Response-Screenings am Zebrafisch werden vier Systeme entwickelt, um die vollautomatische Schnittstelle im Touch-Response-Screening zu implementieren: eine automatisierte Datenerfassungsplattform für Einzellarven, für mehrere Larven, eine automatisierte Inspektionspipeline für gesammelte Daten, und eine Pipeline für die automatisierte Musteranalyse der Inspektionsergebnisse. Es werden angepasste Bewertungsmetriken definiert, um jedes vorgeschlagene System zu bewerten und nach wünschenswerten Parametern für die Systeme zu suchen. Verifizierungsexperimente zeigen, dass die vorgeschlagenen Systeme automatisch das Touch-Response Verhalten von Zebrafischlarven untersuchen und die erwarteten Ergebnisse erzeugen können.

Abstract

The automated biological labs stand out because of the complexities and meanings in the life science. The biological labs require a strict protocol of the experiment, and many parameters can influence the readouts. Traditionally, the biological experiment is still finished manually, making the lab work time-consuming, painful, and inefficient. However, human intervention is not desired, or even not allowed in some special and complex processes, to prevent subjective results. Robotics and artificial intelligence can be applied to such processes, releasing the manual workload and optimizing the throughput and reliability of the experimental readouts.

The present work aims to provide a common guidance to fulfil the fully automated interface, including automatically acquiring, inspecting, and analyzing data, which can be achieved by the modern advances, e.g., in the fields of computer vision, machine learning, and statistics. This research is applied to a specific experiment protocol: touch-response screening on zebrafish model. Four systems are proposed to implement the fully automated interface in touch-response screening, including the automated data acquisition platforms for the single-larva and multi-larva cases, respectively, the automated inspection pipeline for the collected data, and the pattern analysis pipeline of the inspection results for the drug screening task. In addition, the customized evaluation metrics are defined to evaluate each proposed system and to search for desirable parameters related to the systems. The verification experiments are conducted to prove that the proposed systems can automatically screen the touch-response behavior of zebrafish larvae and generate the expected results.

Danksagung

Es war eine Herausforderung und eine unvergessliche Erfahrung, in Deutschland forschen und leben zu dürfen. Ich habe diesen Weg gewählt und bin im Nachhinein sehr glücklich, die Entscheidung so getroffen zu haben. Die Herausforderungen auf diesem Weg habe ich mithilfe meiner Familie, Freunden, Kollegen und meines unterstützenden und zuverlässigen Betreuers sowie der wichtigen Finanzierung durch den China Scholarship Council erfolgreich gemeistert.

In den letzten vier Jahren habe ich meine Forschungsfähigkeiten erheblich verbessert, und insbesondere die sorgfältige, zuverlässige und hilfreiche Betreuung durch meinen Betreuer Prof. Markus Reischl haben zu dem Gelingen der Arbeit beigetragen. Er unterstützte mich nicht nur in der Forschung, sondern auch in meinem Leben in Deutschland. Es war für mich ein großes Glück, einen Betreuer wie ihn in Deutschland zu haben. Insbesondere für die veröffentlichten Artikel und beim Schreiben meiner Doktorarbeit hat er viel mehr getan, als ich von ihm erwartet habe, was mich beeindruckt hat und wofür ich sehr, sehr dankbar bin. Ich werde mein Leben lang von der Kooperation mit ihm profitieren.

Ich möchte mich auch herzlich bei meinen Korreferenten Prof. Hagenmeyer und Prof. Foulkes dafür bedanken, meine Doktorendarbeit mitzubetreuen. Außerdem haben mir auch meine Kollegen im IAI und IBCS am KIT sehr geholfen, nicht nur während der Arbeit, sondern auch bei alltäglichen Problemen.

Zusätzlich möchte ich mich bei meinen Eltern und Freunden bedanken. Ich war seit der schwierigen Zeit ab Februar 2020 jahrelang nicht mehr bei meiner Familie, aber sie unterstützte mich unentwegt aus der Ferne. Mein Freund, meine Freunde und meine Freundinnen sind inzwischen so etwas wie meine Familie in Deutschland, und ich schätze mich sehr glücklich, eine solche Familie und solche Freunde zu haben.

Karlsruhe, den 22. Feb, 2023,

Yanke Wang.

Contents

Kurzfassung	iii
Abstract	v
Danksagung	vii
Acronyms and symbols	xiii
1 Introduction	1
1.1 Background	1
1.2 Full automation interface	2
1.2.1 Overview	2
1.2.2 Robot-assisted systems	3
1.2.3 Components of full automation interface	5
1.3 Automation in laboratories	10
1.3.1 Overview	10
1.3.2 Related automated laboratories	11
1.3.3 Zebrafish experimentation	13
1.3.4 Touch response of zebrafish larvae	18
1.3.5 Open questions	20
1.4 Goals and main contributions	22
1.5 Organization of the thesis	23
2 Conceptualization and parameters	25
2.1 General process concept	25
2.1.1 Process design	25
2.1.2 Evaluation and optimization	28
2.2 Touch-screening process	29
2.2.1 Conceptualization	29
2.2.2 Touch screening on zebrafish larvae	31
2.3 Full automation interface	35
2.3.1 Requirements	35
2.3.2 Sensor	36

2.3.3	Sample holder	37
2.3.4	Manipulator	40
2.3.5	Software package	41
2.3.6	Data processing methods	43
2.3.7	Touching strategy	45
2.4	Evaluation metrics	47
2.4.1	Overview	47
2.4.2	Evaluation of image processing	47
2.4.3	Evaluation of touching task	52
2.4.4	Evaluation of inspection	53
2.4.5	Evaluation of pattern recognition	53
3	Data acquisition for single larvae	55
3.1	Overview	55
3.2	System design	57
3.2.1	Actuator system	57
3.2.2	Coordinate system	58
3.2.3	Image processing and touching trajectory	60
3.2.4	Graphical User Interface	63
3.2.5	Touching strategy	64
3.3	Evaluation of SiLTAP	65
3.3.1	Experiment definition and parameter setting	65
3.3.2	Evaluation of image processing and hardware error	68
3.3.3	Evaluation on the touching tasks	70
3.3.4	Discussions	71
4	Data acquisition for multiple larvae	73
4.1	Overview	73
4.2	Multi-larva touch-response data acquisition platform	75
4.2.1	Reusable shadow-covering rings	75
4.2.2	Image processing	76
4.2.3	Touching strategy	77
4.3	Evaluation of MuLTAP	79
4.3.1	Experiment definition and parameter setting	79
4.3.2	Selection of segmentation methods	80
4.3.3	Parameter determination	83
4.3.4	Evaluation and comparison of SiLTAP and MuLTAP	89
4.3.5	Discussions	92
5	Touch-response behavior inspection	95
5.1	Task description and challenges	95

5.2	Multi-larva touch-response inspection pipeline	96
5.2.1	Overview	96
5.2.2	Tracking and segmentation	97
5.2.3	Quantification	99
5.3	Quantification experiment and evaluation of AMTIP	101
5.3.1	Parameter setting of AMTIP	101
5.3.2	Quantification experiment definition and parameter setting	102
5.3.3	Results of zebrafish larvae's touch response	107
5.3.4	Evaluation of AMTIP	109
5.3.5	Discussions	111
6	Touch-response pattern recognition	113
6.1	Task description and challenges	113
6.2	AI-based drug effect pattern database	115
6.2.1	Overview	115
6.2.2	Feature engineering	115
6.2.3	Hierarchical clustering and pattern mapping	117
6.2.4	Pattern proposing	117
6.3	Drug screening experiment to evaluate PRTD	118
6.3.1	Experiment setup	118
6.3.2	Evaluation results	121
6.3.3	Discussions	123
7	Conclusions	127
A	Appendix	131
A.1	Pipeline of image processing in multi-larva case	131
A.2	Training of U-Net	132
A.3	Methods in AMTIP	135
A.3.1	Optical flow for needle tracking	135
A.3.2	Particle filter based larva tracking	135
A.3.3	Region growing based larva segmentation	136
A.4	Iterative curve fitting for the C-Bend analysis	137
	List of Figures	139
	List of Tables	147
	List of Publications	149
	Journal articles	149
	Conference contributions	149

Bibliography **151**

Acronyms and symbols

Acronyms

IAI	Institute for Automation and Applied Informatics
KIT	Karlsruhe Institute of Technology
AI	Artificial Intelligence
SiLTAP	Single-Larva Touch-response data Acquisition Platform
MuLTAP	Multi-Larvae Touch-response data Acquisition Platform
AMTIP	AI-based Multi-larvae Touch-response behavior Inspection Pipeline
PRTD	Pattern Recognition of Touch response based Database
hpf	hours post-fertilization
dpf	days post-fertilization
Approx.	Approximately
PTFE	Polytetrafluoroethylene
GUI	Graphical User Interface
CI	Capture Image
IP	Image Processing
MLT	Move the needle to the Larva based on Trajectory
CV	Capture Video
SV	Save Video
MNW	Move the Needle to the next Well
fps	frames per second

Thre	Threshold based binarization
LRb	Linear Regression based binarization
RGb	Region Growing based binarization
RT	RoTation
CB	Control and Brightness
GN	Gaussian Noise
Wild	Larvae in fish water
DMSO	Larvae in Dimethyl sulfoxide
Dia	Larvae treated by Diazepam
Caffi	Larvae treated by Caffeine
C1, C2, ..., C76	Compound 1, Compound 2, ..., Compound 76

Experiments and datasets

E_{ip}	Experiment on evaluation of image processing
E_{he}	Experiment on hardware error
E_{tt}	Experiment on touching tasks
E_{seg}	Experiment on the selection of segmentation methods
E_{c1}	Experiment on the capacity of well plate
E_{c2}	Experiment on the verification of the capacity of well plate
E_r	Experiment on the effectiveness of PTFE shadow-covering rings
E_{rh}	Experiment on the height of the shadow-covering rings
E_a	Experiment on the ages of the larvae
E_d	Experiment on the effect of dechoriation
E_{tm}	Experiment on the touching task of MuLTAP
E_t	Experiment on the touching strategy

E_{vpw}	Experiment on the average number of videos collected per well
E_p	Experiment on touching positions
E_{db}	Experiment on ages
E_{st}	Experiment on short-term treatment
E_{lt}	Experiment on long-term treatment
E_{pr}	Experiment on pattern recognition of drug effects
DA – Eip	Dataset collected in E_{ip}
DA – Ehe	Dataset collected in E_{he}
DA – Ett	Dataset collected in E_{tt}
DA – Ec1	Dataset collected in E_{c1}
DA – Ec2	Dataset collected in E_{c2}
DA – Er	Dataset collected in E_r
DA – Erh	Dataset collected in E_{rh}
DA – Ea	Dataset collected in E_a
DA – Ed	Dataset collected in E_d
DA – Etm	Dataset collected in E_{tm}
DA – Et	Dataset collected in E_t
DA – Evpw	Dataset collected in E_{vpw}
DA – Ep	Dataset collected in E_p
DA – Edb	Dataset collected in E_{db}
DA – Est	Dataset collected in E_{st}
DA – Elt	Dataset collected in E_{lt}
DA – Epr	Dataset collected in E_{pr}
DA – Epr – TRAIN	Training dataset of DA – Epr
DA – Epr – TEST	Testing dataset of DA – Epr

DA – Seg – Train	Training dataset for comparing the segmentation methods
DA – Seg – Test	Testing dataset for comparing the segmentation methods

Parameters

H_c	Capacity of the sample holder
S_t	Touching force (speed)
P	Touching position
P_h	Head touching position
P_b	Body touching position
P_t	Tail touching position
θ_t	Touching angle
A_t	Age of experiment sample
D_e	Effects of dechoriation
t_1	The time at which needle stops and touch is applied
t_2	The time at which response begins
t_3	The time when the larva has the response amplitude peak
t_4	The time at which response stops
t_l	Latency time
r_a	C-Bend radius average
c_m	C-Bend curvature maximum
t_{cp}	C-Bend peak time
t_r	Response time
d_m	Moving distance
H_{od}	Outer Diameter of the sample holder
H_{id}	Inner Diameter of the sample holder

H_h	Height of the sample holder
r_{od}	Outer Diameter of shadow-covering rings
r_{id}	Inner Diameter of shadow-covering rings
r_h	Height of shadow-covering rings
\mathbb{W}	World coordinate system
\mathbb{N}	Needle coordinate system
\mathbb{I}	Image coordinate system
\mathbb{L}	Larva coordinate system
\mathbb{P}	Percentage coordinate system
w_*	The coordinates in \mathbb{W}
n_*	The coordinates in \mathbb{N}
i_*	The coordinates in \mathbb{I}
l_*	The coordinates in \mathbb{L}
p_*	The coordinates in \mathbb{P}
d_{niz}	Distance between \mathbb{N} and \mathbb{I}
s_{ni}	Scale factor between \mathbb{N} and \mathbb{I}
$\{d_{nix}, d_{niy}, d_{niz}\}$	Offsets between \mathbb{N} and \mathbb{I}
$\{d_{wnx}, d_{wny}, d_{wnz}\}$	Offsets between \mathbb{W} and \mathbb{N}
T_I^N	Transformation matrix from \mathbb{I} to \mathbb{N}
T_N^W	Transformation matrix from \mathbb{N} to \mathbb{W}
T_I^W	Transformation matrix from \mathbb{I} to \mathbb{W}
T_L^I	Transformation matrix from \mathbb{L} to \mathbb{I}
l_g	The coordinate of center of gravity in \mathbb{L}
l_h	The coordinate of head touching position (P_h) in \mathbb{L}
l_b	The coordinate of body touching position (P_b) in \mathbb{L}

l_t	The coordinate of tail touching position (P_t) in \mathbb{L}
ph	The coordinate of head touching position (P_h) in \mathbb{P}
pb	The coordinate of body touching position (P_b) in \mathbb{P}
pt	The coordinate of tail touching position (P_t) in \mathbb{P}
T_{ns}	Threshold for the needle binarization in SiLTAP
T_{ls}	Threshold for the larva binarization in SiLTAP
k_m	Kernel size of dilation and erosion in SiLTAP
T_{ss}	Size threshold in SiLTAP
α	Orientation angle of the larva in \mathbb{I}
P_n	Needle position
\dot{i}_n	The coordinate of needle position (P_n) in \mathbb{I}
P_s	Starting position
\dot{i}_s	The coordinate of starting position (P_s) in \mathbb{I}
d_{st}	Distance between starting position (P_s) and touching position (P)
$\Delta_{n \rightarrow s}$	Displacement from needle position (P_n) to starting position (P_s)
$\Delta_{s \rightarrow t}$	Displacement from starting position (P_s) to touching position (P)
l_n	The perpendicular line to the larva and crossed at the touching position (P) with the larva
c_n	The circle around the touching position (P) with radius being d_{st}
S_{t1}	The speed of $\Delta_{n \rightarrow s}$
S_{t2}	The speed of $\Delta_{s \rightarrow t}$
T_{ms}	Size threshold in MuLTAP
k_{op}	The kernel size of the adjacent area in optical flow of AMTIP
k_{opn}	The kernel size of the adjacent area in optimized optical flow of AMTIP
$\{X_j^n, Y_j^n, t_j\}$	The coordinate of the needle of frame t_j in \mathbb{I}
$\{X_j^l, Y_j^l, t_j\}$	The coordinate of the larva of frame t_j in \mathbb{I}

T_d	Threshold of image difference in particle filter of AMTIP
T_g	Threshold of image gradient in region growing of AMTIP
T_l	Low threshold of image binarization in region growing of AMTIP
T_h	High threshold of image binarization in region growing of AMTIP
T_{qs}	Size threshold in AMTIP
T_{nl}	Threshold for the distance between the larva and needle of the quantification in AMTIP
T_{mq}	Threshold for the movement of the larvae used in the quantification of AMTIP
Q_c	Quantification criteria result for control group generated by AMTIP
Q	Quantification criteria result for experiment group generated by AMTIP
f	Features generated by the L_2 Wasserstein distance method.
T_d	Data pattern dendrogram generated by PRTD
B	Branches of the pattern tree (also the clusters)
H	Hypotheses of pattern
P	Proposal of data patterns
T_{pr}	Threshold for the hierarchical clustering of PRTD
T_{Thre}	Threshold for Thre
T_{LRb}	Threshold for output of LRb
EpO_{LRb}	Training epoch selected for testing LRb
T_{RGb}	Threshold for the image gradient of RGB
T_{UNet}	Threshold for the output of U-Net
$Loss$	Training loss of U-Net
L_1	Cross entropy loss of U-Net
L_2	Dice loss of U-Net
s	Smooth variable for $Loss$

bce	Weight to combine L_1 and L_2
θ	The random rotation angle for data augmentation of DA-Seg-Train
R_θ	The rotation matrix with the random angle θ
β	The contrast variable for data augmentation of DA-Seg-Train
γ	The brightness variable for data augmentation of DA-Seg-Train
μ_x, μ_y	The mean of Gaussian noise for data augmentation of DA-Seg-Train
σ_x^2, σ_y^2	The variance of Gaussian noise for data augmentation of DA-Seg-Train
l_r	Learning rate for the training of U-Net
B_s	Batch size for the training of U-Net
${}_i P_j^l$	The particle i at position $\{{}_i x_j^l, {}_i y_j^l\}$ of frame t_j for larva l
${}_i b_{j+1}^l$	The binary probability at $\{{}_i x_j^l, {}_i y_j^l\}$ of the l th larva in frame t_{j+1} .
${}_i d_{j+1}^l$	Pixel difference at $\{{}_i x_j^l, {}_i y_j^l\}$ between frame t_{j+1} and frame t_j
$p_k^{(*,*)}$	Adjacent point of the region growing in AMTIP
$L \left(p_k^{(*,*)} \right)$	The labeled (segmented) result for the position $p_k^{(*,*)}$
N_p	Number of the particles of the larva tracking in AMTIP
σ_{px}, σ_{py}	Standard deviation of the Gaussian distribution for resampling the particles of the larva tracking in AMTIP
Ed	Euclidean distance

Evaluation metrics

e_{gd}	Center error
R_o	Ratio of overlap
IoU	Intersection over Union
PC	Per-Class accuracy
JI	Jaccard Index

R_r	Ratio of recall
R_p	Ratio of precision
$\#VT$	Number of valid touching
t_{TPL}	Touching efficiency
$\#VPW$	Average number of videos collected per well
e_{tr}	Touching reproducibility error
r_t	Touching robustness
$\#NT$	Number of videos with no larvae touched
$\#QF$	Number of videos with failure of quantification
$\#N_{pp}$	Number of pattern proposals
P_f	Percentage of failure of pattern predictions

1 Introduction

1.1 Background

The world is being robotized at an unprecedented rate owing to the rapid development of robotics and artificial intelligence (AI). Even human-dominated activities are being mechanized [1] in virtual or substantial ways. For example, the program ChatGPT is creating amusing virtual conversations in an inspiring and interactive manner by using AI [2]. In addition, robots are affecting nearly all life-encompassing tasks [3], and automation is becoming reality in supermarkets, check-ins and payment transactions, transportation of deliveries, etc. Along with these developments, the fourth industrial revolution is approaching. Currently, automation technologies and robotic systems are also being applied in complex, dangerous, and challenging scenarios, and this will increase in the future. For instance, in the scientific research areas, start-up firms have stated: "Robotics and software that autonomously record every detail of an experiment can transform the efficiency and reliability of research." as reported by E. Check Hayden [4]. This indicates that a specific experimental protocol can benefit much from robotic technologies through precise control of the experimental conditions. However, work in factories, infrastructures, laboratories, hospitals still requires a great deal of manual input and the intelligent lifestyle is not yet sufficient and still requires long-term exploration.

Robot-assisted automation with no manual input (defined as full automation interface in this thesis) is developed to change the situation above as it has many advantages for work, i.e., no human intervention, high flexibility, inspiration of new output, and for humans, i.e., burden release, safety guarantee, etc. Thus, the focus of current research is progressively attracted by the conceptualization and application of fully automated systems, so that all the sub-parts from the task can be replaced by automated systems and humans can be released from the task. Even though some automated systems have been explored and implemented, the discussion about a general conceptualization of the full automation interface is not sufficient. The thesis aims to design a full automation interface (with the assistance of robots) given a specific process and to provide a template for establishing fully automated processes.

1.2 Full automation interface

1.2.1 Overview

The concept of full automation is defined as no operators', manipulators', or engineers' assistance involved in all operating levels of a process. It contains several sub-processes, each implemented by a separate automation solution, such as a robot or a software. The current research and achievements in industry and academy have moved much forward, such as the development of materials, fabrication, robotics, and intelligent systems. It is a necessity to explore and attempt building full automations and releasing the humans.

The vision of full automation is gradually becoming reality in many busy and demanding processes, as it completes tasks in high efficiency with automated workflows and decision making as well as traceability and identification of products [5]. In various research fields, the concept of full automation has been spreading widely [6]. In a production line, multiple repeatable devices work together to output the desirable products. For example, the concept of full automation can be observed from the fabrication line of lasers, including the design of the laser, the actual fabrication, and the measurement evaluation [7]. The automation in a laboratory is seen from a single device, like the automated cell incubator [8] and cell injection [9]. Some automated laboratories usually include a series of experimental devices as well as the statistical analysis [10]. In the clinical field, for example, a fully automatic pipeline is used in measuring the maximum aortic diameter [11] in the pre-/post-surgical phases. In large hospitals, the full automation requires a great amount of devices, e.g., a robust basophil activation test in a high-throughput and automated manner, including the sample preparation, measurement, and analysis, reducing a large amount of hands-on time [12]. Industry 4.0 has also derived many potential technologies to fulfill fully automated production, but the most of the recent or even future manufacturing processes still require human supervision in the loop of fabrication [13]. Apart from the design of the robot (or device) itself, the development of full automations is also largely limited by the testing and evaluation techniques [14] and analysis technologies. The results of evaluation metrics and data analysis are used to optimize the essential parameters in the designed systems.

Robotic technologies have undertaken the main role of fulfilling full automation interfaces. Thus, the review on the existing robot-assisted systems helps build a general structure (components) together with the involved technologies of implementing full automation interface.

1.2.2 Robot-assisted systems

Robots do not feel exhausted by repeating tasks, and the existence of artificial intelligence makes it possible to complete a task fully automatically with minor or no human intervention. Inspired by the intelligence and accuracy of robots, researchers begin to search for the option of applying robotic technologies in the manufacturing process, assembly pipeline or academic research. Those technologies help in reducing the amount of the work from the side of operators as well as in generating repeatable and reliable outputs. Especially on repeatable tasks, the robots have shown high capability and have been in application to a large extent. Developers have focused on the sub-parts of the task and developed systems in the semi-automated manner with human intervention. The review on robotic technologies is the perspective of industry, scientific experiments, healthcare, and biorobots.

In Industry 4.0, companies and factories are revoluted via the collaboration of the physical, virtual and social worlds [15]. Robots are integrated into repeatable manufacturing or assembly processes to improve efficiency. Common manners can be a robot undertaking a specific part of production tasks in an unsupervised manner or the complex task in a human-robot interactive manner. Without human intervention, the manufacturing process is usually in a fixed program, like the fabrication of predefined and specific patterns (material extrusion) [16]. It works fully automatically, but it is not sufficient to respond to the dynamic changing environment or process owing to the fixed fabrication procedure. The research of the human-robot interaction (HRI) is focused on the safety, usability, performance, speed, user programming interface, intuitive interaction and flexibility [17]. Robots not only work themselves, but work in a collaborative pattern with additional devices (like separate cameras) as well as with human operators to generate more desirable outputs [15]. In human-unfriendly (dirty or dangerous) and complex conditions, robots are considered to improve the working environment of the workers via remote control, like in the waste management, glass manufacturing process [18], and assembly of dangerous objects [19], or by twin robots with the help of the virtual reality (VR) in the industrial assembly [20, 21]. Collaborative robots have the advantages of real-time monitoring, feasibility, and safety.

Scientific experiments are automated by robotic technologies based on predefined experimental protocols that contain the key and fundamental information (structures) of experiments and describe the experimental plans, materials, sources, and processes [22]. In the automated laboratory, a workflow with assistance of robotic systems and softwares improves reproducibility and reliability of the experiments as well as guarantees the scientists' safety. Some workflows with single or multiple robotic systems have been proposed to help chemists prepare chemistry samples, such as the solubility screening and crystallisation [23] and rheometer unit operation [24]. Even in small labs, low-cost automation processes are fulfilled by robots, like the inexpensive 4-axis robot (Dobot Magician) for automated physical tasks: pipetting, autosampling for an atomic absorption

spectroscopy, and inoculation of bacterial cultures [25]. Robot-assisted technologies have also moved biological experiments into a modern and intelligent stage. For instance, microrobots are commonly used for positioning, sensing, transporting of microobjects [26], and optical tweezers can also be used as the *vivo* manipulator of the cells [27]. In addition, automated systems have been developed to conduct touch-evoked experiments for some species, such as *C. elegans* by using a microfluidic platform [28, 29, 30, 31]. In small clinical labs, robots support medical experiments, like the coagulation and clinical chemistry tests of medical samples repeatedly [32]. Besides the automated devices, the softwares designed for the experimental data analysis are also taking a vital role in the automated lab, e.g., the open source Chemotion Electronic Laboratory Notebook (ELN) for acquiring and handling the data from experiments in life science [33] and chemical science [34] into digitalized lab notebooks. An extremely large number of experiments can be automated in the lab, as they are usually highly repeatable and in a highly restrict protocol, but this also means that the robots need to work in a more accurate and robust level to generate reliable and convincing results.

Furthermore, robots are helpful for healthcare, such as the surgery robots, or drug delivery via so-called soft robots that are defined as the robots made of multiple soft and compliant materials [35]. The surgical work requires high accuracy, zero-error, and instant reaction/decision with surgical prior knowledge. Thus, surgery robots usually work in supervised manner under the monitoring of the surgical specialists. Apart from those single robotic systems, similar to the workflows in chemistry as mentioned above, the workflow for healthcare is also becoming popular in clinical use, where robots usually provide an affordable automation solution in a collaborative manner. For example, the concept of the novel workflow for the COVID-19 pandemic achieves prevention, screening, diagnosis, treatment, and home care as an automated pipeline [36]. Although the health science and health automation are becoming a more attractive topic, the robotic technologies can still not work in a full stand-alone mode and the human inputs are required, especially in the collaborative work with several devices.

Biorobots indicate machines that imitate biological systems [37] and take a vital role in scenarios being far accessible to humans. They have the advantage of flexibility with reference to natural creatures, such as birds, snakes, insects, worms, fish, etc. The studies on biorobots are focused on the parts of the animals, like legged robots for walking [38] or the micro-/nanorobots, being applied in medicine, surgery, animal behavior, neurosciences, and infectious diseases [37]. A typical type of animal-level biorobots is found in Boston Dynamics, having built robot platforms for imitating the human motions and undertaking the human tasks, e.g., humanoid robot platform (Atlas), or for imitating animal-like motions, e.g., Spot with ability of agile running, climbing, and jumping [39]. To perform search and rescue tasks as well as to guarantee the safety of the rescue crew, a snake-like robot can be used to manipulate the small objects and navigate the rescue environment [40]. Apart from the single biorobots, the robotic swarm is attracting more attention

from researchers with the inspiration of the swarm behaviors of the animals. According to the robot sizes, the robot swarms develop from meter-scale to micro-/nanoscale [41]. The swarm microrobots have been developed to perform the advanced tasks, such as the targeted delivery of nanoparticles by using a vortex-like paramagnetic nanoparticle swarm [42] and magnetotactic bacteria MC-1 (MTB [43]), the cooperative manipulation of the magnetically actuated mobile microrobots [44], and catalytic antimicrobial robots (CARs) for effective biofilm elimination [45]. The design of a biorobot is usually challenging as a result of discovering and designing the certain functioning of the behaviors from the animals.

1.2.3 Components of full automation interface

The examples of robots listed in Section 1.2.2 scarcely meet the requirements of no human assistance according to the definition of full automation interface in Section 1.2.1. A full automation needs to be a stand-alone solution (or workflow) that fulfills the desirable tasks on an end-to-end basis. As shown in Fig. 1.1, the full automation can be described as an interface between user and process. The user provides initialization for the interface that works with the help of a robot and offline data processing, and obtains the conclusion automatically drawn by the interface. The robot consists of real-time data processing, control policy, actuators, and sensors, which interact directly with the process. According to the feedback from real-time data processing, the control policy sends control commands to the actuators that output motions (or movements) of sensors and process. For example, the control command is expressed as the positions, acceleration, or rotation angles of the actuators. The actual status of process is captured by sensors as sensor data, e.g., image or numerical data, to real-time and offline data processing that are commonly implemented with the application of statistics, machine learning, and computer vision. The feedback generated by the real-time processing from sensor data indicates results of previous control command and provides the reference to next command. Many potential technologies have promoted the development of robots and data processing, and reviews on actuators, sensors, control policy, real-time and offline data processing can help in building a general template of full automation interface as a guidance for the automated processes in different scenarios.

Review on actuators The choice (or design) of actuators and sensors is challenging, as they manipulate and interact directly with the process. The review of potential actuators and sensors provides a guidance for the application of them. Conventional actuators for robots include passive actuators (e.g., brakes, dampers) and active actuators (e.g., electric motors). The combination of them forms new types of actuation setups, like active-passive actuators [46]. The group of Biomedical Engineering & Robotics (BER) at Karlsruhe Institute of Technology (KIT) proposed a body energy harvesting (BEH) device based on an electric motor to generate arm and leg swing motions [47]. Another example of a newly introduced actuator is the vacuum-actuated

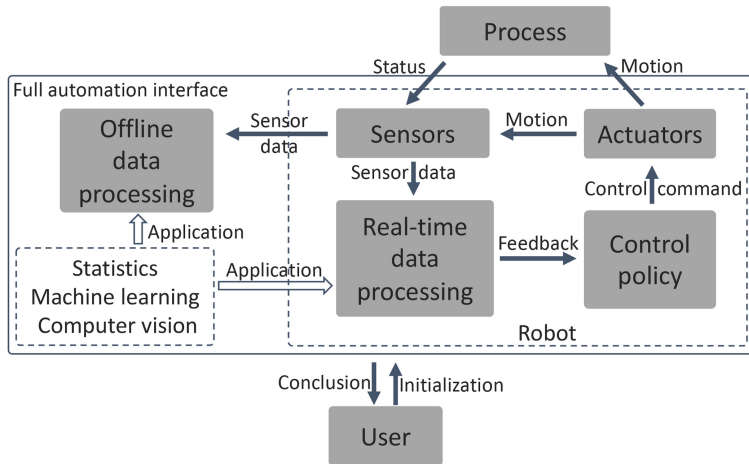


Figure 1.1: Essential parts for a full automation.

rotary actuator used for the wearable soft knee [48]. The variable stiffness actuators (VSAs) are commonly used to provide intrinsic compliance for safe human-robot interaction [49] or for the lower limb robot [50]. The parallel elastic actuators with the help of springs can fulfill the movement of the snake-like robot. Besides, advanced material science and technologies have boosted robots to a new level, like the soft robotics. In those new advances, the actuation is fulfilled by the novel material technologies, mainly including the bio-chemistry materials (water-/pH-/thermo-/photo-/magnetic/electrically responsive actuators [51]). [35] proposed a multi-material soft robot with elastic polymer and magnetic nano-particles (NPs) to handle more complicated scenarios for the drug delivery. There are also other smart actuator systems, e.g., light (infrared laser and phototransistors) induced manipulation in fluidic environments [52] and the magnetocaloric actuation (polymer bilayer actuator) for the soft polymer robots [53]. In terms of the special use in the design of full automation, some new actuators are fabricated via 3D printing [54]. The work described above lists various potential cases of actuators, and they are mainly designed for special use cases, e.g., some specific experiments or robots. However, their use in the design of the general template for full automation interface is still challenging, and it is vital to select right actuators or design new actuators that can be commonly used in automated processes.

Review on sensors The sensing devices are the eyes for the full automation to capture the world, such as the camera, temperature transducers, or rotary encoders. Some new technologies are proposed for special uses of robots, such as the design of multi-modal sensing disk unit (SDU, shape mapping, contact detection, and force sensing) for safe human-robot interaction using continuum robots [55], the design of a soft and thin-film tactile sensor for the force sensing used

in manipulation tasks [56], and the new ultrathin three-axis fiber Bragg grating (FBG) force sensor for the robotic arm [57]. In the automated laboratory, sensors are essential to the development of the automated devices, e.g., the in situ sensors and flow reactors used in the microfluidic technologies [58]. Apart from new designs of sensors, the fusion of multiple sensors also achieves advanced functions, like the rope-measuring sensor fusion (length, angle, and absolute angle) for a dual ascender robot (DAR) [59], the multi-sensor fusion for the self-localization of a miniature underwater robot without GPS [60], the design of a firenose (with three gas sensors, an infrared sensor, a temperature humidity sensor, and a flow sensor) for the use of the robot in the firefighter [61]. The BER group at KIT proposed a measurement chamber with O₂ sensor, temperature and humidity sensors combined with additional 3D-printed parts to conduct the concentration measurement of ozone gas used in the medical treatment [62]. They used polyvinyl chloride (PVC) to fabricate the components that contact directly with ozone gas, and this work makes the best of material science and integrates different existing sensors to achieve new functions. Additionally, material science is used for the sensing design of the bioinspired robots, including biological materials, carbon-based materials, hydrogels, liquid crystalline polymers, etc [51]. Similarly to actuators, in the certain usage scenarios, specific sensors are designed by the combination and adaptation of the work reviewed above or design of sensing systems. However, the common sensor choice in the template full automation interface can be a visual sensing device, e.g., camera, with the help of AI advances. Thus, robot-vision methods via AI is one essential factor to the choice of sensors.

Review on control policy The control policy is the strategy used to gather information from sensors (i.e., the results of real-time processing) and control or coordinate the sensors and actuators according to the tasks of the processes. The design of control policies is booming, e.g., the in-the-loop decision-making process in the hierarchical architecture for the collaborative robots [63] and decision-making via reinforcement learning for human robot interaction [64]. Moreover, as for complex automated devices, the coordination of different parts (actuators, sensors, and processes) is essential to achieve a full automation as visualized in Fig. 1.1. The control strategies reply on the real-time processing of the sensor data, so data analysis methods are important to find a good strategy. With the developments of artificial intelligence (AI) technologies, the search of control policies is becoming smarter and more robust via learning of the environment [65, 66]. The group of High Performance Humanoid Technologies (H²T) at KIT proposed diverse control policies for the manipulation and grasping tasks for Humanoid robots based on machine learning, e.g., the Bayesian recursive state estimation method of robot action [67] and the semi-autonomous policy for the grasping task of prosthetic hands based on the fusion of image processing module and object distance module [68]. Additionally, [69] proposed to finish the picking task of robot via controller considering the image feedback and robot's joint state. The performance of those policies have proven to work well in simulation or a limited real-world cases. However, it faces

more challenges in terms of the application of them in a specific process or in the template design of full automation interface. The review above also hints that the real-time processing is one essential factor to the design of control policies.

Review on real-time and offline data processing Data processing is used in three ways: real-time processing of data generated by sensors during the automation process, offline inspection of data from the process, and offline analysis of the inspection results. The real-time processing of the sensory data generates the essential information to show the status or the behavior of an automated process as feedback for a better control. The raw data collected from the process is hard to visualize and difficult to be understood by humans, so the transformation of the data into understandable variables is important. Additionally, when it comes to a high-efficiency production line, the inspection on a single data point is not reliable to draw a conclusion. With the development of data mining methods, the analysis of the inspection results of the repeated process cases is becoming more interesting and attractive, as it is easier for humans to gain the information of the large-scale dataset via the results of statistics or pattern analysis. Thus, the data processing methods are to extract useful information that can be used for the control policy in the real-time manner (e.g., the feedback with the tracking positions of the targets or the health status of the actuators) or the data analysis results in the offline manner (e.g., the trajectories of the targets, classification, or clusters). The analysis of the data in the complex, continuous, or multi-dimensional data format (e.g., numerical and image data) is the challenging part of the full automation interface. The key is to apply an intelligent data processing pipeline to the complex data generated in the automated process, including preprocessing, feature engineering, result generation, and post-processing as illustrated in Fig. 1.2.



Figure 1.2: The pipeline of data processing.

As the data inputs from sensors usually have noise, the first step is preprocessing of the data to generate better new inputs to the pipeline by filtering. The filtering of the numerical or image data can be done by various filters, e.g., the commonly used median filter [70], or Gaussian filter [71]. The parameters (e.g., filter size) need to be adapted according to the actual data, and researchers usually select those parameters heuristically. To better represent the data, feature engineering is commonly used to extract the key information (feature) from the data output by the process. Many feature engineering methods can be considered, including linear transformation, e.g., principal component analysis (PCA) and independent component analysis (ICA) [72, 73], non-linear mapping or neural networks (auto-encoded architecture [74]). Feature engineering

methods of data in the image format usually include Canny [75], SIFT [76], CNNs [77], etc. Given the feature from feature engineering, the result generation can be achieved by intelligent and advanced methods for the real-time processing or offline data analysis. The result can be classification, detection, segmentation or tracking, which are usually not directly used, and modification (postprocessing) is required, e.g., normalization and unit transformation for the strategic control, and vivid output plotting for the analysis results. The methods above were mainly proposed for a general data processing task and tested on the general datasets. As visualized in Fig. 1.1, the challenge of applying them to the design of real-world full automation interface comes from the interaction with the specifically chosen actuators, sensors, and control policy.

Review on statistics and machine learning The statistics indicates the comparison, curve fitting or description of the data. Given a dataset, researchers usually tend to seek the comparison of them, so the t-test [78], analysis of variance (ANOVA) [79], multivariate analysis of variance (MANOVA) [80] or distance measurement metrics [81] are commonly used. The statistical methods are usually used for the simple comparison of the dataset, but modern processes consider more complex tasks, i.e., the regression, classification, clustering, or data-generation, achieved by intelligent machine learning models. The application of machine learning to the data processing pipeline in Fig. 1.2 is more attractive but challenging. In some data formats, manual supervising information can be provided, like the classification labels, continuous numbers, bounding boxes of the targets from the operators or researchers. In those processes, the supervised learning methods can be an ideal option to train models for classification, object detection, and segmentation, like support vector machine [82], neural networks [83], and Gaussian processes [84]. As the labelling is difficult or impossible for some types of data, unsupervised learning methods are the only way to analyze the data, including the clustering methods like K-Means [85], Mean-Shift [86], density based clustering [87], hierarchical clustering [88], or generative learning methods to map the data points into distributions, like variational auto-encoded neural networks [89] and variational auto-encoded Gaussian processes [90]. When parts of the data points can not be used or the process cases are difficult to repeat too many times (the number of generated data points is limited), active learning [91], self-supervised learning or data synthetization [92] can provide help in the case. The challenge of machine learning methods comes from the modification to a specific process in the real-world case because of their black-box techniques. In other words, researchers can hardly figure out the theoretical principles of the functioning of those methods, resulting in difficulties in adjusting the architectures of the methods and optimizing the parameters involved in the methods.

Review on computer vision Computer vision methods focus on the processing of image or video data. Often, they are used to implement soft sensorics to derive data, which can not be obtained by direct measurements. Apart from the feature extraction mentioned in Fig. 1.2, some other methods are to be considered in various tasks, e.g., object detection, segmentation, and tracking.

The object detection methods aim to search the areas (usually displayed as bounding boxes) of the targeting objects in an image, including the conventional techniques, e.g., Hough transform [93], and deep-learning based advances, e.g., Faster RCNN [94] and YOLO [95] based on CNNs and proposals of bounding boxes by neural networks. The object segmentation methods aim to generate the pixel-wise classification, e.g., binarization [96] based on a single threshold and U-Net [97] built by CNNs in a U-Shape. The U-Net based methods usually require manual annotations (labels) in the training phase, which is not practical in some real cases. However, the Cycle Generative Adversarial Network (CycleGAN) can be used to generate the synthetic annotations used for the training of segmentation methods [98]. Object tracking methods aim to calculate the positions of the targeting objects in each frame of video data, e.g., optical flow for the tracking of slowly moving objects [99, 100], Kalman filter regarding tracking as a linear process with recursive mathematical principles [101], particle filter in nonlinear or non-Gaussian cases [102], and 3D CNNs by applying CNNs on batch of frames [103]. As computer vision technologies can achieve a task in an intelligent manner, those methods are commonly considered in terms of designing a robot or a new automated process. For example, those methods can be integrated into the experimental devices to achieve an intelligent imaging procedure, e.g., the software SerialEM proposed by [104] for automated transmission electron microscopy, which involves objection detection, image registration, etc. Recently, [105] improved the annotation quality of image segmentation issue via uncertainty-aware deep neural networks (DNNs), and [106] combined the results of CNNs, stacked autoencoder, and DNNs to do the automated COVID-19 identification and diagnosis based on computed tomography (CT) images. However, the time consumption of training deep learning based methods as well as the computational costs of those models are usually big disadvantages of computer vision methods, so developers need to make a compromise between performance of the methods and the requirements for real-time processing.

1.3 Automation in laboratories

1.3.1 Overview

Various widely used robots provide the clue that full automation commonly requires collaboration between different devices or softwares, and a customized and specific automated workflow/protocol is usually designed, especially in the automated laboratory where the workflow is required to be well controlled. The experimental protocol in the laboratory or clinical use is required to be strictly carried out, and the parameters involved in the experiments are specifically defined and precisely controlled. Based on this knowledge, a full automation holds the potential to improve the whole protocol both in efficiency and accuracy, and thus, biological, chemical, physical, or

clinical experiments can benefit from the concept of full automation. Transcriptic, a start-up by Max Hodak, is a typical example of automated laboratory [4], which builds up a warehouse, "biology data centre", to conduct the experiment and acquire the corresponding results. With a powerful application programming interface (API), the users can design tens of common experimental protocols finished by the automated workstations, so that the scientists can spend most time on science instead of notebook work. The reasons of automated lab attracting high attention from researchers and engineers are summarized as the following factors:

- A great number of different experiments can be automated, and the experiments in similar categories can follow a general template design of the automated system. Thus, the design of a specific automated lab interface can be extended to the other related interfaces,
- Experiments can be done in a high throughput, e.g., Zymergen, a company called "AI-powered biotech", can conduct 1000 experiments per week [107]. This can reduce an large amount of manual lab work,
- The automation technologies in the lab can make high profits. The laboratory automation held a huge size of global marketing with value \$4,884 million in 2018, which promises to grow up to \$8,424 million by 2026 [108]. The lab automation only in Europe can expect a marketing share of \$3,637.82 million by 2029 [109]. The market in such scale will attract high investments from industry and durable interests from researchers,
- The amount of scientific data is increasing rapidly, which requires development of the data analysis and processing tools [110], and
- The automated lab promotes the emergency of new technologies by the adaptation of current research and methods.

1.3.2 Related automated laboratories

The development of a full automation in laboratories is summarized in three directions:

- Experimental preparation: The experimental samples, e.g., chemical compounds and solutions, tissues, cells, or animals, are prepared, distributed into sample holders (e.g., Petri dish, well plate, beaker, or Erlenmeyer flask), and placed in the experimental devices via automated systems in advance,
- High-throughput experimental recording: The experimental protocol is conducted via automated systems, with experimental data recorded and collected, and

- Lab software development: The automated devices usually need to work with the help of software packages, especially in the experiments with much lab notebook work.

To have a detailed understanding of the automated laboratories and reveal the underlying related research topics, more specific advances in these three directions are investigated.

First, the experimental preparation procedure is the most repeatable and precision-demanding part of the experiment under a fixed workflow. All the lab work relies on the preparation of samples, as the experimental reaction is constrained to the quality of the preparation. In this case, the automation can undertake the preparation work, which usually includes the design of new sample holders, e.g., multi-droplets and new pipetting systems. The German Cancer Research Center (DKFZ) in Heidelberg, for example, showcases the designed "autoSP3" to make the unbiased clean-up of proteins and peptides a simple and fast pipeline [111]. The SP3 method can handle the input samples down to low samples containing only 100 cells or less, with high reproducibility and precision. Most importantly, this design can be extended to other related clinical or non-clinical research fields with the combination of other devices. The promotion of material science is also a key feature in the automated lab owing to its contribution to the high-throughput experiments. The group of Pavel Levkin at KIT reports a droplet microarray (DMA) platform that can be used for miniaturized high-throughput experiments, e.g., the preparation and screening of antibacterial compounds [112] or nanoliter sized hydrogel [113] with few pipetting steps.

Second, the experimental recording is the most time-consuming part of the lab work, especially in the high-throughput experimental recording task. Many high-throughput automated laboratory facilities have shown a satisfactory performance in generating experimental data, mainly including the devices for observing or manipulating the experiment subjects. In terms of observation, devices can be designed for the imaging of alive biological organisms and chemical reactions. For example, the IMAGING Cluster at KIT contains various imaging facilities that can be applied to a massive number of experimental protocols [114]. The advanced synchrotron X-ray imaging modality has been widely used in different biological imaging processes in a high-throughput and semi- or fully automated manner with the help of artificial intelligence methods, e.g., imaging of medaka, zebrafish, and *Xenopus laevis* [115]. In terms of imaging on the freely moving *C. elegans*, [116] proposed to adjust Magnetic Resonance Imaging (MRI) modality to capture the response of the organism stimulated externally and predict their motion via automated image processing. In chemistry, the images of the reactions can be recorded to calculate the solubility and test reaction variables, which can be automatized via the proposed solid- and liquid-dosing modules with the help of computer vision and machine learning [117]. Apart from the observation, manipulation is also a challenging task in experimental protocol where automation technologies can show strong advantages. The liquid-handling stations or devices have been applied widely in automated chemical and biological lab work with manipulation of tips, needles, and nozzles

[118]. The screening center at KIT proposed multiple automated biological experimental systems, e.g., automated embryo sorting [119] and intelligent Olympus ScanR screening microscope [120]. Additionally, the automated systems to touch-evoked experiments on *C. elegans* via microfluidic platforms limited the animal within a small area or even in a channel, causing the limitation of the movements of the animal to control the readouts but accelerating the experiment process. A much more precise system [121] was proposed to measure the force stiffness of specific parts of the *C. elegans* body in nano-scale level. As well, other systems were also proposed to conduct touch-evoked experiments on the freely moving *C. elegans*, like [122], which only focused on the mechanics of *C. elegans* to force-varied touching, and [123], in which the pair of permanent magnets still limited the movements of the animal and the touching angle cannot be well controlled. Those systems prove the potential of applying the robotic technologies in the biological experiments.

Third, the analysis and processing of the experimental data requires the development of lab software and tools, which can also be integrated into the existing devices to promote an automated imaging process. The group of Automation for Laboratories (ATLAS) at KIT is developing various software platforms for the processing of data from diverse scientific labs. For example, in [124], the detection and analysis of heartbeats of zebrafish embryos from videos is fulfilled fully automatically based on an advanced image processing pipeline, and in [125], the phase analysis of X-ray diffraction (XRD) scans is automated by deep learning methods. Besides, [104] designed software tools for automated transmission electron microscopes based on image analysis, which shows high flexibility and is useful and applicable to different automated experimental protocols. The automation in biology is playing a higher role in modern research, as it gathers interdisciplinary technologies, knowledge, and research interests from informatics, robotics, mechatronics, biology, chemistry, etc.

Those automated systems or software packages above mainly focused on the sub-tasks. According to the definition of full automation interface in Section 1.2.1, the lab interface also requires a series of devices that fulfill sub-tasks in an integrated manner with no manual intervention.

1.3.3 Zebrafish experimentation

The study of animal behaviors is attracting more interests from reserachers, including two manners: modelling the animal personality [126] that can be defined as behavioral difference of individuals in one animal species [127, 128], or analyzing common patterns of animal behaviors [129]. Robotics has shown the potential in animal behavioral research by evoking customizable, controllable, and repeatable stimuli, e.g., the study on the interaction of live animals with a robotic

replica [130]. Among the research subjects in the automated biological lab, zebrafish (*Danio rerio*) has become a commonly used organism model in behavior research, genetic studies, and drug discovery as a result of the characteristics of zebrafish, i.e., optical transparency, small size, genetic tractability, measurable behavior, high fecundity [131], and similar genome to humans (70% genetic homology [132]). Zebrafish experiments follow strict protocols, i.e., a detailed description of sequence of tasks and operations for the conduction of experimentation [133]. A great amount of experiments can be conducted on the zebrafish model, and the behavioral screening is becoming an attractive research focus. Zebrafish displays diverse behaviors, the screening of those behaviors indicates the test and analysis of behavioral patterns of zebrafish, which can be used for the detection of genetic mutants [134]. Besides, as a result of the rapid developmental stages and the capability of high generation of the eggs, it is possible to do the high-throughput screening for the functioning and effect pattern of chemical compounds, i.e., drug screening. More than 80% of the genes associative to the human diseases can be found the clue in the zebrafish model [135]. Behaviors of the zebrafish vary at different developmental stages, so the selection of the behavior and developmental stages follow the actual research goal. For instance, the research on the locomotion (movement) of the larvae require elder samples, as the younger samples do not develop complete moving skills.

The zebrafish is housed in tanks made of high-quality glasses¹ with fresh fish water after filtering and the room temperature at constantly $28 \pm 1^\circ\text{C}$ [136, 137]. The adults are fed twice a day. The zebrafish has a high fecundity, as an adult fish can generate hundreds of embryos during the breeding procedure. In breeding, the female and male adult zebrafish (as shown in Table 1.1) spend together for 14-10 h day-night cycle (12-14 hours for the day, 10-12 hours for the night). The embryos are collected in the following morning and are raised in an incubator with fish water (60 mg/mL f.c. sea salts Instant Ocean Spectrum Brands, Blacksburg, VA) and temperature at 29.5°C . The embryos can survive on the nutrients of the yolk for 5 days.



Figure 1.3: The adult zebrafish (left: female, right: male, [138]).

Zebrafish embryos develop rapidly in the early ages (stages) and take only hours from cells to larvae [139], as shown in Table 1.1, where the early developmental stages of the larval zebrafish

¹ In this thesis, all zebrafish larvae used were raised and collected in conformity with the Animal Ethical Requirements.

are outlined. The zygote period of zebrafish with only one cell is defined as 0 hpf (hours post-fertilization), and the embryo develops through cleavage, blastula, gastrula, segmentation, pharyngula, and hatching periods to a larva. Those periods are defined according to the morphology, size, and age of the zebrafish embryos. The size of the embryo grows from cell level to micro-meters before hatching. The hatching of zebrafish embryo indicates the transition process from an embryo to a free-living individual larva [140] out of their chorion, i.e., the acellular envelope surrounding their egg [141], as visualized in Fig. 1.4a and b. The zebrafish embryo spontaneously hatches at 48-72 hpf. However, some specific experiments are conducted before its spontaneous hatching, so a manual dechoriation procedure is considered by removing the chorions of the embryos with a forceps [142] for a better imaging of zebrafish in the embryonic and larval studies [143], as illuminated in Fig. 1.4c.

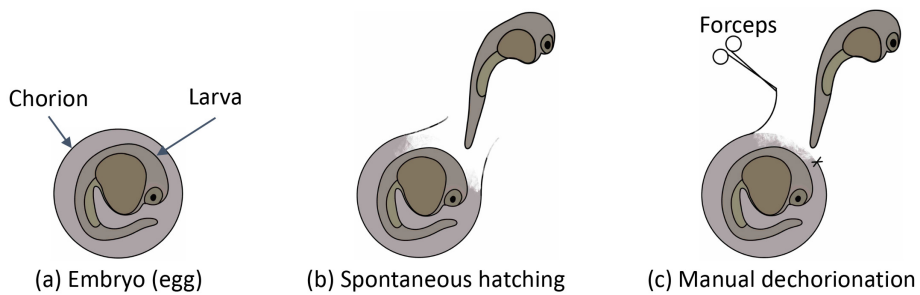











Figure 1.4: (a) Zebrafish embryo (egg). (b) The spontaneous hatching of a zebrafish embryo. (c) The manual dechoriation of a zebrafish embryo.

During the very early stage of development, the embryo of zebrafish has already established sensory and motor circuitry, showing various behaviors and responding to several stimuli. As detailed in Table 1.1, the embryo shows movements of cells in the first 10 hpf, e.g., pronuclear congression [144], ooplasmic segregation [145], cleavage [147], cytokinetic abscission [149], and convergent extension of the mesoderm with changes of cell shapes [151]. Upon those behaviors, many possible experimental protocols have been designed. For example, [146] tested the effect of Venlafaxine (VEN), a selective serotonin and norepinephrine reuptake inhibitor (SSNRI), on the development of zebrafish brain by injecting one-cell embryo (zygote) with VEN. Similarly, [151] analyzed the function of Gravin, a tumor suppressor, on the cell shape and behavior changes (i.e., gastrulation) of zebrafish with the injection of Gravin into zebrafish embryos at the one-cell stage. In [148], the live imaging technique of zebrafish cells at cleavage was proposed to characterize the cytoskeleton (protein filament network) of zebrafish embryonic cells. [150] proposed a procedure to fuse two zebrafish blastulae that can be used to generate parabiotic zebrafish embryos, i.e., two embryos sharing a common body. After 10 hpf, the morphology of the zebrafish becomes

Table 1.1: The developmental stages of zebrafish with the possible behaviors and experiments at each stage.

Stage	Hours	Lucida sketches [139]	Size	Behavior	Experimentation
Zygote period	0 - 0.75		One cell	Pronuclear congression [144]; Ooplasmic segregation [145]	Cell injection [146]
Cleavage period	0.75 - 2.25		2 - 64 cells	Cleavage [147]	Real-time imaging of microtubules and microfilaments [148]
Blastula period	2.25 - 5.25		128 - 1k cells	Cytokinetic abscission [149]	Blastula fusion [150]
Gastrula period	5.25 - 10.33		50% - 100% epitholy	Convergent extension of the mesoderm, cell shape changes [151]	Regulation of Gravin in the migratory behavior of mesodermal cells [151]
Segmentation period	10.33 - 24		EL (embryo length): 0.9 - 1.6 mm	Response of boundary cells to mechanical cues [152]	3D and time imaging of cell divisions, cell tracking [152]
Pharyngula period	24 - 48		EL: 1.9 - 2.9 mm	Photomotor response [153]	Photomotor response screening [119]
Hatching period	48 - 72		EL: 3.1 - 3.5 mm	Shrinking yolk, open mouth [139]	Influencing factors in hatching rhythm [154, 155]
Day 3 - 4	72 - 96		Body length: 3.5 - 3.7 mm	No spontaneous movement, occasional tail flicks [156]; Locomotor movement to touch stimuli [157]	Touch response screening [157]
Day 5	120		Body length: 3.9 mm	Spontaneously motile [156]; Feeding behavior, locomotor movement to vibration [158]	Vibrational response screening [158]

Drug screening

an embryo body from cells, and it begins to respond to external stimuli. For example, [159, 160] conducted the touch-response assay to study the locomotion of zebrafish in response to the external mechanical stimuli. Besides, [119] analyzed the response of zebrafish embryos when exposed to visual stimuli, e.g., a short and bright light flash. In order to test the functioning of Mauthner cells (M-cells) on the initiating fast-start, or "startle", responses to unexpected and abrupt stimuli, [161] screened the response of the normal larvae and larvae lacking M-cells to the vibrational stimuli. The zebrafish larvae can also show response to the chemical stimuli. For example, [162] proposed a microfluidic device that injects pulses of chemical stimuli to the fixed larvae and analyzed the tail behaviors stimulated by the chemicals.

At pharyngula period before hatching, the zebrafish embryo begins to feel visual stimuli. For instance, the embryo has photomotor response (PMR) [153] with a light excitation, which can be used for the drug screening [119] as PMR of zebrafish embryo alters under different neuroactive compounds. The zebrafish begins to respond to external mechanical stimuli at around 21 hpf (segmentation period, slightly earlier than PMR), like the response of boundary cells to mechanical cues, and it shows the movement of group cells or tissues. Therefore, 3D+t (3D over time)-imaging of cell divisions and cell tracking is usually conducted at this stage [152]. Besides, [163] proposed an efficient framework for 3D cell image acquisition over time, generating 4D data of cell movements. After the segmentation period, the touch response of zebrafish is displayed as the movement of the whole body, and the zebrafish keeps its ability of feeling touch stimuli in the following developmental stages. At 3 days post-fertilization (dpf) after hatching, the embryo of zebrafish grows to the larval level, and the ability of zebrafish larva feeling touch is well-established with response as movements and escape behaviors [157]. However, before 4 dpf, the larvae do not show spontaneous movements and keep immotile without touch stimuli [156]. Therefore, the touch-response behavioral screening is commonly conducted within 3 dpf [157]. At 5 dpf, the larvae begin to be spontaneously motile [156] and develop feeding behavior and movement to vibration [158]. Therefore, the experiment to analyze the vibrational response (movement) of zebrafish larvae can be conducted at this stage [158]. During stage 2-5 dpf, the larvae can move reaching 80-95 (near-) cyclic moving events, i.e., the movements of zebrafish larvae exhibiting cyclic, with the speed at 200-300 mm/s [164].

One application of experiments described above is drug screening, which can be used as the way to understand the pathogenesis (i.e., the process of disease developing) of inherited human diseases, like kidney diseases [135], visual impairment [165], and to discover and verify the potential treatment approaches to each disease. The way of doing the drug screening on zebrafish can be diverse:

- observation of morphological changes of zebrafish bodies or internal organs compared with control groups (i.e., the wild-type larvae), like the pathological changes or the defects in the head, brain, ears, liver, and heart, etc [166],
- the changes of the zebrafish behaviors after treatment by drugs, like the external stimuli evoked behaviors of larvae under treatment [167] compared with control groups.

The behaviors of the zebrafish alter under the treatments of chemical compounds, so the preparation procedure of the chemical treatment on zebrafish is vital to the drug screening experiments. Basically, the changes of the behaviors is related to the treatment time and concentration of the compounds. Therefore, in terms of time, the animal can be treated in a short term (30 minutes to 2 hours) or long term (over 10 hours or even days). Besides, the dose screening (treatment with different concentrations of the compounds: e.g., 5, 25, 50, or 100 $\mu\text{mol/mL}$) is commonly conducted for observing the drug effects as a function of concentration. During the preparation of a compound in a certain concentration, the dimethyl sulfoxide (DMSO) is used to dissolve the compound followed by dilution with fish water [168]. Therefore, the experiments of drug screening are usually conducted on zebrafish larvae without any treatments (i.e., as wild type as control group), with DMSO (also as control group), and with chemical treatments (experimental group), for the comparison of their differences.

1.3.4 Touch response of zebrafish larvae

Among the experiments described in Section 1.3.3, the touch-response screening experiment on zebrafish at larval level is increasing the interests from researchers, which is conducted by the collaboration of biologists, animal behavioral scientists, chemists, engineers, and data scientists. Conventionally, the "touching" is elicited to zebrafish larvae manually with a needle or the end of a regular Pasteur pipette or similar instruments [159, 160, 169], as visualized in Fig. 1.5. The larva is touched at an uncertain position of the trunk with visual localization. The response can be captured in images or videos by using CCD camera, fluorescent microscopy, or light microscope [170, 171, 172, 173, 174]. In [159], the manual touching is conducted on the non-fixed zebrafish larvae at 2 dpf to study the movement performance concerning muscle function for the wild-type larvae without genetic mutants and those with genetic mutants. The larvae so not have spontaneous movements, and they are placed in the middle of a Petri dish, touched, and recorded manually, and each experiment is replicated only for 15 individual fish. The quantification of the movement behaviors, i.e., the movement acceleration, is done via automatic tracking of the trajectory based on the manually initialized positions and moving areas of larvae. Similarly, [170] proposed to screen two types of drugs via manually touch-evoked behaviors on zebrafish larvae at 3 dpf, and the collected videos are analyzed via ImageJ generating the movement distance.

Obviously, in the manual experiments on touching the zebrafish larva, operators need to select the larva and limit the measurement area visually, and usually only one larva is focused for the touching at once. The quantification of the behaviors is limited according to the functions of the chosen software packages that do not provide detailed analysis of the behavior in each frame. In addition, the behaviors of zebrafish endure thousands of frames, so screening on a large number of larvae causes a high burden on the operators.

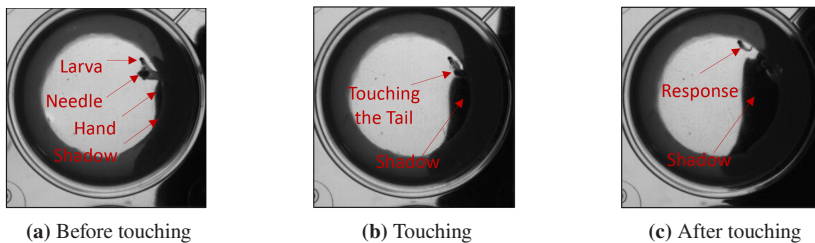


Figure 1.5: The visualization of the problems caused in the manual touch-response experiment [175].

To recap, the manual approach of the touch-response experiments of zebrafish larvae, as shown in Fig. 1.5, has some obvious drawbacks:

- **High time consumption:** To guarantee the precision of touching, operators need much time to locate the larvae and apply the touching carefully. Due to the high time consumption and thus low experiment numbers, it is almost impossible to obtain sufficient results,
- **Bad quality of video data:** As visualized in Fig. 1.5, the image quality is bad because of the hands, making the needle and larvae unclear to observe,
- **Heavy pain:** To collect enough experimental data, operators usually devote many efforts to the experiments and long-time experimental work. Thus, experiments are not often conducted as people do not want "to lose their time",
- **Lack of repeatability:** Even the expert operators cannot assure to conduct the experiment precisely, and it is not known whether experiments deliver the same outputs,
- **Uncontrollable touching parameters:** In manual experiments, the essential touching parameters, e.g., the location of the touch input on the zebrafish body and the force with which the touch input is applied, cannot be strictly controlled for consistency. Influence of varying touching parameters is not known,

- Unreliable conclusions: The response elicited manually varies owing to the significant variations of the input touching parameters even by the same operator and thereby might not result in convincing conclusions, and
- External factors: The hand or the needle may cause shadows that make the response of the larvae not observable, or fluid motions that shock the larvae away, so the results in this way are usually not reliable and not comparable for specific conclusions or knowledge discovery.

As it is impossible to manually leverage the zebrafish model to scientifically perform high-throughput behavioral screening, a fully automated and intelligent system (a full automation interface of the touch-response experimental process) is needed to accurately provide the touch stimulus, record the data at a high speed and resolution, and automatically inspect and analyze the response behavior itself. The touch-response experimentation on zebrafish larvae helps life scientists, neuroscientists, animal scientists, and geneticists understand the genetic expression, neural functioning, behaviors of zebrafish model which can be referred to humans. This experiment can be conducted weekly, so the touch-response interface can be used frequently. Besides, one essential application of the interface is automated drug screening, and the market of drug screening is estimated to grow in size by \$10,944.1 million between 2021 and 2028 [176]. Therefore, a full automation interface of touch response used for drug screening is expected to play a vital role in the drug market.

1.3.5 Open questions

Designing a customized automated interface for touch-response experiments on zebrafish larvae still presents numerous challenges. Common automated systems for *C. elegans* as described in Section 1.3.2 cannot be used for zebrafish larvae, as they have different sizes (*C. elegans* in micro scale [123] and zebrafish in 2.5-3.7 mm from 30 hpf to 96 hpf [139]) in length, and their movement speeds are different (*C. elegans*: less than 1 mm/s [177], zebrafish larva: 200-300 mm/s [164]). Therefore, the features to be derived are different, such as the behaviors to quantify and analyze. In the robotic designs mentioned in Section 1.2.2, few researchers have considered a full automation interface in the lab work for data acquisition, behavioral inspection as well as pattern analysis in an integrated experimental system, so the open questions and requirements are raised as:

1. The current touch-response experiments on zebrafish are still conducted manually, so the experimental throughput is low. The robotic touch-response experimental data recording system is missing. A workflow with high-throughput screening, i.e., recording the responses

of a large number of the targets (zebrafish larvae) in a fast and unsupervised manner, is demanding. Besides, current interfaces on other animals still involve manual work and do not meet the requirements for the application to the zebrafish model. The choices of actuators, manipulators, and sensors are not clear, and real-time data processing methods, as well as control policies, are still open to be explored, selected, and adapted to achieve the automated recording of touch response.

2. The inspection of touch response is still done manually or semi-automatically, which can only generate a few simple quantification outcomes of the touch-evoked behaviors. Those outcomes are not objective and not repeatable. The automatic offline data processing software is still missing. The software on other animals is not directly applicable to zebrafish, as they show different types of responses or behaviors. It is still open to apply the technologies in statistics, machine learning, and computer vision to the offline data processing pipeline.
3. The design of a fully automated touch-response interface with minimal human intervention is not fulfilled. The key requirements for automated (data-driven) laboratories include the data generation (acquisition) processes and data processing (analysis) pipeline [178] in an end-to-end manner. Current interfaces in the experiments on other animals only focus on the data acquisition part or the data analysis part, resulting from the challenges and limitations of sensors, actuators, data processing, and control policies. The methods and techniques listed in Section 1.2.3 can help build a fully automated touch-response experimental interface in the lab. However, the theoretical research or simulation of those methods cannot consider all experimental conditions, so those methods are not easily replicated in the real world [179]. As the robotic systems can be used for the design of the interface, it is still not explicit to apply the robot to the experimental protocol and integrate the output generated from the robots into the interface.
4. The input parameters and experimental conditions affecting the touch-response readouts and the output parameters to quantify the touch-response behaviors are not clear. In the manual work, many touching parameters in the pipeline affecting the experimental results are not controllable. The data acquisition procedure in the experiments is expected to have a high accuracy and good robustness (repeatability) - touching the same location of the targets with the same speed (force) in each experiment. In the robotic touch-response system, definitions of parameters, influence of the parameters on the readouts, and choices of the optimal experimental conditions are missing.
5. The customized evaluation of the inspected and analyzed results for readouts and potential conclusions are missing. Optimization of the parameters in the interface as well as choices

of experimental conditions aim to improve the performance and effectiveness of the interface, but the definitions of good performance and high effectiveness are not clear. The evaluation metrics are required to generate specific numbers to validate the effectiveness of the interface.

1.4 Goals and main contributions

The design of the automated touch-screening interface on the larval level of zebrafish requires minimal or no human intervention, easy controllability and extensibility, full automation, and safety, including the exploration of a series of robotic platforms. It needs a touch-response experiment protocol in a fully automated manner including data acquisition, later-on quantification, and analysis of data patterns (Fig. 1.6). The goals and main contributions of this thesis are summarized as:

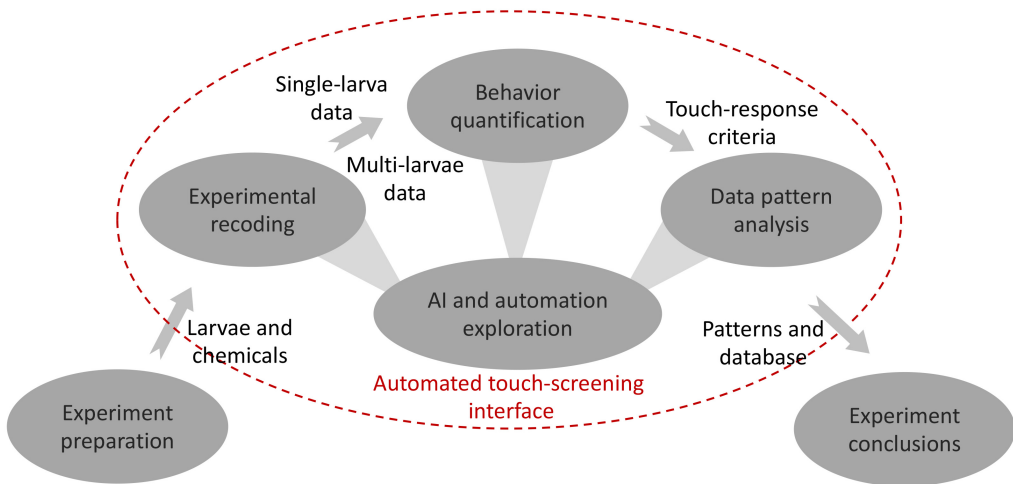


Figure 1.6: The goal of the thesis, i.e., design of the fully automated touch-screening interface on zebrafish larvae.

1. The general concept for full automation is proposed and extendable to other related processes,
2. Automatic experimental recording systems are designed to collect the touch-response experimental data in the process with minor human intervention involved, including both single- and multi-larva cases,

3. The exploration for the experimental recording includes mechanical design, i.e., manipulators, actuators, and sensors, and software design, i.e., control policies and real-time data processing methods,
4. Behaviors of zebrafish larvae are quantified automatically according to the data collected in the process with no human intervention involved,
5. Offline data processing methods are discussed for the optimal choices used in the behavior quantification,
6. Patterns shown in the data from the process are analyzed automatically with no human intervention involved,
7. The pattern analyzing and data mining methods are chosen for the analysis task,
8. The work above is integrated as a fully automated touch-screening interface for the first time, as visualized in Fig. 1.6,
9. New concepts and parameters are introduced for the touch-screening interface, including the conceptualization of general touch-response parameters to conduct and quantify the touch-response behaviors,
10. Evaluation metrics are defined to evaluate the proposed touch-response screening interface,
11. Customized experimental protocols are designed for verifying the controllability of the parameters and effectiveness (e.g., efficiency, accuracy, repeatability, robustness, etc.) of the proposed interface, and
12. The strategies are proposed for guaranteeing the safety of the operators, manipulators, experimental samples, and experimental devices.

1.5 Organization of the thesis

The organization of this thesis is as follows. Chapter 2 delivers the concepts of the automated interface in modern processes, and introduces the proposed new concepts and parameters used for the design and evaluation of the touch-screening interface. Chapter 3 describes the details of the design for the data acquisition in single-larva case, as well as corresponding experiments and results. Chapter 4 upgrades the data acquisition system to the multi-larva case with corresponding experiments and results. As for the data collected by these two proposed platforms, Chapter 5 provides the automated inspection pipeline with verification experiments as well. Based on proposed data acquisition and inspection platforms, Chapter 6 provides the details of pattern

recognition strategies based on statistical methods and machine learning and applies the pipeline to the experiments on drug screening for the analysis and prediction of the drug effect patterns shown by the touch-response behaviors. The conclusion is drawn in Chapter 7.

2 Conceptualization and parameters

The fully automated interface can be defined as process with automatic data acquisition, inspection, and analysis. The components (parts) of the process are necessary to be conceptualized and each part is designed, adapted and optimized based on manual and robotic optimizers. The proposed template process is applied to touch-response interface, and corresponding parameters are defined. The essential parts of touch-response interface and possible solutions to each part are introduced and compared. Besides, the metrics are defined for the evaluation of the touch-response interface.

2.1 General process concept

2.1.1 Process design

The tasks in a process can be explained by an exemplary pharmaceutical production process that combines both the industrial and academic worlds. Fig. 2.1 gives the general flow of producing a medical drug. After initial *research*, the chemists or pharmacologists propose the *protocol* to formulate the procedure and requirements used in production process of the medical drug, and design the process accordingly. The *protocol* is implemented into the *production* line, where the *medicine* products are obtained by the corresponding reaction tanks designed by the factory and mechanical engineers. Some samples of the *medical drug* are used for *testing* to derive control *variables*. Based on those variables, *analysis* is done with the help of data analysis software. The analysis results are used for the later-on *optimization* of the process or reported to the potential users and customers. According to the process described above, the main character of modern processes is that they are strongly interdisciplinary, i.e., the pharmaceutical production process in Fig.2.1 requires technologies and contributions from chemists, pharmacologists, factory designers, mechanical engineers, software developers, etc. Therefore, the automation is vital to the process.

Obviously, there is a need for a common architecture for different types of processes. In the industrial or academic world, the design of a whole process or task involves many questions,

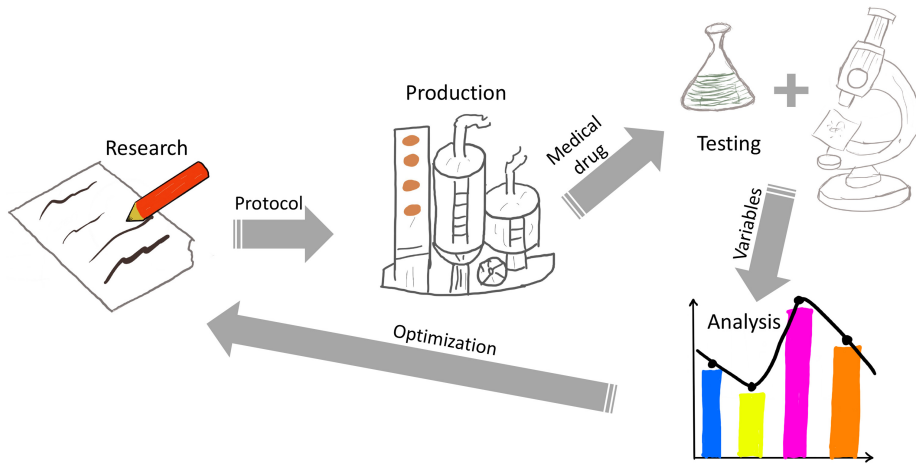


Figure 2.1: The pharmaceutical production process.

including design of the constructional blocks in the process, definition of the parameters used for the process, and decoupling the conflicts raised in the design of the process as well as the solutions to the conflicts. A constructional diagram in Fig 2.2 illustrates the general blocks (components) that can be considered for the design of a process: *acquisition*, *inspection*, *analysis*, *domain expert*, and corresponding optimization. The **user inputs** specify the goal of the whole process, like the requirements or existing open questions. The **initial parameters** come together as the inputs for the *acquisition* of the **data** where the actual task or action occurs (i.e., the actual machines or devices are running). The raw **data** from the machines is mostly hard to interpret, so it is *inspected* into **variables** that are intuitive to humans. In order to generate the conclusions for the process, a statistical *analysis* of the **variables** is essential, and the resulting **readouts** are combined with the *domain expert*¹ to draw the **conclusions**. In order to improve the process, the optimization of all parameters according to the evaluation criteria used in all blocks is required to improve the performance of the process.

Conventionally, as shown in the blue dashed box of Fig. 2.2, the optimization is done with a *visual assessment* of the **process parameters** as well as the **configurable parameters** used in the *inspection* and *analysis*. The parameters used in the *acquisition* procedure, e.g., production rate, are defined as **process parameters** that can be adjusted to output the desired **data** outputs. The parameters used in the *inspection* and *analysis*, e.g., the rate of samples used for testing, are defined as **configurable parameters** that can be adjusted to successfully fulfill the *inspection* and *analysis*

¹ The final outputs of the processes still need the prior knowledge from the expert in each domain. For example, in the chemical reaction process, the researchers need to draw the experimental conclusions based on their chemical knowledge and experience.

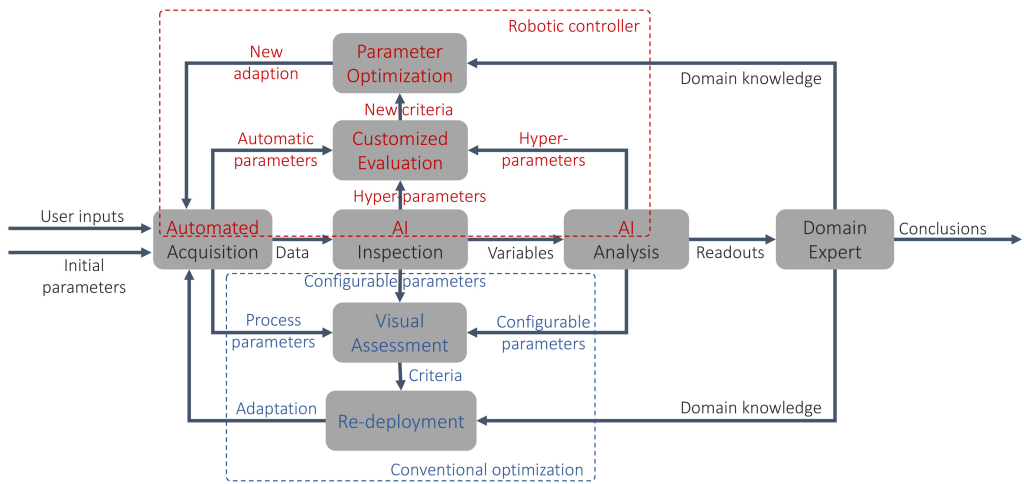


Figure 2.2: The general blocks for the design of a process with the conventional optimization by manually as well as the new optimization based on a robotic controller.

of the output **data**. Those parameters mentioned above are evaluated by *visual assessment* to form into **criteria** that are commonly used to *re-deploy* the process via potential **adaptation**. During the *re-deployment*, **domain knowledge** is essential as it guides the optimization by the *domain expert*. The *conventional optimization* can only improve the process in a limited level, as merely simple and basic parameters can be *assessed visually* without an optimal optimization procedure.

By comparison, as shown in the red dashed box of Fig. 2.2, the *robotic controller* based optimization considers intelligent methods to achieve *automated acquisition*, *AI-based inspection* and *analysis*. Apart from the parameters mentioned in the *conventional optimization*, new methods contain new parameters, like **automatic parameters** defined in the *automated acquisition* as well as **hyper-parameters** defined in the *AI-based inspection* and *analysis*. The **automatic parameters** are used to control the process conditions for the automated acquisition of data, e.g., the movement (positions and velocities) of the acquisition devices and the parameters of the methods applied to the acquisition procedure. In the AI methods applied to *inspection* and *analysis* tasks, the **hyper-parameters** are used to control the learning of the AI models, e.g., the learning rate and training epochs used in the deep learning methods or the size of neural networks. In the *robotic controller*, the concept of automation is not only considered in the *automated acquisition*, but the *AI-based inspection* and *analysis* are also fulfilled automatically. Additionally, the challenge of adjusting **automatic parameters** and **hyper-parameters** is to evaluate the performance of the process given the chosen parameters. Therefore, a *customized evaluation* for the *robotic*

controller is developed for assessing those parameters and **new criteria** are generated accordingly. Based on the **new criteria**, the *parameter optimization* is done with help of the additional **domain knowledge** as well. This optimization needs to output the **new adaption** that can be used to optimize the corresponding blocks of the automated process.

In simple and less challenging processes, some blocks (*acquisition, inspection, or analysis*) have been accomplished by the *robotic controller* to reduce the efforts required in the *conventional optimization*. However, in complex scenarios, the processes are usually too high-tech-contained to be optimized. Those special tasks are normally still taken manually by human beings or semi-automatically, which is undoubtedly not optimal. Therefore, according to the design of the *robotic controller* in Fig. 2.2, the modern advances can be adapted or inserted into the *robotic controller* to run the whole process in a fully automated manner. Based on the proposed common architecture of the template process, the blocks shown in Fig. 2.2 can be designed and adapted according to the application case.

2.1.2 Evaluation and optimization

The parameters of the methods used in acquisition, inspection, and analysis are the core of the optimization. As the process can be fulfilled in manual and automated manners as mentioned in Section 2.1.1, the optimizers also have two different potential styles, i.e., the manual and robotic optimizers.

The manual optimizer is normally based on the visual assessment of the parameters in the process, as visualized in Fig. 2.2. In addition, Fig. 2.3 shows the way the operators visually observe and control the quality of products (from the screen of the data by the sensors) by comparing and analyzing the inspection figure and the reference value. Also, the operators evaluate the inspection results and consider (e.g., by brainstorming) adjusting the input parameters of the process to output the desired products.

In terms of the robotic optimizer, the automatic parameters in the automation technologies and hyper-parameters in the AI methods are the core part to optimize. According to the architecture in Fig. 1.1, the robot is designed to manipulate the process by actuators and monitor the status of the process by sensors. Fig. 2.4 visualizes one potential example to insert the robot in the process for an automated manner. The screen displays the AI inspection results from the data generated by sensors. As for the optimization, the analysis and evaluation are implemented as an "AI box" to do the evaluation of the parameters according to the customized evaluation metrics and criteria. The evaluation results are used to adjust those parameters and fed back to the robot-assisted process.

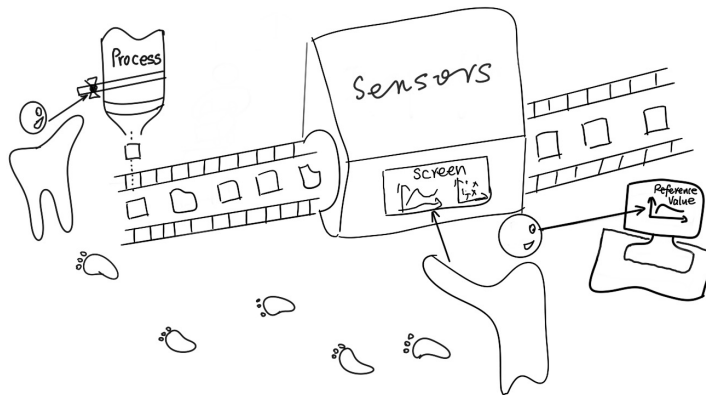


Figure 2.3: The manual optimizer. The operators adjust the process by visually comparing the sensor data and reference values.

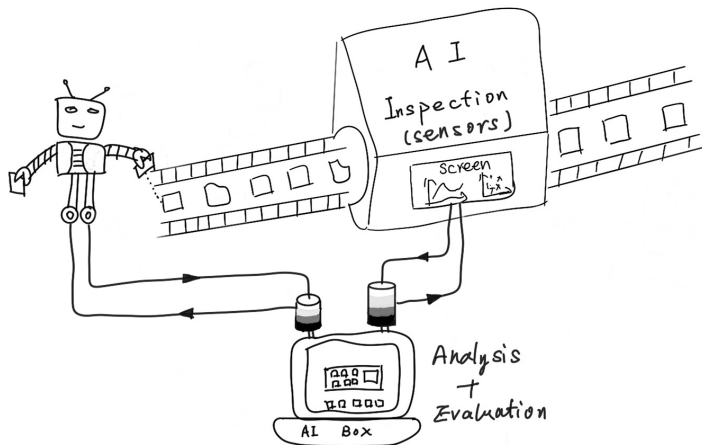


Figure 2.4: The robotic optimizer. The inspection, analysis, and evaluation are fulfilled given the sensor data automatically via AI methods and fed back to the robot to adjust the process.

2.2 Touch-screening process

2.2.1 Conceptualization

Mechanical stimuli, e.g., touch or startle input, can evoke the response (behaviors) of animals. Ahead of discussion of touch response on a specific animal model, the touch screening is conceptualized generally, which can be applied to various animal models, e.g., zebrafish, medaka, mouse, and *Caenorhabditis elegans* (*C. elegans*). The proposed concept for fully automated process in Fig. 2.2 can be adapted to touch-screening process as shown in Fig. 2.5. Given the **user inputs** and

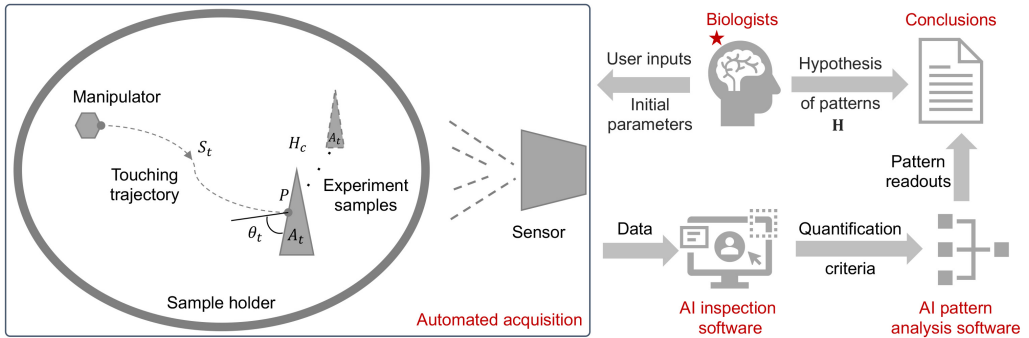


Figure 2.5: The illustration of touch-response screening, including biologists (starting point marked with a red "star"), automated acquisition, AI inspection software, AI pattern analysis software, and conclusions.

initial parameters from the *biologists* (marked with a red "star"), the touch-screening process is conducted as: at a given space (**sample holder**), a **manipulator** is controlled with a defined angle (θ_t) and speed (S_t) to touch a defined position (P) of the **experiment samples** (one or more, the number of samples denoted as H_c) at a defined age (A_t). In the *automated acquisition* procedure, the **touching trajectory** can be calculated with the help of image processing methods, i.e., the detection of **manipulator** and **experiment samples** as well as the segmentation of **experiment samples**. The behaviors of the **experiment samples** are captured by a **sensor** device, generating the touch-screening behavioral **data**. The *AI inspection software* is used to transfer the behavioral **data** into **quantification criteria**, followed by *AI pattern analysis software* that generates **pattern readouts**, i.e., the visualization of the behavioral data patterns. The **hypothesis of data patterns** (denoted as H) given by the *biologists*, i.e., the prior behavioral data patterns based on the biological knowledge, is compared with the **pattern readouts** to generate the **conclusions**. The most essential components of a complete touch screening include: *biologists*, *automated acquisition* (example sample, manipulator, touching strategy with a trajectory, and sensor), *AI inspection* and *pattern analysis softwares*, and **conclusions**.

The involved parameters of touch-response screening include input parameters and readout parameters. The input parameters ($\theta_t, S_t, P, H_c, A_t$) are used to control the experimental conditions during the automated acquisition. The readout parameters, i.e., quantification criteria and pattern readouts, are used to quantify the touch response and analyze the behavioral data patterns. The details are outlined as follows:

- Touching angle θ_t : One important parameter in terms of the touch-evoked response is the angle of touching θ_t . It is defined as the angle in which the manipulator approaches the samples.

- Touching speed (force) S_t : The evoked cells or neurons of the alive samples by the touch can be altered in a different touching speeds. A too gentle force does not evoke the response and a too strong one might harm the animals. Approximately, the touching force is monotonically related to the speed of the manipulator. Therefore, the touching speed S_t is defined as the optimal speed of the needle to conduct the touching task.
- Touching position P : The position of the sample where the touching is applied is denoted as P . Touching inputs at different positions can generate different reactions of the samples.
- Capacity of sample holder H_c : As the touch-response experiments are designed in high-throughput screening manner, single sample and multiple samples can be considered in the experiments. As for the multi-sample case, in order to collect more data as well as to minimize the interference between the samples, the capacity of sample holder H_c is defined as the maximum number of samples in the given sample holder.
- Age of experiment sample A_t : The samples can show different shapes and behaviors during the development, thereby reacting diversely with the mechanical stimuli.
- The quantification criteria of the evoked behaviors of the samples: The behaviors of the samples can be captured in different ways according to the types of the sensors, like time-series data, images, videos, etc. Those data need to be inspected by the AI inspection software regarding predefined parameters (quantification criteria) to vividly display the behaviors of the samples evoked by touch input.
- The pattern readouts of touch-response behaviors: Given different experimental conditions defined by the input parameters, experimental samples show various touch-response behaviors. The data pattern readouts can be generated by the AI pattern analysis software based on the quantification criteria. The readouts can be classification results or a tree structure by clustering, which depends on the specific aim of application case of touch screening.

The touch-screening concept can be applied to different alive animals or even tissues, and more involved readout parameters can be defined accordingly.

2.2.2 Touch screening on zebrafish larvae

Experiments on zebrafish have been considered in the behavioral and drug screening. Particularly, the touch-response behavior has emerged to help in investigating the mechanisms underlying locomotion [169], learning and memory [180], as well as in identifying mutations [172]. Zebrafish start responding to touching early (21 hpf, Section 1.3.3) when they are still inside the chorion (egg shell [181]) by demonstrating vigorous body coiling.

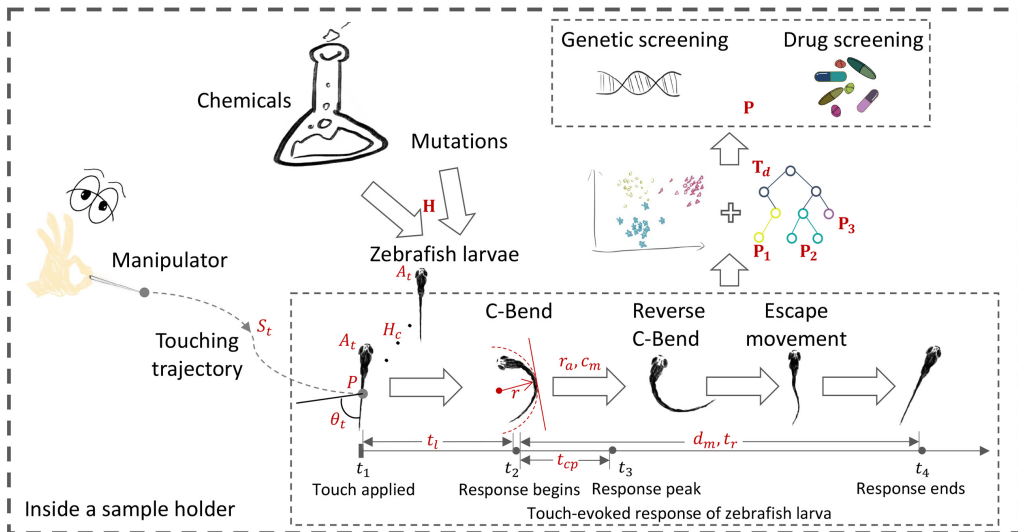


Figure 2.6: The schematic demonstration of the touch-response experiment on zebrafish larvae by manually [182].

The touch screening evokes a response to a touching input on zebrafish larvae and aims to quantify their behaviors, as this behavior delivers the information for the functioning of the locomotors of the larvae. Before 4 dpf, as shown in Table 1.1, zebrafish larvae do not exhibit spontaneous locomotion and there is no movement before the application of tactile stimuli or with minor mechanical stimuli. This guarantees that touch-response experiment does not need to consider distinguishing the spontaneous and touch-evoked movements, simplifying the later-on inspection and analysis procedure. However, if subjected to touching the zebrafish shows an escape response. Fig. 2.6 derives from the *automated acquisition* of Fig. 2.5 and visualizes the workflow of touch screening on zebrafish larvae with two typical applications: the screening of **chemicals** and **mutations**. The zebrafish larva treated by **chemicals** or having genetic **mutations** is touched by the **manipulator** according to the **touching trajectory**, causing **touch-evoked response of zebrafish larva**. The responses show different patterns under different experimental conditions (i.e., varying **chemicals** or **mutations**), which are analyzed to do the **genetic and drug screening**. The **touch-evoked response of zebrafish larva** consists of three components [181]:

- **C-Bend:** The larva bends the body as a C-Shape, shortened as **C-Bend**,
- **Reverse C-Bend:** The larva reverses the body and displays **C-Bends** continuously, and
- **Escape movement:** The larva escapes from the position where the input was applied. When the response stops, the larva keeps immobile.

Zebrafish larvae are usually dechorionated at 27 hpf and the touch screening needs to be conducted within 3 dpf (Section 1.3.3). Furthermore, the movements of the larvae are essentially restricted to two dimensions (2D)². This response is highly repeatable and distinguishable, namely the response occurs similarly to larvae under the same condition and is altered significantly in larvae exposed to neuroactive or neurodegenerative substances and in larvae that have sensory circuit and/or muscular mutations [183]. Consequently, understanding the touch-response behavior promises help in revealing the functioning of sensory and motor circuits along with the functioning of the neural architecture.

Apart from the input parameters (θ_t , S_t , P , H_c , A_t) introduced in Section 2.2.1, the readout parameters customized for the touch-response screening of zebrafish model are also to be defined. As described in Fig. 1.4, the dechoriation needs to be considered for a long-term chemical treatment in some experimental design. The effects of dechoriation (denoted as D_e) are to be tested. Furthermore, many unanswered questions can be revealed by the locomotor behaviors of zebrafish larvae during response to the touch stimuli, which needs collaborative work of a variety of cells or neuron types. The quantification criteria are the key to the behaviors displayed in the touch screening, and the behavioral data pattern readouts are to be generated for genetic or drug screening.

Quantification criteria Throughout the experiment, as shown in the touch-evoked response of zebrafish larva in Fig. 2.6, four time points of importance are

- the time at which needle stops and touch is applied (t_1),
- the time at which response begins (t_2),
- the time when the larva has the response amplitude peak (t_3), and
- the time at which response stops (t_4).

In total, six parameters (denoted as a vector \mathbf{Q}) are described to quantify the larva's behaviors to display the latency, amplitude, and duration of this response:

- Latency (latency time t_l): The larva takes some time to respond after being touched ($t_l = t_2 - t_1$), and this time indicates the reaction of the larva to the stimuli. This reaction can be delayed or intensified under different experimental conditions, e.g., varying ages of larvae or treatments of varying chemicals.

² Zebrafish larvae normally start swimming in three dimensions (3D) after 6 days post-fertilizations.

- Amplitude (C-Bend radius average r_a , C-Bend curvature maximum c_m , C-Bend peak time t_{cp}): During the response, the radius of the C-Bends r ($r > 0$, shown at time point t_2 in Fig. 2.6) during the response is also quantified. The larva shapes into a C-Bend [184] between t_2 and t_4 , and the amplitude of it indicates the strength of the larva's response, which can be denoted by the radius of the C-Bend r . The average radius of the C-Bends ($r_a = \text{ave}\{r_i\}_{i \in \Omega}$, where Ω indicates the whole response) is calculated. Apart from r_a , for better quantification of the response strength, the maximum of the C-Bend curvature that the larvae shape into (C-Bend curvature maximum, $c_m = \text{max}\{1/r_i\}_{i \in \Omega}$) and the C-Bend peak time ($t_{cp} = t_3 - t_2$) are also calculated.
- Duration (response time t_r and moving distance d_m): The response time ($t_r = t_4 - t_2$) and total distance of the larva moving (d_m) illuminate the duration of the touch response, which also differ under various stimuli [185, 186] and chemicals.

Pattern readouts As shown in Fig. 2.6, the touch screening can be applied to genetic screening, drug screening, etc. In detail, the chemicals or mutations are applied to zebrafish larvae, resulting in different touch-response behavioral patterns. With the hypothesis of data patterns (**H**) from biologists, the behaviors can be displayed as a data pattern dendrogram \mathbf{T}_d , i.e., the behavioral data pattern in a tree structure. The goal of genetic and drug screening based on touch response is to predict the data pattern for the new unknown chemicals and mutations (denoted as **P**). Therefore, two essential parameters involved in touch-response behavioral pattern analysis include:

- Data pattern dendrogram \mathbf{T}_d : The behavioral data of zebrafish under different input experimental conditions can be clustered or classified by the modern statistical or machine learning methods (Section 1.2.3) into a tree structure, namely \mathbf{T}_d . Besides, the data patterns in \mathbf{T}_d can be named according to specific experimental samples with the help of **H**, i.e., the names (reference) of the drug patterns or mutants provided by biologists.
- Predictions of data pattern **P**: According to the touch screening results (\mathbf{T}_d) of known genes and chemicals, the patterns of new unknown genes and chemicals can be predicted by a list of proposals of data patterns, i.e., $\mathbf{P} = \{\mathbf{P}_1, \mathbf{P}_2, \dots\}$, where $\mathbf{P}_1, \mathbf{P}_2, \dots$ are the pattern names in \mathbf{T}_d . This list hints at potential functioning mechanisms of new chemicals and mutations.

To summarize, these six quantification criteria $\mathbf{Q} = \{t_l, r_a, c_m, t_{cp}, d_m, t_r\}$ and two pattern readouts \mathbf{T}_d, \mathbf{P} will be used to construct a working model that represents the response pattern of zebrafish to touching or other mechanical stimuli.

2.3 Full automation interface

2.3.1 Requirements

The general concept of full automation conceptualized in Section 2.1 can be applied to the touch-response automation interface on zebrafish larvae. The interface ensures a high-throughput experimental pipeline, which provides possibilities of building a statistical behavioral model for the zebrafish. The high-throughput screening of touch-evoked behaviors is also ideal for the drug screening task, from which the pharmaceutical world can benefit for the discovery and production of new drugs. Various possibilities of the components of the touch screening are to be discussed, including sensors, sample holders, manipulators, softwares, as well as the strategies for touching, as displayed in Fig. 2.5. Besides, the robotic system used of the interface can be implemented by the methods in statistics, machine learning, and compute vision, as described in Fig. 1.1. The potential methods in those fields are also discussed. The requirements of designing a full automation interface for the touch-response experimental process on zebrafish larva can be identified according to the components of the touch-response screening described in Section 2.2.1:

- I. The larvae is in 3.5-3.7 mm, so the holder of the sample is to be large enough to provide the space for the locomotion movements,
- II. The observation of behaviors at the edge of sample holder needs to be avoided, since the optical effects between the edge of the well plate and the fish water make the imaging difficult,
- III. The larvae move with high velocities (200-300 mm/s, as introduced in Section 1.3.3), so the sensor needs to work fast enough to capture the behaviors,
- IV. The capturing of the videos and the conduction of the experiments need to be done at the same time,
- V. The experimental process needs be able to automatically localize the instruments and target zebrafish larva, and generate the touching strategy for touching,
- VI. The data generated in the process is required to be possible to do automatic analysis, and
- VII. The software packages used to implement the interface are required to be easily extended to functions of the real-time and offline data processing.

2.3.2 Sensor

The choice of sensors has two goals: low costs and enough resolution and frame rate to capture the touch-response behaviors of zebrafish larvae. The possible sensor devices that can be used in the observation of the behaviors of the zebrafish model, as listed in Table 2.1, can be divided into three categories according to the observed samples, i.e., low-speed camera (e.g., Canon EOS 600D and Panasonic WV-BP330), microscope (e.g., SP5 Leica confocal microscope), and high-speed camera (e.g., MotionScope 2000S and Ximea 1.3 MP Mono NIR Enhanced Camera MQ013RG-E2).








Table 2.1: The possible sensors for capturing the animal behaviors.

Sensors	Photo	Lens	Resolution & frames per second (fps)	Price	Reference & literature
Canon EOS 600D		EF-S 55-250 mm, Canon	640 × 480 30 fps	approx. 500 €	Ref. [187]
Panasonic WV-BP330		S6E stereomicroscope	752 × 582 30 fps	approx. 1,500 €	Ref. [172]
SP5 Leica confocal microscope		HC PLAN APO, PH2 0.17/C 0.59	512 × 512 250 fps	approx. 60,000 €	Ref. [152]
MotionScope 2000S		MZ6 dissecting microscope	480 × 420 2000 fps	approx. 3,000 €	Ref. [157]
Ximea 1.3 MP Mono NIR Enhanced Camera MQ013RG-E2		Lensation CMFA1520ND	480 × 480 1000 fps	approx. 2,500 €	Ref. [158]

The low-speed camera is commonly used to observe the escape behaviors of zebrafish in a long-term manner [172, 187], which works in a low frame rate (30 frames per second, fps). The

microscope can observe changes or movements of cells or tissues [152, 157], which commonly works together with fluorescence to study the organic or inorganic substances of animal model. The high-speed camera is chosen to capture the details of the fast response of the zebrafish [157, 158], which usually works in 1000 to 2000 fps. The frame rate of each imaging devices can also be changed according to the resolution, namely the higher the resolution, the lower the frame rate. Additionally, the choice of the sensors also involves the costs. The low-speed cameras cost the least, approximately 500 to 1,500 €, and the microscope is commonly more expensive, approximately 60,000 €. The high-speed cameras cost higher than low-speed cameras, approximately 2,500 to 3,000 €. Therefore, the trade-off between the performance (resolution and frame rate) and the cost is important. As the speed of the zebrafish larva moving is in millisecond scale, the high-speed cameras are optimal. The choice of sensors depends on the actual design of the interface, which will be discussed in the system design.

Table 2.2: The possible sampler holders.

Holders	Type	Photos	H_{id} (mm)	H_{od} (mm)	H_h (mm)
24-well plate	Transparent		15.5	16.2	19.0
	Black [188]		14.0	14.0	19.0
12-well plate	Transparent		22.05	22.9	18.95
	Black		22.05	22.9	18.95
6-well plate	Transparent		34.94	35.77	19.0
	Black [189]		34.94	35.77	19.0
Petri dish	-		-	94.0	16.0

2.3.3 Sample holder

In terms of sample holders, a few requirements are considered:

- The larvae need enough space to display their behaviors (response),
- The throughput needs to be guaranteed, i.e., the sample holder needs to hold enough number of larvae, and

- The sample holder needs to be easy for imaging.

The possible sample holders are outlined in Table 2.2. The size of the sample holder can be described by three parameters, i.e., Inner Diameter as H_{id} , Outer Diameter as H_{od} , and Height as H_h . For the consideration of the size of the zebrafish larvae, four different sizes of the sample holders can be possibly chosen: 24-well plate, 12-well plate, 6-well plate, and Petri dish with corresponding H_{od} , H_{id} , and H_h listed in Table 2.2. For the consideration of light effects (illumination, reflection, refraction, etc), two types of well plates can be possibly compared: transparent and black. The Petri dish is commonly used in experiments with groups of zebrafish larvae.

Table 2.3: The possible sample holders under different conditions used in touch-response experiment. "+" indicates the image quality is good or the larva is observable, and "-" indicates the image quality is not good or the larvae is not observable.

Conditions	Image quality	Larva observable
Black well with low brightness	+	-
Black well with high brightness	-	-
Transparent well with low brightness	+	-
Transparent well with high brightness	-	-
Transparent well with agarose rings	+	+
Transparent well with PTFE rings	+	+

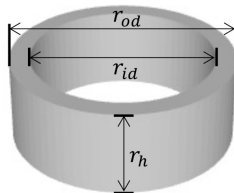


Figure 2.7: The dimensions of the shadow-covering rings: outer diameter as r_{od} , inner diameter as r_{id} , and height as r_h .

In terms of the light effects, especially the optical effects at the edge of sample holder (mentioned in Section 2.3.1), two possible types of the shadow-covering rings can be installed at the edge of the sample holder to ensure that the larva only stays within the view of the camera (the center of the well), i.e., agarose rings and re-usable 3D printed rings (Material: Polytetrafluoroethylene, PTFE). Thus, the rings need to be put under the water surface, otherwise optical effects between the water and the edge of the rings can disturb the imaging. The size of the rings can be formulated

with outer diameter as r_{od} , inner diameter as r_{id} , and height as r_h , as visualized in Fig. 2.7. As the shadow-covering rings are put under the fish water, the larvae have possibilities to move over the rings after touching (especially in the multi-larva case), causing the response unseen. Thus, the height of the ring r_h is an important parameter to be determined.

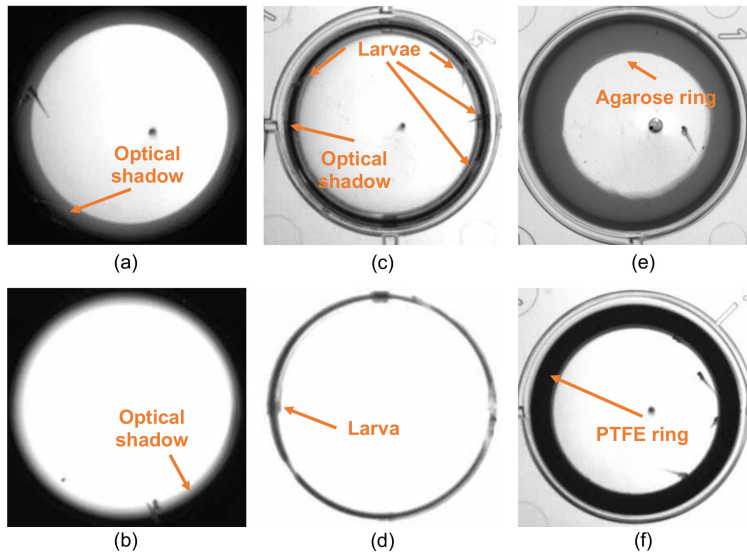


Figure 2.8: The comparison between possibilities of sample holders (well plates). (a) Black well plate with low brightness. (b) Black well plate with high brightness. (c) Transparent well plate with low brightness. (d) Transparent well plate with high brightness. (e) Transparent well plate with installed agarose rings. (f) Transparent well plate with installed PTFE rings.

For the choice of the right sample holder, different holders under various conditions are compared as visualized in Table 2.3 and Fig. 2.8. The brightness of the image captured by the camera can be adjusted by varying the exposure time of the camera. The black and transparent well plates with and without rings are used to capture images with high and low brightnesses, and the details are summarized as:

- Black well plate with low brightness: As visualized in Fig. 2.8a, the image quality is high (+), but the larva is not observable (-) as it easily hides in the optical shadow,
- Black well with high brightness: As visualized in Fig. 2.8b, the image quality is bad (-), and the larva is not observable (-) as it easily hides in the optical shadow,
- Transparent well with low brightness: As visualized in Fig. 2.8c, the image quality is high (+), but the larva is not observable (-) as it easily hides in the optical shadow,

- Transparent well with high brightness: As visualized in Fig. 2.8d, the image quality is bad (-), and the larva is not observable (-) as it easily hides in the optical shadow,
- Transparent well with agarose rings: As visualized in Fig. 2.8e, the image quality is high (+), and the larva is observable (+), and
- Transparent well with PTFE rings: As visualized in Fig. 2.8f, the image quality is high (+), and the larva is observable (+).

Two options of sample holders, i.e., "Transparent well with agarose rings" and "Transparent well with PTFE rings" can be chosen with good image quality (+) and larva observable (+), which will be discussed in the actual system design.

2.3.4 Manipulator

The choice of manipulators follows three requirements:

- The size of the manipulator needs to be comparable to the size of zebrafish larvae,
- The manipulator cannot harm the larvae, and
- The manipulator is able to evoke enough force for touching.

Zebrafish show different sizes and behaviors during the development, so possible manipulators for applying the mechanical stimuli on the animals can be various, e.g., fluidic force microscopy (FluidFM) in nanoscale (Fig. 2.9a), glass pipette in microscale (Fig. 2.9b), and blunt needle in milliscale (Fig. 2.9c). The FluidFM, i.e., an atomic force microscope with a microfluidic probe [190], can undertake the tasks of injecting the cells and analyzing the stiffness of the cell surface (the interaction between the cell and the FluidFM tip). The glass pipette is ideal to inject the tissues of animals and apply the touch input to animal embryo. However, it has disadvantages of damaging the embryos. The blunt needle can work to apply the touch stimuli on the animal larva in several millimeters, e.g., the zebrafish larvae in 3.5-3.7 mm. The integration of the manipulator to the other devices is also important in terms of the design of an automation interface. The FluidFM is usually provided as "black box", so that it is only possible to re-develop basic tasks based on initially integrated functions. The glass pipette is easily broken, so the safety of the tip is to be considered in the design. Therefore, the blunt needle is the optimal manipulator for touch screening on zebrafish larvae.

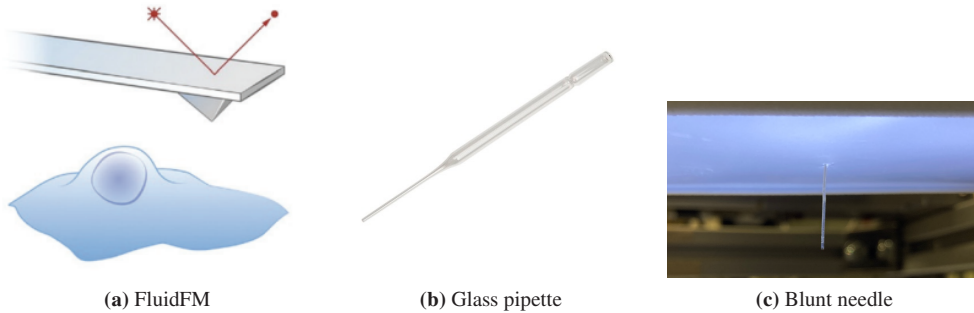


Figure 2.9: Possible manipulators used in the design of the touch-response interface. (a) FluidFM [190]: an atomic force microscopy (AFM) in microfluidic environment. The size of FluidFM is usually from hundreds of nanometers to several micrometers. (b) Glass pipette [191]: size from tens to hundreds of micrometers. (c) Blunt needle: size over hundreds of micrometers.

2.3.5 Software package

The objective is to choose a software package with following advantages:

- implementation of current advances and reprogrammable to new functions,
- implementation in the desired programming languages.

The development of machine learning, computer vision, and statistics have promoted the processing of data in bio-medical research. A great amount of softwares and libraries are being proposed to make the image and data processing more user-friendly, easy-to-operate, highly accurate, etc. Table 2.4 outlines the typical software tools that are potentially helpful for the design of touch-screening interface, including Python Imaging Library (PIL) [192], OpenCV [193], Tensorflow [194, 195], PyTorch [196], Scikit-learn [197], and SciPy [198]. Both PIL and OpenCV contain image processing algorithms, but PIL is mainly implemented in Python with only basic image processing functions. OpenCV is implemented in different languages, i.e., C++, C#, Python, and Java, and it contains most of recent advances in computer vision. Tensorflow is used for the general research in deep learning with the supportive and friendly APIs to implement nearly all deep learning methods in C++, C#, Python, Java, etc. Particularly, the trained model by Tensorflow can be easily or even directly applied to other platforms owing to its unique model format. PyTorch is a package mainly based on Python, but possible to be extended to C/C++ with the help of an API. It has been frequently used in the state of the arts in the research of deep learning and makes it simple to implement complex methods. Scikit-learn is Python based tool with implementation of recent machine learning algorithms. It provides a friendly high-level interface for users to adjust

Table 2.4: The possible software tools that can be applied to the design of automation interface.

Packages	Languages	Functions	Literature
Python Imaging Library (PIL)	Python	- Image reading, displaying, and saving, - Basic image processing functionality.	[192]
OpenCV	C++, C#, Python, Java, etc.	- Different image processing tasks, - Well-established interface and parameter setting.	[193]
Tensorflow	C++, C#, Python, Java, etc.	- Friendly APIs, - Easily adaptable to nearly all deep learning methods, and - Models easily applied to different platforms without further modification.	[194, 195]
PyTorch	Python with extension API to C/C++.	- High simplicity and ease of use, - Implementation of basic elements of machine learning and deep learning methods.	[196]
Scikit-learn	Python	- Implementation of advances in machine learning methods, - High-level interface with easily setting of hyper-parameters for fine-tuning machine learning.	[197]
SciPy	Python	- Implementation of methods in different scientific problems with numeric functionality, - Diverse algorithms, including mathematical modelling, signal/image processing, machine learning, statistics, etc.	[198]

the hyper-parameters to fine-tune the performance of the methods. Scipy, i.e., science in Python, is designed for different scientific problems, e.g., mathematics, signal/image processing, machine learning, and statistics. Those software packages discussed above are reprogrammable, i.e., easily extended to new functions. The choices of the packages described above will be discussed in the actual system design.

2.3.6 Data processing methods

The data processing methods undertake the tasks of real-time (e.g., the detection and segmentation of the zebrafish larvae and manipulator) and offline data processing (e.g., the implementation of the behavioral inspection and pattern analysis softwares). The choices of them obey two goals:

- able to fulfill the tasks,
- possible to be implemented by the software packages discussed in Section 2.3.5.

First, the detection and segmentation of the zebrafish larvae and manipulator can be done by the segmentation methods. A few possible algorithms are to be compared to select the optimal one for the segmentation task:

- Binarization with a single threshold: The threshold based binarization (shortened as Thre) is obtained by a heuristic threshold (T_{Thre}).
- Otsu's method: Different from the single fixed threshold based binarization, Otsu's method selects the threshold automatically according to the histogram of the whole image.
- Linear regression based binarization: Linear regression based binarization (named as LRb) is a binary classification method for each proposed image patch. The image patches are simply proposed one by one at each position inside the well area, and the linear regression model is trained by the image patches with binary labels (object or background). The pre-trained model is used to compute the probability of being the object (larva or needle) for each image patch at each position inside the well area. Based on the probability at each position, a threshold is set for binarization (T_{LRb}).
- Region growing based binarization: The region growing based binarization method (named as RGb) is applied within the well area, searching for all potential objects based on the image gradient threshold (T_{RGb}). The seed point of the region growing is set as the center of the well area. The results can exceed the actual number of real objects, so the binary image from region growing is optimized by the morphological operators (erosion and dilation) to filter the small objects (noise).
- U-Net based binarization: U-Net [97] is a convolutional neural networks based method. A U-Net mainly consists of down-sampling blocks - two convolutional layers (Conv) and one max pooling (Max-pool) - for feature extraction and up-sampling blocks - one deconvolutional layer (deConv) and two convolutional layers, as visualized in Fig. 2.10. The output of U-Net is designed as binary images of the needle and larvae given the input grayscale image. A high-dimensional feature map is extracted with down-sampling blocks

and then re-sampled back to the binarization (larvae and needle separately) in the size of the original image with up-sampling blocks. Similarly to LRb, a threshold is also set for binarization (T_{UNet}) based on the output probability at each position of the image.

Those methods can be implemented by the software packages in Section 2.3.5, and will be chosen according to their performance on the segmentation task based on the evaluation metrics.

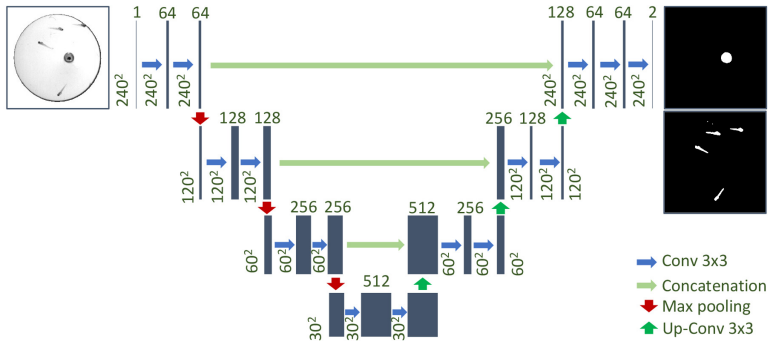


Figure 2.10: The structure of U-Net based segmentation [97, 199]. Blue arrows indicate layers of convolutional neural networks with 3×3 kernel, denoted as Conv 3×3 . Cyan arrows indicate the concatenation of the results from the input layers. Red arrows indicate the down-sampling layer, i.e., max pooling. Green arrows indicate up-sampling convolutional neural networks with 3×3 kernel, denoted as Up-Conv 3×3 .

Second, many tracking procedures can be used in the offline behavioral inspection software, e.g., optical flow, Kalman filter, and particle filter, as follows:

- Optical flow is defined as the pattern of brightness changes caused by the movements of objects [200]. In the tracking task, optical flow assumes that the object does not move rapidly, and that the neighboring pixels around the object have the same motions [99, 100]. It estimates the motion of the object via the difference (gradient) of neighboring pixels in an adjacent area. Therefore, the optical flow is sensitive to the size of the adjacent area.
- Kalman filter can be defined as mathematical transformations (i.e., matrix) that estimate the past, present, and even future states of a system [201]. In terms of application to tracking task, Kalman filter assumes an initial state and the noise covariance matrix for calculating the motion vector of the object [202]. Therefore, Kalman filter can perform well if the noise is identified, e.g. Gaussian noise, but is sensitive to unknown types of noises [203].
- Particle filter is a probabilistic model that can be used in nonlinear and non-Gaussian scenarios [102]. In the tracking task, particle filter has no special constraints or assumptions. The particles are defined with a potential position and corresponding score of the position that is commonly obtained by image feature or similarity. It estimates the position of the

object by sampling a group of particles in the neighboring area of the object and computing the weighted average position of the particles. Therefore, the accuracy and computational costs of particle filter are sensitive to the number of particles and computation of scores.

Third, the machine learning methods in supervised or unsupervised manners can be explored to do the offline pattern analysis:

- Supervised learning works on the data source with each data sample labeled by the correct classification. The learning models perform well if the data has distinctive characteristics and the training and testing data are consistent. Supervised learning algorithms are sufficient for linear and non-linear learning tasks, e.g., classification, forecasting, or prediction [204].
- Unsupervised learning is a self-organizing model identifying hidden patterns in unlabeled data source [204]. Thus, the model performs well if the training and testing data have the same distribution and structures. Unsupervised learning is considered in clustering and reducing data dimensionality [205], e.g., representing the neurobiological behavioral information in a low dimension [204].

The choices of methods used for offline data processing will be discussed according to the data format in the actual system design.

2.3.7 Touching strategy

The touching strategies are the core of the automation interface to finish a touching task. As zebrafish have developed locomotion in the larval level, they are capable to move in a 2D space evoked by the external stimulus. Therefore, touch-response experiments of zebrafish can be conducted in two ways like the experiments on *C. elegans* were done:

1. touching the fixed larva while observing brain activity by using fluorescent reporters, [160, 173, 174, 206], or other tissues [207, 208],
2. touching any anatomical positions as well as observing the response of the larvae in the free space to understand the escape behavior itself and the kinematics [169].

According to the research topics, as outlined in Table 2.5, the possible strategies of touching vary in three aspects:

1. The larvae can be fixed (+) or self-positioning (-) for the research on the activities of the brain or some other related issues and locomotion behaviors of the zebrafish model

Table 2.5: The possible touching strategies.

Strategies	Larva number	Larvae fixed	Repeated touching	Research goal
I	Single	+	+	Individual behaviors
II	Single	+	-	
III	Single	-	+	
IV	Single	-	-	
V	Multiple	-	+	Interactive behaviors
VI	Multiple	-	-	
Research goal		+: brain activity; -: locomotion behaviors [180]	+: habitation [180] -: instant locomotion	

respectively [180]. Fixing multiple larvae together is the same as the single-larva case. Therefore, only the self-positioning (-) larvae in the multi-larva case are considered,

2. The larvae can be touched repeatedly (+) or only once (-) for the research on the habitation [180] and instant locomotion of the zebrafish model respectively, and
3. As the sample holder can hold different numbers of larvae, the experiments can be done in single-larva and multi-larva cases for the research of individual and interactive behaviors of the zebrafish respectively.

According to those aspects, six strategies of touching (Table 2.5) are commonly used in the touch-response screening:

- I. single fixed (+) larva with touching repeated (+),
- II. single fixed (+) larva with touching not repeated (-),
- III. single unfixed (-) larva with touching repeated (+),
- IV. single unfixed (-) larva with touching not repeated (-),
- V. multiple unfixed (-) larvae with touching repeated (+), and
- VI. multiple unfixed (-) larvae with touching not repeated (-).

In the design of the full automation interface for the touch response, the strategies also include the control and coordination of the movements of the manipulators, sensors, and image processing (softwares).

2.4 Evaluation metrics

2.4.1 Overview

The evaluation metrics take the role of assessing the performance of each component as described in Section 2.2. The evaluation serves the goal (expectations) of:

- high accuracy of the image processing,
- high effectiveness of the touching task,
- high effectiveness of the quantification criteria, and
- high accuracy of the pattern analysis.

The metric for each goal is outlined in Table 2.6 including:

- Evaluation of the image processing: center error e_{gd} , ratio of overlap R_o , intersection over union IoU , per-class accuracy PC_i , Jaccard index JI_i , ratio of recall R_r , ratio of precision R_p ,
- Evaluation of touching task: number of valid touching $\#VT$, touching efficiency t_{TPL} , average number of videos collected per well $\#VPW$, touching reproducibility error e_{tr} , touching robustness r_t ,
- Evaluation of inspection: number of videos with no larvae touched $\#NT$, number of videos with failure of quantification $\#QF$, and
- Evaluation of pattern analysis: number of pattern proposals $\#Npp$, percentage of failures of pattern predictions P_f .

2.4.2 Evaluation of image processing

The image processing method undertakes the main role in the implementation of the real-time data processing of the interface. In particular, in terms of the touch-response interface, the image processing is used for the detection of the zebrafish larvae and manipulator (e.g., the blunt needle) and segmentation of the larvae. The evaluation of image processing includes the performance of the detection and segmentation, as visualized in Fig. 2.11. The zebrafish larvae and manipulator in a sample holder are captured by a high-speed camera into an image, followed

Table 2.6: The evaluation metrics for the full automation interface of touch response.

Evaluation metric	Symbol	Comment
Center error	e_{gd}	Evaluation of image processing
Ratio of overlap	R_o	
Intersection over Union	IoU	
Per-Class accuracy	PC_i	
Jaccard Index	JI_i	
Ratio of recall	R_r	
Ratio of precision	R_p	
Number of valid touching	$\#VT$	Evaluation of touching task
Touching efficiency	t_{TPL}	
Average number of videos collected per well	$\#VPW$	
Touching reproducibility error	e_{tr}	
Touching robustness	r_t	
Number of videos with no larvae touched	$\#NT$	Evaluation of inspection
Number of videos with failure of quantification	$\#QF$	
Number of pattern proposals	$\#N_{pp}$	Evaluation of pattern analysis
Percentage of failures of pattern predictions	P_f	

by the developer annotating³ and image processing. The manual ground-truth labels from the developer are compared with the segmentation and detection results from the image processing via the evaluation metrics. Fig. 2.12 visualizes the detection and segmentation results of a zebrafish larva from developer annotating and image processing. The bounding box is defined as the minimum rectangle box that covers the area of the larva.

The center locations of the larvae or manipulator are one of the essential results to be evaluated in terms of detection methods. As for the larva detection, for example, the center error between

³ The software developer annotates the image manually to obtain ground-truth labels with the help of domain experts.

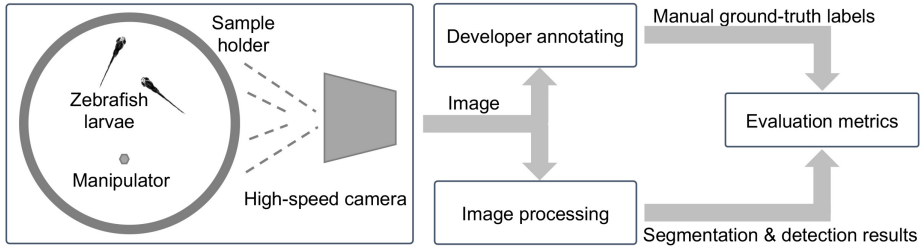


Figure 2.11: The workflow of evaluating the image processing, containing the comparison between the result from developer annotating and image processing for the image of zebrafish larvae.

manual ground-truth center (assumed as $\{g_x, g_y\}$, green dot in Fig. 2.12d) and detected center (assumed as $\{d_x, d_y\}$, red dot in Fig. 2.12d) is calculated as

$$e_{gd} = \sqrt{(g_x - d_x)^2 + (g_y - d_y)^2}. \quad (2.1)$$

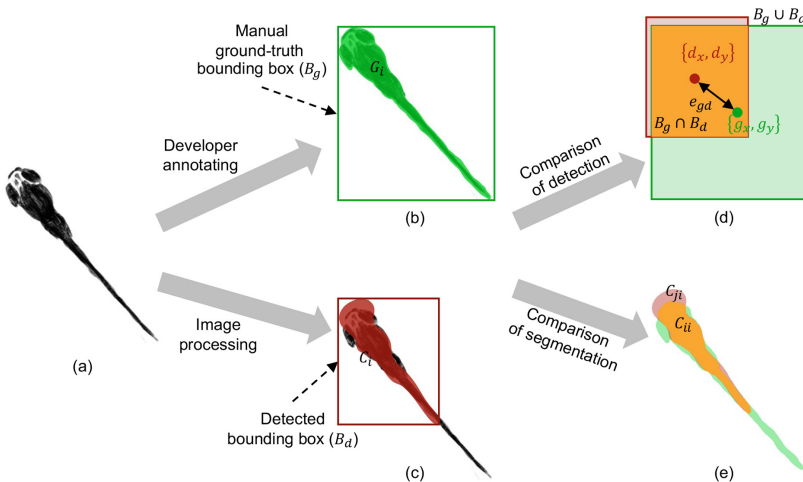


Figure 2.12: The illustration of the segmentation and bounding boxes from developer annotating and image processing. (a) The original image of zebrafish larva captured by the camera. (b) The manual ground-truth bounding box (B_g , green box) and segmentation area (G_i , in green) from developer annotating. i indicates classes, e.g., the larva, manipulator, or image background. (c) The detected bounding box (B_d , red box) and segmentation area (C_i , in red) from image processing. (d) The comparison of detection (i.e., B_g and B_d). The overlap area of B_g and B_d is in orange ($B_d \cap B_g$). (e) The comparison of segmentation (i.e., G_i and C_i). The overlap area of G_i and C_i is in orange (C_{ii}).

Additionally, the detected bounding box of the larva is also vital to localization of the touching position in the touch screening. Assumed the detected bounding box of the larva is B_d (red box

in Fig. 2.12d), its ratio of overlap (R_o) with manual ground-truth bounding box B_g (green box in Fig. 2.12d) is computed as

$$R_o = \frac{A\{B_d \cap B_g\}}{A\{B_g\}} \quad (2.2)$$

where $A\{*\}$ is the area, and $B_d \cap B_g$ is overlap of B_d and B_g (filled in orange in Fig. 2.12d). Accordingly, a threshold (T_o) of R_o is defined to determine whether the larva is successfully detected (recalled). R_o gives the “degree” of how the detection output covers the ground-truth bounding box. Thus, the threshold (T_o) of R_o is the minimum “degree” determining whether the detection output is successful.

Intersection over Union (IoU) is also commonly used for the evaluation of detection methods. The detected bounding box of the larva is B_d , so its IoU with manual ground-truth bounding box B_g is computed as

$$IoU = \frac{A\{B_d \cap B_g\}}{A\{B_d \cup B_g\}} \quad (2.3)$$

where $B_d \cup B_g$ is the union of B_d and B_g (red, orange, and green areas in Fig. 2.12d).

In terms of the segmentation for the larvae and the needle at the same time, the pixel-wise accuracy is one evaluation index of the multi-class segmentation methods, i.e., Per-Class accuracy (PC) [209]. The proportion of correctly segmented pixels for each class (e.g., the larva, needle, or background) is computed according to the manual ground-truth pixel labels. The PC for class i , i.e., PC_i , is computed as

$$PC_i = \frac{C_{ii}}{G_i} \quad (2.4)$$

where C_{ii} is the number of the pixels correctly segmented (the label of which is i with prediction being also i , orange area in Fig. 2.12e), and G_i is the number of the ground-truth pixels with label being i , green area in Fig. 2.12e. Another pixel-wise index is the Jaccard Index (JI) that is generated by the IoU between the segmented area for each class and the manual ground-truth area [209]. For class i , the equation of JI_i is

$$JI_i = \frac{C_{ii}}{G_i + C_{ji} - C_{ii}} \quad (2.5)$$

where C_{ji} is the number of the pixels falsely predicted (with label being j but predicted as i , red area in Fig. 2.12e).

The image processing methods are required to retrieve as many larvae as possible, so the ratio of recall (R_r) of the detection result is defined. Fig. 2.13 gives 5 examples of segmentation

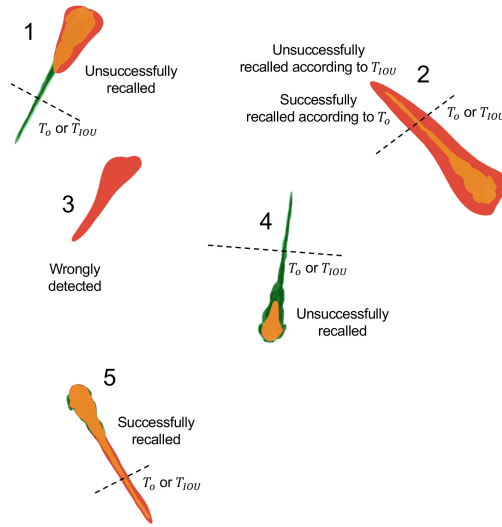


Figure 2.13: The illustration of successful and unsuccessful recalls of larvae according to T_o and T_{IOU} with 5 examples. Example "1": unsuccessfully recalled. Example "2": unsuccessfully recalled according to T_{IOU} but successfully recalled according to T_o . Example "3": wrongly detected. Example "4": unsuccessfully recalled. Example "5": successfully recalled.

(detection) results, where the manual ground-truth area is in green, the detected area by the image processing is in red, and the intersection of them is filled in orange. A successful recall of the object is defined that ratio of overlap (R_o) or Intersection over Union (IoU) is over a threshold, i.e., T_o and T_{IOU} (dash line in Fig. 2.13). For example, number "5" in Fig. 2.13 is successfully recalled as the intersection area is over T_o or T_{IOU} . However, as for number "2" in Fig. 2.13, the ratio of overlap R_o is high but IoU is low, generating two different results. Therefore, the ratio of recall based on both R_o and IoU are considered to better demonstrate the performance of the image processing. The ratio of recall with T_o is defined as

$$R_r = \left. \frac{N_d}{N_g} \right|_{T_o} \quad (2.6)$$

where N_d is the number of the successfully recalled larvae (e.g., larva "5" in Fig. 2.13), and N_g is the number of the ground-truth larvae that are expected to be detected e.g., totally 4 larvae to be detected (larvae "1", "2", "4", "5") in Fig. 2.13. The ratio of recall (R_r) with IoU is defined as

$$R_r = \left. \frac{N_d}{N_g} \right|_{T_{IOU}} . \quad (2.7)$$

Similarly to R_r , the ratio of precision is also defined based on T_o or T_{IOU} as

$$R_p = \frac{N_s}{N_s + N_w} \Big|_{T_o} \quad \text{or} \quad R_p = \frac{N_s}{N_s + N_w} \Big|_{T_{IOU}} \quad (2.8)$$

where N_w is the wrongly detected larvae (e.g., larvae "3" in Fig. 2.13), and $N_s + N_w$ is the total number of the detected larvae, e.g., 5 detected objects in Fig. 2.13. Apart from the metrics above, the segmentation of the larvae need to be evaluated on the performance of touching different positions of the larvae, with details in Section 2.4.3.

2.4.3 Evaluation of touching task

Touching is the main task of the designed interface, and evaluation metrics are required to evaluate the effectiveness of touching, including:

- Touching validation: The data with valid touching are defined as valid videos, in which the larvae are successfully touched after visual inspection. The number of valid touching is denoted as $\#VT$.
- Touching efficiency: The automation interface to conduct the touch-response experiments consists of different parts according to the touching strategy, and in each part, the average time used is important to estimate the efficiency of the data acquisition systems. Additionally, t_{TPL} indicates the average time used for touching each larva, which is the important evaluation metric for the strategies. As the aim of this automation interface is to generate as many data as possible in a high-throughput manner, the evaluation metric - average number of videos collected per well ($\#VPW$) - is also to be evaluated.
- Touching reproducibility and robustness: The touching reproducibility (e_{tr}) and robustness (r_t) refer to the capabilities of the automation interface generating the reliable and consistent touch application (positions) to the larvae in the repeated experiments. The touching is repeated with a predefined position (P), and mean μ and standard deviation s of the actual applied touching positions are calculated. The touching reproducibility (e_{tr}) is calculated as $|P - \mu|$, and s denotes the touching robustness (r_t).

2.4.4 Evaluation of inspection

The evaluation metrics for the inspection (quantification) software also need to be defined. The detected errors are to be analyzed, e.g., the inaccuracy caused by the segmentation method and missing objects in the processing of the video data by the quantification software. As well, the collected data contain some unquantifiable videos, such as the larvae were not touched (according to the definition of valid touching in Section 2.4.3), and the larvae or needle might not be detected. Thus, the analysis of detected errors is given by using the number of videos with no larvae touched ($\#NT$) as well as those with failure of quantification ($\#QF$).

2.4.5 Evaluation of pattern recognition

The touch-response interface can generate pattern readouts to analyze patterns of the touch-response behaviors. The performance of pattern analysis software can be evaluated by comparing the patterns generated by the interface and those provided by the domain experts (the hypothesis of patterns \mathbf{H}), as discussed in Section 2.2.2. The interface can predict a list of proposals of the potential data patterns $\mathbf{P} = \{\mathbf{P}_1, \mathbf{P}_2, \dots\}$ as the pattern readouts. Ideally, the users expect the interface to generate the correct pattern for each data point within a limited number of proposals. Thus, two evaluation metrics are defined to validate the accuracy of the pattern recognition software, i.e., number of pattern proposals $\#N_{pp}$ and the percentage of failures of pattern predictions P_f formulated as

$$\#N_{pp} = \frac{1}{N} \sum_{i=1}^N N_{ppi} \quad (2.9)$$

$$P_f = \frac{\#PF}{N} \quad (2.10)$$

where N is the number of the data points, N_{ppi} is the number of pattern proposals for data point i , and $\#PF$ is the number of data points with wrong predictions (namely none of the pattern proposals is matched with the hypothesis of pattern \mathbf{H}_i for the data point i , $\mathbf{H}_i \notin \mathbf{P}$)⁴. In general, the lower $\#N_{pp}$ and P_f , the better.

⁴ A wrong pattern prediction indicates $\mathbf{P}_1 \neq \mathbf{H}_i, \mathbf{P}_2 \neq \mathbf{H}_i, \dots$, i.e., $\mathbf{H}_i \notin \mathbf{P}$

3 Data acquisition for single larvae

3.1 Overview

The first block of a full automation interface is an automated acquisition system (displayed in Fig. 2.2) to collect the experimental data. A touch-response system on the self-positioning¹ zebrafish larvae (Single-Larva Touch-response data Acquisition Platform, SiLTAP)² is proposed, including the choices of sensor, manipulator, image processing software, and touching strategy. An automated acquisition system needs to be fed with user inputs and initial parameters to generate the experimental data, as described in Section 2.1. Fig. 3.1 illustrates three essential parts in the pipeline of SiLTAP, and gives solutions to each part (according to the comparisons in Section 2.3):

- **Sample preparation:** The larvae are prepared for the use in the experiments according to the larva preparation protocol detailed in Section 1.3.3. During the preparation, the larvae are dechorionated at 27 hpf. The larvae without any treatments are used as control group as comparison with those treated by drugs.
- **Hardware:** The locomotion behavior analysis is the goal, so the FluidFM, as listed in Section 2.3.4, is not sufficient to apply enough force for evoking the touch response. Besides, the glass pipette is much smaller than the zebrafish larva and the sharp tip can damage the larva body. Thus, a blunt needle is chosen as the manipulator. As the larva can move 200-300 mm per second, a high-speed camera (1000 fps, with an objective Lensation CMFA1520ND) is needed to capture the touch-evoked response of zebrafish larvae. Besides, Section 2.3.3 introduces that the transparent well plate is preferred as sample holder. Given that the sample holder is expected to hold numbers of larvae (regarding the throughput), the 24-well plate is not ideal for possible extensibility (e.g., for multiple larvae). Besides, the 6-well plate and Petri dish can not guarantee sufficient repeated

¹ Self-positioning means that the larvae are immobile at a specific position in the well, and do not move until stimulated externally.

² The touch screening of zebrafish larvae can be conducted with single or multiple larvae in the sample holder. This chapter focuses on the single-larva case, and Chapter 4 focuses on the multi-larva case.

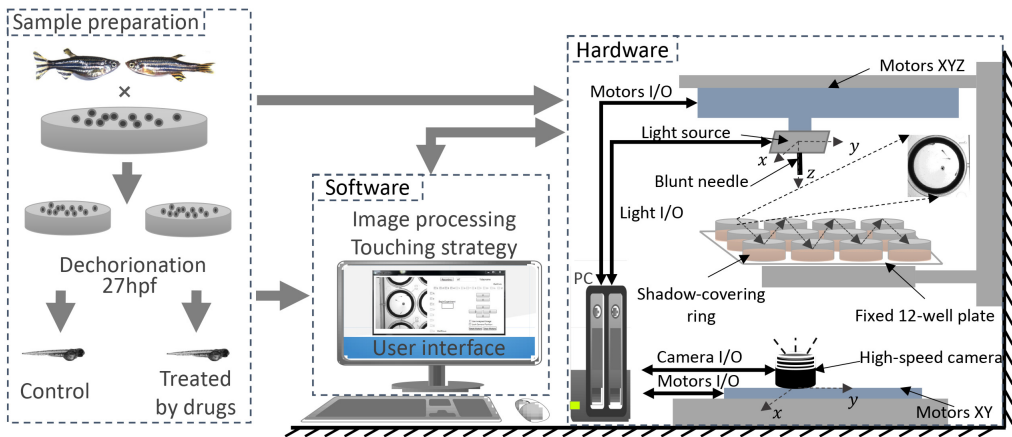


Figure 3.1: The overall architecture of the automated touch-response data acquisition system (SiLTAP), including three parts in three dashed boxes (sample preparation [175], software, and hardware [210]). In the hardware part of SiLTAP, the gray blue boxes indicate the actuator systems, and the orange pieces in the 12-well plate are the shadow-covering rings. More details about the three parts of SiLTAP are discussed in Section 3.1.

experiments. Therefore, the transparent 12-well plate is chosen as the sample holder. As the shadow-covering agarose rings help increase the image quality, they are installed in each well of the plate. Given those solutions above, the design of the hardware for SiLTAP is shown in Fig. 3.1. The video capturing and needle touching are to be conducted at the same time, so the needle mounted with a light source is set over the well plate with the high-speed camera under the well plate. The movements of them are controlled by the actuator system.

- **Software:** The software development includes the user interface, image processing algorithm, and touching strategy. The user interface interacts with users to initialize the parameters used in the experiments according to the preparation procedure of the samples. As the image data is to be collected by the camera, the software library OpenCV is chosen to develop a customized image processing pipeline. According to the goal of SiLTAP, Touching Strategy IV (single unfixed larva with touching not repeated, as in Table 2.5) of touching is chosen. The data interaction between software and hardware is based on the data input/output (I/O) Bus.

Based on the overall architecture in Fig. 3.1, the hardware images of SiLTAP are visualized in Fig. 3.2, including linear actuators, stepping motors, shadow-covering rings, linear drives, high-speed camera, magnetic set, light source, manipulator, and 12-well plate, designed by [158]. Apart from the elements introduced in Fig. 3.1, more setups are needed accordingly. The linear drives are used to transfer the circular motions of the stepping motors to the linear motions of

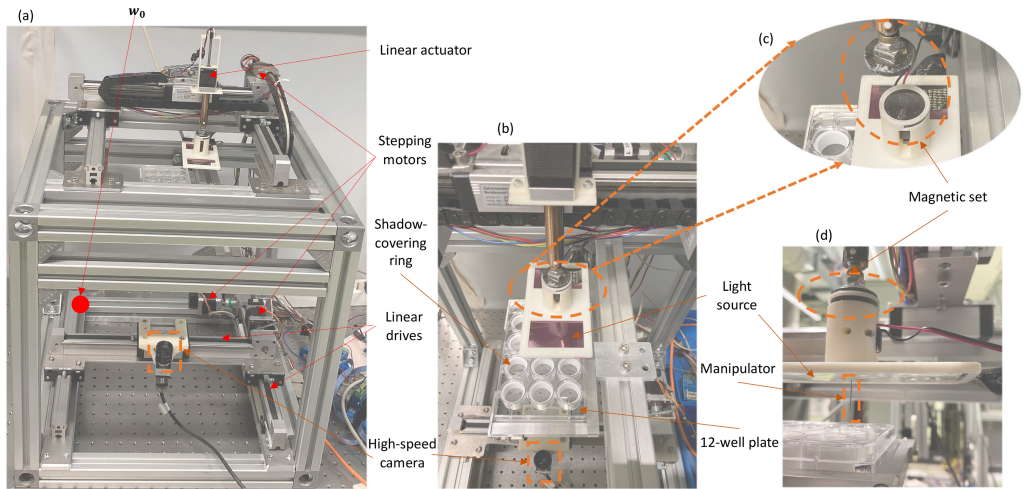


Figure 3.2: The hardware design of SiLTAP [175, 210]. (a) The whole image of SiLTAP, including the actuator system (linear actuator and stepping motors), linear drives, and high-speed camera. (b) The partial image of (a), i.e., the light source with manipulator and 12-well-plate with shadow-covering rings. (c) The partial image of (b), i.e., the magnetic set. (d) The detailed image of the light source with the manipulator in (b).

the manipulator and high-speed camera. A magnetic set (Fig. 3.2c and d) is used between the needle and the light source for safety in case of the needle breaking the well plate. Compared with the manual experiments in Fig. 1.5, the parameters including touching position P , angle θ_t , and speed (force) S_t are controllable via SiLTAP.

3.2 System design

3.2.1 Actuator system

To provide the movement of the manipulator (the blunt needle) and the positioning of the sensor (the high-speed camera), the actuators used in the SiLTAP include four stepping motors (ST4118L1804-B – STEPPER MOTOR – NEMA 17, Nanotec - Munich) and one linear actuator. Two stepping motors, mounted by two linear drives (LEZ 1 with toothed belt drive, isel Germany AG) orthogonally, work together to move in the 2D plane (for the needle and camera) and the linear actuator is used to achieve the movement in z -axis (for the needle, as shown in Fig. 3.2a). Two switches are fixed at the initial and end positions of each actuator for providing a correct positioning as well as for avoiding the movements of the actuators exceeding the length of the linear drives. After moving to the well plate in z -axis (which is also triggered by a switch), the needle only moves in 2D horizontally and the larvae are touched only once. The resolution of

each stepping motor is $0.3\text{mm}/\text{step}$, which is sufficient for touching the zebrafish larva with size in 2.5-3.7 mm mentioned in Section 1.3.5.

3.2.2 Coordinate system

In touch-response experiments, the larvae are expected to respond differently with different anatomical positions touched, e.g., the head, the body (trunk of the larvae), or the tail as shown in Fig. 3.3. For a precise description of the system design, five coordinate systems are used in the proposed SiLTAP, including world coordinate system \mathbb{W} , needle coordinate system \mathbb{N} , image coordinate system \mathbb{I} , larva coordinate system \mathbb{L} , and percentage coordinate system \mathbb{P} , details visualized in Fig. 3.3. The world coordinate system (\mathbb{W}) is defined by the robotic system with

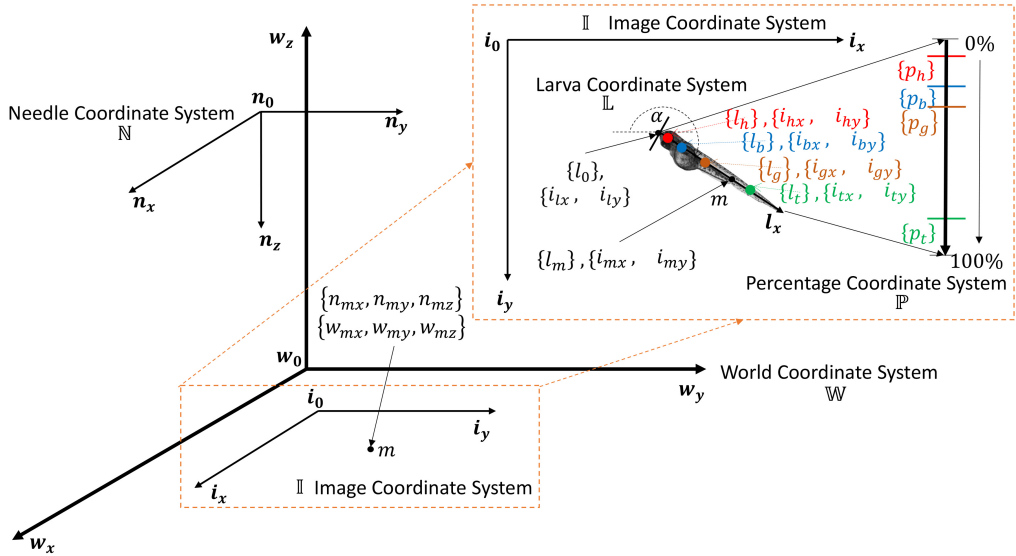


Figure 3.3: The coordinate systems used in SiLTAP [175], including the world coordinate system (\mathbb{W}) with origin w_0 , the needle coordinate system (\mathbb{N}) with the origin n_0 , the image coordinate system (\mathbb{I}) with the origin i_0 , the larva coordinate system (\mathbb{L}) with the origin l_0 . The coordinates of the larva head part are in red, body in blue, gravity in brown, tail in green. An example point m with corresponding coordinates are shown in the image and larva coordinate systems.

the coordinates formulated as $w_* = \{w_{*x}, w_{*y}, w_{*z}\}$, and the origin w_0 is defined at the bottom corner of the robot, see the red dot in Fig. 3.2. The origin of the needle coordinate system (\mathbb{N}) is the end position of the needle on the plane of light source, and the z -axis is in the opposite direction of z in \mathbb{W} as the needle moves from up to down pointing to the sample holder (shown in Fig. 3.1). The coordinates in \mathbb{N} are formulated as $n_* = \{n_{*x}, n_{*y}, n_{*z}\}$ with the origin n_0 . The

image coordinate system (\mathbb{I} , captured by the camera) is a 2D coordinate system (coordinates as $\mathbf{i}_* = \{i_{*x}, i_{*y}\}$) with the origin \mathbf{i}_0 being the top left corner of the image. The larva coordinate system (\mathbb{L}) is a 1D coordinate system (coordinates as $\mathbf{l}_* = \{l_*\}$) with the origin \mathbf{l}_0 at the very top point of the head and x -axis from the head to the tail. The percentage coordinate system (\mathbb{P}) is mapped and scaled from \mathbb{L} in the range of [0%, 100%], and the percentages in \mathbb{P} are formulated as $\mathbf{p}_* = \{p_*\}$. Ideally, it is possible for the robot to touch any region of the larva.

Assumed that α is the orientation angle of the larva in \mathbb{I} and that the origin of \mathbb{L} is expressed as $\mathbf{i}_l = \{i_{lx}, i_{ly}\}$ in \mathbb{I} , an example point m ($\mathbf{l}_m = \{l_m\}$ in \mathbb{L})³, shown in Fig. 3.3, is transferred to \mathbb{I} by

$$\mathbf{i}_m = \begin{bmatrix} i_{mx} \\ i_{my} \end{bmatrix} = \mathbf{T}_L^I \begin{bmatrix} l_m \\ 1 \end{bmatrix}, \quad \text{with } \mathbf{T}_L^I = \begin{bmatrix} -\cos \alpha & i_{lx} \\ -\sin \alpha & i_{ly} \end{bmatrix} \quad (3.1)$$

where \mathbf{T}_L^I is the transformation matrix from \mathbb{L} to \mathbb{I} . Given that the distance between \mathbb{N} and \mathbb{I} is fixed (denoted by d_{niz}) and that the X-Y planes in \mathbb{N} , \mathbb{I} , and \mathbb{W} are parallel, the scale factor between \mathbb{N} and \mathbb{I} is assumed as s_{ni} , the offsets between \mathbb{N} and \mathbb{I} are $\{d_{nix}, d_{niy}, d_{niz}\}$, and the offsets between \mathbb{W} and \mathbb{N} are $\{d_{wnx}, d_{wny}, d_{wnz}\}$. Thus, the point m defined above can be transferred from \mathbb{I} to \mathbb{N} and then to \mathbb{W} by

$$\mathbf{n}_m = \begin{bmatrix} n_{mx} \\ n_{my} \\ n_{mz} \end{bmatrix} = \mathbf{T}_I^N \begin{bmatrix} \mathbf{i}_m \\ 1 \end{bmatrix}, \quad \text{with } \mathbf{T}_I^N = \begin{bmatrix} s_{ni} & 0 & d_{nix} \\ 0 & s_{ni} & d_{niy} \\ 0 & 0 & d_{niz} \end{bmatrix} \quad (3.2)$$

$$\mathbf{w}_m = \begin{bmatrix} w_{mx} \\ w_{my} \\ w_{mz} \end{bmatrix} = \mathbf{T}_N^W \begin{bmatrix} \mathbf{n}_m \\ 1 \end{bmatrix}, \quad \text{with } \mathbf{T}_N^W = \begin{bmatrix} 1 & 0 & 0 & d_{wnx} \\ 0 & 1 & 0 & d_{wny} \\ 0 & 0 & -1 & d_{wnz} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.3)$$

³ \mathbf{l}_m and \mathbf{i}_m indicate the coordinate vectors of point m in \mathbb{L} and \mathbb{I} . l_m and i_{mx}, i_{my} indicate the values of the coordinates. \mathbb{L} is in one dimension, i.e., $\mathbf{l}_m = \{l_m\}$, and \mathbb{I} is in two dimensions, i.e., $\mathbf{i}_m = \{i_{mx}, i_{my}\}$.

According to the above, the transformation matrix from \mathbb{I} to \mathbb{W} is

$$\mathbf{T}_I^W = \mathbf{T}_N^W \begin{bmatrix} \mathbf{T}_I^N & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} s_{ni} & 0 & d_{nix} & d_{wnx} \\ 0 & s_{ni} & d_{niy} & d_{wny} \\ 0 & 0 & -d_{niz} & d_{wnz} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.4)$$

To recap, the coordinates of the needle and the target larva in \mathbb{I} are the important task for the image processing software if the system aims to control and move the needle to touch the animals. Therefore, the coordinates can be transformed to \mathbb{W} for calculating the touching trajectory.

3.2.3 Image processing and touching trajectory

The most important task of SiLTAP is the generation of the touching trajectory from the needle to the larva⁴, which is a relative displacement that controls the movement of the stepping motors. Thus, the positioning of the needle and larva by image processing are optimal⁵ to compute the displacement in \mathbb{I} which can be transferred to \mathbb{W} and control the movements of the actuators. As mentioned in Section 1.3.3, only the early stages of zebrafish larvae are to be characterized when they are still immotile and do not exhibit any spontaneous locomotion, and movements are elicited only with tactile inputs. Thus, the image processing is not repeated a second time after the larva and the needle are detected for each well.

The image captured by the high-speed camera is in gray scale with pixel value at the range of [0, 255]. For the single-larva case, only one larva as well as the needle is to be detected. Fig. 3.4 visualizes the pipeline of the image processing and the generation of the touching trajectory, including four parts:

- Preprocessing: A circle Hough transform [211] is used to detect the target well area, and the image is cropped within the well. This preprocessing is also used to determine the safety area (the well area) of the needle movements,

⁴ The actuator system introduced in Section 3.2.1 can give a position of the manipulator (blunt needle), but the position of the larva in the well plate can only be provided by the image from the high-speed camera.

⁵ Even though the position of the needle in \mathbb{W} is known by the movements of the stepping motors, the coordinates of the needle still have errors in \mathbb{I} which cause the unsuccessful touching. Thus, the positioning of the needle by image processing is better.

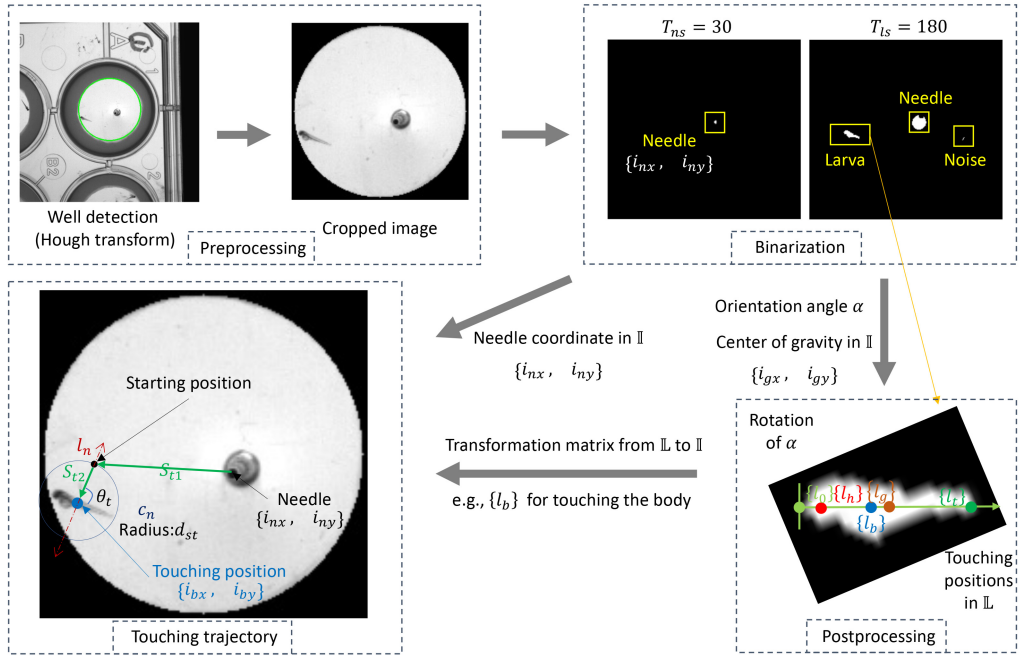


Figure 3.4: Pipeline of image processing in SiLTAP [175], including preprocessing, binarization, postprocessing, and touching trajectory.

- Binarization: The threshold based binarization method (Thre) is used for the detection of the needle and the larva with corresponding thresholds (T_{ns} : binarization of the needle and T_{ls} : binarization of the larva, respectively) chosen heuristically,
- Postprocessing: The coordinates of the detected larva are rotated with the orientation angle α to locate the touching positions in \mathbb{L} , and
- Touching trajectory: The coordinates of the needle and the touching position (e.g., touching the body in Fig. 3.4) are used to generate the touching trajectory.

The needle and the larva are the two darkest areas, and the pixel value of the needle area is lower than that of the larva. Thus, two different heuristic thresholds are used for the binarization of them. A needle binarization threshold T_{ns} (pixel value, e.g., $T_{ns} = 30$ shown in Fig. 3.4) generates a clear needle area (in the yellow box). The position of the needle is computed as the center of gravity of the found area $i_n = \{i_{nx}, i_{ny}\}$ in the image coordinate system.

As for the larva, a higher heuristic larva binarization threshold T_{ls} (for example $T_{ls} = 180$ in Fig. 3.4) is applied to the cropped image, and the resulting binary image contains two objects (the needle and the larva). This threshold also generates noise caused by uneven distribution of the

light and the larva. Thus, the resulting image is optimized by morphological operators (dilation and erosion with kernel size as k_m) and other noises are filtered out according to the size (with a size threshold T_{ss}) as the noise is smaller than the needle and larva area (shown in yellow boxes). Here, the area of the larva is located as the position of the needle is already known. From this area, orientation angle α (in the range of $[-180^\circ, 180^\circ]$) of the larva is calculated. After a rotation of α , the center of gravity $\mathbf{l}_g = \{l_g\}$ in \mathbb{L} is computed by averaging the coordinates of the whole detected larva area. In addition, the head part of the larva has more pixels than the transparent tail part, so the center of gravity of the larva is closer to the head part. Thus, with the rotation angle α , the very top of the larva head (with more pixels) is rotated to the left side, as $\mathbf{l}_0 = \{l_0\}$ in \mathbb{L} , and three touching positions of the larva, e.g., head, body, and tail in the larva coordinate system, are calculated by using fixed ratio:

$$l_h = (l_g - l_o) \cdot \frac{p_h}{p_g} \quad (3.5)$$

$$l_b = (l_g - l_o) \cdot \frac{p_b}{p_g} \quad (3.6)$$

$$l_t = (l_g - l_o) \cdot \frac{p_t}{p_g} \quad (3.7)$$

where p_h, p_b, p_t are the coordinates of the head, body, and tail of the larva in \mathbb{P} , and l_h, l_b, l_t are the corresponding coordinates in \mathbb{L} . The three touching positions are then transformed from \mathbb{L} to \mathbb{I} ($\mathbf{i}_h = \{i_{hx}, i_{hy}\}, \mathbf{i}_b = \{i_{bx}, i_{by}\}, \mathbf{i}_t = \{i_{tx}, i_{ty}\}$) according to the transformation matrix \mathbf{T}_L^I mentioned in Section 3.2.2. The thresholds used in the needle (T_{ns}) and larva binarization (T_{ls}) can be adapted according to the image brightness, e.g., higher T_{ns} and T_{ls} under a higher brightness.

The aim of the image processing for the needle and the larva is to generate a touching trajectory and control the actuators correspondingly. The touching angle θ_t and speed S_t are two important parameters for the touch screening as introduced in Section 2.2.1. Additionally, the larva can be shocked by the fluid motions (caused by the approaching of the manipulator), so the manipulator needs to approach slowly (defined as approaching speed) and apply the touching rapidly (defined as touching speed). Thus, a two-step moving strategy is designed from the actual needle position ($P_n: \mathbf{i}_n = \{i_{nx}, i_{ny}\}$ in \mathbb{I}) to a starting position (P_s formulated as $\mathbf{i}_s = \{i_{sx}, i_{sy}\}$ in \mathbb{I}), then to the touching position (e.g., the fish body part, P_b). The starting position P_s is defined with a heuristic distance d_{st} to the touching position, and the speed of the needle is increased at P_s to apply the touching rapidly. Besides, the touching angle θ_t is chosen by users, and the touching trajectory is generated according to chosen θ_t . It contains two displacements (movements): one from the needle position to the starting position $\Delta_{n \rightarrow s}$, and the other from the starting position to the touching position $\Delta_{s \rightarrow t}$. To compute the starting position, an auxiliary line l_n is used for explanation, crossed with the body of the larva with the angle θ_t at the touching position,

shown in Fig. 3.4. As the distance between the starting position and the touching position is d_{st} , an auxiliary circle c_n around the touching position is also used with radius being d_{st} , which has two crossing points with l_n . The closer point to the needle is P_s , so that the larva is not on the movement path of the needle to avoid shocking the larva before touching. Based on the computed P_s and the positions of the needle and the larva obtained by the image processing, the touching trajectory is determined. For example, for the body touching of the larva (formulated as $i_b = \{i_{bx}, i_{by}\}$ in \mathbb{I}), the touching trajectory is computed as

$$\Delta_{n \rightarrow s} = T_I^W(i_s - i_n) = \begin{bmatrix} s_{ni}(i_{sx} - i_{nx}) \\ s_{ni}(i_{sy} - i_{ny}) \end{bmatrix} \quad (3.8)$$

$$\Delta_{s \rightarrow t} = T_I^W(i_b - i_s) = \begin{bmatrix} s_{ni}(i_{bx} - i_{sx}) \\ s_{ni}(i_{by} - i_{sy}) \end{bmatrix} \quad (3.9)$$

where the speed of $\Delta_{n \rightarrow s}$ is set at S_{t1} , and then $\Delta_{s \rightarrow t}$ at S_{t2} .

3.2.4 Graphical User Interface

The Graphical User Interface (GUI) needs to be designed for the interaction (as illustrated in Fig. 3.1) between the user and SiLTAP, including a GUI for user initialization before experiments, named as SETTING_GUI (Fig. 3.5a), and a GUI used for the user interaction during the experiments, named as MAIN_GUI (Fig. 3.5b). After the preparation of the sample, the operator uses SETTING_GUI to initialize the expected touching position of the larva, type of well plate⁶, visualized in Fig. 3.5a. The second GUI is the main interactive interface (MAIN_GUI, in Fig. 3.5b) during the experiment. The MAIN_GUI contains two main areas:

- Visualization area: The video captured by the camera is shown in real time when the motors move,
- Operation area: There is a group of buttons (1-12 in columns, A-H in rows) that are designed for a 12-well plate in touch-screening experiment on zebrafish larvae as well as for the possible extensibility to other experiments. Given an actually chosen well type in SETTING_GUI, specific target wells can be enabled and the others are disabled⁷. The "Begin Experiment" button (in green) is used to start the experiments, the "Motor Steps" visualizes how many steps the motors move on, and the "Stop Motors" button is designed

⁶ Even though the 12-well plate is chosen, SiLTAP is still designed for different possibilities of well plates for the sake of extensibility.

⁷ For instance, only the buttons 1-4 in columns and A-C in rows are enabled if 12-well plate is chosen.

for stopping the experiments during the process. After experiments are finished, the "Reset Motors" button can be used to move the needle and camera back.

The two GUIs provide the possibility for the non-image-processing experts to conduct the touch-response experiments via SiLTAP.

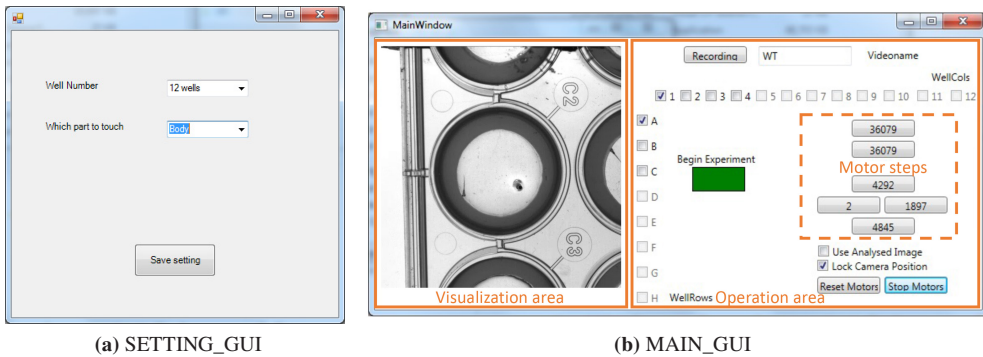


Figure 3.5: The graphical user interfaces used in the touch-response experimentation system [175].

3.2.5 Touching strategy

Based on the actuator system, image processing, generation of touching trajectory, and the interactive user interface, the touch screening is executed via a touching strategy. As shown in Fig 3.6, the touching strategy (adapted from Touching Strategy IV in Table 2.5) for single-larvae touching of SiLTAP is proposed, including the following tasks for each larva: i.e., capturing image (CI), image processing (IP), moving the needle to the larva based on trajectory (MLT), capturing video (CV), saving video (SV), move needle slightly upward (MNU), and move to the next well (MNW). Given the user inputs, SiLTAP initializes the system and moves the needle to the well one by one, where the needle is set to move upward slightly (MNU) when arriving at the bottom (triggered by the switch described in Section 3.1), thus avoiding damage to the wells and shock to the larvae. One image is captured for the image processing to detect the needle and the larva. Without any larva found, the needle and camera are moved to the next well (MNW) to repeat the steps above. With a larva found, the trajectory is generated, and the needle is moved to apply the touching. At the same time, the video is captured. During the saving of the video, the needle is moved to the next well to repeat the experiment. Based on the touching strategy above, the videos are acquired fully automatically by SiLTAP as the touch-response experimental data.

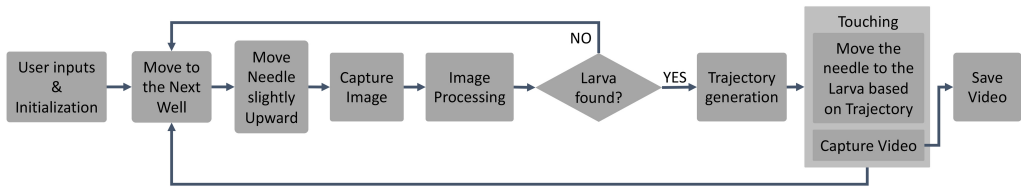


Figure 3.6: The touching strategy of SiLTAP.

3.3 Evaluation of SiLTAP

3.3.1 Experiment definition and parameter setting

The data collected by SiLTAP is essential to the biologists, so SiLTAP needs to be evaluated, including the evaluation of the software, hardware, and the touching task. The definitions of evaluation experiments and setup of parameters used in SiLTAP are discussed as follows:

- Evaluation of software (image processing): The image processing method (detection of the needle and the larva) is the key of the software, so the performance of the methods affects the effectiveness of the touch screening. Thus, experiments are to be conducted to evaluate the image processing method (shortened as E_{ip}), including two evaluation metrics: i) the center errors (e_{gd}) between the manually labeled center by the developer of SiLTAP and the detected center by the image processing (as defined in Section 2.4.2); ii) the recall ratio (R_r) of larva detection based on the comparison between the manually labeled larva area by the developer of SiLTAP and the segmented larva area by the image processing.
- Evaluation of hardware: As the actuator system undertakes the task of moving the needle to conduct the touching task, the error caused by the actuators is to be evaluated. Instead of evaluating the actuators directly, a simpler way is to quantify the error of the needle movements in \mathbb{I} . The needle is moved to a randomly predefined position, so the hardware error is analyzed by the center error e_{gd} between the actual position the needle moves to and the predefined position. This experiment is denoted as E_{he} .
- Evaluation of the touching task: The reliability, repeatability, and robustness of the proposed SiLTAP are verified by evaluating the touching task according to the evaluation metrics defined in Section 2.4.3. Ideally, SiLTAP is able to touch any position of the larva, but the errors occur in practice caused by software and hardware of the system, e.g., inaccuracy of the movements of manipulator and touching positions. Importantly, biological operators expect to use the touch-response system to deliver comparable touching positions to the animals, which is essential to generate comparable experiment results. Thus, experiments

(E_{tt}) are needed to evaluate the touching positions that are generated by the proposed SiLTAP. The evaluation metrics, touching reproducibility error e_{tr} and touching robustness r_t are used to evaluate the experiment result of E_{tt} .

Table 3.1 outlines the details of the corresponding datasets generated in Experiments E_{ip} , E_{he} , E_{tt} , including the size and the corresponding evaluation metrics (defined in Section 2.4.2 and 2.4.3). To evaluate the methods of detection in single-larva case mentioned in Section 3.2.3, an evaluation dataset is collected in Experiment E_{ip} (randomly collected 806 videos, denoted as DA-Eip) and the first frame of each video (with manual annotations) is used to quantify the performance of these methods according to the evaluation metrics e_{gd} , R_r . The hardware error caused by the stepping motors is quantified by the average of the center errors e_{gd} between 12 randomly chosen positions and the actual positions where the needle moves in \mathbb{I} (Experiment E_{he} with dataset denoted as DA-Ehe). To evaluate the repeatability and robustness e_{tr} , r_t of SiLTAP for touching tasks, the touched points in \mathbb{P} from 300 videos in Experiment E_{tt} (dataset denoted as DA-Ett) are analyzed by comparison with the predefined touching positions (p_h , p_b , p_t) in Table 3.2. The protocols of the experiments E_{ip} and E_{tt} are visualized in Fig. 3.7 on a timeline. Basically, the embryos are collected at 9:00 am (as 0 hpf) one day after crossing, and are dechorionated at 27 hpf, since the experiments are conducted even before they normally hatch (on Day 2). Experiments E_{ip} and E_{tt} are conducted at 15:00 on Day 1, 2, and 3 (30 hpf, 54 hpf, 78 hpf).

Table 3.1: The datasets of Experiments E_{ip} , E_{he} , and E_{tt} . \downarrow indicates that a low value of the evaluation metric is desired, and \uparrow indicates a high value of the evaluation metric is desired.

Description	Dataset	Size	Evaluation metric
Evaluation of software (image processing, Experiment E_{ip})	DA-Eip	806 videos	$e_{gd} \downarrow$, $R_r \uparrow$
Evaluation of hardware error (Experiment E_{he})	DA-Ehe	12 random moving points	$e_{gd} \downarrow$
Evaluation of touching tasks (Experiment E_{tt})	DA-Ett	300 videos	$e_{tr} \downarrow$, $r_t \downarrow$

In the design of SiLTAP, as described in Section 3.2, many parameters are introduced. Table 3.2 outlines the values of those parameters chosen during the experiments. The image size of the captured video data is set 480×480 pixels to ensure a high frame rate (1000 fps). The touching positions can be given by the biological operators (head, body, and tail) and are set here to 5%, 30%, and 65% in \mathbb{P} , and those values can be changed according to the requirements of the operators. For each collected video, the maximum duration is 15 s. The distance between starting position and the touching position is set to 30 pixels in order to not shock the larva before touching.

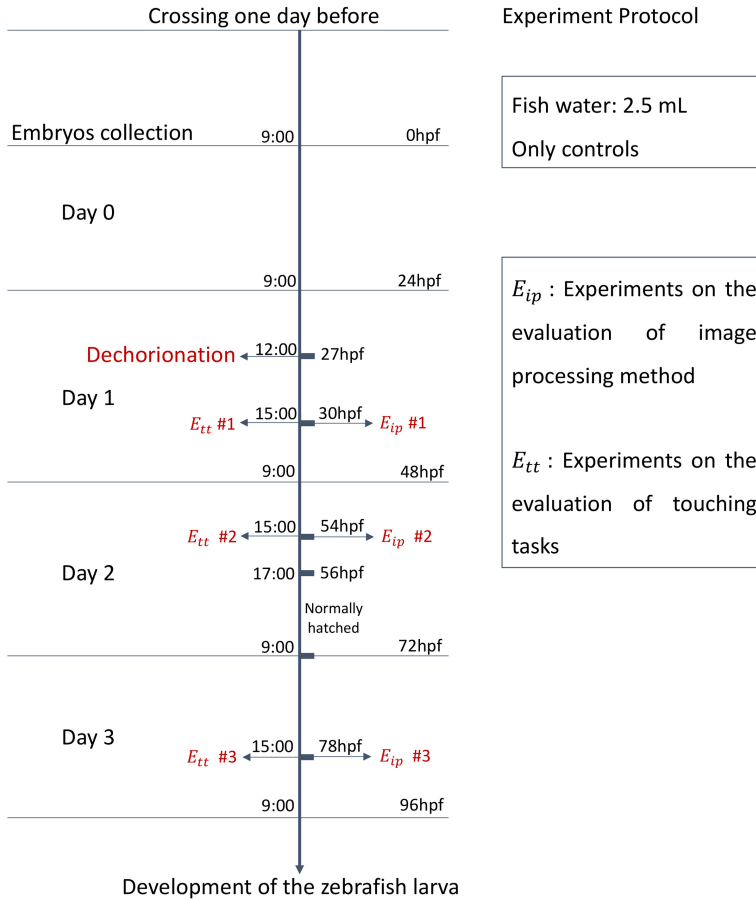


Figure 3.7: Protocol of the verification experiments E_{ip} and E_{tt} [175].

In addition, to introduce less changing factors, θ_t is fixed at 90° , S_{t1} at 1.5 mm/s (lower speed to not shock the larva), and S_{t2} at 7.5 mm/s (higher speed to apply the touching), which are selected heuristically. The parameters used in the image processing pipeline of SiLTAP (Section 3.2.3) are detailed as follows. The thresholds used for the needle and larva binarization (pixel value) are set as $T_{ns} = 30$, $T_{ls} = 180$ respectively⁸. The kernel size used in the morphology (dilation and erosion) of the image processing pipeline is set as $k_m = 5 \times 5$ pixels. The size threshold is set as $T_{ss} = 12$ pixels for filtering the noise out as mentioned in the image processing pipeline. The control of the actuator system, image processing together with touching strategy used in SiLTAP are coded in C# based on the OpenCV library.

⁸ The range of pixel value in one image captured by the high-speed camera is $[0, 255]$.

Table 3.2: The parameters used in SiLTAP.

Symbol	Quantity	Chosen value
–	Age of larvae being dechorionated	27 hpf
–	Image size	480 × 480 pixels
–	Type of well plate	12-well plate
p_h	Head position in \mathbb{P}	5%
p_b	Body position in \mathbb{P}	30%
p_g	The center of gravity in \mathbb{P}	30%
p_t	Tail position in \mathbb{P}	65%
–	Frame rate	1000 fps
–	Maximum duration of videos	15 s
d_{st}	Distance between P_s and P_t	30 pixels
θ_t	The angle of touching	90°
S_{t1}	Approaching speed from P_n to P_s	1.5 mm/s
S_{t2}	Touching speed from P_s to P_t	7.5 mm/s
T_{ns}	Threshold for needle binarization used in the SiLTAP (Section 3.2.3, pixel value)	30
T_{ls}	Threshold for larvae binarization used in the SiLTAP (Section 3.2.3, pixel value)	180
k_m	Morphology kernel size used in the image processing of SiLTAP (Section 3.2.3)	5 × 5 pixels
T_{ss}	Size threshold used in the image processing of SiLTAP (Section 3.2.3)	12 pixels

3.3.2 Evaluation of image processing and hardware error

The evaluation of the image processing pipeline (software of SiLTAP) discussed in Section 3.2.3 includes the detection of the needle and the larva, as well as the segmentation of the larva:

- The center errors e_{gd} of the needle and larva detection are shown in Table 3.3. Both e_{gd} are less than 3 pixels, which ensures an accurate detection of the needle and larva,
- The performance of the segmentation of the larva is analyzed by the ratio of recall R_r (according to Equ. 2.6 and 2.7 in Section 2.4.2) that specifies the percentage of the larvae

Table 3.3: The error evaluation. \downarrow indicates that a low value of the evaluation metric is desired.

Evaluation Method	Value
Center error e_{gd} for needle on dataset DA-Eip (pixels)	1.04 \downarrow
Center error e_{gd} for larva on dataset DA-Eip (pixels)	2.64 \downarrow
Hardware center error e_{gd} on dataset DA-Ehe (pixels)	2.52 \downarrow
Head touched reproducibility error e_{tr1} on dataset DA-Ett (%)	0.82 \downarrow
Body touched reproducibility error e_{tr2} on dataset DA-Ett (%)	1.13 \downarrow
Tail touched reproducibility error e_{tr3} on dataset DA-Ett (%)	4.46 \downarrow

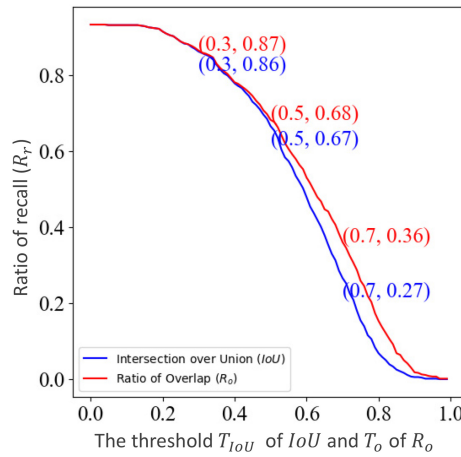


Figure 3.8: The ratio of recall of the larva segmentation with respect to threshold of IoU (T_{IoU}) and ratio of overlap (T_o), evaluated on dataset DA-Eip [175]. According to Equ. 2.6 and 2.7, the thresholds T_{IoU} and T_o are chosen for successful recall of the larva detection, and the ratio of recall drops with the increase of the thresholds T_{IoU} and T_o .

that are successfully segmented⁹. The ratio of recall R_r based on both IoU and R_o can be different as introduced in Section 2.4.2. However, as visualized in Fig. 3.8, the two figures of ratio of recall ($R_r|_{T_o}$ in red and $(R_r|_{T_{IoU}}$ in blue) have only slight differences. This indicates that they have the consistent results. In details, R_r is between 90% to 100% if $T_{IoU}(T_o) \leq 0.3$ (but the segmentation of transparent tail part is probably missing) and decreases significantly from $T_{IoU}(T_o) = 0.5$ and converges to 0 after $T_{IoU}(T_o) \geq 0.9$. This indicates that the binarization method may miss some information of the larva, so

⁹ The larva is regarded as successfully segmented if the segmented area has a higher IoU or R_o with the manually labeled area than the threshold T_{IoU} or T_o , as details defined in Section 2.4.2.

that only a part of larva can be segmented, e.g., some part of the larva tail is probably not segmented. Thus, it is reasonable and practical to locate the different parts of the larva by using Equ. 3.5 to 3.7.

The center error of the hardware based on dataset DA-Ehe is outlined in Table 3.3. Only an error of 2.52 pixels on average can be observed from the movement of the needle driven by the stepping motors, which ensures to accurately apply the touching.

3.3.3 Evaluation on the touching tasks

To evaluate the reproducibility (e_{tr}) and robustness (r_t) of SiLTAP for touching tasks, Experiment E_{tt} is conducted by touching the predefined positions (p_h, p_b, p_t) in Table 3.2. In dataset DA-Ett, the touched point in each video is visually inspected and mapped to the coordinate in \mathbb{P} . The coordinates of those touched points are figured as a histogram in Fig. 3.9, including the fitted normal distributions (with mean μ and standard deviation s).

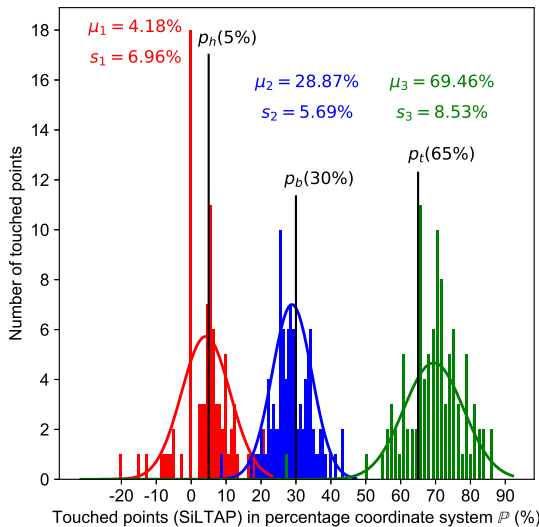


Figure 3.9: The distribution of the touched points with the proposed SiLTAP tested on dataset DA-Ett [175]. The touched points are mapped to \mathbb{P} (seen in Fig. 3.4, red for the head, blue for the body, green for the tail, and black solid lines as the predefined touching positions (p_h, p_b, p_t)). Minus value indicates that SiLTAP does not touch the larva successfully.

The reproducibility of the touching task (e_{tr1} , e_{tr2} , e_{tr3} for head, body, and tail touching respectively) can be represented by the difference between predefined touching positions in Table 3.2 and the mean of the distributions of the touched points (μ_1, μ_2, μ_3) in Fig. 3.9. As outlined in Table 3.3, the head touching has the highest accuracy with touched point error $e_{tr1} = |\mu_1 - p_h| = 0.82\%$, the body touching has an accuracy of $e_{tr2} = |\mu_2 - p_b| = 1.13\%$, and the tail touching has an accuracy of $e_{tr3} = |\mu_3 - p_t| = 4.46\%$. The robustness of three parts touching can be observed from the standard deviations of the fitted normal distributions $r_t = \{s_1, s_2, s_3\}$ in Fig. 3.9. The body touching has a better robustness ($s_2 = 5.69\%$) than the other two ($s_1 = 6.96\%$, $s_3 = 8.53\%$). Importantly, these three parts touching are clearly in different positions of the larva, thereby leading to comparable inputs to the experiments compared with manually touching. The tail touching is still the worst with the highest standard deviation ($s_3 = 8.53\%$) since the calculation of center of gravity may be slightly disturbed because of the missing segmentation for the transparent tail part. A small error at the center of gravity may be enlarged at the other two parts as a result of Equ. 3.5 to 3.7.

3.3.4 Discussions

The experiments as well as the corresponding results verify the effectiveness of the proposed SiLTAP, including the performance of the software and hardware as well as the reproducibility and robustness of the touching task. From the results, SiLTAP achieves the desired touching task at different positions of the larvae in an automated manner. The robustness of the system ensures the reliable output of touch-response experimental data acquisition on zebrafish larvae. Nonetheless, some drawbacks still exist. Firstly, the single larva in each well limits the number of the collected data (the throughput of experiments), and the image processing method has potentials to miss the needle or the larvae that furthermore reduces the throughput. Also, the agarose shadow-covering rings need to be prepared in each experiment, which is still time-consuming and does not meet the goal of the full automation interface. The potential improvements shall be considered in the next step as discussed in Chapter 4.

4 Data acquisition for multiple larvae

A Multi-larva Touch-response data Acquisition Platform (MuLTAP) with a multi-larva touching strategy is proposed to automatically conduct the touch-response experiments on multiple zebrafish larvae. A new touching strategy as well as a multi-larva image processing pipeline is proposed, and a series of experiments are conducted to prove the improvement of MuLTAP compared with SiLTAP.

4.1 Overview

In SiLTAP, only a single larva can be considered in each well. The throughput is limited by the sequential moving of the needle between wells, as the touching instrument (needle) needs to move between wells of the sample holder (the well plate, discussed in Section 2.3.3) and only one video can be collected for each well. The solution of this problem is to put more larvae in each well to increase throughput and efficiency of the touch-response screening experiments. Some problems need to be considered:

- The agarose shadow-covering rings in SiLTAP are not reusable and need to be prepared in each experiment, causing unnecessarily repeated work. As the improvement of the experimental throughput and efficiency is the main goal of MuLTAP, the agarose shadow-covering rings are not optimal as the preparation of them is time-consuming,
- The larvae can touch each other after the application of the touching from the manipulator. This interactive response is not desired in touch-response experiments, as they are not simulated externally. Thus, the optimal number of larvae in each well needs to be chosen, and
- In SiLTAP, the threshold based binarization is sufficient for the segmentation of the single larva and manipulator. However, it can miss the segmentation of tail part of the larva, causing a low reproducibility and robustness of tail touching experiment, detailed in Section 3.3.3. Thus, in MuLTAP, a better image processing pipeline is required for segmenting multiple larvae and classifying them from manipulator.

Compared with SiLTAP, the goals of MuLTAP are to find the solutions to the conflicts discussed above:

- introducing reusable shadow-covering rings,
- upgrading image processing pipeline for the segmentation of multiple larvae and needle, and
- designing an adapted multi-larva touching strategy to achieve a higher throughput.

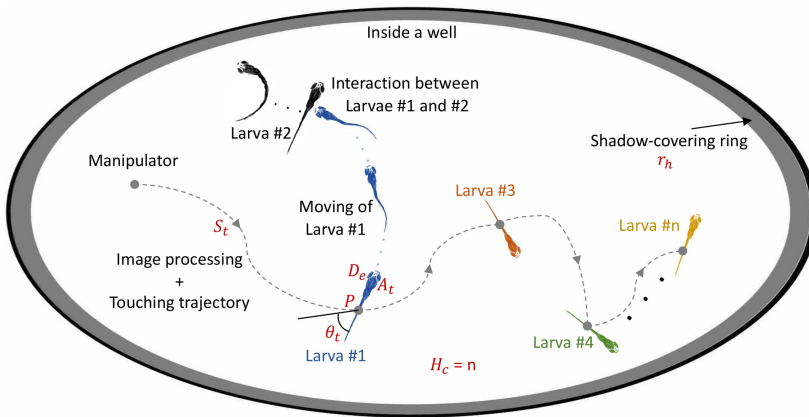


Figure 4.1: The desired workflow of MuLTAP. The larvae are marked in different colors to better distinguish them, i.e., Larva #1 in blue, Larva #2 in black, Larva #3 in orange, Larva #4 in green, and Larva #n in yellow. The parameters used in the multi-larva touch screening (S_t , θ_t , P , A_t , H_c , r_h) are marked in red, where the capacity of the well $H_c = n$. The angle of touching θ_t is set to 90° as SiLTAP does.

As more than one larva is put inside the well (installed with a shadow-covering ring), as visualized in Fig. 4.1, the manipulator is moved to touch all larvae one by one to collect more video data. The multi-larva data acquisition has to handle more complex scenes and the larvae have more interactive behaviors. The tasks of multi-larva touching strategy are summarized as:

- The manipulator is moved according to the results of image processing and touching trajectory to touch all larvae in the well one by one,
- The touching task needs to be skipped for the larva that has interaction (movements) with the touched larva by the manipulator. For example, in Fig. 4.1, Larva #1 (in blue) is evoked to move by touching and the movement of Larva #1 simulates Larva #2 (in black) to move as well. In this case, the touching on Larva #2 needs to be ignored, as this movement is not considered as mechanical touch-evoked response, and

- The video data for the touching of each larva needs to be saved before touching the next one. For example, the video of touching Larva #3 (in orange) is saved before touching Larva #4 (in green).

4.2 Multi-larva touch-response data acquisition platform

4.2.1 Reusable shadow-covering rings

In SiLTAP, high-throughput experiments are not considered, so agarose rings are sufficient in single-larva experiments. However, in MuLATP, the experimental throughput is the focus of the system design, so a re-usable 3D printed ring¹ (material: Polytetrafluoroethylene (PTFE)) is introduced to improve the experimental efficiency as shown in Fig. 4.2. The associative information of the PTFE rings is outlined in Table 4.1. The outer diameter $r_{od} = 22$ mm of the ring is chosen according to the inner diameter of 12-well plate ($H_{id} = 22.05$ mm) as listed in Table 2.2. The inner diameter r_{id} of the ring is chosen heuristically as 18 mm. The larvae may move over the rings if the height of the ring r_h is too small. Thus, the parameter r_h is to be explored. As the fish water is required to be over the ring (mentioned in Section 2.3.3), the height of the ring needs to be lower than the height of 12-well plate ($H_h = 18.95$ mm). Thus, two possible r_h of the ring chosen heuristically (6 and 12 mm) are outlined in Table 4.1. A verification experiment is needed to search for the optimal height r_h .

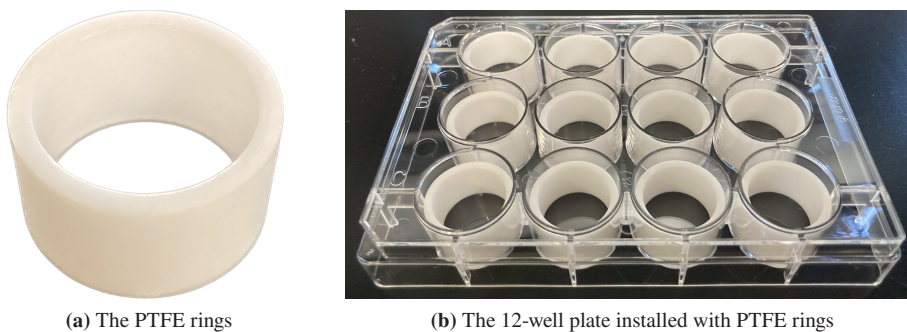


Figure 4.2: The reusable 3D printed shadow-covering rings installed in each well of 12-well plate.

¹ The 3D printed shadow-covering rings are designed via Creo CAD Software, and printed by the 3D printer at KIT-IAI.

Table 4.1: The information of PTFE rings.

Symbol	Quantity	Value
r_{od}	Outer diameter of shadow-covering ring	22 mm
r_{id}	Inner diameter of shadow-covering ring	18 mm
r_h	Height of shadow-covering ring	6 mm or 12 mm

4.2.2 Image processing

In SiLTAP, the segmentation of only a single larva is needed and only two objects (one larva and manipulator) need to be detected. Thus, the binarization method is sufficient for SiLTAP to reduce the complexity of the system design. However, this method has the potential for improvements, as described in Section 4.1. In contrast to the single-larva case, image processing for multiple larvae in each well bring much more challenges as not only one larva needs to be detected, and the needle point is required to be classified among those larvae. Therefore, experiments are needed to compare the methods listed in Section 2.3.6 and select the optimal one for the segmentation task.

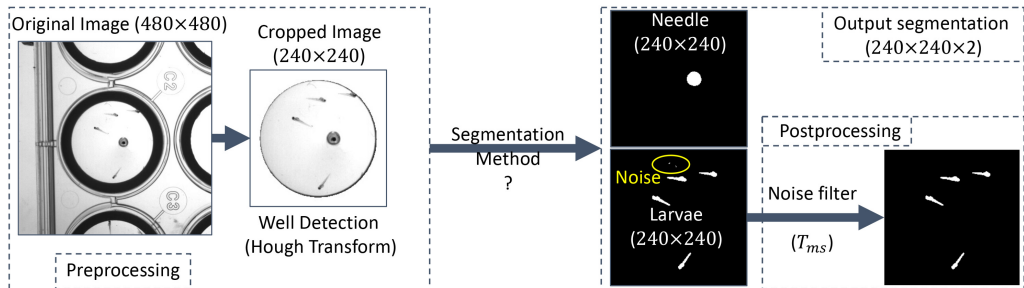


Figure 4.3: Pipeline of image processing in the multi-larva case, including preprocessing, segmentation method to be determined, output segmentation.

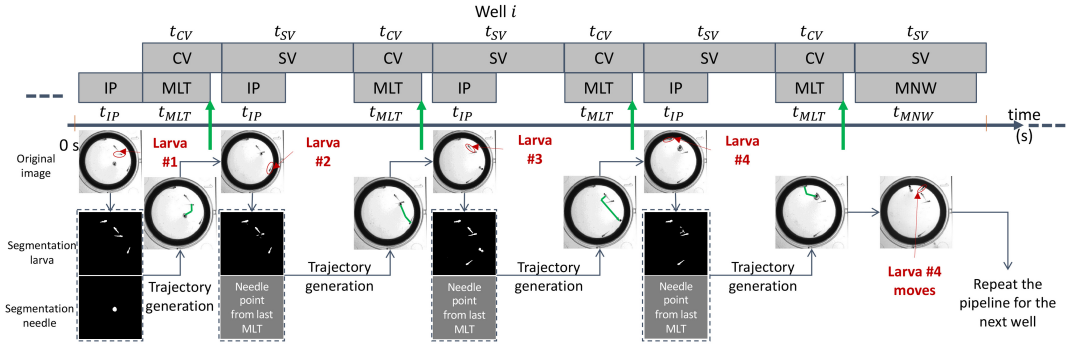
The image processing pipeline used for MuLTAP is visualized in Fig. 4.3. In the preprocessing of the pipeline, similar to SiLTAP (Section 3.2.3), a Hough transform is used to detect the well area from the captured image by the camera, cropping the image from 480×480 to 240×240 . The segmentation task of multiple larvae needs to be applied to the cropped image for generating the binary images for the needle and larva separately. Thus, the verification experiment is needed for the comparison of the methods above to determine the optimal one. In the binary images, the noise (in the yellow circle of Fig. 4.3) is possible to occur in the segmentation results, so a size threshold (T_{ms}) is used as postprocessing to delete the smaller objects (noise).

4.2.3 Touching strategy

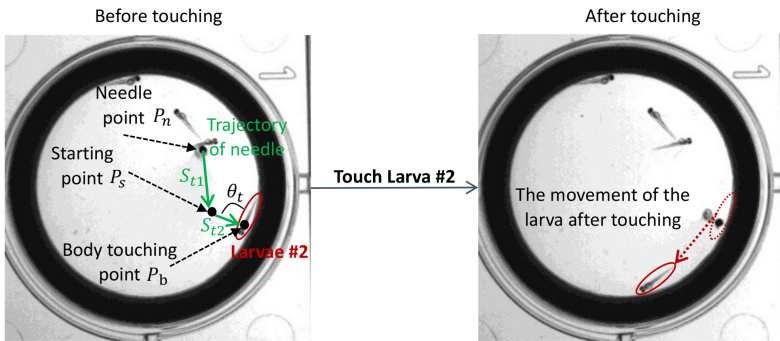
As visualized in Fig. 4.1, the touching strategy in the multi-larva case needs to be organized carefully, since the larvae can be evoked by each other to move even if only one larva is touched. This evoked movement by the other larvae is not considered in touch-response screening. Fig. 4.1 is adapted to a multi-larva touching strategy, as visualized in Fig. 4.4, including image processing (IP), moving the needle to the larva based on trajectory (MLT), capturing video (CV), saving video (SV), and moving the needle to the next well (MNW). To introduce this strategy, four larvae ($H_c = 4$), as an example, are put into the well to be touched one after another (each larva with one video of the whole well from the beginning of the MLT to the end of CV). The touch strategy is conducted as follows:

- The camera captures an image and the image processing (IP) pipeline in Section 4.2.2 conducts the segmentation of larvae and needle, as the first segmentation,
- Larva #1 is defined as the closest larva to the needle position according to the segmentation by using IP, followed by the generation of the touching trajectory (Section 3.2.3),
- The needle is moved to the Larva #1 based on trajectory (MLT) to apply the touching, and at the same time, the video of the larva response is captured (CV) followed by saving the video (SV),
- At the beginning of SV, another image captured by the camera is used for the second IP to obtain a new segmentation of larvae, as the second segmentation,
- The segmented areas of the larvae in the first and second segmentation are compared, e.g., by using intersection over union of the segmented areas, to detect larvae that are not stimulated by the first touching. The larvae that are stimulated to move by the touched larva are ignored,
- The needle position after touching Larva #1 is known according to the touching trajectory for Larva #1, so the segmentation for the needle is no more needed in the second segmentation. The touching trajectory for Larva #2 is calculated based on the second segmentation and needle position. The touching is conducted and video is saved for Larva #2 the same as done to Larva #1, and
- After all larvae are touched, the needle is moved to the next well (MNW) to repeat the experiments.

One essential parameter of the touching strategy to be evaluated by the experiment is the experimental efficiency, i.e., the times used for each part of the strategy t_{IP} , t_{MLT} , t_{CV} , t_{SV} , and t_{MNV} , as shown in Fig. 4.4.



(a) Multi-larva touching strategy in well i



(b) The touching of Larva #2

Figure 4.4: The proposed multi-larva touching strategy of MuLTAP [210]. (a) The pipeline of the multi-larva touching strategy consists of five main parts, including image processing (IP), moving the needle to the larva based on trajectory (MLT), capturing video (CV), saving video (SV), and moving the needle to the next well (MNV), with details in Section 4.2.3. The time for each part is denoted as t_{IP} , t_{MLT} , t_{CV} , t_{SV} , and t_{MNV} . The larvae are marked in red and the application of the touching is shown as the green arrows. (b) Example of touching Larva #2 in (a). The generation of the touching trajectory with a starting point is displayed in green, and the movement of Larvae #2 is shown in red. The approaching speed S_{t1} , touching speed S_{t2} , and touching angle θ_t are defined in Section 2.2.1 and 3.2.3.

4.3 Evaluation of MuLTAP

4.3.1 Experiment definition and parameter setting

The concept and design of MuLTAP described in Section 4.1 and 4.2 introduces many parameters to determine, so three groups of experiments are to be conducted:

- Selection of segmentation methods: The segmentation method needs to be selected and the size threshold T_{ms} in the image processing pipeline of MuLTAP in Section 4.2.2 is to be determined,
- Parameter determination: MuLTAP aims to collect as many valid videos as possible (as defined in Section 2.4.3). The factors that affect the throughput of the touch-response screening are capacity of well plate H_c , height of shadow-covering rings (PTFE) r_h , ages of larvae A_t , and effect of dechoriation D_e . Those parameters are to be determined, and
- Evaluation and comparison: The comparison of SILTAP and MuLTAP regarding the touching reproducibility, robustness, efficiency, and throughput is needed. Based on the parameters searched above, experiments are to be conducted for evaluating MuLTAP.

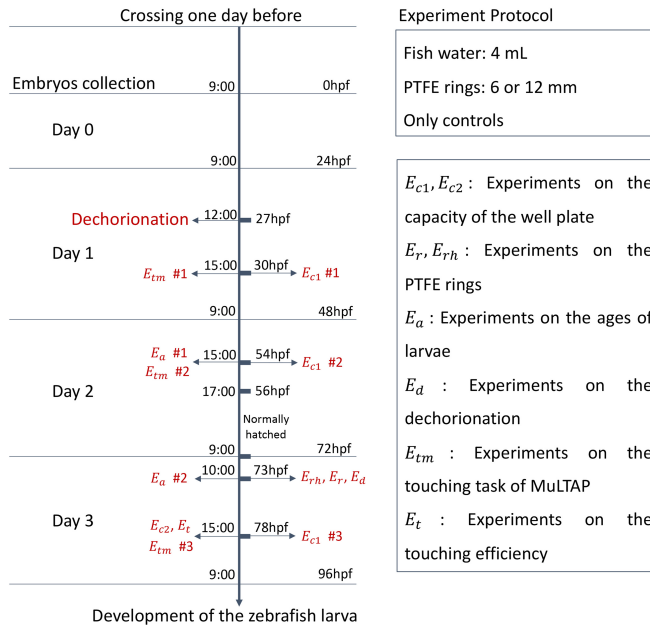


Figure 4.5: Protocol of the verification experiments (E_{c1} , E_{c2} , E_r , E_{rh} , E_a , E_d , E_{tm} , and E_t) for MuLTAP. The experiments are marked in red.

A series of experiments are conducted with the protocol visualized in Fig. 4.5 on a timeline. The experiments are defined with three groups:

Selection of segmentation methods:

- experiment on the selection of segmentation methods (named as E_{seg}).

Parameter determination:

- experiment on the capacity of well plate (named as E_{c1}),
- experiment on the verification of the capacity of well plate (named as E_{c2}),
- experiment on the effectiveness of PTFE shadow-covering rings (named as E_r),
- experiment on the height of the shadow-covering rings (named as E_{rh}),
- experiment on the ages of the larvae (named as E_a), and
- experiment on the effect of dechoriation (named as E_d).

Evaluation and comparison of SiLTAP and MuLTAP:

- experiment on the touching task of MuLTAP (named as E_{tm}),
- experiment on the touching efficiency (named as E_t), and
- experiment on the average number of videos collected per well (named as E_{vpw}).

The datasets collected in Experiments E_{seg} , E_{c1} , E_{c2} , E_r , E_{rh} , E_a , E_d , E_{tm} , E_t , and E_{vpw} as well as the parameters and evaluation metrics are outlined in Table 4.2.

4.3.2 Selection of segmentation methods

The segmentation methods are to be evaluated according to the evaluation metrics, i.e., Per-Class accuracy (PC), Jaccard Index (JI), ratio of recall (R_r), and ratio of precision (R_p), defined in Section 2.4.2. For the testing of the performance of the methods on the segmentation of zebrafish larvae, datasets with 528 real images containing different number of larvae are collected, including 204 images for the training (the dataset denoted as DA-Seg-Train) and 324 images for the testing (the dataset denoted as DA-Seg-Test), as denoted in Table 4.2. The parameters used for Thre, Otsu, LRb, Rgb, and U-Net² are outlined in Table 4.4, and the evaluation results of the those

² The training details of the U-Net are described in Appendix A.2.

Table 4.2: The datasets collected in Experiments E_{seg} , E_{c1} , E_{c2} , E_r , E_{rh} , E_a , E_d , E_t , and E_{vpw} .

Description	Dataset denotation	Number	Parameters	Evaluation metrics
Experiment on the selection of segmentation methods E_{seg}	DA-Seg-Train	204 images	-	PC, JI, R_r , R_p
	DA-Seg-Test	324 images		
Experiment on the capacity of well plate E_{c1}	DA-Ec1	383 videos	H_c	#VT
Experiment on the verification of the capacity of well plate E_{c2}	DA-Ec2	134 videos	H_c	#VT
Experiment on the effectiveness of PTFE shadow-covering rings E_r	DA-Er	24 images	-	R_r , R_p
Experiment on the height of the shadow-covering rings E_{rh}	DA-Erh	74 videos	r_h	#VT
Experiment on the ages of the larvae E_a	DA-Ea	83 videos	A_t	#VT
Experiment on the effect of dechoriation E_d	DA-Ed	146 videos	D_e	#VT
Experiment on the touching task of MuLTAP E_{tm}	DA-Etm	109 videos	-	e_{tr} , r_t
Experiment on the touching efficiency E_t	DA-Et	111 videos	-	t_{TPL}
Experiment on the average number of videos collected per well E_{vpw}	DA-Evpw	57 videos	-	#VPW

methods based on the evaluation metrics PC, JI, R_r , and R_p are shown in Table 4.5 and Fig. 4.6a and 4.6b.

The U-Net based segmentation over-performs the other methods in PC, JI, and R_r , and only has lower R_p than RGB. However, the RGB can not classify the larvae and needle, and performs worse than U-Net in JI. Thus, the U-Net based segmentation method is selected, and based on the U-Net, the image processing pipeline for MuLTAP is designed for generating two binary images (the needle and the larvae), as shown in Fig. A.1. In the pipeline of image processing visualized in Fig. 4.3, a threshold T_{ms} for size based filter in the postprocessing needs to be determined.

Table 4.3: The experiment setup of E_{c1} , E_{c2} , E_r , E_{r_h} , E_a , E_d , E_{tm} , E_t , and E_{vpw} . "ND" indicates no dechlorination.

Experiment	E_{c1}	E_{c2}	E_r	E_{r_h}	E_a	E_d	E_{tm}	E_t	E_{vpw}
Volume of water	2 mL	2 mL	4 mL	2 mL, 4 mL	4 mL	4 mL	2 mL	4 mL	2.5 mL, 4 mL
Type of rings	PTFE	PTFE	PTFE	PTFE	PTFE	PTFE	PTFE	PTFE	Agarose, PTFE
r_h	6 mm	6 mm	no rings, 12 mm	6 mm, 12 mm	12 mm	12 mm	6 mm	12 mm	4 mL, 12 mm
Age of larvae	30, 54, 78 hpf	78 hpf	73 hpf	73 hpf	54, 73 hpf	73 hpf	30, 54, 78 hpf	78 hpf	54 hpf
Dechlorinated	27 hpf	27 hpf	27 hpf	27 hpf	27 hpf	27 hpf, ND	27 hpf	27 hpf	ND
Touching positions	head, body, tail	head, body, tail	-	body	body	body	head, body, tail	head, body, tail	body
Number of larvae per well	2, 3, 4, 5	2, 3, 4, 5	4	4	4	4	4	4	1, 4
Total number of larvae	42	42	96	48	48	192	12	60	60

The choice of the age of larvae is heuristic.

If different groups of larvae are compared, the age of larvae for each group needs to be consistent.

Table 4.4: The parameters used for the comparison of segmentation methods.

Parameter	Explanation	Value
T_{Thre}	Threshold for Thre	180
T_{LRb}	Threshold for output of LRb	0.95
Epo_{LRb}	Training epoch selected for testing LRb	700000
T_{RgB}	Threshold for the image gradient of RgB	5
T_{UNet}	Threshold for the output of U-Net	0.9

Table 4.5: Comparison of segmentation methods. \uparrow indicates that a high value of the evaluation metric is desired, and \downarrow indicates that a low value of the evaluation metric is desired.

Method	Thre	Otsu	LRb	RgB	U-Net
Training set	-	-	DA-Seg-Train	-	DA-Seg-Train
Testing set	DA-Seg-Test	DA-Seg-Test	DA-Seg-Test	DA-Seg-Test	DA-Seg-Test
PC \uparrow	0.60	0.69	0.36	0.82	0.85
JI \uparrow	0.21	0.21	0.25	0.39	0.76
Time \downarrow	0.024 s	0.024 s	0.910 s	4.210 s	3.187 s
Classification of needle and larva	No	No	No	No	Yes

T_{ms} is tested with the U-Net model to visualize the PC and JI only for the larvae segmentation in Fig. 4.6c. After comparison, $T_{ms} = 12$ is selected to filter the noise out.

4.3.3 Parameter determination

Capacity of well plate The number of the larvae in each well defines the throughput of MuLTAP. However, a large number of larvae can cause interactive behaviors between larvae, as visualized in Fig. 4.1, which is not desired. Thus, the number of the larvae (capacity H_c , defined in Section 2.2.1) in each well is to be quantified by conducting the experiments on the capacity of well plate, i.e., Experiment E_{c1} and E_{c2} . In order to increase the experimental throughput and decrease the interactions, two objectives need to satisfied:

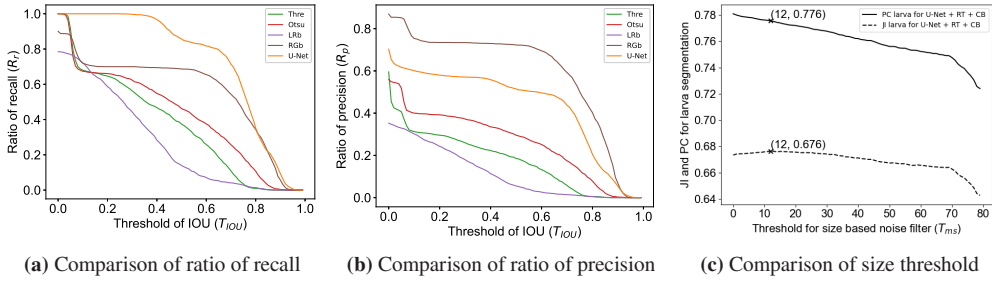


Figure 4.6: (a) and (b) Comparison of Thre, Otsu, LRb, RGB, and U-Net by ratio of recall and precision. All methods are evaluated without any post-processing. (c) Comparison of size threshold (T_{ms}) on the PC and JI of larvae segmentation by using the U-Net. If T_{ms} is set as 12, the JI has highest value with PC still satisfactory, so 12 is selected to be T_{ms} in the experiment.

1. Only one larva moves when one larva is touched. Experiment E_{c1} is to calculate the number of larvae that have movements if only one larva is touched. Ideally, only one moves after touching,
2. As many videos as possible need to be collected. As the larvae that have interactive behaviors are ignored, as described in Section 4.2.3, the number of collected data does not increase even if more larvae are put into the well. Experiment E_{c2} is to verify the maximum number of larvae at which the number of collected data does not increase.

In Experiment E_{c1} , multiple numbers (2, 3, 4, or 5) of larvae are put into the well to conduct the touching on the head, body, and tail of only one larva in each well at 30 hpf, 54 hpf, and 78 hpf, respectively. The parameters of Experiment E_{c1} are outlined in Table 4.3, and 383 videos (with the dataset denoted as DA-Ec1 in Table 4.2) are collected in Experiment E_{c1} , including 98 videos with 2 larvae, 97 videos with 3 larvae, 93 videos with 4 larvae, and 95 videos with 5 larvae. In Experiment E_{c1} , only one larva is touched among different number of larvae in each well, and the number of larvae that have interactive behaviors are counted as shown in Fig. 4.7. The details

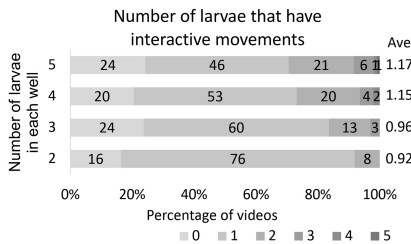


Figure 4.7: Comparison of the number of larvae that have interactive movements after only one larva touched, with 2, 3, 4, and 5 larvae in each well respectively [210]. The "Ave" indicates the average number of larvae that have interactive movements in each well.

of results for Experiment E_{c1} in Fig. 4.7 are listed as:

- With 2 larvae in the well, if one larva is touched, no larva responds in 16% of videos, 1 larva has movements in 76% of videos, and 2 larvae have movements in 8% of videos,
- With 3 larvae in the well, if one larva is touched, no larva responds in 24% of videos, 1 larva has movements in 60% of videos, 2 larvae have movements in 13% of videos, and 3 larvae have movements in 3% of videos,
- With 4 larvae in the well, if one larva is touched, no larva responds in 20% of videos, 1 larva has movements in 53% of videos, 2 larvae have movements in 20% of videos, 3 larvae have movements in 4% of videos, and 4 larvae have movements in 2% of videos, and
- With 5 larvae in the well, if one larva is touched, no larva responds in 24% of videos, 1 larva has movements in 46% of videos, 2 larvae have movements in 21% of videos, 3 larvae have movements in 6% of videos, 4 larvae have movements in 1% of videos, and 5 larvae have movements in 1% of videos.

Moreover, in the well plate with 3 larvae, 0.96 larvae on average respond after touching ("Ave" in Fig. 4.7), and in the well plate with 4 larvae, 1.15 larvae on average respond after touching. Thus, 3-larva and 4-larva cases are two possible choices of capacity of well plate according to Experiment E_{c1} , as only one larva moving is preferred with one larva touching. Another experiment is needed to determine the better one among these two cases.

In Experiment E_{c2} , multiple numbers (2, 3, 4, or 5) of larvae are put into the well to conduct the touching on all the larvae in the well plate at only 78 hpf (as outlined in Table 4.3). In total, 134 valid videos ($\#VT$, as defined in Section 2.4.3) are collected, as shown in Table 4.6, including 25 videos in 2-larvae case, 31 videos in 3-larvae case, 39 videos in 4-larvae case, and 39 videos in 5-larvae case (with the dataset denoted as DA-Ec2 in Table 4.2). The touching in 4-larvae case collects more videos than those in 3-larvae case. Therefore, 4-larva case meets the aim of Experiment E_{c2} . The conclusion of E_{c1} and E_{c2} can be drawn that 4 larvae ($H_c = 4$) are the optimal choice to collect more videos.

Table 4.6: Number of valid videos collected via MuLTAP in 2-larvae, 3-larvae, 4-larvae, and 5-larvae cases by touching all larvae in each well.

Number of larvae in each well	2	3	4	5	Total
$\#VT$	25	31	39	39	134

Shadow-covering rings As the PTFE shadow-covering rings are newly introduced in Section 4.2.1, the experiment on the effectiveness³ of PTFE shadow-covering rings (named as E_r) is conducted by comparing the data with and without shadow-covering rings. In Experiment E_r , videos of the larvae are captured at 73 hpf (Table 4.3). Four larvae are in each well and 12 images are collected with and without rings in the wells respectively, with the dataset denoted as DA- E_r in Table 4.2. The trained U-Net model (with $T_{ms} = 12$, mentioned in Section 4.2.2) is applied to the images to test the possibilities of larvae being detected with two evaluation metrics (ratio of recall R_r and ratio of precision R_p as defined in Section 2.4.2). The results are shown in Fig. 4.8. If no rings are installed into the well, no more than 50% of the larvae can be detected (R_r , dashed line in Fig. 4.8a) and the precision of the detection (R_p , dashed line in Fig. 4.8b) is below 10%. However, if the PTFE rings are installed into the well, more than 90% of larvae can be detected (R_r , solid line in Fig. 4.8a) and the precision of the detection (R_p , solid line in Fig. 4.8b) is around 70%. The results of R_r and R_p with rings installed are both higher than those without rings, as the larvae hide at the edge of the well if no ring is installed, making it impossible to be detected by the image processing pipeline. The result verifies that the PTFE shadow-covering rings are necessary in the experiments of MuLTAP.

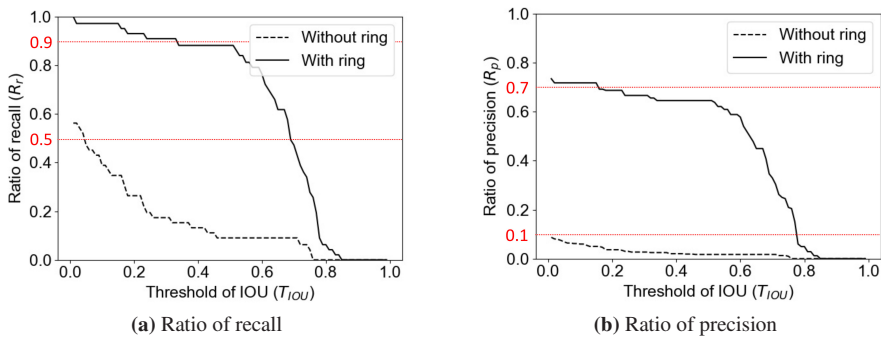


Figure 4.8: The ratios of recall and precision of the data without rings and with rings in the wells, based on larva segmentation method (U-Net model with $T_{ms} = 12$) with 4 larvae in each image [210]. The red lines help compare the results of both cases.

Section 4.2.1 introduces two possible heights of the ring, and the experiment on the height of the PTFE rings (named as E_{r_h}) is conducted to choose the optimal one. The well plate installed with the shadow-covering rings in different heights (r_h : 6 and 12 mm) and filled with corresponding volumes of fish water (2 and 4 mL) are used to conduct the touching on the body of the larvae at

³ The effectiveness of the PTFE shadow-covering rings indicate that the rings can help in improving the performance of the detection of the larvae, thereby increasing the throughput of the experiments.

73 hpf (as outlined in Table 4.3). The dataset collected in Experiment E_{r_h} is denoted as DA-Erh in Table 4.2 including 74 videos (36 videos for 6mm-rings and 38 videos for 12mm-rings). Thus, all collected videos (number denoted as $\#C$) are visually screened to generate the number of the videos with larvae touched validly (number denoted as $\#VT$), the videos with larvae having movements (number denoted as $\#M$) and moving over the rings (number denoted as $\#O$) as well as the percentage (denoted as $PO = \#O/\#M$). The percentage of the videos with larvae moving over the rings ($\#PO$) needs to be reduced for improving the throughput. The results are outlined in Table 4.7 with the following details:

- 6mm-rings: Among 36 collected videos ($\#C$), 31 videos have a valid touching ($\#VT$). In 29 videos ($\#M$), larvae have movements after touching input, and 15 videos ($\#O$) have larvae moving over the rings with the percentage as $\#PO = 51.5\%$,
- 12mm-rings: Among 38 collected videos ($\#C$), 34 videos have a valid touching ($\#VT$). In 27 videos ($\#M$), larvae have movements after touching input, and 9 videos ($\#O$) have larvae moving over the rings with the percentage as $\#PO = 33.3\%$.

A lower number of larvae ($\#O = 9$) move over the 12mm-rings than 6mm-rings ($\#O = 15$). Besides, the percentage ($\#PO = 33.3\%$) of the videos with larvae moving over the 12mm-rings is also lower than the percentage ($\#PO = 51.5\%$) with 6mm-rings. Thus, the result of Experiment E_{r_h} verifies that $r_h = 12$ mm can reduce PO and is chosen in the subsequent experiments.

Age of larvae The larvae in later developmental stages may move more strongly than the younger larvae and move over the shadow-covering rings with higher probability, making the behaviors of the larvae hidden by the rings and reducing the experimental throughput. Thus, the experiment on the age of larvae (named as E_a) is conducted by touching 4 larvae at different ages in each well for the choice of the age of larvae for touch-response experiments (A_t , defined in Section 2.2.1). In Experiment E_a , the well plate installed with the shadow-covering rings in height $r_h = 12$ mm (chosen as above) and filled with corresponding 4 mL fish water is used to conduct the touching on the body of the larvae at different ages (54 hpf and 73 hpf), as outlined in Table 4.3. The dataset collected in Experiment E_a is denoted as DA-Ea in Table 4.2 including 83 videos (45 videos for larvae at 54 hpf and 38 videos for larvae at 73 hpf). The results of Experiment E_a are outlined in Table 4.7 with the following details:

- Larvae at 54 hpf: Among 45 collected videos ($\#C$), 43 videos have a valid touching ($\#VT$). In 34 videos ($\#M$), larvae have movements after touching input, and 0 videos ($\#O$) have larvae moving over the rings with the percentage as $\#PO = 0\%$,

Table 4.7: The experiment results of Experiments E_{rh} , E_a , E_d , and E_{vpw} .

Experiment	Type	#C	#VT	#M	#O	PO ↓
Experiment on the height of the shadow-covering rings (E_{rh})	$r_h = 6$ mm	36	31	29	15	51.5%
	$r_h = 12$ mm	38	34	27	9	33.3%
Experiment on the ages of the larvae (E_a)	54 hpf	45	43	34	0	0%
	73 hpf	38	34	27	9	33.3%
Experiment on the effect of dechoriation (E_d)	With dechoriation	72	64	51	16	31.4%
	Without dechoriation	74	58	43	4	9.3%
Experiment on the average number of videos collected per well (E_{vpw})	Single larva	12	12	#VPW = 1		
	Multiple larvae	45	43	#VPW = 3.58		

#C: number of the collected videos.

#VT: number of the videos with larvae touched (valid touching).

#M: number of the videos with larvae moving.

#O: number of the videos with larvae moving over the rings.

$PO = \#O/\#M$.

#VPW: average number of videos collected per well.

↓: the lower, the better (higher experimental throughput).

- Larvae at 73 hpf: Among 38 collected videos (#C), 34 videos have a valid touching (#VT). In 27 videos (#M), larvae have movements after touching input, and 9 videos (#O) have larvae moving over the rings with the percentage as #PO = 33.3%.

More larvae at 73 hpf (#PO = 33.3%) move over the rings than those at 54 hpf (#PO = 0%). The result of Experiment E_a verifies that younger age of larvae ($A_t = 54$ hpf) is preferably chosen to reduce PO.

Dechoriation Section 1.3.3 introduces that the procedure of dechoriation is needed if the experiments are conducted before the larvae hatch, and the effect of dechoriation on the throughput of MuLTAP is not known. Thus, the experiment (named as E_d) is conducted on touching 4 larvae in each well with and without dechoriation at 27 hpf, aiming to discover the effect of dechoriation (D_e as described in Section 2.2.2). In Experiment E_d , the larvae are used to conduct the touching on the body at 73 hpf, as outlined in Table 4.3. The larvae without dechoriation can hatch spontaneously (out of the chorion) at 48-72 hpf before the experiments. The dataset collected in Experiment E_d is denoted as DA-Ed in Table 4.2 including 72 videos for larvae with dechoriation and 74 videos for larvae without dechoriation. The results of Experiment E_d are outlined in Table 4.7 with the following details:

- With dechoriation: Among 72 collected videos ($\#C$), 64 videos have a valid touching ($\#VT$). In 51 videos ($\#M$), larvae have movements after touching input, and 16 videos ($\#O$) have larvae moving over the rings with the percentage as $\#PO = 31.4\%$,
- Without dechoriation: Among 74 collected videos ($\#C$), 58 videos have a valid touching ($\#VT$). In 43 videos ($\#M$), larvae have movements after touching input, and 4 videos ($\#O$) have larvae moving over the rings with the percentage as $\#PO = 9.3\%$.

Less larvae ($\#PO = 9.3\%$) without dechoriation move over the rings than those with dechoriation ($\#PO = 31.4\%$). The result of E_d verifies that the larvae without dechoriation are preferred to reduce PO . Besides, the effect of dechoriation D_e is concluded as increasing the touch response of zebrafish larvae according to PO .

4.3.4 Evaluation and comparison of SiLTAP and MuLTAP

Touching task of MuLTAP In order to prove that MuLTAP can replicate the touching task the same as SiLTAP, the experiment on the touching task by using MuLTAP (denoted as E_{tm}) is conducted under the same conditions as Experiment E_{tt} in Section 3.3.3 by touching the head, body, and tail of larvae at 30 hpf, 54 hpf, and 78 hpf in 4-larvae case, as outlined in Table 4.3. The predefined positions (head: p_h , body: p_b , tail: p_t) are outlined in Table 3.2, and 109 videos are collected in Experiment E_{tm} (with dataset denoted as DA-Etm in Table 4.2), including 39 videos for head touching, 40 videos for body touching, and 30 videos for tail touching. The reproducibility e_{tr} and robustness r_t (defined in Section 2.4.3) of MuLTAP for touching tasks are evaluated by visually mapping the actually touched point in each video of dataset DA-Etm to the coordinate in \mathbb{P} . Fig. 4.9 visualizes the coordinates of the touched points as a histogram with the fitted normal distributions (mean μ and standard deviation s). The touching reproducibility is represented by the difference between predefined touching points (p_h, p_b, p_t) and the mean of

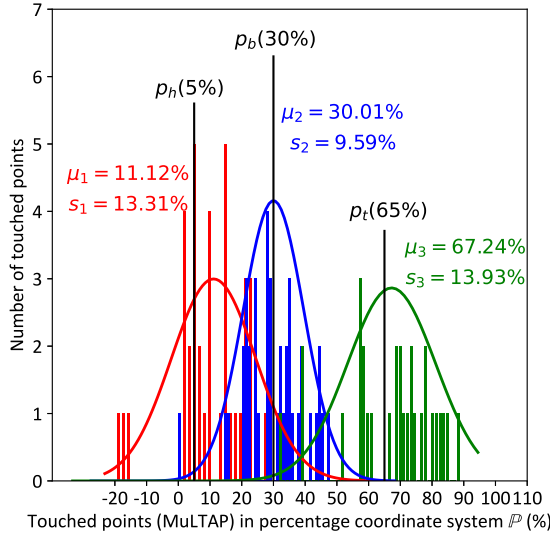


Figure 4.9: The distribution of the touched points with the proposed MuLTAP tested on dataset DA-Etm. The touched points are mapped to \mathbb{P} (seen in Fig. 3.4, red for the head, blue for the body, green for the tail, and black solid lines as the predefined touching positions (p_h, p_b, p_t)). Minus value indicates that MuLTAP does not touch the larva successfully.

touched point distribution via MuLTAP (μ_1, μ_2, μ_3) , with results in Table 4.8. The robustness r_t of MuLTAP for touching task is revealed by the standard deviation s_1, s_2, s_3 of the distribution. The details of the results in Fig. 4.9 and Table 4.8 contain:

- **Reproducibility:** The tail and body touching have higher accuracy, i.e., $e_{tr2} = |\mu_2 - p_b| = 0.01\%$ and $e_{tr3} = |\mu_3 - p_t| = 2.24\%$, compared with the head touching $e_{tr1} = |\mu_1 - p_h| = 6.12\%$,
- **Robustness:** The body touching has higher robustness, i.e., $s_2 = 9.59\%$, compared with the head and tail touching $s_1 = 13.31\%$ and $s_3 = 13.93\%$.

The body and tail touching via MuLTAP have higher reproducibility than that via SiLTAP, but lower in terms of head touching, with details in Table 4.8. The robustness of touching task via MuLTAP is lower than SiLTAP, but Fig. 4.9 shows comparable touching position inputs similar to Fig. 3.9 of SiLTAP. The results above prove that MuLTAP can replicate the touching task via SiLTAP and achieve higher accuracy on body and tail touching by using the multi-larva image processing pipeline.

Table 4.8: The reproducibility of touching task via SiLTAP and MuLTAP. \downarrow indicates that a low value of the evaluation metric is desired.

Evaluation Method	SiLTAP on dataset DA-Ett	MuLTAP on dataset DA-Etm
Head touched reproducibility error e_{tr1} (%)	0.82 \downarrow	6.12 \downarrow
Body touched reproducibility error e_{tr2} (%)	1.13 \downarrow	0.01 \downarrow
Tail touched reproducibility error e_{tr3} (%)	4.46 \downarrow	2.24 \downarrow

Table 4.9: The comparison of the average time used for each step of single- and multi-larva touching strategies used in SiLTAP and MuLTAP.

Time	Part of multi-larva touching strategy	Dataset	Average Time ($\pm\Delta$)
t_{IP}	Image Processing	DA-Et	5.66 s \pm 0.37 s
t_{MLT}	Move the needle to the Larva based on Trajectory	DA-Et	6.30 s \pm 2.73 s
t_{CV}	Capture Video	DA-Et	9.29 s \pm 2.71 s
t_{SV}	Save Video	-	22.29 s \pm 6.26 s
t_{MNW}	Move to the Next Well	-	18.71 s \pm 1.25 s
t_{TPL_s}	Average time used for touching each larva (SiLTAP)	-	55.95 s
t_{TPL_m}	Average time used for touching each larva (MuLTAP)	-	33.00 s

Δ : standard deviation.

Touching efficiency One of the evaluation metrics for the touch-response data acquisition systems is the touching efficiency as defined in Section 2.4.3. The experiments (named as E_t) are conducted to compute the average times used in each part of the touching strategy (defined in Section 4.2.3) and compare the touching efficiency of SiLTAP and MuLTAP. In Experiment E_t , the larvae at 78 hpf are put in the well plate with the shadow-covering rings in height $r_h = 12$ mm and filled with corresponding 4 mL fish water to conduct the touching on the head, body, and tail, respectively (outlined in Table 4.3). The dataset collected in Experiment E_t is denoted as DA-Et, including 111 videos with the results outlined in Table 4.9. For the average time used for saving video and moving the needle to the next well t_{SV} and t_{MNW} , no dataset is used, and they are obtained by the average time of 7 trials of saving videos with different length and moving from one well to

another. According to the results in Table 4.9, the average time per larva in multi-larva case is $t_{TPL_m} = 33.00$ s, which is lower than that of single-larva case $t_{TPL_s} = 55.95$ s, verifying that MuLTAP increases the touch-response experimental data acquisition efficiency compared with SiLTAP.

Average number of videos per well The touch-response experimental throughput is quantified by the average number of videos collected per well ($\#VPW$, defined in Section 2.4.3). In order to prove that MuLTAP increases the experimental throughput compared with SiLTAP, the experiment (named as E_{vpw}) is conducted to count the average number of videos collected for each well. In Experiment E_{vpw} , single larva and 4 larvae are put in the well plate installed with agarose rings (2.5 mL fish water and 4 mL agarose, used in SiLTAP) and PTFE rings (4 mL fish water and rings with $r_h = 12$ mm, used in MuLTAP) for the touching on the body of the larvae (outlined in Table 4.3). The dataset collected in Experiment E_{vpw} is denoted as DA-Evpw, and the results are outlined in Table 4.7 ($\#VPW = \#VT/12$ for 12-well plate) with the following details:

- Single larva: Among 12 collected videos ($\#C$) from 12-well plate, 12 videos have a valid touching ($\#VT$). Therefore, the average number of videos per well in single-larva case is $\#VPW = 1$,
- Multiple larvae: Among 45 collected videos ($\#C$) from 12-well plate, 43 videos have a valid touching ($\#VT$). Therefore, the average number of videos per well in multi-larva case is $\#VPW = 3.58$.

More videos can be collected in the multi-larva case ($\#VPW = 3.58$) than single-larva case ($\#VPW = 1$). The result of Experiment E_{vpw} , as well as Experiment E_t , verifies that MuLTAP increases the throughput of touch-response experimental data acquisition with more videos (3.58 on average) collected in each well and less time used for each larva (33.00 s in Table 4.9). However, not each larva among the 4 larvae in each well can be collected, as some of them are possibly ignored as mentioned in Section 4.2.3 when they have interactive behaviors after the other larvae are touched. Thus, the compromise between the throughput and the number of invalid larvae needs to be considered before the design of the experiments.

4.3.5 Discussions

The experiments in Section 4.3.3 and Section 4.3.4 test the essential parameters to improve the experimental throughput of the touch-response data acquisition procedure:

- The capacity of the well plate is chosen as 4 larvae in each well,

- The height r_h of the proposed PTFE shadow-covering rings is chosen as 12 mm,
- The age of larvae is advised to choose $A_t = 54$ hpf, and
- The dechoriation is needed if experiments of long-term treatments with chemicals on the larvae are conducted, and not suggested if only short-term treatments are considered.

Furthermore, this work proves that MuLTAP can replicate the touching task regarding the reproducibility and robustness, and generate comparable touching positions inputs, similar to SiLTAP. Two more essential evaluation metrics (average time used for touching each larva t_{TPL} and average number of videos per well $\#VPW$) are used for multi-larva touching experiments to be compared with single-larva case, as they represent the essential evaluation of experimental throughput, i.e., the experimental efficiency and number of collected data. These two metrics verify that the proposed MuLTAP improves the efficiency and throughput of touch-response data acquisition. If all parameters are chosen as in Tables 4.3 and A.1, users can expect consistent results.

5 Touch-response behavior inspection

From the touch-response experiments, operators can compare the difference between individual animals based on the data generated by the automated experimental data acquisition process. Particularly, MuLTAP can generate a large-scale dataset, making the visual inspection difficult and painful for operators. In addition, in the touch-response experiments, there are parameters (e.g., the curvature of the C-Bends, as described in Section 2.2.2) being impossible to be quantified by visual inspection. Therefore, the automated inspection procedure (quantifying the touch-evoked behaviors) is needed. According to the proposed two data acquisition systems (SiLTAP and MuLTAP) in Chapter 3 and 4, the customized behavior inspection platform is to be designed. For this goal, this chapter aims to propose an AI-based Multi-larva Touch-response behavior Inspection Pipeline (AMTIP). AMTIP is applied to both single-larva (SiLTAP) and multi-larva (MuLTAP) cases without further modifications.

5.1 Task description and challenges

According to the design of the automated process in Section 2.1, the inspection procedure transfers the raw data collected by the acquisition platform into variables (criteria) that make sense to humans. In the video data collected by SiLTAP and MuLTAP, varying numbers of larvae are in the well, causing complexities of distinguishing larvae during the inspection. As the collected videos have a high frame rate (1000 fps), an efficient tracking procedure is the core of the automated inspection. The design of the inspection procedure faces several challenges:

- The segmentation method (e.g., U-Net) for each frame is insufficient to process the high-frame-rate video, e.g., one 5000-frame video takes approximately 4 hours by using a U-Net. Thus, an algorithm is needed to efficiently generate the positions and segments of the needle and larvae for each frame of the video, e.g., an efficient tracking procedure,
- An accurate initialization of the tracking procedure is important for the following frames of the video, and

- The inspection procedure needs to generate the quantification criteria \mathbf{Q} , i.e., latency time t_l , C-Bend radius average r_a , C-Bend curvature maximum c_m , C-Bend peak time t_{cp} , response time t_r , and moving distance d_m .

Many tracking procedures can be used in the inspection, e.g., optical flow, Kalman filter, and particle filter, as introduced in Section 2.3.6. Besides, different segmentation methods can be used for the initialization of the tracking methods and local segmentations of larvae as discussed in Section 4.2.2. The choices of methods above help in designing AMTIP.

5.2 Multi-larva touch-response inspection pipeline

5.2.1 Overview

Fig. 5.1 visualizes the architecture of AMTIP that transfers the video data collected by SiLTAP and MuLTAP offline and generates the quantification criteria of the touch-response behaviors as defined in Section 2.2.2.

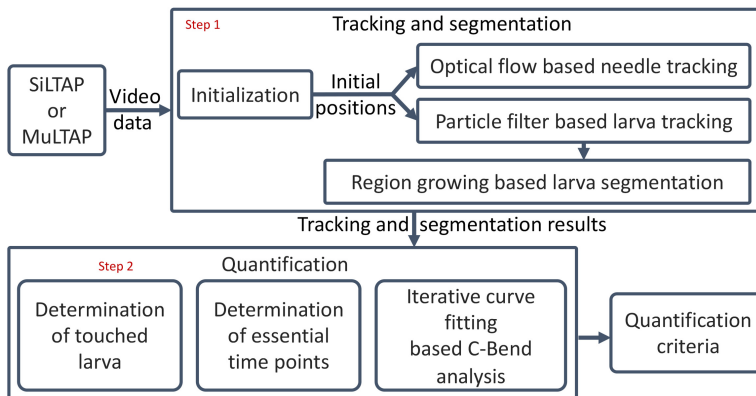


Figure 5.1: The diagram of the AI-based multi-larva touch-response behavior inspection pipeline (AMTIP) [182], including initialization, tracking and segmentation, quantification, and quantification criteria. The steps are marked in red.

AMTIP contains two essential steps:

1. Tracking and segmentation (Step 1 in Fig. 5.1): The initialization is provided by the image processing of SiLTAP and MuLTAP. Given the initial positions, an optical flow is optimized to do the tracking of the needle, a particle filter is optimized to do the tracking of the larvae,

and a region growing is optimized to do the segmentation of the larvae. The results of the tracking and segmentation are used for the following quantification.

2. Quantification (Step 2 in Fig. 5.1): As varying number of larvae are in the well, the touched larva is to be determined. As introduced in Section 2.2.2, the essential time points are determined. Besides, an iterative curve fitting algorithm is proposed to do the C-Bend analysis. The quantification criteria \mathbf{Q} are generated in this step.

5.2.2 Tracking and segmentation

Initialization The initial positions of the needle and larvae obtained from the first frame are vital to the accuracy of the tracking procedure in AMTIP. The proposed SiLTAP and MuLTAP provide the segmentation of the needle and larvae while collecting each video based on a U-Net (as described in Section 4.2.2). In Part 1 of Fig. 5.2, the results of U-Net are used directly to provide the needle position $\{X_0^n, Y_0^n\}$ and larvae positions $\{X_0^l, Y_0^l\}_{l=1,2,\dots}$ in the first frame ($t = 0$), where n indicates the needle and l indicates the larvae.

Optical flow based needle tracking As the needle moves slightly in the video, the tracking of the needle is done by using the optical flow method as mentioned in Part 2 of Fig. 5.2 [100, 200]. Optical flow is based on Taylor series, and the next position of the needle is estimated according to the gradients. As described in Appendix A.3.1, the next needle position $\{X_j^n + u, Y_j^n + v\}$ at frame t_{j+1} is obtained according to needle movements u, v within an adjacent area (with the kernel size denoted as k_{opn}) of the needle position $\{X_j^n, Y_j^n\}$ at frame t_j . However, the estimates from optical flow may miss the needle in the frames where the needle is close to the larva, since the pixels around the larvae are similar to those around the needle. Thus, the optical flow tracking strategy is to be optimized by

$$\{X_{j+1}^n, Y_{j+1}^n\} = \operatorname{argmin}_{\{x_i, y_i\}} \{f(x_i, y_i)\}, \quad \{x_i, y_i\} \in U\{X_j^n + u, Y_j^n + v, t_{j+1}\}_{k_{opn}} \quad (5.1)$$

where $\{X_{j+1}^n, Y_{j+1}^n\}$ is the optimized needle position at frame t_{j+1} , $\{X_j^n + u, Y_j^n + v, t_{j+1}\}$ is the estimated needle position at frame t_{j+1} by optical flow, U indicates the adjacent area, and k_{opn} indicates the kernel size of the adjacent area used during the optimization.

Particle filter based larva tracking For the movement of the larva, optical flow is not useful as the larva moves rapidly. Therefore, a separate tracking method needs to be used. As multiple larvae are considered for collecting touch-response data by using MuLTAP, a multi-larva larva tracking method is used in AMTIP based on the optimization of particle filter. Compared with Kalman filter, particle filter does not rely on a specific assumption [203], and the tracking result is dependent on the score of each particle sampled randomly according to the prior knowledge (the

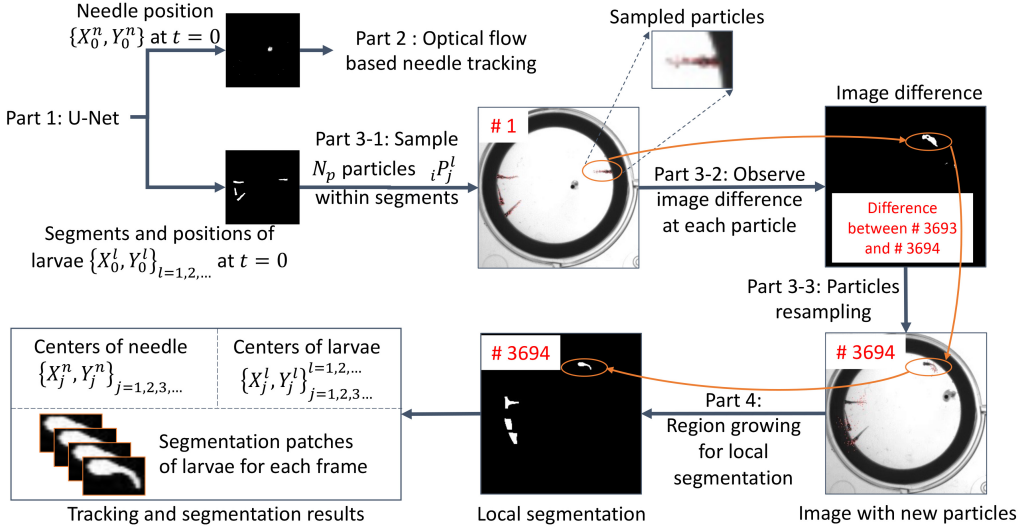


Figure 5.2: The pipeline of the tracking and segmentation of AMTIP [199], including initialization, an optimized optical flow for needle tracking, an optimized particle filter for larva tracking, and an optimized region growing for larva segmentation. The frame numbers of the images are in red and the touched larva is marked in orange circles. Part 1: A U-Net is used to segment the larvae and needle with two binary images for initialization. Part 2: Optical flow based needle tracking is used for predicting the positions of the needle. Part 3: Particle filter based multi-larva tracking. Part 3-1: Particles (marked in orange) are sampled according to the segmented larvae areas. Part 3-2: According to the position of each particle, the image difference between two frames (with an example of the image difference between Frame #3693 and Frame #3694) is observed for the binary probability of the corresponding particle. Part 3-3: The particles with binary probability equal to 0 are resampled around the larvae positions, details in Appendix A.3.2. Part 4: For each larva, the segmentation is achieved by local region growing, discussed in Section 5.2.2. The outputs of the tracking contain the image patches of all larvae as well as the series of positions of the larvae and needle.

previous positions of the larvae in this case). The particle in AMTIP is defined as the position of the larva with corresponding binary probability (either 0 or 1). The particle i at position $\{i x_j^l, i y_j^l\}$ of frame t_j for larva l is denoted as $iP_j^l = \{i x_j^l, i y_j^l, t_j\}$. The binary probability $b\{i x_j^l, i y_j^l, t_j\}$ indicates whether the particle $\{i x_j^l, i y_j^l\}$ belongs to the l -th larva area, shortened as $i b_j^l$. As shown in Part 3-1 of Fig. 5.2, the particles (with number N_p) are sampled randomly according to the segment of each larva to do the following tracking procedure. The binary probability is computed by the image difference based on a pixel-changing threshold T_d chosen heuristically. As described in Appendix A.3.2, if the image difference at particle iP_j^l is over the pixel-changing threshold T_d , the probability $i b_j^l$ is set as 1, otherwise as 0. Compared with the computation of the probability by CNNs in [203], the binary probability computation based on image difference is more sufficient. The particle iP_j^l with the binary probability value equal to zero ($i b_{j+1}^l = 0$) is discarded and resampled in a Gaussian distribution as Equ. A.10 with a heuristic variance

$\Sigma_p = \begin{bmatrix} \sigma_{px}^2 & 0 \\ 0 & \sigma_{py}^2 \end{bmatrix}$ for the range of resampling the particles. The retained particles are used to store the previous information of the positions of the larvae, and the resampled particles are used to search for new potential positions of the larvae.

Region growing based larva segmentation The particle filter only gives the larva position, but the segment for each larva is required for the analysis of the behaviors. Thus, a region growing is optimized to do the local segmentation for each larva, as shown in Fig. A.3. The estimated position ($\{X_{j+1}^l, Y_{j+1}^l\}$) of the larva from the particle filter is used as the seed point of the region growing. In order to reduce the computational costs, the binary label (either 0 or 1) for each pixel of the region growing is obtained according to the pixel value and image gradient based on low and high thresholds of pixel value T_l, T_h and gradient threshold T_g chosen heuristically. As described in Appendix A.3.3, the pixel with value between T_l and T_h and gradient value less than T_g is labeled (segmented) as 1, otherwise as 0. The pixel labeled as 1 is used for growing the region. As the larva area is connected with other objects or noise, the iteration may not stop even if the region covers the larva. Thus, a size threshold (T_{qs}) is set to end the iterations. As the growing of the regions only occurs in the local areas of the larvae, the computational cost is much lower than the global region growing for the entire image or deep learning based methods [212, 213].

5.2.3 Quantification

With the results of the tracking and segmentation, the quantification is done as visualized in Fig. 5.3. The needle and larva positions are used for the determination of touched larva and essential time points, and segmentation patches of larvae for each frame are used for the iterative curve fitting based C-Bend analysis. The quantification criteria \mathbf{Q} (latency time t_l , C-Bend radius average r_a , C-Bend curvature maximum c_m , C-Bend curvature peak time t_{cp} , response time t_r , and moving distance d_m , as defined in Section 2.2.2) are generated by following three steps:

Determination of touched larva The videos contain a varying number of larvae (Larva #1, Larva #2, Larva # l , etc.), so the larva that is actually touched by the needle needs to be firstly distinguished among the larvae in the video by AMTIP for the following quantification (Step 1 in Fig. 5.3). The needle stops at the larva position after touch is applied, so the initial position of the touched larva (X_0^l at $t = 0$) is the closest to the final position of the needle ($X_{t_f}^n$ at $t = t_f$, i.e., time point when video ends).

Determination of essential time points According to the definitions of the latency time t_l and response time t_r (details in Section 2.2.2), the essential time points, including t_1 (touch applied),

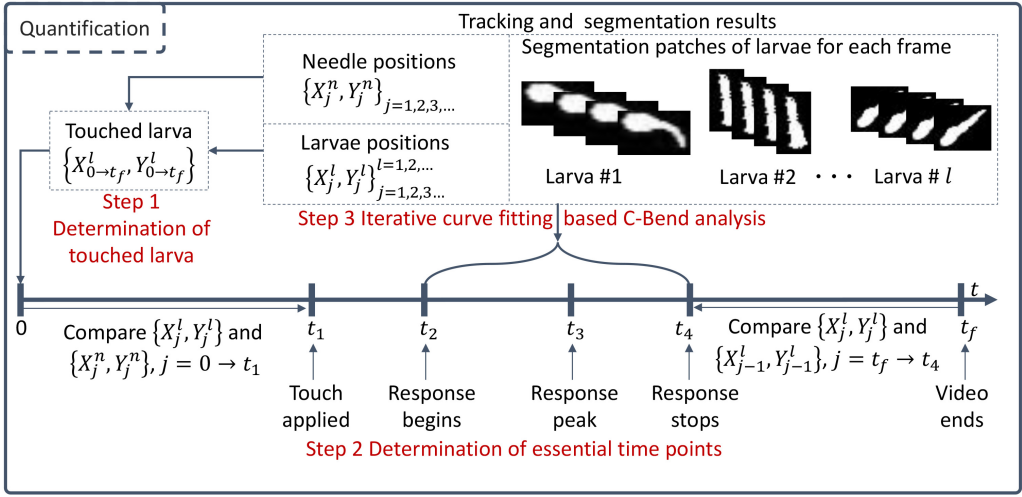


Figure 5.3: The diagram of quantification procedure [182], and the steps are marked in red.

t_2 (response begins), and t_4 (response ends), need to be computed. As shown in Step 2 of Fig. 5.3, the distance between the needle and larva for each frame is computed from $t = 0$ until the time point with the distance lower than a needle-larva distance threshold T_{nl} chosen heuristically, as t_1 (when touch is applied). The time point t_2 is obtained as the time point when the larva moves and response begins (with a larva-moving threshold T_{lm} chosen heuristically). Similarly, the time point t_4 is searched from the time point t_f reversely until the time point when response stops. Consequently, the latency time is computed as $t_l = t_2 - t_1$, and the response time is computed as $t_r = t_4 - t_2$. The moving distance is computed by the sum of the distances between the larva positions in the frames from t_2 to t_4 as

$$d_m = \sum_{j=t_2}^{j=t_4-1} \sqrt{(X_j^l - X_{j+1}^l)^2 + (Y_j^l - Y_{j+1}^l)^2}. \quad (5.2)$$

Iterative curve fitting based C-Bend analysis To quantify the amplitude of the touch-response behaviors, the radius (curvature) of the C-Bend for each frame is to be analyzed, i.e., the exact bending curvature needs to be computed. The average radius of the C-Bends conveys information related to the amplitude of the touch response in the average level, and C-Bend curvature maximum c_m shows the highest response amplitude with the time point at t_3 . Fig. 5.4 shows one example of C-Bend analysis. Given the image patch ("Binary" in Fig. 5.4) for the larva in each frame generated by Section 5.2.2, the larva is skeletonized ("Skeleton" in Fig. 5.4) [214] and an iterative curve fitting algorithm is used to compute the radius (curvature) of the C-Bend in each frame, detailed in Appendix A.4. Fig. 5.4 shows the result of one fitted curve. This algorithm outputs the C-Bend curve to compute the radius and curvature (by using the least square method) for the

l -th larva in each frame. The average of the C-Bend radii r_a and the maximum of the C-Bend curvatures c_m are then computed, as well as the time point of c_m (t_3 , response peak), generating C-Bend curvature peak time $t_{cp} = t_3 - t_2$.

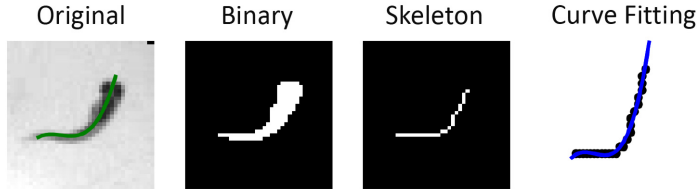


Figure 5.4: Curve fitting includes three steps [175]: binarization, skeleton detection and iterated curve fitting (described in Appendix A.4). The fourth picture shows the third order curve (in blue) fitted after iterations, which is mapped to the original image patch in a green curve.

5.3 Quantification experiment and evaluation of AMTIP

5.3.1 Parameter setting of AMTIP

The characteristics of zebrafish larvae's touch response are still unknown:

- how the larvae respond when different positions are touched,
- how the development of the larvae affects their touch-response behaviors, and
- whether and how the treatment of chemicals can affect the touch-response behaviors.

These questions can be answered by the touch-response experiments on zebrafish larvae quantified by AMTIP. Besides, the effectiveness of the proposed AMTIP is verified by analyzing the detected errors (as defined in Section 2.4.4). The parameters used in AMTIP (as defined in Section 5.2) are outlined in Table 5.1. The kernel sizes of the adjacent areas (k_{op} and k_{opn}) used in the optimized optical flow based needle tracking in Section 5.2.2 are set as 3×3 and 7×7 pixels. The number of the particles used in the particle filter based larva tracking (N_p) is 50 and the standard deviation of the Gaussian distribution used for resampling the particles (σ_{px}, σ_{py}) is set as 7 pixels. In the region growing based local segmentation of the larva (Section 5.2.2), the thresholds T_g , T_l , and T_h are set as 100, 50, and 220, respectively. The parameters above are chosen heuristically and to be verified by the experiments. The average size of larvae is 162.66 pixels, computed by

Table 5.1: The parameters used in AMTIP.

Symbol	Description	Value
k_{op}	Kernel size of the adjacent area of the optical flow based needle tracking in Section 5.2.2 (pixels)	3×3
k_{opn}	Kernel size of the adjacent area of the optimized optical flow in Section 5.2.2 (pixels)	7×7
N_p	Number of the particles of the larva tracking in Section 5.2.2	50
σ_{px}, σ_{py}	Standard deviation of the Gaussian distribution for resampling the particles of the larva tracking in Section 5.2.2 (pixels)	7
T_g	Threshold for the image gradient used in the region growing based local larva segmentation in Section 5.2.2 (pixel value)	100
T_l	The lower threshold for the image value used in the region growing based local larva segmentation in Section 5.2.2 (pixel value)	50
T_h	The higher threshold for the image value used in the region growing based local larva segmentation in Section 5.2.2 (pixel value)	220
T_{qs}	Size threshold for the region growing based local larva segmentation in Section 5.2.2 (pixels)	200
T_{nl}	Needle-larva distance threshold of the quantification in Section 5.2.3 (pixels)	10
T_{lm}	Larva-moving threshold used in the quantification of AMTIP (Section 5.2.3)	50%

the manual annotations of 320 images (4 larvae in each), so the size threshold T_{qs} is set as 200 pixels for safety. Thresholds used in the quantification of AMTIP (T_{nl} and T_{lm} , Section 5.2.3) are chosen heuristically as 10 pixels and 50%.

5.3.2 Quantification experiment definition and parameter setting

In order to quantify the characteristics of zebrafish larvae's touch response and evaluate AMTIP, four sets of quantification experiments are conducted on the zebrafish larvae with the protocol

visualized in Fig. 5.5 on a timeline, and the experimental setup is outlined in Table 5.2. The experiments are defined with two groups:

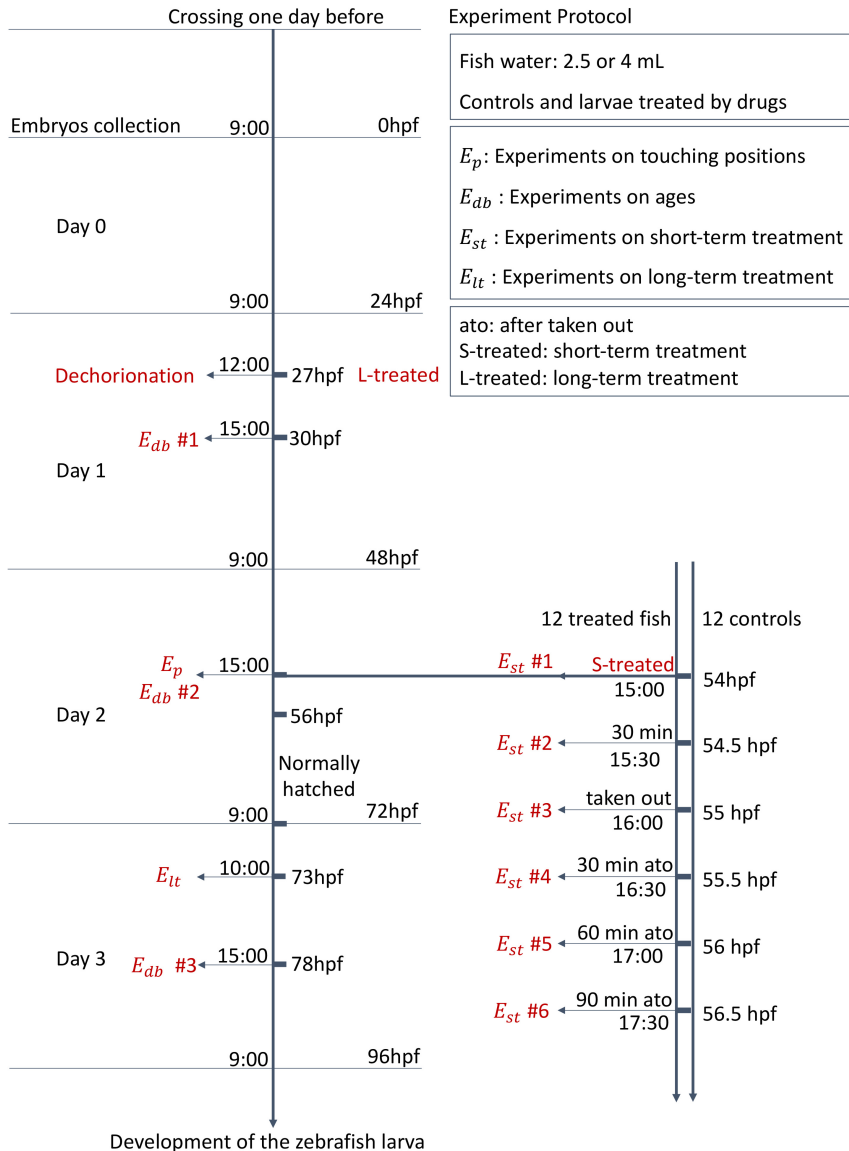


Figure 5.5: Protocol of the quantification experiments E_p , E_{db} , E_{st} , and E_{lt} which are marked in red [175].

Characterization of zebrafish larvae's touch response:

- Experiments on touching positions (denoted as E_p): The touching on different positions of the larvae is expected to evoke different behaviors. Thus, Experiment E_p is conducted by touching wild-type larvae's head, body, and tail at 54 hpf. The proposed AMTIP is used to quantify the data collected in Experiment E_p for verifying that AMTIP can differentiate the touch-response behaviors from the touching of different positions of the larvae. The expected readouts are the quantification criteria Q showing differences of touching positions based on touch-response behaviors.
- Experiments on ages (denoted as E_{db}): As described in Section 1.3.3, the zebrafish larvae display different behaviors during the development (ages). Thus, Experiment E_{db} is conducted by touching the body of wild-type larvae, and three ages are considered, including 30 hpf, 54 hpf, and 78 hpf, as the touch response of zebrafish larvae at different ages are to be characterized. The proposed AMTIP is used to quantify the data collected in Experiment E_{db} for verifying that AMTIP can show results displaying temporal changes of the touch-response behaviors with the development of the larvae. The expected readouts are the quantification criteria Q showing differences of touch-response behaviors based on larvae's development.
- Experiments on short-term treatment (denoted as E_{st}): One of the factors that affect behaviors of larvae is the treatment of chemical compounds [162, 172, 183], e.g., Tricaine (an anesthetic), and the influence of the chemicals can change temporally. Thus, Experiment E_{st} of the short-term treatment on the zebrafish larvae is conducted to analyze whether known behavioral changes can be mapped to the touch-evoked behaviors concerning the treatment time of the chemicals. In Experiment E_{st} , larvae are treated with Tricaine (concentration: $100 \mu\text{mol}/\text{mL}$) at 54 hpf and taken out from the treatment after 60 minutes at 55 hpf. The experiments are conducted every 30 minutes by touching the body of the wild-type larvae and treated larvae at 54 hpf, 54.5 hpf, 55 hpf, 55.5 hpf, 56 hpf, 56.5 hpf, respectively. The proposed AMTIP is used to quantify the data collected in Experiment E_{st} for verifying that AMTIP can generate results showing the changes of the chemical effects temporally on touch-response behaviors in the short term. The expected readouts are the quantification criteria Q showing the changes of the chemical effects in a temporal figure based on touch-response behaviors.

Evaluation of AMTIP:

- Experiments on long-term treatments (denoted as E_{lt}): The effects of the long-term treatments with chemicals are different from those in the short term. In Experiment E_{lt} , the larvae at 73 hpf are put in the well plate to touch the body. Details of the treatments are

Table 5.2: The experiment setup of Experiments E_p , E_{db} , E_{st} , and E_{lt} .

Experiment	E_p	E_{db}	E_{st}	E_{lt}
Data acquisition system	SiLTAP	SiLTAP	SiLTAP	MuLTAP
Volume of water	2.5 mL	2.5 mL	2.5 mL	4 mL
Type of rings	agarose	agarose	agarose	PTFE
r_h : Height of rings	10 mm (4 mL)	10 mm (4 mL)	10 mm (4 mL)	12mm
Age of larvae	54 hpf	30, 54, 78 hpf	54, 54.5, 55, 55.5 56, 56.5 hpf	73 hpf
Dechoriation	27 hpf	27 hpf	27 hpf	27 hpf
Touching positions	head, body, tail	body	body	body
Number of larvae per well	1	1	1	4
Type of larvae	<i>Wild</i> type	<i>Wild</i> type	<i>Wild</i> type and treated by Tricaine	<i>Wild</i> and treated by <i>DMSO</i> (1%), <i>Dia</i> , <i>Caffi</i>
Concentration of chemicals	-	-	100 $\mu\text{mol/mL}$	100 $\mu\text{mol/mL}$
Total number of larvae	12	12	24	113

Wild: Larvae in fish water. *DMSO*: Larvae in Dimethyl sulfoxide.

Dia: Larvae treated by Diazepam. *Caffi*: Larvae treated by Caffeine.

outlined in Table 5.2. Experiment E_{lt} aims to verify whether AMTIP is able to generate the quantification criteria \mathbf{Q} showing reduction (suppression) of touch response of zebrafish larvae with treatments by chemicals. Thus, four experiments are conducted on: *wild* type (without treatment), larvae with Dimethyl sulfoxide (*DMSO*, Section 1.3.3)¹, as well as larvae treated by Diazepam (*Dia*) to reduce the movements [215], and Caffeine (*Caffi*) for

¹ As each treatment is prepared with *DMSO*, the experiments on the larvae with only *DMSO* (1%) are also conducted as controls.

also reduction of movements [215], respectively. Each treatment is in a concentration of $100 \mu\text{mol}/\text{mL}$ for the demonstration. The larvae are dechorionated and treated at 27 hpf for long-term treatment. Besides, the detected errors, i.e, number of videos with no larvae touched $\#NT$ and number of videos with failure of quantification $\#QF$ (definitions in Section 2.4.4), are analyzed to evaluate the effectiveness of AMTIP.

In the experiments above, AMTIP aims to generate the expected readouts as described in the corresponding experiments. In detail, Experiments E_p , E_{db} , and E_{st} are conducted by SiLTAP with agarose ring (4 mL). Experiments E_p , E_{db} are designed to characterize the zebrafish larvae's touch response without any chemical treatments as negative controls, which users of AMTIP can use to compare with their own results for a verification. Besides, Experiment E_{lt} is conducted by MuLTAP with PTFE rings ($r_h = 12 \text{ mm}$) as more chemicals are considered in Experiment E_{lt} and more videos are required to collect at 73 hpf in a higher throughput. Table 5.3 outlines the details of the corresponding datasets generated by Experiments E_p , E_{db} , E_{st} , and E_{lt} as well as the quantification criteria.

Table 5.3: The datasets of Experiments E_p , E_{db} , E_{st} , and E_{lt} .

Description	Dataset denotation	Number		Quantification criteria
Collected in Experiment E_p (on touching positions)	DA-Ep	36 videos		t_l, r_a, t_r, d_m
Collected in Experiment E_{db} (on ages)	DA-Edb	36 videos		t_l, r_a, t_r, d_m
Collected in Experiment E_{st} (on short-term treatment)	DA-Est	144 videos		t_l, r_a, t_r, d_m
Collected in Experiment E_{lt} (on long-term treatment)	DA-Elt	<i>Wild type</i>	24	$t_l, c_m, t_{cp}, t_r, d_m$
		<i>DMSO</i>	27	
		<i>Dia</i>	38	
		<i>Caffi</i>	24	
		113 videos		

In Experiment E_p , 36 videos are collected by touching the head, body, and tail of 12 wild-type larvae at 54 hpf, with the dataset denoted as DA-Ep. In Experiment E_{db} , 36 videos are collected by touching the body of 12 wild-type larvae at 30 hpf, 54 hpf, 78 hpf, respectively, with the dataset denoted as DA-Edb. In Experiment E_{st} , 144 videos (dataset denoted as DA-Est) are collected by

touching the body of 12 controls (wild type) and 12 treated larvae at 54 hpf, 54.5 hpf, 55 hpf, 55.5 hpf, 56 hpf, 56.5 hpf, respectively. Experiment E_{lt} is conducted to collect 113 videos (24 videos for *Wild*, 27 videos for *DMSO*, 38 videos for *Dia*, 24 videos for *Caffi*, with the dataset denoted as DA-Elt). The quantification criteria for Experiments E_p , E_{db} , and E_{st} include latency time t_l , C-Bend radius average r_a , response time t_r , and moving distance d_m for a demonstration of single-larva case. The quantification criteria for Experiment E_{lt} include latency time t_l , C-Bend curvature maximum c_m , C-Bend curvature peak time t_{cp} , response time t_r , and moving distance d_m for the multi-larva case.

5.3.3 Results of zebrafish larvae's touch response

Touch-response analysis on different touching positions Fig. 5.6 displays quantification criteria of each touching position (head, body, and tail), including latency time t_l , C-Bend radius average r_a , response time t_r , and moving distance d_m . As the result of latency time in Fig. 5.6a shows, the head touching causes the quickest response of the larvae, and the tail touching causes the slowest response. Fig. 5.6b shows the result of the averaged radius of the C-Bends that the larvae shape into, which gives a clue of their response strength (amplitude). The larvae with head and tail touched respond in C-Bend with a larger radius compared with the body touching. The larger the radius, the less the response strength. Fig. 5.6c and Fig. 5.6d show the results of response time and moving distance during the whole response. These two results are consistent, i.e., more response time with more moving distance. In short, the head touching causes the larva to have a quick and lasting response, but the body touching makes the larva respond more strongly but less lasting. The results above verify that different inputs of the touching positions deliver different readouts of the behaviors via the inspection by the proposed AMTIP.

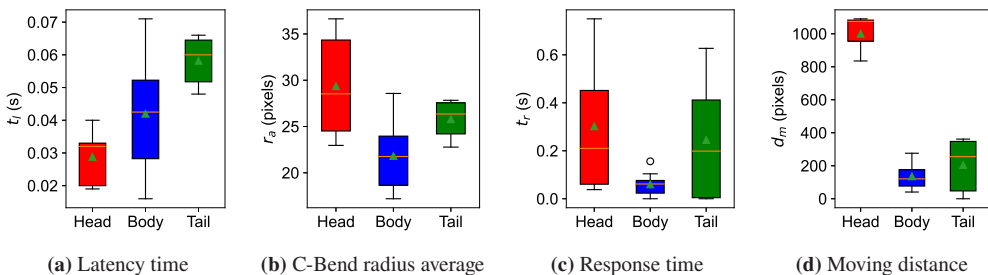


Figure 5.6: Results of the behavior quantification on dataset DA-Ep with different positions touched at 54 hpf of the wild-type larvae, including four quantification criteria, i.e., latency time t_l , C-Bend radius average r_a , response time t_r , and moving distance d_m [175].

Touch-response analysis on different ages The quantification results of the experiment on ages of zebrafish larvae (E_{db}) are sketched in Fig. 5.7. The temporal line charts, with corresponding mean value (black triangle) and the error bar, display the tendency of four quantification criteria (latency time, C-Bend radius average, response time, and moving distance) with the development of the larva. Fig. 5.7a shows no obvious changes in latency time with the larva growing older. Fig. 5.7b, 5.7c, and 5.7d describe a consistent trend, older larvae (78 hpf) respond with less amplitude (larger C-Bend radius) but longer lasting (more response time and moving distance) compared with younger larvae (30 hpf and 54 hpf). The temporal increases shown in Fig. 5.7b, 5.7c, and 5.7d confirm that the age of the larvae is an essential factor to their touch-evoked response.

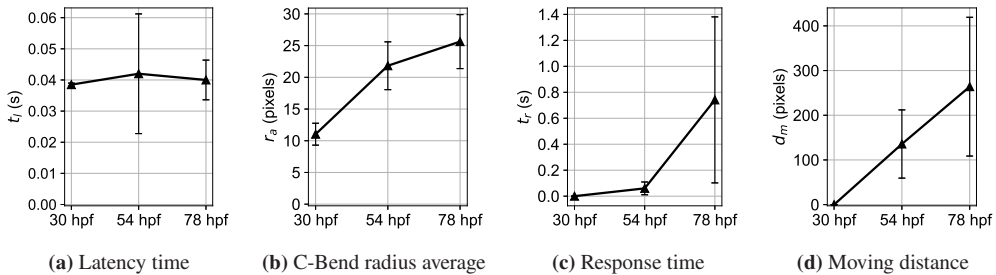


Figure 5.7: The temporal line charts of the behavior quantification on dataset DA-db with the body touched at 30 hpf, 54 hpf, and 78 hpf of the wild-type larvae [175].

Touch-response analysis with short-term treatment by chemicals The experiment on short-time treatment (E_{st}) assumes that AMTIP can generate temporal changes of the touch-response behaviors of zebrafish larvae treated by Tricaine in the short term. The treated larvae are expected to rarely move within the treatment (being anesthetized) and begin to wake up after being taken out from the treatment. The controls are assumed to have no noticeable behavior changes in a short time. Fig. 5.8 visualizes the experimental results of wild-type larvae (controls) and larvae treated by Tricaine. The response of the controls differs only slightly from 54 hpf to 56.5 hpf, but the treated larvae scarcely have the response within the treatment (at 54 hpf and 54.5 hpf) and begin to respond more quickly (less latency time), less strongly (higher C-Bend radius average) and more lasting (more response time and moving distance) after being taken out from the treatment at 55 hpf. The experiment results (at 54 hpf and 54.5 hpf) generated by AMTIP indicate that the short-time treatment with Tricaine can reduce the touch response of zebrafish larvae. Besides, the temporal changes in the line charts of Fig. 5.8 verify the assumption of E_{st} in Section 5.3.2.

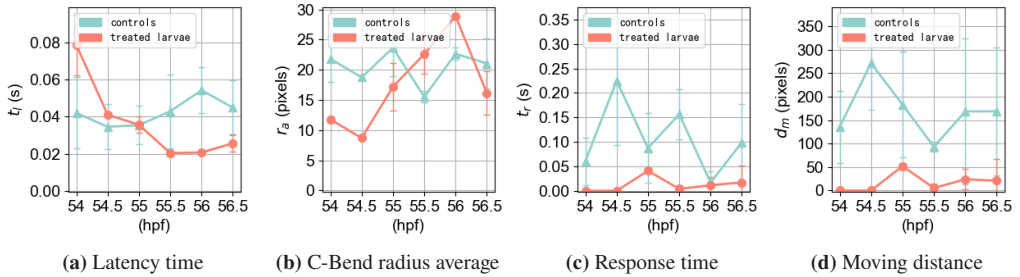


Figure 5.8: Results of the behavior quantification on dataset DA-Est with the body touched at 54 hpf, 54.5 hpf, 55 hpf, 55.5 hpf, 56 hpf, 56.5 hpf of the controls and treated larvae, including four quantification criteria, i.e., latency time t_l , C-Bend radius average r_a , response time t_r , and moving distance d_m [175].

5.3.4 Evaluation of AMTIP

The experiment on long-term treatment (E_{lt}) is conducted and the quantification is run by using AMTIP with the results visualized in Fig. 5.9. The consistent result is shown within each chemical, i.e., the larvae with a longer latency time have lower response strength (lower c_m), shorter time to shape the C-Bend peak (lower t_{cp}), and less response duration (lower t_r and d_m), examples seen from the cases of *Dia* and *Caffi*. This result also proves that the larvae under the treatments of *Dia* and *Caffi* respond less compared with *wild* and *DMSO*, i.e., the reduction of touch response by the two chemicals (assumption of Experiment E_{lt} in Section 5.3.2).

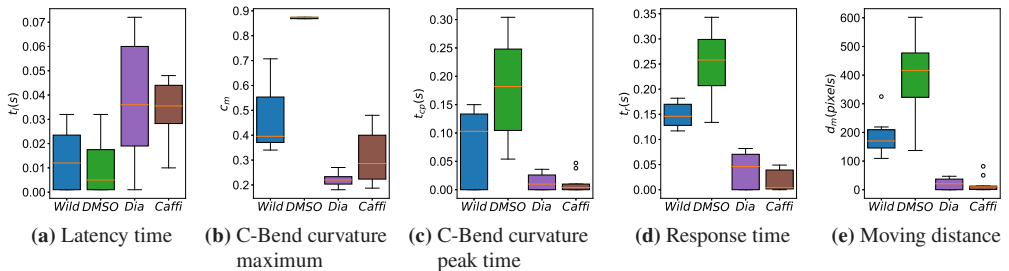


Figure 5.9: Five quantification criteria on dataset DA-Elt with four experiment cases (*wild*, *DMSO*, *Dia*, and *Caffi*) generated by AMTIP, including latency time t_l , C-Bend curvature maximum c_m , C-Bend curvature peak time t_{cp} , response time t_r , and moving distance d_m [199].

Dataset DA-Elt collected in Experiment E_{lt} is used to analyze the detected errors, i.e., number of videos with no larvae touched $\#NT$ and number of videos with failure of quantification $\#QF$. The manually generated $\#NT$ is compared with that generated by AMTIP for a verification.

Table 5.4: The analysis of the detected errors (failure cases) of the proposed AMTIP.

Type	#C	#NT _g	#NT _p	E _{NT}	#QF	E _{QF}
<i>Wild</i>	24	4	7	12.5%	1	5%
<i>DMSO</i>	27	3	8	18.5%	0	0%
<i>Dia</i>	38	4	1	7.9%	8	23.5%
<i>Caffi</i>	24	7	5	8.3%	3	17.6%
Average	-	-	-	11.8%	-	11.5%

#C: number of the collected videos.

#NT_g: ground-truth number of the videos with no larvae touched, generated by visual inspection.

#NT_p: predicted number of the videos with no larvae touched, generated by AMTIP.

E_{NT}: error percentage of the videos without larvae touched between #NT_g and #NT_p.

$$E_{NT} = |\#NT_g - \#NT_p| / \#C.$$

#QF: number of failure of quantification.

E_{QF}: percentage of failure of quantification. $E_{QF} = \#QF / |\#C - \#NT_g|$.

Among the videos collected (#C), shown in Table 5.4, the ground-truth numbers of the videos with no larvae touched (#NT_g, generated by visually screening) are compared with the numbers output from AMTIP (#NT_p), with the percentage ($E_{NT} = |\#NT_g - \#NT_p| / \#C$). As well, the number of failure of quantification (#QF) is given with the percentage ($E_{QF} = \#QF / |\#C - \#NT_g|$). The number of videos with failure of quantification #QF with the corresponding percentage E_{QF} evaluates the failure rate of AMTIP caused by various factors, such as the overlapping of the larvae and needle or similar brightness of the larvae to the background. Besides, AMTIP can generate #NT and #QF automatically and find 88.2% (i.e., 1-11.8%, "Average" in Table 5.4) of videos without any larvae touched on average. Besides, 11.5% of valid videos (#C - #NT_g, "Average" in Table 5.4) cannot be quantified by AMTIP (failure cases). In addition, the larvae under the treatment of *Dia* are assumed to have a response scarcely. Thus, the output of latency time is expected to be infinite, and the other quantification criteria (C-Bend curvature maximum, C-Bend curvature peak time, response time, moving distance) are expected to be 0. However, AMTIP can only generate finite numbers less than 15 s (the duration of videos set by SiLTAP and MuLTAP, as listed in Table 3.2). From Fig. 5.9a, the latency time of *Dia* is still useful to be compared with the controls as it shows a much longer latency time than

those of *wild* and *DMSO*. Furthermore, the results of *Dia* and *Caffi* in Fig. 5.9b-5.9e are above zero, caused by the following reasons:

- some larvae still have a slight response,
- the movements of the needle can push the larvae away (fake response), and
- the tracking procedure generates the changes of the larvae positions because of the slight environment changes.

Nonetheless, the results of *Dia* and *Caffi* in Fig. 5.9b-5.9e are much lower than those of *wild* and *DMSO*. In other words, even with slight variance, the proposed AMTIP verifies the assumption on treatments *Dia* and *Caffi* that reduce the touch response of zebrafish larvae. Finally, with the proposed efficient tracking and segmentation procedure, AMTIP achieves the quantification in a high efficiency (frame rate: on average 63 ms per frame on CPU).

5.3.5 Discussions

The results verify that the proposed inspection pipeline AMTIP works as automated quantification tool for the single-larva or multi-larva touch-response data in a high frame rate. AMTIP has the following advantageous strategies:

- The segmentation via U-Net for the first frame of the video data (initialization of AMTIP) is more accurate than the region growing based local segmentation in the following frames. Thus, it is advantageous to determine the time point t_1 (when the touch is applied), as well as the actually touched larva, by the position of the needle in the final frame and the initialized positions of the larvae in the first frame,
- The centers of the larvae can change slightly but constantly during the tracking procedure. In other words, AMTIP may generate a large moving distance, even if the larvae do not move. Thus, it is advantageous to use the change of each particle used in the particle filter based larva tracking (Section 5.2.2) to represent the movements of the larvae, instead of using the change of the larva center, and
- The larva can move slowly (no significant changes of pixels) for a moment and start moving strongly again. Thus, it is advantageous to determine the time point t_4 (when the touch response stops) by searching the stop of the larva's movement from the last frame reversely to the previous frames. Besides, it is better to achieve the quantification after the tracking and segmentation of all frames in the video, making it possible to consider the global information of the video.

However, some drawbacks still need to be considered when users apply AMTIP. The tracking procedure and local segmentation of the larvae are the keys of AMTIP, but they may fail in the following cases:

- the larvae overlap with each other when moving,
- the well edge area has similar brightness with the larvae, and
- the needle overlaps with the larvae.

As shown in Table 5.4, AMTIP fails to quantity 11.5% of valid videos, and these videos are not considered in the experimental results. It is essential to conduct the experiments on a large scale, so the proposed MuLTAP and AMTIP are vital in such cases.

6 Touch-response pattern recognition

Drug screening experiments are commonly used to screen the behaviors of zebrafish larvae, which is also essential to the pharmaceutical industry [216]. The interface of touch-response screening on zebrafish larvae is implemented based on the introduction of SiLTAP, MuLTAP, and AMTIP. The feasibility of the interface is proved by the application of it to drug screening. The experiment results for AMTIP prove that the chemical compounds can change the touch-response behaviors of zebrafish larvae, i.e., increasing or suppressing the movements of the larvae. A drug effect pattern is defined as the consistent changes of zebrafish behaviors with treatments by chemical compounds (as experimental group) compared with the wild-type larvae (without any treatments, as control group). Thus, the compounds showing the same functioning mechanisms, i.e., changing the behaviors of zebrafish larvae to the same extent, are defined to have the same drug effect pattern. Besides, many chemical compounds change behaviors to different extents according to their specific functioning mechanisms. Screening on numerous chemical compounds generates various drug effect patterns forming into a database, which is used to predict the potential functioning mechanisms for unknown compounds. The generation and prediction procedures mentioned above are known as pattern recognition of drug effects. This chapter aims to propose an AI-based pipeline for the Pattern Recognition of Touch response based Database (PRTD) to analyze the patterns from the inspection results Q of AMTIP. Each drug pattern in PRTD is defined as the change of quantification criteria Q in experimental group compared with Q of control group. PRTD is expected to output the effect pattern database for known chemical compounds and predict the effect patterns for unknown chemical compounds. Besides, drug screening experiments are conducted to prove whether the proposed PRTD generates the expected outcomes.

6.1 Task description and challenges

The task of PRTD is visualized in Fig. 6.1. Given the inputs from AMTIP and domain expert, PRTD generates two outputs to users. PRTD has two inputs with following details:

- Quantification criteria \mathbf{Q} : AMTIP generates a group of criteria to quantify the touch-response behaviors, i.e., $\{t_l, r_a, c_m, t_{cp}, d_m, t_r\}$, and a vector \mathbf{Q} is used to present those criteria,
- Hypotheses of drug effect patterns \mathbf{H} : In the design of full automation interface, domain expert provides essential domain knowledge to help draw the conclusions (Fig. 2.2), and the domain knowledge from biologists in drug screening experiments are the known pattern name of each drug obtained according to their biological theories. The pattern names of all drugs are used as hypotheses of drug effect patterns, denoted by a vector \mathbf{H} .

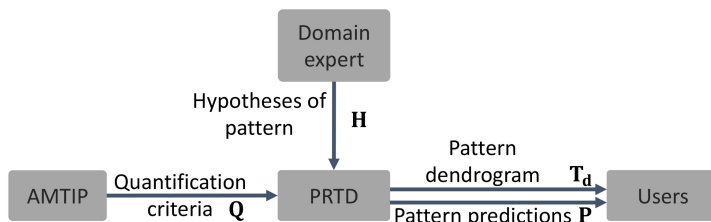


Figure 6.1: The inputs and outputs of PRTD. Inputs: quantification criteria \mathbf{Q} from AMTIP and hypotheses of drug effect patterns \mathbf{H} from domain expert. Outputs: pattern dendrogram \mathbf{T}_d and pattern predictions \mathbf{P} to users.

Based on those two inputs, PRTD aims to generate two outputs:

- a pattern dendrogram of the drug effects, denoted as \mathbf{T}_d ,
- the pattern predictions of the unknown drug effects, denoted as \mathbf{P} .

The design of PRTD faces two problems:

- The patterns shown in the data collected in the process rely on many factors, and the hypotheses of the patterns can be inconsistent to the actually collected data. For each chemical compound, the biological researchers provide the hypothesis of the drug effect on the zebrafish, e.g., increasing (\mathbf{H}_1) and decreasing (\mathbf{H}_2) the control variable (here, latency time t_l in Fig. 6.2). However, the actual effect pattern of Drug b is increasing the latency time compared with control group, inconsistent with the hypothesis \mathbf{H}_2 , similar to Drug c.
- The individual data points show variance even under the same conditions, and different individuals under various conditions can show the same patterns. For example, as shown in Fig. 6.2, two individual larvae show the effect pattern of decreasing the latency time compared with control group, which is inconsistent to the other larvae under the same treatment of Drug a.

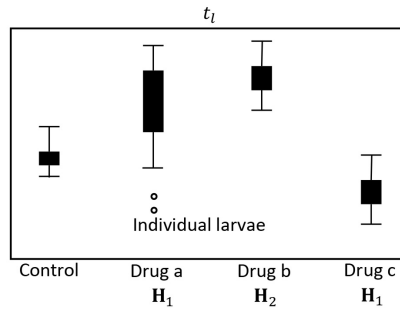


Figure 6.2: The illustration of the data used for PRTD in box plots: latency time t_l of zebrafish larvae under the treatments of three example drugs (Drug a, Drug b, Drug c) compared with control group. The hypotheses of effect patterns for Drug a, Drug b, Drug c are H_1 , H_2 , and H_1 .

Therefore, it is a challenging task to choose the method to do the processing of the data with the inconsistency above between the hypotheses and the actual data patterns. Machine learning methods in supervised or unsupervised manners are to be explored according to the description in Section 2.3.6.

6.2 AI-based drug effect pattern database

6.2.1 Overview

Exploring supervised learning based classifiers with annotations from the domain expert (hypotheses of effect pattern H) is challenging for the design of PRTD since the data is not consistent as discussed in Section 6.1, causing a wrong classifier. Thus, PRTD proposes to complete the pattern analysis task by using an unsupervised learning model based on quantification data Q output from AMTIP. The overall workflow of PRTD is visualized in Fig. 6.3. The quantification criteria from AMTIP for control group and experimental group are denoted as Q_c and Q , respectively. PRTD aims to generate two outputs, i.e., the pattern dendrogram of the drug effects and pattern predictions of the unknown drug effects as described in Section 6.1, by four steps: feature engineering, hierarchical clustering, pattern mapping, and pattern proposing.

6.2.2 Feature engineering

According to the definition of drug effect pattern, PRTD aims to analyze the changes (differences) between the experimental and control groups (Q and Q_c). Because of the inconsistency within the data, a statistical distance method is preferred to compute the differences instead of comparing

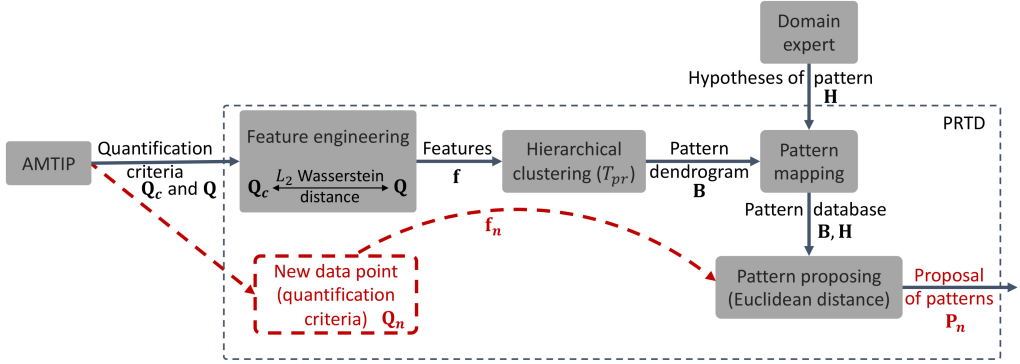


Figure 6.3: The workflow of PRTD, including feature engineering, hierarchical clustering, domain expert, pattern mapping, and pattern proposing. Q_c and Q : quantification criteria results for control group and experiment group, respectively. f : features generated by the L_2 Wasserstein distance method. B : branches of the pattern dendrogram (also the clusters). H : hypotheses of effect pattern. The pattern database contains B and H . The prediction for a new data point Q_n is marked in red with the proposal of patterns P_n .

the individual data points. A L_2 Wasserstein distance method helps in this case owing to its robustness to outliers and low computational cost. Besides, the L_2 Wasserstein distance method considers both the mean and variance of data for the computation of the distance [81]. The distance results of L_2 Wasserstein distance method are used as features f for the data of the experimental group Q . Assumed that the data in the control group and experimental group Q_c and Q are in D dimensions, i.e., $Q_c \in R^{N_c \times D}$ and $Q \in R^{N \times D}$, where N_c and N are the number of the data points in Q_c and Q respectively. The features ($f \in R^{1 \times D}$) of data for the experimental group Q are computed for each dimension separately, formulated as

$$f_i(Q_i, Q_{ci}) = s \cdot \left(\|\bar{Q}_i - \bar{Q}_{ci}\|_2^2 + \sigma_i + \sigma_{ci} - 2\sqrt{\sigma_{ci}\sigma_i}\sqrt{\sigma_{ci}} \right),$$

$$\text{with } i = 1, 2, \dots, D, \quad s = \begin{cases} 1 & \text{if } \bar{Q}_i > \bar{Q}_{ci} \\ -1 & \text{else} \end{cases} \quad (6.1)$$

where $f_i \in R^1$ is the feature for dimension i , $Q_i \in R^{N \times 1}$ and $Q_{ci} \in R^{N_c \times 1}$ are the dimension i of Q and Q_c , \bar{Q}_i and \bar{Q}_{ci} are the mean of Q_i and Q_{ci} , and σ_i and σ_{ci} are the variance of Q_i and Q_{ci} . $f = \{f_i\}_{i=1,2,\dots,D}$ are the features of data for the experimental group Q . In particular, the features of control group are zero, i.e., $f_c = 0$, as the distance of control group to itself is zero.

6.2.3 Hierarchical clustering and pattern mapping

PRTD aims to generate a pattern database of numerous chemical compounds, and the features of each compound are generated by the feature engineering as described in Section 6.2.2, e.g., $\mathbf{f}_1 \sim \mathbf{f}_{11}$ in Fig. 6.4. The pattern database is built by generating a pattern dendrogram \mathbf{T}_d and mapping the hypotheses of the drug effect patterns \mathbf{H} to the pattern dendrogram.

Pattern dendrogram A hierarchical clustering (Ward’s method, [88]) is commonly used to generate a pattern dendrogram displaying the data patterns in a tree structure \mathbf{T}_d . The clustering is applied to the features \mathbf{f} of all chemical compounds based on the Euclidean distance:

$$\text{Ed}(\mathbf{f}_a, \mathbf{f}_b) = \sqrt{\mathbf{f}_a \cdot \mathbf{f}_a - 2(\mathbf{f}_a \cdot \mathbf{f}_b) + \mathbf{f}_b \cdot \mathbf{f}_b} \quad (6.2)$$

where \mathbf{f}_a and \mathbf{f}_b are the features of two experimental groups a and b , respectively. The hierarchical clustering computes distance for the data of all experimental groups. The clusters are determined by a distance threshold T_{pr} , as visualized in Fig. 6.4 (orange line), where different colors indicate different clusters. For example, $\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3$ are close to each other, therefore they are clustered in one branch \mathbf{B}_1 in purple. Those branches ($\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3, \mathbf{B}_4$) are used as the pattern dendrogram of the drug screening data.

Pattern mapping Each branch of the pattern dendrogram \mathbf{B} needs to be named by the hypotheses of the drug effect patterns \mathbf{H} provided by biologists, which is done by the pattern mapping between \mathbf{B} and \mathbf{H} in Fig. 6.4. Each data in the experimental group (with features \mathbf{f}_a) belonging to the branch \mathbf{B}_j has its corresponding hypothesis of effect pattern \mathbf{H}_k . Besides, a branch can be probably mapped to different hypotheses of effect patterns. For example, Branch \mathbf{B}_3 (green in Fig. 6.4) contains three data points with features $\mathbf{f}_7, \mathbf{f}_8, \mathbf{f}_9$, and two hypotheses of patterns \mathbf{H}_1 (rectangle) and \mathbf{H}_2 (triangle) are mapped to Branch \mathbf{B}_3 .

6.2.4 Pattern proposing

The final output of PRTD is the prediction of drug effect pattern for an unknown chemical compound. As the actual patterns shown in the data are not consistent to the hypotheses of the drug effect patterns given by biologists, PRTD aims to propose different potential patterns for each unknown compound instead of predicting only one pattern. Based on the mapping results between branches of the pattern dendrogram \mathbf{B} and hypotheses of the drug effect patterns \mathbf{H} in Section 6.2.3, a database of drug screening is established. The prediction of effects for unknown chemical compounds is done as visualized in Fig. 6.4. Given the features \mathbf{f}_n of a new chemical compound, the potential patterns \mathbf{P}_n are generated according to the Euclidean distance between

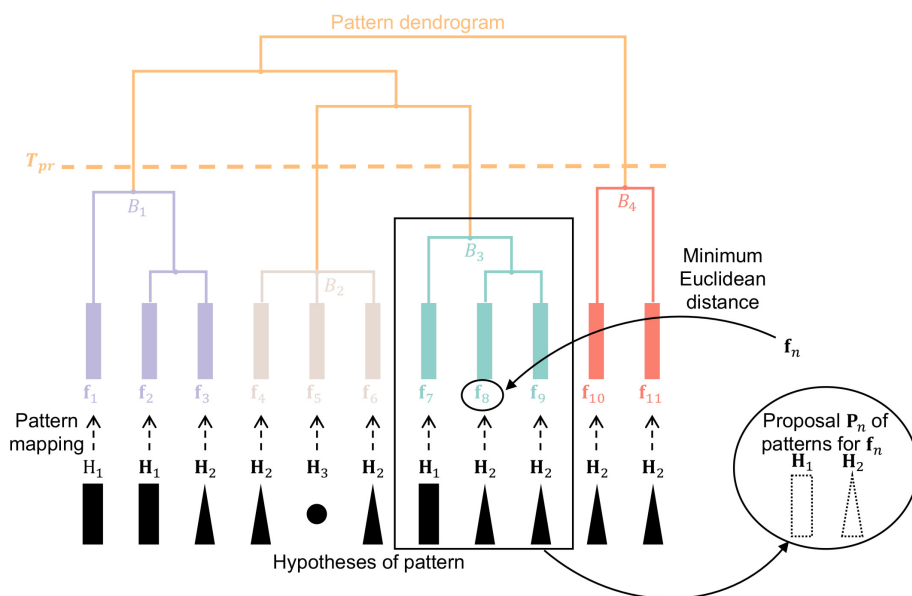


Figure 6.4: The database built by the proposed PRTD. The branches (clusters) of the pattern dendrogram are marked in different colors, and different shapes in black denote the hypotheses pattern from the domain expert. Four clusters (B_1 , B_2 , B_3 , B_4) and three patterns from hypotheses (H_1 , H_2 , H_3) are demonstrated. The dashed arrows indicate the pattern mapping.

the features f_n of the new chemical compound and the features of all the compounds in the database, as Equ. 6.2. The features in the database with a minimum Euclidean distance to the features of the new data f_n are selected (f_8 in Fig. 6.4). All the hypotheses of effect patterns in Branch B_3 are proposed for the new data f_n , i.e., $P_n = \{H_1, H_2\}$ in dashed shapes. Each branch may contain more than one hypothesis of pattern, so that PRTD can propose several potential patterns for each new chemical compound.

6.3 Drug screening experiment to evaluate PRTD

6.3.1 Experiment setup

PRTD aims to analyze the data patterns of the data collected in the automated interface and can be applied to the touch-response screening. The drug screening experiments based on touch-response behaviors of zebrafish larvae are conducted to verify performance of PRTD on pattern recognition, and the experiments are denoted as E_{pr} . The larvae are treated by a group of chemicals, each with its corresponding hypothesis of effect pattern H_i . The experiments are conducted by MuLTAP,

and the resulting data are inspected by AMTIP which outputs the quantification criteria to PRTD. The aim of the experiments is to evaluate whether PRTD can predict the same drug effect patterns to hypotheses of effect patterns \mathbf{H} provided by biologists. Besides, PRTD is expected to generate a low number of pattern proposals with a low failure percentage. The protocol of Experiment E_{pr} is visualized in Fig. 6.5 on a timeline.

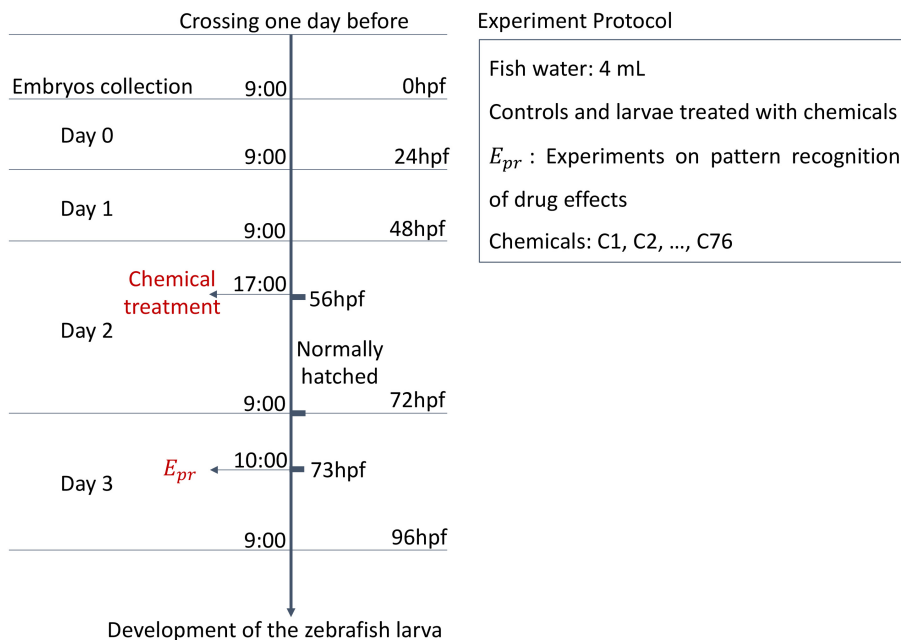


Figure 6.5: The protocol of Experiment E_{pr} , which is marked in red.

The wild-type larvae (controls, denoted as C0) and larvae treated with 76 chemical compounds (denoted as C1 - 76) are considered in Experiment E_{pr} . The hypotheses of the chemical compound effect patterns \mathbf{H} are listed in Table 6.1, where C* indicates the chemical compounds and totally 10 different hypotheses of the chemical compound effect patterns are considered. The dataset generated in Experiment E_{pr} is outlined in Table 6.2. The data are collected via MuLTAP and quantified via AMTIP. Each data point contains five quantification criteria, i.e., latency time t_l , C-Bend curvature maximum c_m , C-Bend curvature peak time t_{cp} , response time t_r , and moving distance d_m , and is represented by \mathbf{Q} . The dataset is split into training and testing datasets (denoted as DA-Epr-TRAIN and DA-Epr-TEST, respectively). As visualized in Fig. 6.6, dataset DA-Epr-TRAIN is used to build the effect pattern database of the chemical compounds via PRTD as described in Section 6.2, and dataset DA-Epr-TEST is used to test the capability of PRTD predicting effect patterns for unknown compounds. Thus, the compounds in dataset

Table 6.1: The hypotheses of effect pattern \mathbf{H} of chemical compounds C0 - C76 provided by biologists.

Compounds	\mathbf{H}
C3, C6, C8, C15, C16, C18, C23, C44, C46, C50, C55, C62, C67, C73	GABAA pore blocker
C4, C5, C19, C20, C21, C22	vesicular ACh transport antagonist
C12, C13, C10, C11, C14, C17	nAChR orthosteric agonist
C24, C25, C35, C41, C43, C56, C64, C65, C63, C66	nAChR orthosteric antagonist
C1, C9, C26, C30, C33, C51, C52, C53, C57, C58, C59, C60, C61, C68, C69, C74, C75, C76	TRPV agonist
C27, C28, C29, C34, C36, C37, C38, C39, C40, C42, C45, C47, C48, C49, C54, C70, C72	GABAA allosteric antagonist
C31, C32	RyR agonist
C71	Na channel
C2, C7	Unknown; likely neurotoxin
C0	Wild type (controls)

DA-Epr-TRAIN are not the same as those in dataset DA-Epr-TEST, and both DA-Epr-TRAIN and DA-Epr-TEST have the ground-truth effect patterns, i.e., the hypotheses of effect patterns \mathbf{H} provided by the biological expert. The database built by dataset DA-Epr-TRAIN includes the hypotheses of patterns (\mathbf{H} , different black shape in Fig. 6.6) mapped to the chemical compounds with features \mathbf{f} . PRTD uses the database to propose the drug effect patterns (\mathbf{P}_n , different dashed shapes in Fig. 6.6) for the testing dataset (DA-Epr-TEST), i.e., the unknown compounds with features \mathbf{f}_n . Therefore, the evaluation of PRTD can be done by comparing the predictions \mathbf{P}_n from PRTD and ground-truth effect patterns \mathbf{H}_n provided by biologists. Two evaluation metrics are proposed to evaluate the performance of PRTD, including the number of pattern proposals $\#N_{pp}$ and percentage of failure of pattern predictions P_f on dataset DA-Epr-TEST, as defined in Section 2.4.5. A desirable PRTD is expected to generate a low number of pattern proposals $\#N_{pp}$ and low percentage of failures of pattern predictions P_f on dataset DA-Epr-TEST.

Additionally, section 4.3.3 indicates that experiments without the dechoriation procedure generate more experimental data, so the treatments are applied to the larvae at 56 hpf (when the larvae hatch spontaneously without the dechoriation) for the long-term treatment, and Experiment E_{pr} is conducted at 73 hpf. The concentration of the chemical compounds is set to 50 $\mu\text{mol}/\text{mL}$.

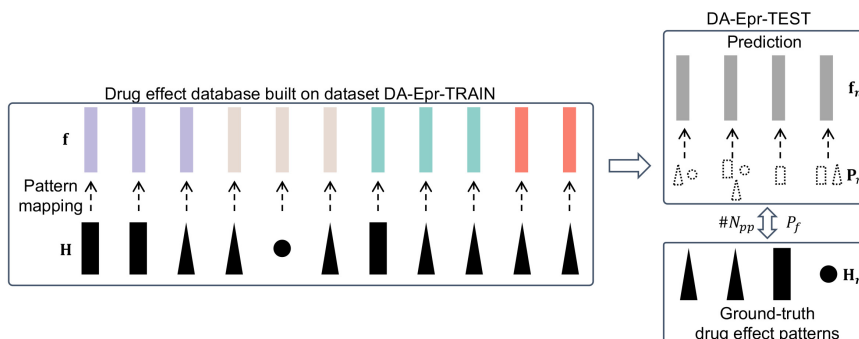


Figure 6.6: The evaluation procedure of PRTD on the drug effect prediction. The drug effect database is built on dataset DA-Epr-TRAIN, which is tested on dataset DA-Epr-TEST by comprising the prediction \mathbf{P}_n from PRTD and ground-truth \mathbf{H}_n of drug effect patterns.

Table 6.2: The dataset and evaluation metrics of Experiment E_{pr} , including the training and testing datasets from C0 to C76. \downarrow indicates that a low number of pattern proposals $\#N_{pp}$ and a low percentage of failures of pattern predictions P_f are desired.

Description	Dataset denotation	Number of compounds	Number of data	Evaluation metrics
Training dataset collected in E_{pr}	DA-Epr-TRAIN	61	322	$\#N_{pp} \downarrow$, $P_f \downarrow$
Testing dataset collected in E_{pr}	DA-Epr-TEST	15	85	

6.3.2 Evaluation results

Fig. 6.7 visualizes the results for number of pattern proposals $\#N_{pp}$ (left y -axis) and percentage of failures of pattern predictions P_f (right y -axis) concerning the threshold T_{pr} (x -axis) used in the hierarchical clustering of PRTD (details in Section 6.2.3). For each threshold T_{pr} , the

corresponding number of pattern proposals $\#N_{pp}$ and percentage of failures of pattern predictions P_f are generated by the comparison between the predictions from PRTD and ground-truth drug effect patterns (visualized in Fig. 6.6) according to Equ. 2.9 and 2.10. Low values of $\#N_{pp}$ and P_f are desired in PRTD. The results of Experiment E_{pr} on dataset DA-Epr-TEST in Fig. 6.7 are

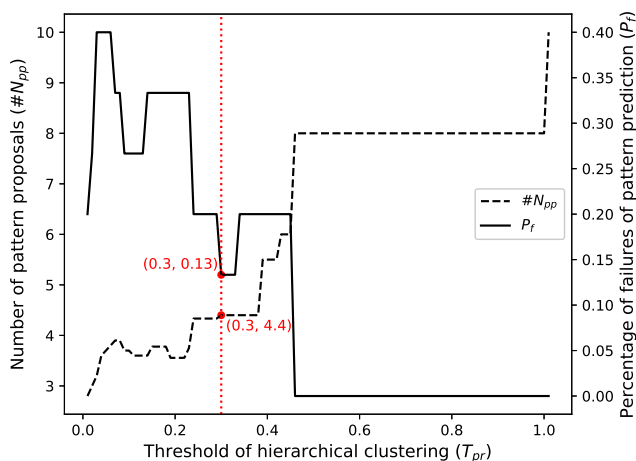


Figure 6.7: Evaluation result of Experiment E_{pr} , including the number of pattern proposals $\#N_{pp}$ and failure of pattern predictions P_f concerning the threshold used in the hierarchical clustering (T_{pr}) in PRTD.

summarized as:

- With the increase of threshold T_{pr} , the number of pattern proposals $\#N_{pp}$ increases but the percentage of failures of pattern predictions P_f decreases.
- With a low threshold, e.g., $T_{pr} = 0 \sim 0.2$, PRTD can generate 3 \sim 4 pattern proposals for each chemical compound on average ($\#N_{pp}$, dashed line in Fig. 6.7 with left y -axis), and the percentage of failures of pattern predictions (P_f , solid line in Fig. 6.7 with right y -axis) is around 25% \sim 40%. This result indicates that PRTD proposes a low number of possible drug pattern proposals $\#N_{pp}$ with a low threshold T_{pr} , but resulting in a high percentage of failures of pattern predictions P_f .
- With a high threshold, e.g., $T_{pr} = 0.6 \sim 1.0$, PRTD can generate 8 \sim 10 pattern proposals for each chemical compound on average ($\#N_{pp}$, dashed line in Fig. 6.7 with left y -axis), and the percentage of failures of pattern predictions (P_f , solid line in Fig. 6.7 with right y -axis) is under 5%. This result indicates that PRTD proposes almost all possible drug pattern proposals (totally 10 patterns in Table 6.1) with a high threshold T_{pr} in hierarchical

clustering, which is not optimal as a low number of pattern proposals is desired. However, a high threshold T_{pr} can guarantee a low percentage of failure of pattern predictions P_f .

- A compromise between the number of pattern proposals $\#N_{pp}$ and percentage of failures of pattern predictions P_f is made at $T_{pr} = 0.3$, where $\#N_{pp} = 4.4$, $P_f = 0.13$, which can be adjusted according to the requirements of users. Increasing the threshold T_{pr} causes the increase of $\#N_{pp}$ and P_f , and decreasing T_{pr} causes the increase of P_f as well. Thus, $T_{pr} = 0.3$ is an optimal choice to generate low $\#N_{pp}$ and P_f . The drug pattern effect dendrogram generated on dataset DA-Epr-TRAIN with $T_{pr} = 0.3$ is shown in Fig. 6.8, with the corresponding ground-truth **H**. With $T_{pr} = 0.3$, PRTD generates 4.4 possible drug pattern proposals ($\#N_{pp}$) on average for each compound in dataset DA-Epr-TEST, and PRTD fails to predict drug effects for 13% of the compounds in dataset DA-Epr-TEST.

The results above evaluate PRTD with two evaluation metrics P_f and $\#N_{pp}$. PRTD clusters the data of known chemical compounds into patterns as shown in Fig. 6.8 and predicts the potential drug effects for unknown chemicals within 4.4 possible drug pattern proposals and in a 13% failure rate.

6.3.3 Discussions

The drug screening experimental results in Section 6.3.2 prove the proposed PRTD can predict the drug effect patterns for the unknown compounds in dataset DA-Epr-TEST within a limited failure rate. The pattern prediction strategy of PRTD has the following advantages:

- PRTD considers the statistical distance between the data of treatments and controls instead of the individual data points of each compound, reducing the influence of individual differences,
- PRTD uses an unsupervised learning method (hierarchical clustering) to build the drug effect dendrogram and predicts the drug effect for unknown compounds according to the hypotheses of drug effect patterns **H** provided by the biologists. This method is chosen owing to the inconsistency between the actual chemical effects and **H**, instead of classification methods that use **H** as classification labels, and
- The threshold T_{pr} of PRTD (validated on dataset DA-Epr-TEST) is chosen to adjust the number of drug effect proposals generated by PRTD and the failure cases of predictions, so that the users can select the customized T_{pr} according to the application case. For example, if the prediction accuracy is more important than the number of pattern proposals in a specific usage case, a higher threshold T_{pr} is desired.

Experiment E_{pr} only provides a demonstration for proving the effectiveness of PRTD, but more potential and meaningful experiments can be conducted and analyzed via PRTD, e.g., increasing the number of the compounds and varying the concentrations of the compounds.

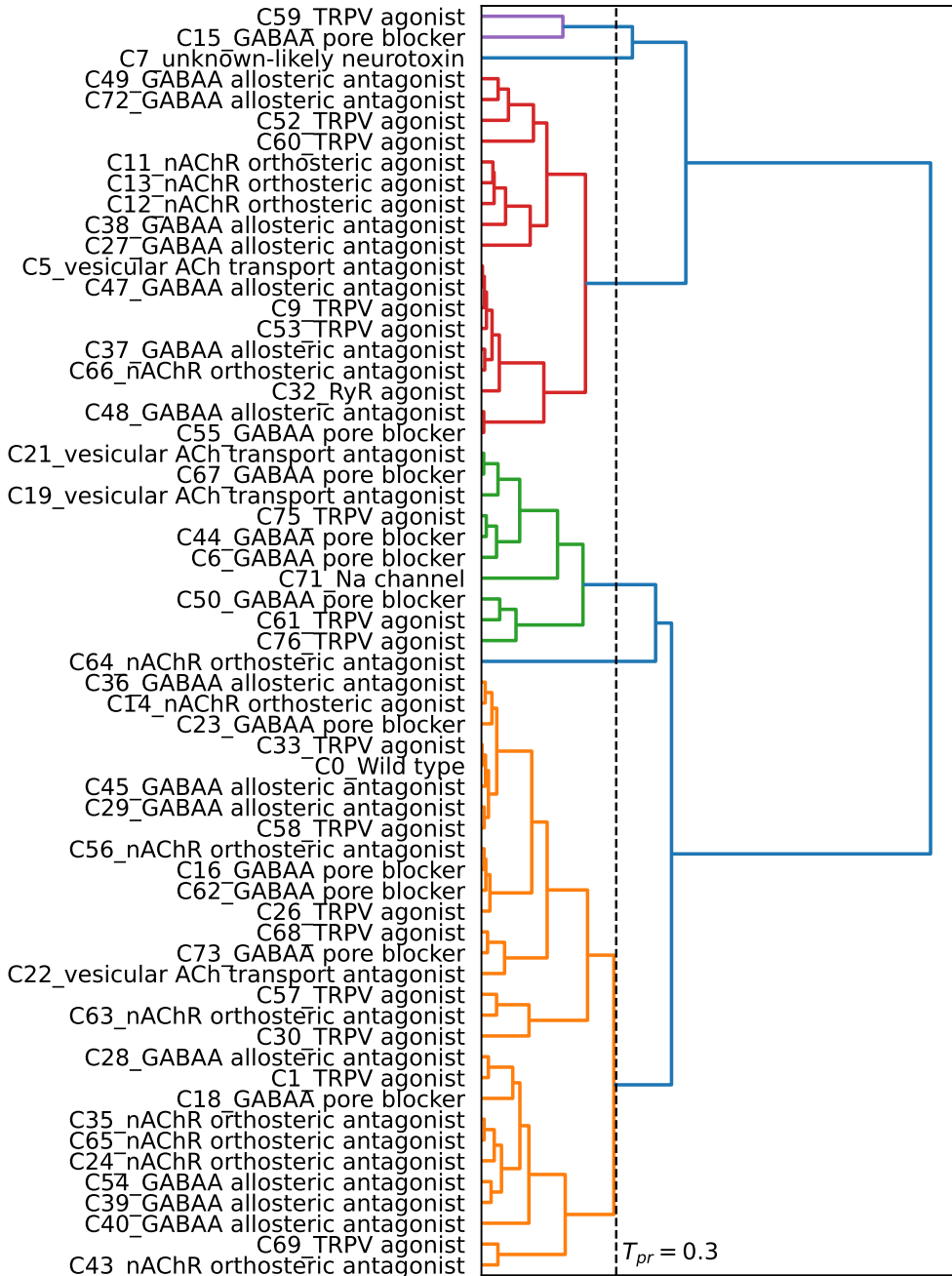


Figure 6.8: The drug effect pattern dendrogram generated on dataset DA-Epr-TRAIN with $T_{pr} = 0.3$. Different colors indicate different clusters. The y -axis outlines the name of each compound together with the corresponding hypothesis of effect pattern H_i .

7 Conclusions

This thesis conceptualizes full automation in automatic laboratories and proposes an automated interface in the touch-response experimental process based on modern AI methods, including robot-assisted data acquisition, AI-based inspection and analysis. Four new systems work together to achieve the full automated touch-response experimental process: Single-Larva Touch-response data Acquisition Platform (SiLTAP), Multi-Larva Touch-response data Acquisition Platform (MuLTAP), AI-based Multi-larva Touch-response behavior Inspection Pipeline (AMTIP), Pattern Recognition of Touch response based Database (PRTD). SiLTAP is proposed for the first time to make the touch-response experiment automated, and MuLTAP improves the throughput of SiLTAP. SiLTAP and MuLTAP generate a large amount of touch-response data, which requires a customized platform to inspect the behaviors of the larvae. Thus, AMTIP is designed to inspect the data collected from SiLTAP and MuLTAP to generate the behavior inspection (quantification) results automatically. With these inspection results, PRTD is introduced to analyze the patterns of touch-response behaviors. PRTD generates the data pattern tree and assigns the patterns to new unknown data points. Given these proposed systems, new findings on touch-response screening are summarized as:

1. A general full automation interface is conceptualized, including the automated acquisition, AI-based inspection and analysis, evaluation, parameter optimization, and comparison with the traditional manual process,
2. New concepts and parameters used in the automated touch-response interface are defined. A general touch screening pipeline is proposed with the controllable parameters. The metrics for evaluating the image processing, touching task, inspection, and pattern analysis in the proposed interface are defined in Section 2.4,
3. The data acquisition systems for single- and multi-larva touching screening (SiLTAP and MuLTAP) are proposed, with the mechanical design, including the manipulator (a blunt needle), actuators (the stepping motors with five coordinate systems), sample holder (12-well plate with shadow-covering rings), and sensor (high-frame-rate camera) as well as the software design, including the binarization method (U-Net) of the needle and larvae to finish the single- and multi-larva touching strategies. The touch screening can be conducted via SiLTAP and MuLTAP fully automatically with minor human intervention,

4. The pipeline for the touch-response data inspection (AMTIP) is developed by exploring tracking algorithms. The data collected by SiLTAP and MuLTAP is inspected via AMTIP fully automatically without human intervention,
5. Pattern analysis for the touch-response data (PRTD) is implemented. The statistical distance measurement and hierarchical clustering methods are explored to fulfill an automated pipeline for pattern analysis and prediction based on touch-response screening,
6. The proposed sub-systems work together on an end-to-end basis and provide users with the option of conducting touch-response-based drug screening experiments without manual intervention, and
7. The experimental protocols in the automation interface are designed in the corresponding sections, including evaluations of the image processing, touching task, and hardware error, choices of the controllable parameters, touch-response behavior quantification and drug screening experiments.

Based on the above knowledge, the proposed interface makes contributions to different research fields, i.e., robotics, artificial intelligence, as well as computational and automated biology. Robotic technologies and AI methods (deep learning, computer vision, and data mining) have attracted much attention thanks to their potential applications to real-world cases. The exploration of this work, with the proposal of one concept design of a modern process, brings the theoretical research of the recently proposed high-tech methods to practices. This work will arouse the interest of researchers working in the application of robotics in special and complex scenarios, e.g., the biological experimental process. The work not only focuses on automated acquisition in experiments, but subsequent inspection and analysis are also included. Thus, the interface starts to bring biology into a modern age at an intelligent level. Furthermore, drug screening is playing a vital role in human lives, and this work is expected to assist the screening procedure in a highly efficient and reliable manner. The interface completes the lab work for drug screening without pain and automatically generates the analysis results (patterns of drug effects). The conclusions drawn by the interface are more convincing with fewer errors. Therefore, the well-rounded interface will reduce the costs of screening drugs.

One limitation of the proposed interface is the inaccuracy of the segmentation and tracking algorithms. Secondly, the throughput still needs to be higher, which has the potential to be improved. Therefore, one exploration is to put more larvae into each well and to classify / evaluate the collective. The robot is controlled to touch the groups of larvae, and behaviors of this amount of larvae can be quantified and analyzed together. In this case, a new detection method during the real-time data acquisition and a new quantification pipeline are to be proposed or adapted with the existing methods. This idea is potentially helpful in improving the throughput

of screening drugs. However, the number of output data is not higher, and only the mean values from the collective are obtained.

Given the capability of the proposed interface on drug screening and the possible extension of it to other experiments, the commercialization of the interface is to be considered. A tremendous amount of new drugs are produced each year, and screening of the drugs at animal behavioral level using the interface is becoming more essential.

A Appendix

A.1 Pipeline of image processing in multi-larva case

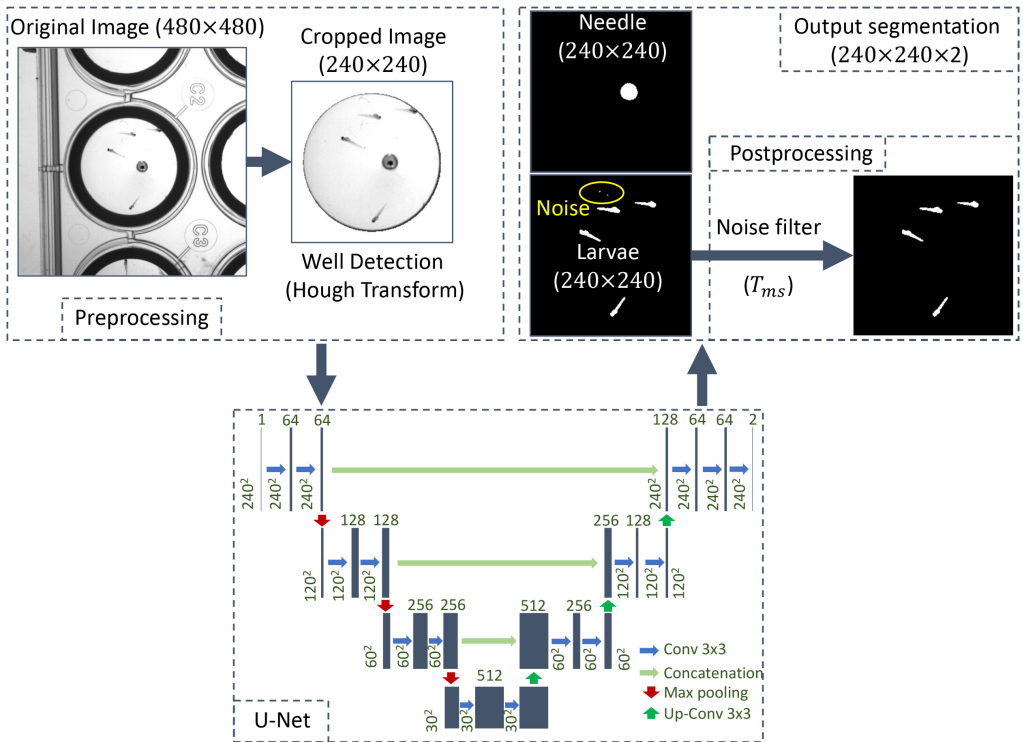


Figure A.1: Pipeline of image processing in the multi-larva case based on a U-Net [199].

The U-Net based multi-larva image processing pipeline is shown in Fig. A.1. The captured image (size: $480^2 \times 1$) by the camera is preprocessed by a Hough transform to detect the well area (cropped into $240^2 \times 1$). The preprocessed image within the well area is input to the U-Net to generate the binarization of the larvae and needle, and the noise (in the yellow circle of Fig. A.1) is possible to occur in the segmentation results, so a size threshold (T_{ms}) is used to delete the

smaller objects (noise). With the segmented binary images for the needle and larvae, the larvae are selected to be touched one by one, and the coordinates of different positions of the larvae as well as the generation of the touching trajectory are the same to the SILTAP as described in Section 3.2.3.

A.2 Training of U-Net

The training loss of U-Net is combined with two losses as follows,

$$\begin{aligned}
 Loss &= L_1 \cdot bce + L_2 \cdot (1 - bce) \\
 L_1 &= - \sum_{x \in \Omega} (y(x) \log p(x) + (1 - y(x)) \log(1 - p(x))) \\
 L_2 &= 1 - \frac{2 \cdot \sum_{x \in \Omega} (y(x) \cdot p(x)) + s}{\sum_{x \in \Omega} (y(x) + p(x)) + s}
 \end{aligned} \tag{A.1}$$

where $x \in \Omega$ is each pixel of the image, L_1 is the cross entropy loss [217], used to minimize the error between each pixel-wise prediction ($p(x)$) and the manually labeled ground-truth value ($y(x)$), L_2 is the dice loss, used to maximize the similarity (dice coefficient) of $p(x)$ and $y(x)$ [217], s is the manually chosen smooth variable, and bce is a manually chosen weight to combine L_1 and L_2 . As the data set (DA-Seg-Train) used for training of U-Net is limited, augmentation methods are used to help build a more robust U-Net model for the segmentation of larvae and needle. After considering the potential environment changes, three kinds of augmentation strategies are tested, including rotation, contrast and brightness, and Gaussian noise as follows.

Rotation (RT) As the larvae only stay within the well but possibly at different positions and with different orientations, a random rotation of the image about the center of the well (denoted as $\mathbf{i}_w = \{i_{wx}, i_{wy}\}$) in \mathbb{I}) can to a large extent cover almost all cases mentioned above. Given a rotation angle θ , the pixel value at a position E ($\mathbf{i}_e = \{i_{ex}, i_{ey}\}$ in \mathbb{I} , as an example) of the image is reformulated as follows,

$$f(\mathbf{i}'_e) = f(R_\theta(\mathbf{i}_e - \mathbf{i}_w)), \quad \text{with } R_\theta = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \tag{A.2}$$

where $f(*)$ is the pixel value, and R_θ is the rotation matrix with the random angle θ .

Contrast and Brightness (CB) The illumination in the experiment can be affected by the lab environment as well, causing the changes of the contrast and brightness of the image. Thus, this

is also considered for the augmentation of the training data set. The pixel at the position E (i_e , the same example to RT) with changes of contrast and brightness can be formulated as follows,

$$f'(i_e) = \beta \cdot f(i_e) + \gamma \quad (\text{A.3})$$

where β is the variable for changing the contrast, and γ is the variable for changing brightness.

Gaussian Noise (GN) As the noise that occurs in the experiments is not possible to quantify, Gaussian noise is considered to enhance the diversity of the data set. Given the position E (i_e , the same example to RT), the pixel with Gaussian noise is formulated as follows,

$$f'(i_e) = f(i_e) + g, \quad \text{with } g \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}),$$

$$\boldsymbol{\mu} = \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \boldsymbol{\Sigma} = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix} \quad (\text{A.4})$$

where g is subjected to a two-dimensional Gaussian distribution with mean vector $\boldsymbol{\mu}$ and variance matrix $\boldsymbol{\Sigma}$.

Table A.1: The parameters used in the training of the U-Net.

Symbol	Quantity	Value
θ	The random rotation angle for data augmentation	$0^\circ \sim 360^\circ$
β	The contrast variable for data augmentation	$0 \sim 1$
γ	The brightness variable for data augmentation	$0 \sim 50$
μ_x, μ_y	The mean of Gaussian noise for data augmentation	$\mu_x = \mu_y = 0$
σ_x^2, σ_y^2	The variance of Gaussian noise for data augmentation	$\sigma_x^2 = \sigma_y^2 = 100$
-	Maximum epoch for the training of U-Net	30000
l_r	Learning rate for the training of U-Net	0.0001
B_s	Batch size for the training of U-Net	12
bce	Loss weight for the training of U-Net	0.5

The hyperparameters used in the training phase (including the parameters used in the augmentation strategies) of U-Net based segmentation method (described in Section 4.2.2) are as Table A.1 shows. In order to decide the augmentation strategy that is helpful to increase the diversity of the training dataset (DA-Seg-Train) and build up a more robust model, the U-Net with different augmentations are compared, i.e., without augmentation (U-Net), U-Net + RT, U-Net + CB, U-Net + GN, U-Net + RT + CB, U-Net + RT + GN, U-Net + CB + GN, and U-Net + RT + CB

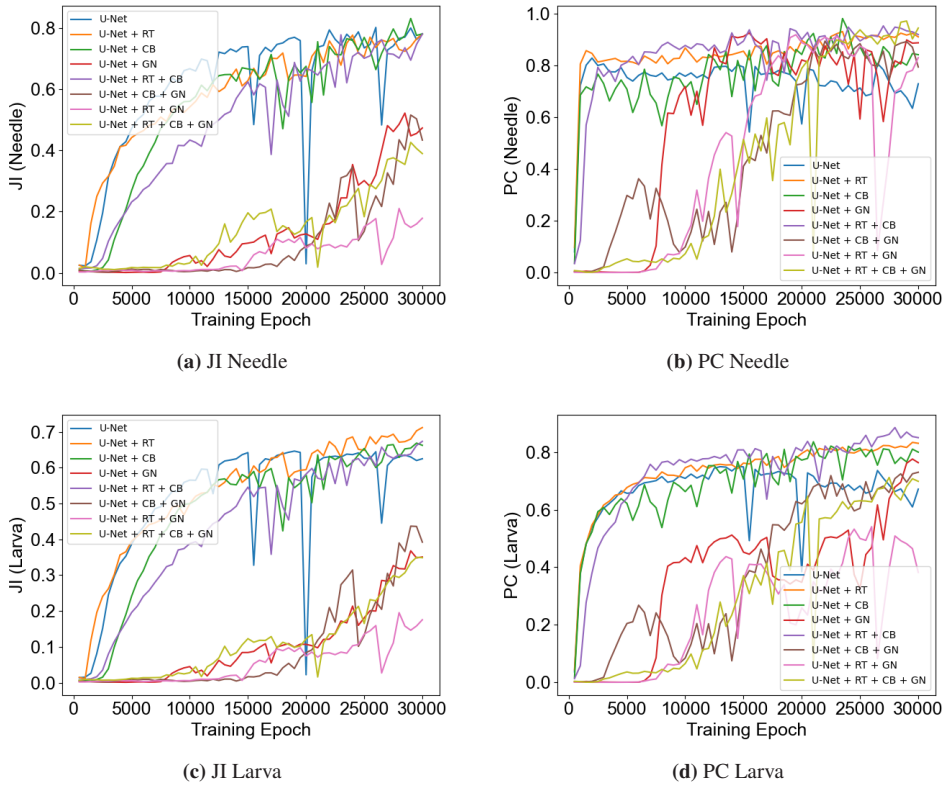


Figure A.2: Comparison of different augmentation strategies, including PC, JI for the larvae and needle of U-Net, U-Net + RT, U-Net + CB, U-Net + GN, U-Net + RT + CB, U-Net + RT + GN, U-Net + CB + GN, and U-Net + RT + CB + GN, respectively.

+ GN, respectively. According to the evaluation metrics in Section 2.4.2, the results of PC and JI (for the needle and larva) are visualized separately in Fig. A.2. The augmentation with Gaussian noise performs much worse than the other two augmentation strategies, as the results of U-Net + GN, U-Net + RT + GN, U-Net + CB + GN, and U-Net + RT + CB + GN show, especially the JI of the needle and larva. The results in Fig. A.2 indicate that the random rotation (RT) together with contrast and brightness (CB) is the best to build a more general model of U-Net, so the model U-Net + RT + CB at epoch 30000 (U-Net-RT-CB-model30000) is considered to conduct the segmentation of the needle and larvae.

A.3 Methods in AMTIP

A.3.1 Optical flow for needle tracking

Let $f(X_j^n, Y_j^n, t_j)$ in \mathbb{I} be the pixel value at the old needle position at frame t_j , and $f(X_{j+1}^n, Y_{j+1}^n, t_{j+1})$ be the pixel value at the new needle position at frame t_{j+1} (where n indicates needle), then the position change between these two frames (t_j and t_{j+1}) is estimated as

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum_i f_{x_i}^2 & \sum_i f_{x_i} f_{y_i} \\ \sum_i f_{x_i} f_{y_i} & \sum_i f_{y_i}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i f_{x_i} f_{t_i} \\ -\sum_i f_{y_i} f_{t_i} \end{bmatrix}, \quad x_i, y_i \in U\{X_j^n, Y_j^n\}_{k_{op}} \quad (\text{A.5})$$

$$f_x = \frac{\partial f}{\partial x}, f_y = \frac{\partial f}{\partial y}, f_t = \frac{\partial f}{\partial t} \quad (\text{A.6})$$

$$u = \frac{dx}{dt}, v = \frac{dy}{dt} \quad (\text{A.7})$$

where u and v are the estimated changes in x -axis and y -axis, and they are calculated according to the partial derivatives within an adjacent area of the old needle position (denoted as k_{op}).

A.3.2 Particle filter based larva tracking

Assumed that ${}_i P_j^l = \{{}_i x_j^l, {}_i y_j^l, t_j\}$ is the particle i at position $\{{}_i x_j^l, {}_i y_j^l\}$ of frame t_j for larva l , the binary probability $b\{{}_i x_j^l, {}_i y_j^l, t_j\}$ indicates whether the particle $\{{}_i x_j^l, {}_i y_j^l\}$ belongs to the l -th larva area, shortened as ${}_i b_j^l$. The new position of the larva l at frame t_{j+1} ($\{X_{j+1}^l, Y_{j+1}^l, t_{j+1}\}$) in \mathbb{I} is estimated as

$$X_{j+1}^l = \frac{1}{\sum_{i=1}^{N_p} {}_i b_{j+1}^l} \sum_{i=1}^{N_p} ({}_i x_j^l \cdot {}_i b_{j+1}^l), \quad Y_{j+1}^l = \frac{1}{\sum_{i=1}^{N_p} {}_i b_{j+1}^l} \sum_{i=1}^{N_p} ({}_i y_j^l \cdot {}_i b_{j+1}^l) \quad (\text{A.8})$$

where ${}_i b_{j+1}^l$ is the binary probability at $\{{}_i x_j^l, {}_i y_j^l\}$ of the l th larva in frame t_{j+1} . The binary probability is computed according to the image difference, namely

$${}_i b_{j+1}^l = \begin{cases} 1 & \text{if } {}_i d_{j+1}^l > T_d \\ 0 & \text{else} \end{cases}, \quad {}_i d_{j+1}^l = |f_{j+1}({}_i x_j^l, {}_i y_j^l) - f_j({}_i x_j^l, {}_i y_j^l)| \quad (\text{A.9})$$

where ${}_i d_{j+1}^l$ is the pixel difference at $\{{}_i x_j^l, {}_i y_j^l\}$ between frame t_{j+1} and frame t_j , $f_{j+1}({}_i x_j^l, {}_i y_j^l)$ is the pixel value at $\{{}_i x_j^l, {}_i y_j^l\}$ of frame t_{j+1} , $f_j({}_i x_j^l, {}_i y_j^l)$ of frame t_j , and T_d is a heuristic

threshold for the image difference to determine the pixels with changes. The particles ${}_i P_j^l$ with the binary probability value equal to zero (${}_i b_{j+1}^l = 0$) are resampled in a Gaussian distribution as

$${}_i \hat{P}_{j+1}^l = \{{}_i x_{j+1}^l, {}_i y_{j+1}^l, t_{j+1}\}, \quad \{{}_i x_{j+1}^l, {}_i y_{j+1}^l\} \sim \mathcal{N}(\boldsymbol{\mu}_{j+1}^l, \boldsymbol{\Sigma}_p)$$

$$\boldsymbol{\mu}_{j+1}^l = \begin{bmatrix} X_{j+1}^l \\ Y_{j+1}^l \end{bmatrix}, \quad \boldsymbol{\Sigma}_p = \begin{bmatrix} \sigma_{px}^2 & 0 \\ 0 & \sigma_{py}^2 \end{bmatrix} \quad (\text{A.10})$$

where ${}_i \hat{P}_{j+1}^l$ is the updated (resampled) particle at $\{{}_i x_{j+1}^l, {}_i y_{j+1}^l\}$ of the new frame t_{j+1} , $\boldsymbol{\mu}_{j+1}^l$ is the new position of the larva estimated by (A.8), and $\boldsymbol{\Sigma}_p$ is the heuristic variance for the range of resampling the particles.

A.3.3 Region growing based larva segmentation

The region growing starts at the estimated position ($\{X_{j+1}^l, Y_{j+1}^l\}$) of the larva by the particle filter based tracking method and labels each pixel in a 3×3 adjacent area according to the pixel value and image gradient. Assumed that $p_k = \{p_{kx}, p_{ky}\}$ is the center of the adjacent area in each iteration (starting iteration: $p_0 = \{X_{j+1}^l, Y_{j+1}^l\}$), the label of each adjacent point $p_k^{(*,*)} = \{p_{kx}^{(*,*)}, p_{ky}^{(*,*)}, * = -1, 0, +1\}$ is calculated as

$$L\left(p_k^{(*,*)}\right) = \begin{cases} 1 & \text{if } g\left(p_{kx}^{(*,*)}, p_{ky}^{(*,*)}\right) < T_g \\ & \text{and } T_l < f\left(p_{kx}^{(*,*)}, p_{ky}^{(*,*)}\right) < T_h, \\ 0 & \text{else} \end{cases} \quad (\text{A.11})$$

$$g\left(p_{kx}^{(*,*)}, p_{ky}^{(*,*)}\right) = |f\left(p_{kx}^{(*,*)}, p_{ky}^{(*,*)}\right) - f(p_{kx}, p_{ky})|$$

where $L\left(p_k^{(*,*)}\right)$ is the labeled (segmented) result for the position $p_k^{(*,*)}$ at k -th iteration, $g\left(p_{kx}^{(*,*)}, p_{ky}^{(*,*)}\right)$ is the image gradient at $p_k^{(*,*)}$, $f\left(p_{kx}^{(*,*)}, p_{ky}^{(*,*)}\right)$ is the pixel value at $p_k^{(*,*)}$, and T_g, T_l, T_h are the heuristic thresholds chosen for the binarization. The position $p_k^{(*,*)}$ labeled as 1 is the next center of the adjacent area at iteration $k + 1$ for the growing of the region, until all new centers are labeled as 0.

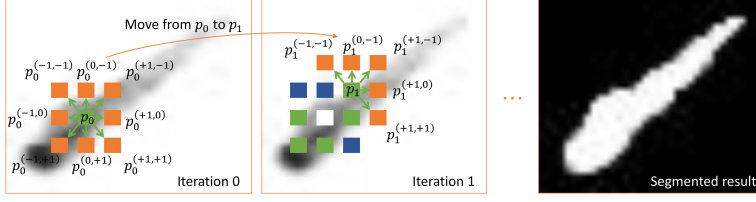


Figure A.3: The principle of region growing based local segmentation of the larva [199]. Region growing begins with an initialized point p_0 and grows according to the adjacent pixel values until no more new pixels meet the requirements, details in Section 5.2.2. The adjacent points of p_k are denoted as $p_k^{(*,*)}$, e.g. $p_k^{(-1,-1)}$ is the top left point of p_k . The pixels in green are the next centers for iterations with the adjacent points in orange, and the pixels that meet the requirements are labeled as white and otherwise as blue.

A.4 Iterative curve fitting for the C-Bend analysis

Assumed that there are totally n pixel points along the skeleton and that a m^{th} order based curve needs to be fitted, the algorithm of the iterated curve fitting comes as Algorithm 1 shows. Fig. 5.4 shows the result of one fitted curve. This algorithm outputs the inliers of one C-Bend curve to compute the radius and curvature (by using the least square method) for the l -th larva in each frame.

Algorithm 1

Iterated curve fitting for C-Bend detection and radius computation.

Input: $\{X_i, Y_i\}_{i=1:n}$ (Skeleton points)

Initialization:

$\{p_j = 0.1\}_{j=1:m}$

$\epsilon \leftarrow 10$

$threshold \leftarrow 2.5$

while $\epsilon > threshold$ **do**

 Optimize $\{p_j\}_{j=1:m}$

$\hat{Y} \leftarrow p_0 + p_1 \times X + \dots + p_m \times X^m$

$\delta Y \leftarrow Y - \hat{Y}$

$\epsilon \leftarrow \max\{\delta Y_i\}_{i=1:n}$

$I \leftarrow \operatorname{argmax}\{\delta Y_i\}_{i=1:n}$

if $\epsilon > threshold$ **then**

 remove $\{X_I, Y_I\}$

$n \leftarrow n - 1$

else

 break

end if

end while

return $\{p_j\}_{j=1:m}$, with inliers of the curve

List of Figures

1.1	Essential parts for a full automation.	6
1.2	The pipeline of data processing.	8
1.3	The adult zebrafish (left: female, right: male, [138]).	14
1.4	(a) Zebrafish embryo (egg). (b) The spontaneous hatching of a zebrafish embryo. (c) The manual dechoriation of a zebrafish embryo.	15
1.5	The visualization of the problems caused in the manual touch-response experiment [175].	19
	(a) Before touching	19
	(b) Touching	19
	(c) After touching	19
1.6	The goal of the thesis, i.e., design of the fully automated touch-screening interface on zebrafish larvae.	22
2.1	The pharmaceutical production process.	26
2.2	The general blocks for the design of a process with the conventional optimization by manually as well as the new optimization based on a robotic controller.	27
2.3	The manual optimizer. The operators adjust the process by visually comparing the sensor data and reference values.	29
2.4	The robotic optimizer. The inspection, analysis, and evaluation are fulfilled given the sensor data automatically via AI methods and fed back to the robot to adjust the process.	29
2.5	The illustration of touch-response screening, including biologists (starting point marked with a red "star"), automated acquisition, AI inspection software, AI pattern analysis software, and conclusions.	30
2.6	The schematic demonstration of the touch-response experiment on zebrafish larvae by manually [182].	32
2.7	The dimensions of the shadow-covering rings: outer diameter as r_{od} , inner diameter as r_{id} , and height as r_h	38

2.8	The comparison between possibilities of sample holders (well plates). (a) Black well plate with low brightness. (b) Black well plate with high brightness. (c) Transparent well plate with low brightness. (d) Transparent well plate with high brightness. (e) Transparent well plate with installed agarose rings. (f) Transparent well plate with installed PTFE rings.	39
2.9	Possible manipulators used in the design of the touch-response interface. (a) FluidFM [190]: an atomic force microscopy (AFM) in microfluidic environment. The size of FluidFM is usually from hundreds of nanometers to several micrometers. (b) Glass pipette [191]: size from tens to hundreds of micrometers. (c) Blunt needle: size over hundreds of micrometers.	41
	(a) FluidFM	41
	(b) Glass pipette	41
	(c) Blunt needle	41
2.10	The structure of U-Net based segmentation [97, 199]. Blue arrows indicate layers of convolutional neural networks with 3×3 kernel, denoted as Conv 3×3 . Cyan arrows indicate the concatenation of the results from the input layers. Red arrows indicate the down-sampling layer, i.e., max pooling. Green arrows indicate up-sampling convolutional neural networks with 3×3 kernel, denoted as Up-Conv 3×3	44
2.11	The workflow of evaluating the image processing, containing the comparison between the result from developer annotating and image processing for the image of zebrafish larvae.	49
2.12	The illustration of the segmentation and bounding boxes from developer annotating and image processing. (a) The original image of zebrafish larva captured by the camera. (b) The manual ground-truth bounding box (B_g , green box) and segmentation area (G_i , in green) from developer annotating. i indicates classes, e.g., the larva, manipulator, or image background. (c) The detected bounding box (B_d , red box) and segmentation area (C_i , in red) from image processing. (d) The comparison of detection (i.e., B_g and B_d). The overlap area of B_g and B_d is in orange ($B_d \cap B_g$). (e) The comparison of segmentation (i.e., G_i and C_i). The overlap area of G_i and C_i is in orange (C_{ii}).	49

2.13	The illustration of successful and unsuccessful recalls of larvae according to T_o and T_{IoU} with 5 examples. Example "1": unsuccessfully recalled. Example "2": unsuccessfully recalled according to T_{IoU} but successfully recalled according to T_o . Example "3": wrong detected. Example "4": unsuccessfully recalled. Example "5": successfully recalled.	51
3.1	The overall architecture of the automated touch-response data acquisition system (SiLTAP), including three parts in three dashed boxes (sample preparation [175], software, and hardware [210]). In the hardware part of SiLTAP, the gray blue boxes indicate the actuator systems, and the orange pieces in the 12-well plate are the shadow-covering rings. More details about the three parts of SiLTAP are discussed in Section 3.1.	56
3.2	The hardware design of SiLTAP [175, 210]. (a) The whole image of SiLTAP, including the actuator system (linear actuator and stepping motors), linear drives, and high-speed camera. (b) The partial image of (a), i.e., the light source with manipulator and 12-well-plate with shadow-covering rings. (c) The partial image of (b), i.e., the magnetic set. (d) The detailed image of the light source with the manipulator in (b).	57
3.3	The coordinate systems used in SiLTAP [175], including the world coordinate system (\mathbb{W}) with origin w_0 , the needle coordinate system (\mathbb{N}) with the origin n_0 , the image coordinate system (\mathbb{I}) with the origin i_0 , the larva coordinate system (\mathbb{L}) with the origin l_0 . The coordinates of the larva head part are in red, body in blue, gravity in brown, tail in green. An example point m with corresponding coordinates are shown in the image and larva coordinate systems.	58
3.4	Pipeline of image processing in SiLTAP [175], including preprocessing, binarization, postprocessing, and touching trajectory.	61
3.5	The graphical user interfaces used in the touch-response experimentation system [175].	64
	(a) SETTING_GUI	64
	(b) MAIN_GUI	64
3.6	The touching strategy of SiLTAP.	65
3.7	Protocol of the verification experiments E_{ip} and E_{tt} [175].	67
3.8	The ratio of recall of the larva segmentation with respect to threshold of IoU (T_{IoU}) and ratio of overlap (T_o), evaluated on dataset DA-Eip [175]. According to Equ. 2.6 and 2.7, the thresholds T_{IoU} and T_o are chosen for successful recall of the larva detection, and the ratio of recall drops with the increase of the thresholds T_{IoU} and T_o	69

3.9	The distribution of the touched points with the proposed SiLTAP tested on dataset DA-Ett [175]. The touched points are mapped to \mathbb{P} (seen in Fig. 3.4, red for the head, blue for the body, green for the tail, and black solid lines as the predefined touching positions (p_h, p_b, p_t)). Minus value indicates that SiLTAP does not touch the larva successfully.	70
4.1	The desired workflow of MuLTAP. The larvae are marked in different colors to better distinguish them, i.e., Larva #1 in blue, Larva #2 in black, Larva #3 in orange, Larva #4 in green, and Larva #n in yellow. The parameters used in the multi-larva touch screening ($S_t, \theta_t, P, A_t, H_c, r_h$) are marked in red, where the capacity of the well $H_c = n$. The angle of touching θ_t is set to 90° as SiLTAP does.	74
4.2	The reusable 3D printed shadow-covering rings installed in each well of 12-well plate.	75
	(a) The PTFE rings	75
	(b) The 12-well plate installed with PTFE rings	75
4.3	Pipeline of image processing in the multi-larva case, including pre-processing, segmentation method to be determined, output segmentation.	76
4.4	The proposed multi-larva touching strategy of MuLTAP [210]. (a) The pipeline of the multi-larva touching strategy consists of five main parts, including image processing (IP), moving the needle to the larva based on trajectory (MLT), capturing video (CV), saving video (SV), and moving the needle to the next well (MNW), with details in Section 4.2.3. The time for each part is denoted as $t_{IP}, t_{MLT}, t_{CV}, t_{SV}$, and t_{MNW} . The larvae are marked in red and the application of the touching is shown as the green arrows. (b) Example of touching Larva #2 in (a). The generation of the touching trajectory with a starting point is displayed in green, and the movement of Larvae #2 is shown in red. The approaching speed S_{t1} , touching speed S_{t2} , and touching angle θ_t are defined in Section 2.2.1 and 3.2.3.	78
	(a) Multi-larva touching strategy in well i	78
	(b) The touching of Larva #2	78
4.5	Protocol of the verification experiments ($E_{c1}, E_{c2}, E_r, E_{rh}, E_a, E_d, E_{tm}$, and E_t) for MuLTAP. The experiments are marked in red.	79
4.6	(a) and (b) Comparison of Thre, Otsu, LRb, Rgb, and U-Net by ratio of recall and precision. All methods are evaluated without any post-processing. (c) Comparison of size threshold (T_{ms}) on the PC and JI of larvae segmentation by using the U-Net. If T_{ms} is set as 12, the JI has highest value with PC still satisfactory, so 12 is selected to be T_{ms} in the experiment.	84
	(a) Comparison of ratio of recall	84
	(b) Comparison of ratio of precision	84

	(c) Comparison of size threshold	84
4.7	Comparison of the number of larvae that have interactive movements after only one larva touched, with 2, 3, 4, and 5 larvae in each well respectively [210]. The "Ave" indicates the average number of larvae that have interactive movements in each well.	84
4.8	The ratios of recall and precision of the data without rings and with rings in the wells, based on larva segmentation method (U-Net model with $T_{ms} = 12$) with 4 larvae in each image [210]. The red lines help compare the results of both cases.	86
	(a) Ratio of recall	86
	(b) Ratio of precision	86
4.9	The distribution of the touched points with the proposed MuLTAP tested on dataset DA-Etm. The touched points are mapped to \mathbb{P} (seen in Fig. 3.4, red for the head, blue for the body, green for the tail, and black solid lines as the predefined touching positions (p_h, p_b, p_t)). Minus value indicates that MuLTAP does not touch the larva successfully.	90
5.1	The diagram of the AI-based multi-larva touch-response behavior inspection pipeline (AMTIP) [182], including initialization, tracking and segmentation, quantification, and quantification criteria. The steps are marked in red.	96
5.2	The pipeline of the tracking and segmentation of AMTIP [199], including initialization, an optimized optical flow for needle tracking, an optimized particle filter for larva tracking, and an optimized region growing for larva segmentation. The frame numbers of the images are in red and the touched larva is marked in orange circles. Part 1: A U-Net is used to segment the larvae and needle with two binary images for initialization. Part 2: Optical flow based needle tracking is used for predicting the positions of the needle. Part 3: Particle filter based multi-larva tracking. Part 3-1: Particles (marked in orange) are sampled according to the segmented larvae areas. Part 3-2: According to the position of each particle, the image difference between two frames (with an example of the image difference between Frame #3693 and Frame #3694) is observed for the binary probability of the corresponding particle. Part 3-3: The particles with binary probability equal to 0 are resampled around the larvae positions, details in Appendix A.3.2. Part 4: For each larva, the segmentation is achieved by local region growing, discussed in Section 5.2.2. The outputs of the tracking contain the image patches of all larvae as well as the series of positions of the larvae and needle.	98
5.3	The diagram of quantification procedure [182], and the steps are marked in red.	100

5.4	Curve fitting includes three steps [175]: binarization, skeleton detection and iterated curve fitting (described in Appendix A.4). The fourth picture shows the third order curve (in blue) fitted after iterations, which is mapped to the original image patch in a green curve.	101
5.5	Protocol of the quantification experiments E_p , E_{db} , E_{st} , and E_{lt} which are marked in red [175].	103
5.6	Results of the behavior quantification on dataset DA-Ep with different positions touched at 54 hpf of the wild-type larvae, including four quantification criteria, i.e., latency time t_l , C-Bend radius average r_a , response time t_r , and moving distance d_m [175].	107
	(a) Latency time	107
	(b) C-Bend radius average	107
	(c) Response time	107
	(d) Moving distance	107
5.7	The temporal line charts of the behavior quantification on dataset DA-db with the body touched at 30 hpf, 54 hpf, and 78 hpf of the wild-type larvae [175].	108
	(a) Latency time	108
	(b) C-Bend radius average	108
	(c) Response time	108
	(d) Moving distance	108
5.8	Results of the behavior quantification on dataset DA-Est with the body touched at 54 hpf, 54.5 hpf, 55 hpf, 55.5 hpf, 56 hpf, 56.5 hpf of the controls and treated larvae, including four quantification criteria, i.e., latency time t_l , C-Bend radius average r_a , response time t_r , and moving distance d_m [175].	109
	(a) Latency time	109
	(b) C-Bend radius average	109
	(c) Response time	109
	(d) Moving distance	109
5.9	Five quantification criteria on dataset DA-Elt with four experiment cases (<i>wild</i> , <i>DMSO</i> , <i>Dia</i> , and <i>Caffi</i>) generated by AMTIP, including latency time t_l , C-Bend curvature maximum c_m , C-Bend curvature peak time t_{cp} , response time t_r , and moving distance d_m [199].	109
	(a) Latency time	109
	(b) C-Bend curvature maximum	109
	(c) C-Bend curvature peak time	109
	(d) Response time	109
	(e) Moving distance	109

6.1	The inputs and outputs of PRTD. Inputs: quantification criteria \mathbf{Q} from AMTIP and hypotheses of drug effect patterns \mathbf{H} from domain expert. Outputs: pattern dendrogram \mathbf{T}_d and pattern predictions \mathbf{P} to users.	114
6.2	The illustration of the data used for PRTD in box plots: latency time t_l of zebrafish larvae under the treatments of three example drugs (Drug a, Drug b, Drug c) compared with control group. The hypotheses of effect patterns for Drug a, Drug b, Drug c are \mathbf{H}_1 , \mathbf{H}_2 , and \mathbf{H}_3	115
6.3	The workflow of PRTD, including feature engineering, hierarchical clustering, domain expert, pattern mapping, and pattern proposing. \mathbf{Q}_c and \mathbf{Q} : quantification criteria results for control group and experiment group, respectively. \mathbf{f} : features generated by the L_2 Wasserstein distance method. \mathbf{B} : branches of the pattern dendrogram (also the clusters). \mathbf{H} : hypotheses of effect pattern. The pattern database contains \mathbf{B} and \mathbf{H} . The prediction for a new data point \mathbf{Q}_n is marked in red with the proposal of patterns \mathbf{P}_n	116
6.4	The database built by the proposed PRTD. The branches (clusters) of the pattern dendrogram are marked in different colors, and different shapes in black denote the hypotheses pattern from the domain expert. Four clusters ($\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3, \mathbf{B}_4$) and three patterns from hypotheses ($\mathbf{H}_1, \mathbf{H}_2, \mathbf{H}_3$) are demonstrated. The dashed arrows indicate the pattern mapping.	118
6.5	The protocol of Experiment E_{pr} , which is marked in red.	119
6.6	The evaluation procedure of PRTD on the drug effect prediction. The drug effect database is built on dataset DA-Epr-TRAIN, which is tested on dataset DA-Epr-TEST by comprising the prediction \mathbf{P}_n from PRTD and ground-truth \mathbf{H}_n of drug effect patterns.	121
6.7	Evaluation result of Experiment E_{pr} , including the number of pattern proposals $\#N_{pp}$ and failure of pattern predictions P_f concerning the threshold used in the hierarchical clustering (T_{pr}) in PRTD.	122
6.8	The drug effect pattern dendrogram generated on dataset DA-Epr-TRAIN with $T_{pr} = 0.3$. Different colors indicate different clusters. The y -axis outlines the name of each compound together with the corresponding hypothesis of effect pattern \mathbf{H}_j	125
A.1	Pipeline of image processing in the multi-larva case based on a U-Net [199].	131
A.2	Comparison of different augmentation strategies, including PC, JI for the larvae and needle of U-Net, U-Net + RT, U-Net + CB, U-Net + GN, U-Net + RT + CB, U-Net + RT + GN, U-Net + CB + GN, and U-Net + RT + CB + GN, respectively.	134
	(a) JI Needle	134
	(b) PC Needle	134
	(c) JI Larva	134

	(d) PC Larva	134
A.3	The principle of region growing based local segmentation of the larva [199]. Region growing begins with an initialized point p_0 and grows according to the adjacent pixel values until no more new pixels meet the requirements, details in Section 5.2.2. The adjacent points of p_k are denoted as $p_k^{(*,*)}$, e.g. $p_k^{(-1,-1)}$ is the top left point of p_k . The pixels in green are the next centers for iterations with the adjacent points in orange, and the pixels that meet the requirements are labeled as white and otherwise as blue.	137

List of Tables

1.1	The developmental stages of zebrafish with the possible behaviors and experiments at each stage.	16
2.1	The possible sensors for capturing the animal behaviors.	36
2.2	The possible sampler holders.	37
2.3	The possible sample holders under different conditions used in touch-response experiment. "+" indicates the image quality is good or the larva is observable, and "-" indicates the image quality is not good or the larvae is not observable.	38
2.4	The possible software tools that can be applied to the design of automation interface.	42
2.5	The possible touching strategies.	46
2.6	The evaluation metrics for the full automation interface of touch response.	48
3.1	The datasets of Experiments E_{ip} , E_{he} , and E_{tt} . ↓ indicates that a low value of the evaluation metric is desired, and ↑ indicates a high value of the evaluation metric is desired.	66
3.2	The parameters used in SiLTAP.	68
3.3	The error evaluation. ↓ indicates that a low value of the evaluation metric is desired.	69
4.1	The information of PTFE rings.	76
4.2	The datasets collected in Experiments E_{seg} , E_{c1} , E_{c2} , E_r , E_{rh} , E_a , E_d , E_t , and E_{vpw}	81
4.3	The experiment setup of E_{c1} , E_{c2} , E_r , E_{rh} , E_a , E_d , E_{tm} , E_t , and E_{vpw} . "ND" indicates no dechoriation.	82
4.4	The parameters used for the comparison of segmentation methods.	83
4.5	Comparison of segmentation methods. ↑ indicates that a high value of the evaluation metric is desired, and ↓ indicates that a low value of the evaluation metric is desired.	83
4.6	Number of valid videos collected via MuLTAP in 2-larvae, 3-larvae, 4-larvae, and 5-larvae cases by touching all larvae in each well.	85
4.7	The experiment results of Experiments E_{rh} , E_a , E_d , and E_{vpw}	88
4.8	The reproducibility of touching task via SiLTAP and MuLTAP. ↓ indicates that a low value of the evaluation metric is desired.	91

4.9	The comparison of the average time used for each step of single- and multi-larva touching strategies used in SiLTAP and MulTAP.	91
5.1	The parameters used in AMTIP.	102
5.2	The experiment setup of Experiments E_p , E_{db} , E_{st} , and E_{lt}	105
5.3	The datasets of Experiments E_p , E_{db} , E_{st} , and E_{lt}	106
5.4	The analysis of the detected errors (failure cases) of the proposed AMTIP.	110
6.1	The hypotheses of effect pattern \mathbf{H} of chemical compounds C0 - C76 provided by biologists.	120
6.2	The dataset and evaluation metrics of Experiment E_{pr} , including the training and testing datasets from C0 to C76. \downarrow indicates that a low number of pattern proposals $\#N_{pp}$ and a low percentage of failures of pattern predictions P_f are desired.	121
A.1	The parameters used in the training of the U-Net.	133

List of Publications

Journal articles

- [1] Y. Wang, N. K. Kanagaraj, C. Pylatiuk, R. Mikut, R. Peravali, and M. Reischl, “High-throughput data acquisition platform for multi-larvae touch-response behavior screening of zebrafish,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 858–865, 2021.
- [2] Y. Wang, D. Marcato, V. Tirumalasetty, N. K. Kanagaraj, C. Pylatiuk, R. Mikut, R. Peravali, and M. Reischl, “An automated experimentation system for the touch-response quantification of zebrafish larvae,” *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 4, pp. 3007–3019, 2021.
- [3] N. Sun, Z. Chen, Y. Wang, S. Wang, Y. Xie, and Q. Liu, “Random fractal-enabled physical unclonable functions with dynamic AI authentication,” *Nature Communications*, vol. 14, no. 1, p. 2185, 2023.
- [4] Y. Wang, C. Pylatiuk, R. Mikut, R. Peravali, and M. Reischl, “A fully automated touch-response behavior inspection pipeline on zebrafish larvae,” *at - Automatisierungstechnik (accepted)*, 2023.

Conference contributions

- [1] Y. Wang, C. Pylatiuk, R. Mikut, R. Peravali, and M. Reischl, “Quantification platform for touch response of zebrafish larvae using machine learning,” in *The proceedings of the 31st Workshop on Computational Intelligence*, vol. 25, p. 37, 2021.
- [2] Y. Wang, N. K. Kanagaraj, C. Pylatiuk, R. Mikut, R. Peravali, and M. Reischl, “High-throughput data acquisition platform for multi-larvae touch-response behavior screening of zebrafish,” in *International Conference on Robotics and Automation (ICRA)*, 2022.
- [3] Y. Wang, A. Dolfini, M. Reischl, E. Konukoglu, and K. Flouris, “Vertebrae high-quality mri reconstructions with latent diffusion energy based models,” in *The 26th International*

Conference on Medical Image Computing and Computer Assisted Intervention (submitted), 2023.

Bibliography

- [1] G. Shone and L. Humairah, “Should we be worried about the robotics revolution?,” June 2022. Available: <https://www.euronews.com/embed/1969120>. Accessed on: May 24, 2023.
- [2] H. H. Thorp, “Chatgpt is fun, but not an author,” *Science*, vol. 379, no. 6630, pp. 313–313, 2023.
- [3] S. E. Baker, “Post-work futures and full automation: towards a feminist design methodology,” *Open Cultural Studies*, vol. 2, no. 1, pp. 540–552, 2018.
- [4] E. Check Hayden, “The automated lab,” *Nature*, vol. 516, no. 7529, pp. 131–132, 2014.
- [5] H. Heinrich, A. Deutschländer, F. Zoghlami, and O. K. Sen, “The long journey from standardization to fully fab automation and more,” *IEEE Transactions on Semiconductor Manufacturing*, vol. 32, no. 4, pp. 408–414, 2019.
- [6] S. Y. Nof and S. Y. Nof, *Springer handbook of automation*. Springer Berlin Heidelberg, 2009.
- [7] Y. Kobayashi, T. Takahashi, T. Nakazato, H. Sakurai, H. Tamaru, K. L. Ishikawa, K. Sakaue, and S. Tani, “Fully automated data acquisition for laser production cyber-physical system,” *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 27, no. 6, pp. 1–8, 2021.
- [8] G. Ragazzini, A. Mescola, L. Corsi, and A. Alessandrini, “Fabrication of a low-cost on-stage cell incubator with full automation,” *Journal of Biological Education*, vol. 53, no. 2, pp. 165–173, 2019.
- [9] L. S. Mattos, E. Grant, R. Thresher, and K. Kluckman, “Blastocyst microinjection automation,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, no. 5, pp. 822–831, 2009.
- [10] J. L. Moreno-Camacho, D. Y. Calva-Espinosa, Y. Y. Leal-Leyva, D. C. Elizalde-Olivas, A. Campos-Romero, and J. Alcántar-Fernández, “Transformation from a conventional clinical microbiology laboratory to full automation,” *Laboratory Medicine*, vol. 49, no. 1, pp. e1–e8, 2018.

- [11] C. Adam, D. Fabre, J. Mougin, M. Zins, A. Azarine, R. Ardon, G. d'Assignies, and S. Haulon, "Pre-surgical and post-surgical aortic aneurysm maximum diameter measurement: full automation by artificial intelligence," *European Journal of Vascular and Endovascular Surgery*, vol. 62, no. 6, pp. 869–877, 2021.
- [12] J. Behrends, C. Schwager, M. Hein, T. Scholzen, S. Kull, and U. Jappe, "Innovative robust basophil activation test using a novel gating strategy reliably diagnosing allergy with full automation," *Allergy*, vol. 76, no. 12, pp. 3776–3788, 2021.
- [13] C. J. Turner, R. Ma, J. Chen, and J. Oyekan, "Human in the loop: Industry 4.0 technologies and scenarios for worker mediation of automated manufacturing," *IEEE Access*, vol. 9, pp. 103950–103966, 2021.
- [14] S. Ulewicz and B. Vogel-Heuser, "Industrially applicable system regression test prioritization in production automation," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 4, pp. 1839–1851, 2018.
- [15] P. Poór, T. Broum, and J. Basl, "Role of collaborative robots in industry 4.0 with target on education in industrial engineering," in *2019 4th International Conference on Control, Robotics and Cybernetics (CRC)*, pp. 42–46, IEEE, 2019.
- [16] P. Urhal, A. Weightman, C. Diver, and P. Bartolo, "Robot assisted additive manufacturing: A review," *Robotics and Computer-Integrated Manufacturing*, vol. 59, pp. 335–345, 2019.
- [17] A. Weiss, A.-K. Wortmeier, and B. Kubicek, "Cobots in industry 4.0: A roadmap for future practice studies on human–robot collaboration," *IEEE Transactions on Human-Machine Systems*, vol. 51, no. 4, pp. 335–345, 2021.
- [18] N. Barker and C. Jewitt, "Filtering touch: An ethnography of dirt, danger, and industrial robots," *Journal of Contemporary Ethnography*, vol. 51, no. 1, pp. 103–130, 2022.
- [19] H. Liu and L. Wang, "Remote human–robot collaboration: A cyber–physical system application for hazard manufacturing environment," *Journal of Manufacturing Systems*, vol. 54, pp. 24–34, 2020.
- [20] A. A. Malik, T. Masood, and A. Bilberg, "Virtual reality in manufacturing: immersive and collaborative artificial-reality in design of human-robot workspace," *International Journal of Computer Integrated Manufacturing*, vol. 33, no. 1, pp. 22–37, 2020.
- [21] L. Pérez, S. Rodríguez-Jiménez, N. Rodríguez, R. Usamentiaga, and D. F. García, "Digital twin and virtual reality based methodology for multi-robot manufacturing cell commissioning," *Applied Sciences*, vol. 10, no. 10, p. 3633, 2020.

-
- [22] O. Giraldo, A. Garcia, and O. Corcho, "A guideline for reporting experimental protocols in life sciences," *PeerJ*, vol. 6, p. e4795, 2018.
- [23] H. Fakhrudeen, G. Pizzuto, J. Glowacki, and A. I. Cooper, "Archemist: Autonomous robotic chemistry system architecture," *arXiv preprint arXiv:2204.13571*, 2022.
- [24] M. Reed, "Lab automation for materials chemistry," 2021. Available: <https://www.royce.ac.uk/content/uploads/2021/06/Materials-4.0-Lab-Automation-for-Innovation-in-Materials-Chemistry.pdf>. Accessed on: May 24, 2023.
- [25] N. Rupp, K. Peschke, M. Köppl, D. Drissner, and T. Zuchner, "Establishment of low-cost laboratory automation processes using autoit and 4-axis robots," *SLAS Technology*, 2022.
- [26] B. Ahmad, M. Gauthier, G. J. Laurent, and A. Bolopion, "Mobile microrobots for in vitro biomedical applications: A survey," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 646–663, 2021.
- [27] X. Li, C. Liu, S. Chen, Y. Wang, S. H. Cheng, and D. Sun, "In vivo manipulation of single biological cells with an optical tweezers-based manipulator and a disturbance compensation controller," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1200–1212, 2017.
- [28] P. D. McClanahan, J. H. Xu, and C. Fang-Yen, "Comparing caenorhabditis elegans gentle and harsh touch response behavior using a multiplexed hydraulic microfluidic device," *Integrative Biology*, vol. 9, no. 10, pp. 800–809, 2017.
- [29] Y. Cho, D. A. Porto, H. Hwang, L. J. Grundy, W. R. Schafer, and H. Lu, "Automated and controlled mechanical stimulation and functional imaging in vivo in *c. elegans*," *Lab on a Chip*, vol. 17, no. 15, pp. 2609–2618, 2017.
- [30] Y. Cho, S. A. Lee, Y. L. Chew, K. Broderick, W. R. Schafer, and H. Lu, "Multimodal stimulation in a microfluidic device facilitates studies of interneurons in sensory integration in *c. elegans*," *Small*, vol. 16, no. 10, p. 1905852, 2020.
- [31] A. L. Nekimken, B. L. Pruitt, and M. B. Goodman, "Touch-induced mechanical strain in somatosensory neurons is independent of extracellular matrix mutations in caenorhabditis elegans," *Molecular Biology of the Cell*, vol. 31, no. 16, pp. 1735–1743, 2020.
- [32] R. Heidrich, "Automation in small labs," *Journal of Laboratory Medicine*, vol. 45, no. 4-5, pp. 237–240, 2021.
- [33] D. S. Lütjohann, N. Jung, and S. Bräse, "Open source life science automation: Design of experiments and data acquisition via "dial-a-device"," *Chemometrics and Intelligent Laboratory Systems*, vol. 144, pp. 100–107, 2015.

- [34] P. Tremouilhac, A. Nguyen, Y.-C. Huang, S. Kotov, D. S. Lütjohann, F. Hübsch, N. Jung, and S. Bräse, “Chemotion eln: an open source electronic lab notebook for chemists in academia,” *Journal of Cheminformatics*, vol. 9, no. 1, pp. 1–13, 2017.
- [35] E. B. Joyee and Y. Pan, “Additive manufacturing of multi-material soft robot for on-demand drug delivery applications,” *Journal of Manufacturing Processes*, vol. 56, pp. 1178–1184, 2020.
- [36] A. Di Lallo, R. Murphy, A. Krieger, J. Zhu, R. H. Taylor, and H. Su, “Medical robots for infectious diseases: lessons and challenges from the covid-19 pandemic,” *IEEE Robotics & Automation Magazine*, vol. 28, no. 1, pp. 18–27, 2021.
- [37] S. Ghosh and R. Dasgupta, *Machine Learning in Biological Sciences*. Springer Nature Singapore Pte Ltd, 2022.
- [38] T. Hayakawa, T. Kamimura, S. Kaji, and F. Matsuno, “Autonomous distributed system for gait generation for single-legged modular robots connected in various configurations,” *IEEE Transactions on Robotics*, vol. 36, no. 5, pp. 1491–1510, 2020.
- [39] E. Guizzo, “By leaps and bounds: An exclusive look at how boston dynamics is redefining robot agility,” *IEEE Spectrum*, vol. 56, no. 12, pp. 34–39, 2019.
- [40] S. Han, S. Chon, J. Kim, J. Seo, D. G. Shin, S. Park, J. T. Kim, J. Kim, M. Jin, and J. Cho, “Snake robot gripper module for search and rescue in narrow spaces,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1667–1673, 2022.
- [41] L. Yang, J. Yu, S. Yang, B. Wang, B. J. Nelson, and L. Zhang, “A survey on swarm microrobotics,” *IEEE Transactions on Robotics*, vol. 38, no. 3, pp. 1531–1551, 2021.
- [42] J. Yu, L. Yang, and L. Zhang, “Pattern generation and motion control of a vortex-like paramagnetic nanoparticle swarm,” *The International Journal of Robotics Research*, vol. 37, no. 8, pp. 912–930, 2018.
- [43] D. Loghin, C. Tremblay, M. Mohammadi, and S. Martel, “Exploiting the responses of magnetotactic bacteria robotic agents to enhance displacement control and swarm formation for drug delivery platforms,” *The International Journal of Robotics Research*, vol. 36, no. 11, pp. 1195–1210, 2017.
- [44] X. Dong and M. Sitti, “Controlling two-dimensional collective formation and cooperative behavior of magnetic microrobot swarms,” *The International Journal of Robotics Research*, vol. 39, no. 5, pp. 617–638, 2020.

-
- [45] G. Hwang, A. J. Paula, E. E. Hunter, Y. Liu, A. Babeer, B. Karabucak, K. Stebe, V. Kumar, E. Steager, and H. Koo, "Catalytic antimicrobial robots for biofilm eradication," *Science Robotics*, vol. 4, no. 29, p. eaaw2388, 2019.
- [46] P. Dills, A. Dawson-Elli, K. Gruben, P. Adamczyk, and M. Zinn, "An investigation of a balanced hybrid active-passive actuator for physical human-robot interaction," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5849–5856, 2021.
- [47] L. Petani, S. Essig, J. Zeilfelder, M. Kortenhaus, and C. Pylatiuk, "Simulation and evaluation of a body energy harvesting device for arm and leg swing motions," *AIP Advances*, vol. 11, no. 6, p. 065201, 2021.
- [48] L. Zhang, Q. Huang, K. Cai, Z. Wang, W. Wang, and J. Liu, "A wearable soft knee exoskeleton using vacuum-actuated rotary actuator," *IEEE Access*, vol. 8, pp. 61311–61326, 2020.
- [49] Y. Ning, Y. Liu, F. Xi, K. Huang, and B. Li, "Human-robot interaction control for robot driven by variable stiffness actuator with force self-sensing," *IEEE Access*, vol. 9, pp. 6696–6705, 2020.
- [50] L. Liu, S. Leonhardt, C. Ngo, and B. J. Misgeld, "Impedance-controlled variable stiffness actuator for lower limb robot applications," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 2, pp. 991–1004, 2019.
- [51] M. Ilami, H. Bagheri, R. Ahmed, E. O. Skowronek, and H. Marvi, "Materials, actuators, and sensors for soft bioinspired robots," *Advanced Materials*, vol. 33, no. 19, p. 2003139, 2021.
- [52] H. Ren, H. Jin, J. Shu, J. Xie, E. Wang, D.-A. Ge, S.-Y. Tang, X. Li, W. Li, and S. Zhang, "Light-controlled versatile manipulation of liquid metal droplets: a gateway to future liquid robots," *Materials Horizons*, vol. 8, no. 11, pp. 3063–3071, 2021.
- [53] X. Huang, J. Wei, H. Wei, and L. Zhang, "Magnetocaloric actuation of soft polymer robots," *Journal of Materials Chemistry C*, vol. 9, no. 39, pp. 13635–13639, 2021.
- [54] T. Hainsworth, L. Smith, S. Alexander, and R. MacCurdy, "A fabrication free, 3d printed, multi-material, self-sensing soft actuator," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4118–4125, 2020.
- [55] C. Abah, A. L. Orekhov, G. L. Johnston, and N. Simaan, "A multi-modal sensor array for human–robot interaction and confined spaces exploration using continuum robots," *IEEE Sensors Journal*, vol. 22, no. 4, pp. 3585–3594, 2021.

- [56] K. Tsukamoto, A. Ebisui, T. Goto, Y. Sakakura, K. Kobayashi, S. Sato, T. Kamei, Y. Imai, and K. Nomoto, "Soft-material-based highly reliable tri-axis tactile thin-film sensors for robotic manipulation tasks," in *IEEE Sensors*, pp. 1–4, IEEE, 2021.
- [57] J. Long, Q. Liang, W. Sun, Y. Wang, and D. Zhang, "Ultrathin three-axis fbg wrist force sensor for collaborative robots," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–15, 2021.
- [58] J. Li, H. Šimek, D. Ilioa, N. Jung, S. Bräse, H. Zappe, R. Dittmeyer, and B. P. Ladewig, "In situ sensors for flow reactors—a review," *Reaction Chemistry & Engineering*, vol. 6, no. 9, pp. 1497–1507, 2021.
- [59] M. Seo, S. Yoo, J. Kim, H. S. Kim, and T. Seo, "Dual ascender robot with position estimation using angle and length sensors," *IEEE Sensors Journal*, vol. 20, no. 13, pp. 7422–7432, 2020.
- [60] H. Xing, Y. Liu, S. Guo, L. Shi, X. Hou, W. Liu, and Y. Zhao, "A multi-sensor fusion self-localization system of a miniature underwater robot in structured and gps-denied environments," *IEEE Sensors Journal*, vol. 21, no. 23, pp. 27136–27146, 2021.
- [61] Y. Xing, T. A. Vincent, H. Fan, E. Schaffernicht, V. H. Bennetts, A. J. Lilienthal, M. Cole, and J. W. Gardner, "Firenose on mobile robot in harsh environments," *IEEE Sensors Journal*, vol. 19, no. 24, pp. 12418–12431, 2019.
- [62] L. Petani, D. Wickersheim, L. Koker, M. Reischl, U. Gengenbach, and C. Pylatiuk, "Experimental setup for evaluation of medical ozone gas sensors," in *2022 IEEE Sensors Applications Symposium (SAS)*, pp. 1–6, IEEE, 2022.
- [63] K. Darvish, E. Simetti, F. Mastrogiovanni, and G. Casalino, "A hierarchical architecture for human–robot cooperation processes," *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 567–586, 2020.
- [64] H. Oliff, Y. Liu, M. Kumar, M. Williams, and M. Ryan, "Reinforcement learning for facilitating human-robot-interaction in manufacturing," *Journal of Manufacturing Systems*, vol. 56, pp. 326–340, 2020.
- [65] M. Wagner, A. Bihlmaier, H. G. Kenngott, P. Mietkowski, P. M. Scheickl, S. Bodenstedt, A. Schiepe-Tiska, J. Vetter, F. Nickel, S. Speidel, *et al.*, "A learning robot for cognitive camera control in minimally invasive surgery," *Surgical Endoscopy*, vol. 35, no. 9, pp. 5365–5374, 2021.
- [66] A. Babu and F. Kirchner, "Terrain adaption controller for a walking excavator robot using deep reinforcement learning," in *2021 20th International Conference on Advanced Robotics (ICAR)*, pp. 64–70, IEEE, 2021.

-
- [67] C. Pohl and T. Asfour, "Probabilistic spatio-temporal fusion of affordances for grasping and manipulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3226–3233, 2022.
- [68] J. Starke, P. Weiner, M. Crell, and T. Asfour, "Semi-autonomous control of prosthetic hands based on multimodal sensing, human grasp demonstration and user intention," *Robotics and Autonomous Systems*, vol. 154, p. 104123, 2022.
- [69] I. Radosavovic, T. Xiao, S. James, P. Abbeel, J. Malik, and T. Darrell, "Real-world robot learning with masked visual pre-training," in *Conference on Robot Learning*, pp. 416–426, PMLR, 2023.
- [70] J. C. Church, Y. Chen, and S. V. Rice, "A spatial median filter for noise removal in digital images," in *IEEE SoutheastCon 2008*, pp. 618–623, IEEE, 2008.
- [71] F. Godtliebsen, E. Spj tvoll, and J. Marron, "A nonlinear gaussian filter applied to images with discontinuities," *Journal of Nonparametric Statistics*, vol. 8, no. 1, pp. 21–43, 1997.
- [72] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 433–459, 2010.
- [73] P. Comon, "Independent component analysis," in *Higher-Order Statistics*, Elsevier, 1992.
- [74] A. Ng *et al.*, "Sparse autoencoder," *CS294A Lecture Notes*, vol. 72, no. 2011, pp. 1–19, 2011.
- [75] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 6, pp. 679–698, 1986.
- [76] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [77] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing systems*, vol. 25, 2012.
- [78] L. Brown, "The conditional level of student's t test," *The Annals of Mathematical Statistics*, vol. 38, no. 4, pp. 1068–1071, 1967.
- [79] L. St, S. Wold, *et al.*, "Analysis of variance (anova)," *Chemometrics and Intelligent Laboratory systems*, vol. 6, no. 4, pp. 259–272, 1989.
- [80] A. French, M. Macedo, J. Poulsen, T. Waterson, and A. Yu, "Multivariate analysis of variance (manova)," 2008.

- [81] A. Mallasto and A. Feragen, “Learning from uncertain curves: The 2-wasserstein metric for gaussian processes,” *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [82] C.-C. Chang and C.-J. Lin, “Libsvm: a library for support vector machines,” *ACM transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, pp. 1–27, 2011.
- [83] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [84] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*, vol. 2. MIT press Cambridge, MA, 2006.
- [85] G. A. Wilkin and X. Huang, “K-means clustering algorithms: implementation and comparison,” in *Second International Multi-Symposiums on Computer and Computational Sciences (IMSCCS 2007)*, pp. 133–136, IEEE, 2007.
- [86] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.
- [87] A. Rodriguez and A. Laio, “Clustering by fast search and find of density peaks,” *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.
- [88] F. Nielsen and F. Nielsen, “Hierarchical clustering,” *Introduction to HPC with MPI for Data Science*, pp. 195–211, 2016.
- [89] D. P. Kingma, M. Welling, *et al.*, “An introduction to variational autoencoders,” *Foundations and Trends® in Machine Learning*, vol. 12, no. 4, pp. 307–392, 2019.
- [90] Z. Dai, A. Damianou, J. González, and N. Lawrence, “Variational auto-encoded deep gaussian processes,” *arXiv preprint arXiv:1511.06455*, 2015.
- [91] B. Settles, “Active learning,” *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 6, no. 1, pp. 1–114, 2012.
- [92] L. Rettenberger, M. Schilling, S. Elser, M. Böhlend, and M. Reischl, “Self-supervised learning for annotation efficient biomedical image segmentation,” *IEEE Transactions on Biomedical Engineering*, 2023.
- [93] M. Smereka and I. Duleba, “Circular object detection using a modified hough transform.,” *International Journal of Applied Mathematics & Computer Science*, vol. 18, no. 1, 2008.
- [94] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *Advances in Neural Information Processing Systems*, vol. 28, 2015.

-
- [95] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *arXiv preprint arXiv:1804.02767*, 2018.
- [96] K. R. Castleman, *Digital image processing*. Prentice Hall Press, 1996.
- [97] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical Image Computing and Computer-assisted Intervention*, pp. 234–241, Springer International Publishing, 2015.
- [98] M. Böhlend, T. Scherr, A. Bartschat, R. Mikut, and M. Reischl, “Influence of synthetic label image object properties on gan supported segmentation pipelines,” in *Proceedings 29th Workshop Computational Intelligence*, pp. 289–305, 2019.
- [99] N. Sharmin and R. Brad, “Optimal filter estimation for lucas-kanade optical flow,” *Sensors*, vol. 12, no. 9, pp. 12694–12709, 2012.
- [100] T. Senst, V. Eiselein, and T. Sikora, “Robust local optical flow for feature tracking,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 9, pp. 1377–1387, 2012.
- [101] S.-K. Weng, C.-M. Kuo, and S.-K. Tu, “Video object tracking using adaptive kalman filter,” *Journal of Visual Communication and Image Representation*, vol. 17, no. 6, pp. 1190–1208, 2006.
- [102] J. U. Cho, S. H. Jin, X. Dai Pham, J. W. Jeon, J. E. Byun, and H. Kang, “A real-time object tracking system using a particle filter,” in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2822–2827, IEEE, 2006.
- [103] R. Hou, C. Chen, R. Sukthankar, and M. Shah, “An efficient 3d cnn for action/object segmentation in video,” *arXiv preprint arXiv:1907.08895*, 2019.
- [104] M. Schorb, I. Haberbosch, W. J. Hagen, Y. Schwab, and D. N. Mastronarde, “Software tools for automated transmission electron microscopy,” *Nature Methods*, vol. 16, no. 6, pp. 471–477, 2019.
- [105] M. P. Schilling, T. Scherr, F. R. Münke, O. Neumann, M. Schutera, R. Mikut, and M. Reischl, “Automated annotator variability inspection for biomedical image segmentation,” *IEEE Access*, vol. 10, pp. 2753–2765, 2022.
- [106] K. H. Abdulkareem, S. A. Mostafa, Z. N. Al-Qudsy, M. A. Mohammed, A. S. Al-Waisy, S. Kadry, J. Lee, and Y. Nam, “Automated system for identifying covid-19 infections in computed tomography images using deep learning models,” *Journal of Healthcare Engineering*, vol. 2022, 2022.

- [107] J. Bohannon, “The cyberscientist,” American Association for the Advancement of Science, 2017.
- [108] Allied Market Research, “Laboratory automation market by product type (equipment and software informatics), application (drug discovery, clinical diagnostics, genomics solutions, and proteomics solutions), and end user (biotechnology pharmaceutical industries, research institutes, and others): Global opportunity analysis and industry forecast, 2019-2026,” 2019. Available: <https://www.alliedmarketresearch.com/laboratory-automation-market>. Accessed on: May 24, 2023.
- [109] Data Bridge Market Research, “Europe lab automation market – industry trends and forecast to 2029,” 2022. Available: <https://www.databridgemarketresearch.com/reports/europe-lab-automation-market>. Accessed on: May 24, 2023.
- [110] R. D. King, J. Rowland, S. G. Oliver, M. Young, W. Aubrey, E. Byrne, M. Liakata, M. Markham, P. Pir, L. N. Soldatova, *et al.*, “The automation of science,” *Science*, vol. 324, no. 5923, pp. 85–89, 2009.
- [111] T. Müller, M. Kalxdorf, R. Longuespée, D. N. Kazdal, A. Stenzinger, and J. Krijgsveld, “Automated sample preparation with sp 3 for low-input clinical proteomics,” *Molecular Systems Biology*, vol. 16, no. 1, p. e9111, 2020.
- [112] W. Lei, K. Demir, J. Overhage, M. Grunze, T. Schwartz, and P. A. Levkin, “Droplet-microarray: miniaturized platform for high-throughput screening of antimicrobial compounds,” *Advanced Biosystems*, vol. 4, no. 10, p. 2000073, 2020.
- [113] M. Seifermann, P. Reiser, P. Friederich, and P. Levkin, “High-throughput synthesis and machine learning assisted design of photodegradable hydrogels,” *ChemRxiv (preprint)*, 2023.
- [114] “Imaging cluster: Instrumentation methods for multiscale x-ray imaging in materials science life science,” 2023. Available: <https://www.ips.kit.edu/5925.php>. Accessed on: May 24, 2023.
- [115] S. A. Doyle, “Kit ips-research-imaging technologies for life science and materials research,” 2022. Available: <https://www.ips.kit.edu/5651.php>. Accessed on: May 24, 2023.
- [116] M. Reischl, M. Jouda, N. MacKinnon, E. Fuhrer, N. Bakhtina, A. Bartschat, R. Mikut, and J. G. Korvink, “Motion prediction enables simulated mr-imaging of freely moving model organisms,” *PLoS Computational Biology*, vol. 15, no. 12, p. e1006997, 2019.
- [117] Y. Shi, P. L. Prieto, T. Zepel, S. Grunert, and J. E. Hein, “Automated experimentation powers data science in chemistry,” *Accounts of Chemical Research*, vol. 54, no. 3, pp. 546–555, 2021.

- [118] S. Neubert, S. Junginger, T. Roddelkopf, S.-J. Burgdorf, N. Stoll, and K. Thurow, "Automated system for pouring and filtration tasks in laboratory applications," *Chemie Ingenieur Technik*, vol. 94, no. 4, pp. 530–541, 2022.
- [119] D. Marcato, R. Alshut, H. Breitwieser, R. Mikut, U. Strähle, C. Pylatiuk, and R. Peravali, "An automated and high-throughput photomotor response platform for chemical screens," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 7728–7731, IEEE, 2015.
- [120] R. Peravali, J. Gehrig, S. Giselbrecht, D. Lütjohann, Y. Hadzhiev, F. Müller, and U. Liebel, "Protocol for intelligent high-content screening of zebrafish embryos on a standard widefield screening microscope," *BioTechniques*, vol. 62, no. 1, p. xx, 2017.
- [121] C. L. Essmann, D. Martinez-Martinez, R. Pryor, K.-Y. Leung, K. B. Krishnan, P. P. Lui, N. D. Greene, A. E. Brown, V. M. Pawar, M. A. Srinivasan, *et al.*, "Mechanical properties measured by atomic force microscopy define health biomarkers in ageing *c. elegans*," *Nature Communications*, vol. 11, no. 1, pp. 1–16, 2020.
- [122] S.-J. Park, B. C. Petzold, M. B. Goodman, and B. L. Pruitt, "Piezoresistive cantilever force-clamp system," *Review of Scientific Instruments*, vol. 82, no. 4, p. 043703, 2011.
- [123] B. Ahmad, T. Kawahara, T. Yasuda, and F. Arai, "Microrobotic platform for mechanical stimulation of swimming microorganism on a chip," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4680–4685, IEEE, 2014.
- [124] C. Pylatiuk, D. Sanchez, R. Mikut, R. Alshut, M. Reischl, S. Hirth, W. Rottbauer, and S. Just, "Automatic zebrafish heartbeat detection and analysis for zebrafish embryos," *Zebrafish*, vol. 11, no. 4, pp. 379–383, 2014.
- [125] J. Schuetzke, A. Benedix, R. Mikut, and M. Reischl, "Enhancing deep-learning training for phase identification in powder x-ray diffractograms," *IUCrJ*, vol. 8, no. 3, pp. 408–420, 2021.
- [126] V. Wilson, A. Guenther, Ø. Øverli, M. W. Seltmann, and D. Altschul, "Future directions for personality research: contributing new insights to the understanding of animal behavior," *Animals*, vol. 9, no. 5, p. 240, 2019.
- [127] P. Slater, "Individual differences in animal behavior," *Perspectives in Ethology: Volume 4 Advantages of Diversity*, pp. 35–49, 1981.
- [128] M. Briffa and A. Weiss, "Animal personality," *Current Biology*, vol. 20, no. 21, pp. R912–R914, 2010.

- [129] R. Ferrer-i Cancho, A. Hernández-Fernández, D. Lusseau, G. Agoramoorthy, M. J. Hsu, and S. Semple, “Compression as a universal principle of animal behavior,” *Cognitive Science*, vol. 37, no. 8, pp. 1565–1578, 2013.
- [130] C. Kim, T. Ruberto, P. Phamduy, and M. Porfiri, “Closed-loop control of zebrafish behaviour in three dimensions using a robotic stimulus,” *Scientific Reports*, vol. 8, no. 1, p. 657, 2018.
- [131] E. E. Patton, L. I. Zon, and D. M. Langenau, “Zebrafish disease models in drug discovery: from preclinical modelling to clinical trials,” *Nature Reviews Drug Discovery*, vol. 20, no. 8, pp. 611–628, 2021.
- [132] P. Mudgal, C. Bhasin, A. Joshi, and R. Gupta, “Zebrafish,” *Resonance*, vol. 26, no. 11, pp. 1499–1521, 2021.
- [133] O. Giraldo, A. Garcia, F. Lopez, and O. Corcho, “Using semantics for representing experimental protocols,” *Journal of Biomedical Semantics*, vol. 8, pp. 1–15, 2017.
- [134] M. B. Orger, E. Gahtan, A. Muto, P. Page-McCaw, M. C. Smear, and H. Baier, “Behavioral screening assays in zebrafish,” *Methods in cell biology*, vol. 77, pp. 53–68, 2004.
- [135] J. Gehrig, G. Pandey, and J. H. Westhoff, “Zebrafish as a model for drug screening in genetic kidney diseases,” *Frontiers in Pediatrics*, vol. 6, p. 183, 2018.
- [136] M. Brand, M. Granato, and C. Nüsslein-Volhard, “Keeping and raising zebrafish,” *Zebrafish: A Practical Approach*, vol. 1, pp. 7–37, 2002.
- [137] N. C. Kuznik, V. Solozobova, I. I. Lee, N. Jung, L. Yang, K. Nienhaus, E. A. Ntim, J. T. Rottenberg, C. Muhle-Goll, A. R. Kumar, *et al.*, “A chemical probe for bag1 targets androgen receptor-positive prostate cancer through oxidative stress signaling pathway,” *Iscience*, vol. 25, no. 5, p. 104175, 2022.
- [138] G. Weidinger, “Zebrafish (danio rerio) as model system,” 2016. Available: https://www.uni-ulm.de/fileadmin/website_uni_ulm/tfz/Weiterbildung/Weiterbildung_2016/2016-07-05_GW_Weidinger_Vortrag_Zebrafish__Danio_rerio__Aquaday_2016.pdf. Accessed on: May 24, 2023.
- [139] C. B. Kimmel, W. W. Ballard, S. R. Kimmel, B. Ullmann, and T. F. Schilling, “Stages of embryonic development of the zebrafish,” *Developmental Dynamics*, vol. 203, no. 3, pp. 253–310, 1995.
- [140] J. F. De la Paz, N. Beiza, S. Paredes-Zúñiga, M. S. Hoare, and M. L. Allende, “Triazole fungicides inhibit zebrafish hatching by blocking the secretory function of hatching gland cells,” *International Journal of Molecular Sciences*, vol. 18, no. 4, p. 710, 2017.

- [141] D. Bonsignorio, L. Perego, L. Del Giacco, and F. Cotelli, "Structure and macromolecular composition of the zebrafish egg chorion," *Zygote*, vol. 4, no. 2, pp. 101–108, 1996.
- [142] B. Swinnen, A. Bento-Abreu, T. F. Gendron, S. Boeynaems, E. Bogaert, R. Nuyts, M. Timmers, W. Scheveneels, N. Hersmus, J. Wang, *et al.*, "A zebrafish model for c9orf72 als reveals rna toxicity as a pathogenic mechanism," *Acta Neuropathologica*, vol. 135, pp. 427–443, 2018.
- [143] K. D. H. D. Wesstrom, S. and S. Padilla, "Dechoriation of zebrafish embryos on day 1 post fertilization alters response to an acute chemical challenge at 6 days post fertilization," Carolinas SETAC, Durham, North Carolina, Apr 2018.
- [144] R. E. Lindeman and F. Pelegri, "Localized products of futile cycle/lrmp promote centrosome-nucleus attachment in the zebrafish zygote," *Current Biology*, vol. 22, no. 10, pp. 843–851, 2012.
- [145] R. Fuentes and J. Fernández, "Ooplasmic segregation in the zebrafish zygote and early embryo: pattern of ooplasmic movements and transport pathways," *Developmental Dynamics*, vol. 239, no. 8, pp. 2172–2189, 2010.
- [146] W. A. Thompson and M. M. Vijayan, "Zygotic venlafaxine exposure impacts behavioral programming by disrupting brain serotonin in zebrafish," *Environmental Science & Technology*, vol. 54, no. 22, pp. 14578–14588, 2020.
- [147] R. E. Lindeman and F. Pelegri, "Vertebrate maternal-effect genes: Insights into fertilization, early cleavage divisions, and germ cell determinant localization from studies in the zebrafish," *Molecular Reproduction and Development: Incorporating Gamete Research*, vol. 77, no. 4, pp. 299–313, 2010.
- [148] M. Wühr, N. Obholzer, S. Megason, H. Detrich III, and T. Mitchison, "Live imaging of the cytoskeleton in early cleavage-stage zebrafish embryos," *Methods in Cell Biology*, vol. 101, pp. 1–18, 2011.
- [149] S. Adar-Levor, D. Nachmias, S. T. Gal-Oz, Y. M. Jahn, N. Peyrieras, A. Zaritsky, R. Y. Birnbaum, and N. Elia, "Cytokinetic abscission is part of the midblastula transition in early zebrafish embryogenesis," *Proceedings of the National Academy of Sciences*, vol. 118, no. 15, p. e2021210118, 2021.
- [150] D. L. Demy, Z. Ranta, J.-M. Giorgi, M. Gonzalez, P. Herbomel, and K. Kissa, "Generating parabioc zebrafish embryos for cell migration and homing studies," *Nature Methods*, vol. 10, no. 3, pp. 256–258, 2013.

- [151] D. C. Weiser, U. J. Pyati, and D. Kimelman, “Gravin regulates mesodermal cell behavior changes required for axis elongation during zebrafish gastrulation,” *Genes & Development*, vol. 21, no. 12, pp. 1559–1571, 2007.
- [152] A. Voltes, C. F. Hevia, C. Engel-Pizcueta, C. Dingare, S. Calzolari, J. Terriente, C. Norden, V. Lecaudey, and C. Pujades, “Yap/taz-tead activity links mechanical cues to progenitor cell behavior during zebrafish hindbrain segmentation,” *Development*, vol. 146, no. 14, p. dev176735, 2019.
- [153] K. A. Horzmann, A. M. Portales, K. G. Batcho, and J. L. Freeman, “Developmental toxicity of trichloroethylene in zebrafish (*danio rerio*),” *Environmental Science: Processes & Impacts*, vol. 22, no. 3, pp. 728–739, 2020.
- [154] N. Villamizar, L. Ribas, F. Piferrer, L. M. Vera, and F. J. Sánchez-Vázquez, “Impact of daily thermocycles on hatching rhythms, larval performance and sex differentiation of zebrafish,” *PLoS One*, vol. 7, no. 12, p. e52153, 2012.
- [155] J. Ord, “Ionic stress prompts premature hatching of zebrafish (*danio rerio*) embryos,” *Fishes*, vol. 4, no. 1, p. 20, 2019.
- [156] S. M. Lindsay and R. G. Vogt, “Behavioral responses of newly hatched zebrafish (*danio rerio*) to amino acid chemostimulants,” *Chemical Senses*, vol. 29, no. 2, pp. 93–100, 2004.
- [157] Y.-C. Liu, I. Bailey, and M. E. Hale, “Alternative startle motor patterns and behaviors in the larval zebrafish (*danio rerio*),” *Journal of Comparative Physiology A*, vol. 198, no. 1, pp. 11–24, 2012.
- [158] D. Marcato, *Design and Development of Imaging Platforms for Phenotypic Characterization of Early Zebrafish*. PhD thesis, Karlsruher Institut für Technologie (KIT), 2018.
- [159] T. E. Sztal, A. A. Ruparelia, C. Williams, and R. J. Bryson-Richardson, “Using touch-evoked response and locomotion assays to assess muscle performance and function in zebrafish,” *JoVE (Journal of Visualized Experiments)*, no. 116, p. e54431, 2016.
- [160] T. Pietri, E. Manalo, J. Ryan, L. Saint-Amant, and P. Washbourne, “Glutamate drives the touch response through a rostral loop in the spinal cord of zebrafish embryos,” *Developmental Neurobiology*, vol. 69, no. 12, pp. 780–795, 2009.
- [161] R. C. Eaton and C. B. Kimmel, “Directional sensitivity of the mauthner cell system to vibrational stimulation in zebrafish larvae,” *Journal of Comparative Physiology*, vol. 140, no. 4, pp. 337–342, 1980.

- [162] R. Candelier, M. S. Murmu, S. A. Romano, A. Jouary, G. Debrégeas, and G. Sumbre, “A microfluidic device to study neuronal and motor responses to acute chemical stimuli in zebrafish,” *Scientific Reports*, vol. 5, p. 12196, 2015.
- [163] A. Y. Kobitski, J. C. Otte, M. Takamiya, B. Schäfer, J. Mertes, J. Stegmaier, S. Rastegar, F. Rindone, V. Hartmann, R. Stotzka, *et al.*, “An ensemble-averaged, cell density-based digital model of zebrafish embryo development derived from light-sheet microscopy data with single-cell resolution,” *Scientific Reports*, vol. 5, no. 1, p. 8601, 2015.
- [164] J. L. van Leeuwen, C. J. Voesenek, and U. K. Müller, “How body torque and strouhal number change with swimming speed and developmental stage in larval zebrafish,” *Journal of The Royal Society Interface*, vol. 12, no. 110, p. 20150479, 2015.
- [165] J. Chhetri, G. Jacobson, and N. Gueven, “Zebrafish—on the move towards ophthalmological research,” *Eye*, vol. 28, no. 4, pp. 367–380, 2014.
- [166] C. Parng, W. L. Seng, C. Semino, and P. McGrath, “Zebrafish: a preclinical model for drug screening,” *Assay and Drug Development Technologies*, vol. 1, no. 1, pp. 41–48, 2002.
- [167] S.-S. Zhao, Y.-L. Wang, M.-Z. Sun, L. Lu, Y.-N. Wang, D. Pfister, J. Lee, X. Zhao, X.-Z. Feng, and L. Li, “Drug screening: zebrafish as a tool for studying epileptic-related chemical compounds,” *Protein & Cell*, vol. 6, no. 11, pp. 853–857, 2015.
- [168] F. Sangermano, M. Masi, A. Kumar, R. Peravali, A. Tuzi, A. Cimmino, D. Vallone, G. Giamundo, I. Conte, A. Evidente, *et al.*, “In vitro and in vivo toxicity evaluation of natural products with potential applications as biopesticides,” *Toxins*, vol. 13, no. 11, p. 805, 2021.
- [169] Y. Naganawa and H. Hirata, “Developmental transition of touch response from slow muscle-mediated coilings to fast muscle-mediated burst swimming in zebrafish,” *Developmental Biology*, vol. 355, no. 2, pp. 194–204, 2011.
- [170] L. Guzman, G. Besa, D. Linares, L. González, C. Pont, M. Bartolini, A.-C. Haigis, J. Legradi, D. Muñoz-Torrero, J. Gómez-Catalán, *et al.*, “Evaluation of the effects of acetylcholinesterase inhibitors in the zebrafish touch-evoked response: quantitative vs. qualitative assessment,” *Environmental Sciences Europe*, vol. 32, no. 1, pp. 1–12, 2020.
- [171] S. Van Helvert and P. Friedl, “Strain stiffening of fibrillar collagen during individual and collective cell migration identified by afm nanoindentation,” *ACS Applied Materials & Interfaces*, vol. 8, no. 34, pp. 21946–21955, 2016.
- [172] S. E. Low, K. Amburgey, E. Horstick, J. Linsley, S. M. Sprague, W. W. Cui, W. Zhou, H. Hirata, L. Saint-Amant, R. I. Hume, *et al.*, “Trpm7 is required within zebrafish sensory

- neurons for the activation of touch-evoked escape behaviors,” *Journal of Neuroscience*, vol. 31, no. 32, pp. 11633–11644, 2011.
- [173] L. Saint-Amant and P. Drapeau, “Time course of the development of motor behaviors in the zebrafish embryo,” *Journal of Neurobiology*, vol. 37, no. 4, pp. 622–632, 1998.
- [174] G. B. Downes and M. Granato, “Supraspinal input is dispensable to generate glycine-mediated locomotive behaviors in the zebrafish embryo,” *Journal of Neurobiology*, vol. 66, no. 5, pp. 437–451, 2006.
- [175] Y. Wang, D. Marcato, V. Tirumalasetty, N. K. Kanagaraj, C. Pylatiuk, R. Mikut, R. Peravali, and M. Reischl, “An automated experimentation system for the touch-response quantification of zebrafish larvae,” *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 4, pp. 3007–3019, 2021.
- [176] Research Dive, “Drug screening laboratory services market report,” Jun 2021. Available: <https://www.researchdive.com/8406/drug-screening-laboratory-services-market#:~:text=The%20North%20America%20drug%20screening,and%20drugs%20among%20the%20millennials..> Accessed on: May 24, 2023.
- [177] J. Jung, M. Nakajima, M. Takeuchi, Z. Najdovski, Q. Huang, and T. Fukuda, “Microfluidic device to measure the speed of *c. elegans* using the resistance change of the flexible electrode,” *Micromachines*, vol. 7, no. 3, p. 50, 2016.
- [178] J. Potthoff, D. Lütjohann, and N. Jung, “Trustworthy laboratory automation,” *DBKDA 2014*, p. 106, 2014.
- [179] K. Chatzilygeroudis, V. Vassiliades, F. Stulp, S. Calinon, and J.-B. Mouret, “A survey on policy search algorithms for learning robot controllers in a handful of trials,” *IEEE Transactions on Robotics*, vol. 36, no. 2, pp. 328–347, 2019.
- [180] A. C. Roberts, J. Chornak, J. B. Alzagatiti, D. T. Ly, B. R. Bill, J. Trinkeller, K. C. Pearce, R. C. Choe, C. Campbell, D. Wong, *et al.*, “Rapid habituation of a touch-induced escape response in zebrafish (*danio rerio*) larvae,” *PLoS One*, vol. 14, no. 4, p. e0214374, 2019.
- [181] A. V. Kalueff, M. Gebhardt, A. M. Stewart, J. M. Cachat, M. Brimmer, J. S. Chawla, C. Craddock, E. J. Kyzar, A. Roth, S. Landsman, *et al.*, “Towards a comprehensive catalog of zebrafish behavior 1.0 and beyond,” *Zebrafish*, vol. 10, no. 1, pp. 70–86, 2013.
- [182] Y. Wang, C. Pylatiuk, R. Mikut, R. Peravali, and M. Reischl, “A fully automated touch-response behavior inspection pipeline on zebrafish larvae,” *at - Automatisierungstechnik (accepted)*, 2023.

- [183] M. Granato, F. Van Eeden, U. Schach, T. Trowe, M. Brand, M. Furutani-Seiki, P. Haffter, M. Hammerschmidt, C.-P. Heisenberg, Y.-J. Jiang, *et al.*, “Genes controlling and mediating locomotion behavior of the zebrafish embryo and larva,” *Development*, vol. 123, no. 1, pp. 399–413, 1996.
- [184] S. A. Budick and D. M. O’Malley, “Locomotor repertoire of the larval zebrafish: swimming, turning and prey capture,” *Journal of Experimental Biology*, vol. 203, no. 17, pp. 2565–2579, 2000.
- [185] T. W. Dunn, C. Gebhardt, E. A. Naumann, C. Riegler, M. B. Ahrens, F. Engert, and F. Del Bene, “Neural circuits underlying visually evoked escapes in larval zebrafish,” *Neuron*, vol. 89, no. 3, pp. 613–628, 2016.
- [186] J. Duan, Y. Yu, H. Shi, L. Tian, C. Guo, P. Huang, X. Zhou, S. Peng, and Z. Sun, “Toxic effects of silica nanoparticles on zebrafish embryos and larvae,” *PloS One*, vol. 8, no. 9, 2013.
- [187] G. Audira, B. P. Sampurna, S. Juniardi, S.-T. Liang, Y.-H. Lai, and C.-D. Hsiao, “A simple setup to perform 3d locomotion tracking in zebrafish by using a single camera,” *Inventions*, vol. 3, no. 1, p. 11, 2018.
- [188] Azenta, “24 well assay plate,” 2023. Available: <https://www.azenta.com/products/24-well-assay-plate>. Accessed on: May 24, 2023.
- [189] Corning, “Corning® elplasia® 6-well black/clear round bottom ultra-low attachment, microcavity plate, with lid,” 2020. Available: <https://ecatalog.corning.com/life-sciences/b2b/CH/en/Microplates/Assay-Microplates/Corning%2%AE-Elplasia%2%AE-Plates/p/4440>. Accessed on: May 24, 2023.
- [190] K. D. Duncan, J. Fyrestam, and I. Lanekoff, “Advances in mass spectrometry based single-cell metabolomics,” *Analyst*, vol. 144, no. 3, pp. 782–793, 2019.
- [191] J. Jiang, J. S. Moore, H. F. Edelhauser, and M. R. Prausnitz, “Intrascleral drug delivery to the eye using hollow microneedles,” *Pharmaceutical Research*, vol. 26, no. 2, pp. 395–403, 2009.
- [192] P. Umesh, “Image processing in python,” *CSI Communications*, vol. 23, no. 2, 2012.
- [193] I. Culjak, D. Abram, T. Pribanic, H. Dzapo, and M. Cifrek, “A brief introduction to opencv,” in *2012 Proceedings of the 35th International Convention MIPRO*, pp. 1725–1730, IEEE, 2012.
- [194] P. Goldsborough, “A tour of tensorflow,” *arXiv preprint arXiv:1610.01178*, 2016.

- [195] B. Pang, E. Nijkamp, and Y. N. Wu, “Deep learning with tensorflow: A review,” *Journal of Educational and Behavioral Statistics*, vol. 45, no. 2, pp. 227–248, 2020.
- [196] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, “Pytorch: An imperative style, high-performance deep learning library,” *Advances in Neural Information processing Systems*, vol. 32, 2019.
- [197] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, “Scikit-learn: Machine learning in python,” *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.
- [198] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, *et al.*, “Scipy 1.0: fundamental algorithms for scientific computing in python,” *Nature Methods*, vol. 17, no. 3, pp. 261–272, 2020.
- [199] Y. Wang, C. Pylatiuk, R. Mikut, R. Peravali, and M. Reischl, “Quantification platform for touch response of zebrafish larvae using machine learning,” in *The proceedings of the 31st Workshop on Computational Intelligence*, vol. 25, p. 37, 2021.
- [200] B. K. Horn and B. G. Schunck, “Determining optical flow,” in *Techniques and Applications of Image Understanding*, vol. 281, pp. 319–331, International Society for Optics and Photonics, 1981.
- [201] G. Welch, G. Bishop, *et al.*, “An introduction to the kalman filter,” 1995.
- [202] S. Chen, “Kalman filter for robot vision: a survey,” *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4409–4420, 2011.
- [203] Q. Zhu, Y. Wang, Y. He, and X. Hong, “Object tracking with particles weighted by region proposal network,” *Multimedia Tools and Applications*, vol. 78, no. 9, pp. 12083–12101, 2019.
- [204] R. Sathya, A. Abraham, *et al.*, “Comparison of supervised and unsupervised learning algorithms for pattern classification,” *International Journal of Advanced Research in Artificial Intelligence*, vol. 2, no. 2, pp. 34–38, 2013.
- [205] Z. Ghahramani, “Unsupervised learning,” *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2-14, 2003, Tübingen, Germany, August 4-16, 2003, Revised Lectures*, pp. 72–112, 2004.
- [206] P. Pichler and L. Lagnado, “The transfer characteristics of hair cells encoding mechanical stimuli in the lateral line of zebrafish,” *Journal of Neuroscience*, vol. 39, no. 1, pp. 112–124, 2019.

- [207] Y. Tomizawa, K. Dixit, D. Daggett, and K. Hoshino, “Biocompatible cantilevers for mechanical characterization of zebrafish embryos using image analysis,” *Sensors*, vol. 19, no. 7, p. 1506, 2019.
- [208] A. F. Mead, G. G. Kennedy, B. M. Palmer, A. M. Ebert, and D. M. Warshaw, “Mechanical characteristics of ultrafast zebrafish larval swimming muscles,” *Biophysical Journal*, vol. 119, no. 4, pp. 806–820, 2020.
- [209] G. Csurka, D. Larlus, F. Perronnin, and F. Meylan, “What is a good evaluation measure for semantic segmentation?,” in *British Machine Vision Conference (BMVC)*, vol. 27, pp. 10–5244, Sept. 2013.
- [210] Y. Wang, N. K. Kanagaraj, C. Pylatiuk, R. Mikut, R. Peravali, and M. Reischl, “High-throughput data acquisition platform for multi-larvae touch-response behavior screening of zebrafish,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 858–865, 2021.
- [211] T. J. Atherton and D. J. Kerbyson, “Size invariant circle detection,” *Image and Vision Computing*, vol. 17, no. 11, pp. 795–803, 1999.
- [212] F. Romero-Ferrero, M. G. Bergomi, R. C. Hinz, F. J. Heras, and G. G. De Polavieja, “Idtracker. ai: tracking all individuals in small or large collectives of unmarked animals,” *Nature Methods*, vol. 16, no. 2, pp. 179–182, 2019.
- [213] X. Wang, E. Cheng, I. S. Burnett, Y. Huang, and D. Wlodkovic, “Automatic multiple zebrafish larvae tracking in unconstrained microscopic video conditions,” *Scientific Reports*, vol. 7, no. 1, pp. 1–8, 2017.
- [214] T. Y. Zhang and C. Y. Suen, “A fast parallel algorithm for thinning digital patterns,” *Communications of the ACM*, vol. 27, no. 3, pp. 236–239, 1984.
- [215] H. Richendrfer, S. Pelkowski, R. Colwill, and R. Creton, “On the edge: pharmacological evidence for anxiety-related behavior in zebrafish larvae,” *Behavioural Brain Research*, vol. 228, no. 1, pp. 99–106, 2012.
- [216] X. Lin, X. Li, and X. Lin, “A review on applications of computational methods in drug screening and design,” *Molecules*, vol. 25, no. 6, p. 1375, 2020.
- [217] X. Li, X. Sun, Y. Meng, J. Liang, F. Wu, and J. Li, “Dice loss for data-imbalanced nlp tasks,” *arXiv preprint arXiv:1911.02855*, 2019.
- [218] D. Kokel, J. Bryan, C. Laggner, R. White, C. Y. J. Cheung, R. Mateus, D. Healey, S. Kim, A. A. Werdich, S. J. Haggarty, *et al.*, “Rapid behavior-based identification of neuroactive small molecules in the zebrafish,” *Nature Chemical Biology*, vol. 6, no. 3, pp. 231–237, 2010.

- [219] D. Kokel, T. W. Dunn, M. B. Ahrens, R. Alshut, C. Y. J. Cheung, L. Saint-Amant, G. Bruni, R. Mateus, T. J. van Ham, T. Shiraki, *et al.*, “Identification of nonvisual photomotor response cells in the vertebrate hindbrain,” *Journal of Neuroscience*, vol. 33, no. 9, pp. 3834–3843, 2013.
- [220] A. J. Figueredo and P. S. A. Wolf, “Assortative pairing and life history strategy – a cross-cultural study,” *Human Nature*, vol. 20, pp. 317–330, 2009.
- [221] Z. Hao, A. AghaKouchak, N. Nakhjiri, and A. Farahmand, “Global integrated drought monitoring and prediction system (GIDMaPS) data sets,” 2014. Available: <http://dx.doi.org/10.6084/m9.figshare.853801>. Accessed on: May 24, 2023.
- [222] B. P. Gupta and P. Rezai, “Microfluidic approaches for manipulating, imaging, and screening *c. elegans*,” *Micromachines*, vol. 7, no. 7, p. 123, 2016.
- [223] Y. Cho, D. N. Oakland, S. A. Lee, W. R. Schafer, and H. Lu, “On-chip functional neuroimaging with mechanical stimulation in *caenorhabditis elegans* larvae for studying development and neural circuits,” *Lab on a Chip*, vol. 18, no. 4, pp. 601–609, 2018.
- [224] A. Bounoutas and M. Chalfie, “Touch sensitivity in *caenorhabditis elegans*,” *Pflügers Archiv-European Journal of Physiology*, vol. 454, no. 5, pp. 691–702, 2007.
- [225] D. M. Parichy, M. R. Elizondo, M. G. Mills, T. N. Gordon, and R. E. Engeszer, “Normal table of postembryonic zebrafish development: staging by externally visible anatomy of the living fish,” *Developmental Dynamics*, vol. 238, no. 12, pp. 2975–3015, 2009.
- [226] S. E. Brockerhoff, “Measuring the optokinetic response of zebrafish larvae,” *Nature Protocols*, vol. 1, no. 5, p. 2448, 2006.
- [227] W. Mo, F. Chen, A. Nechiporuk, and T. Nicolson, “Quantification of vestibular-induced eye movements in zebrafish larvae,” *BMC Neuroscience*, vol. 11, no. 1, p. 110, 2010.
- [228] J. C. Way and M. Chalfie, “The *mec-3* gene of *caenorhabditis elegans* requires its own product for maintained expression and is expressed in three neuronal cell types,” *Genes & Development*, vol. 3, no. 12a, pp. 1823–1833, 1989.
- [229] A. C. Hart, S. Sims, and J. M. Kaplan, “Synaptic code for sensory modalities revealed by *c. elegans* *glr-1* glutamate receptor,” *Nature*, vol. 378, no. 6552, pp. 82–85, 1995.
- [230] M. Chatzigeorgiou, S. Yoo, J. D. Watson, W.-H. Lee, W. C. Spencer, K. S. Kindt, S. W. Hwang, D. M. Miller III, M. Treinin, M. Driscoll, *et al.*, “Specific roles for *deg/enac* and *trp* channels in touch and thermosensation in *c. elegans* nociceptors,” *Nature Neuroscience*, vol. 13, no. 7, pp. 861–868, 2010.

- [231] Y. Gao, R. H. Chan, T. W. Chow, L. Zhang, S. Bonilla, C.-P. Pang, M. Zhang, and Y. F. Leung, "A high-throughput zebrafish screening method for visual mutants by light-induced locomotor response," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 11, no. 4, pp. 693–701, 2014.
- [232] Y. Wang, D. Marcato, V. Tirumalasetty, N. K. Kanagaraj, C. Pylatiuk, R. Mikut, R. Peravali, and M. Reischl, "An automated experimentation system for the touch-response quantification of zebrafish larvae," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 4, pp. 3007–3019, 2022.
- [233] H. B. Atakan, T. Alkanat, M. Cornaglia, R. Trouillon, and M. A. Gijs, "Automated phenotyping of caenorhabditis elegans embryos with a high-throughput-screening microfluidic platform," *Microsystems & Nanoengineering*, vol. 6, pp. 1–14, Apr. 2020.
- [234] J. C. Puchalt, A.-J. Sánchez-Salmerón, E. Ivorra, S. Llopis, R. Martínez, and P. Martorell, "Small flexible automated system for monitoring caenorhabditis elegans lifespan based on active vision and image processing techniques," *Scientific Reports*, vol. 11, pp. 1–11, June 2021.
- [235] K. Kaarj and J.-Y. Yoon, "Methods of delivering mechanical stimuli to organ-on-a-chip," *Micromachines*, vol. 10, p. 700, Oct. 2019.
- [236] J. M. Northcott, I. S. Dean, J. K. Mouw, and V. M. Weaver, "Feeling stress: the mechanics of cancer progression and aggression," *Frontiers in Cell and Developmental Biology*, vol. 6, p. 17, Feb. 2018.
- [237] S. Dong, X. Liu, P. Li, X. Tang, D. Liu, M. Kojima, Q. Huang, and T. Arai, "Automated tracking system with head and tail recognition for time-lapse observation of free-moving *c. elegans*," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9257–9262, May 2020.
- [238] X. Zhang, J. Sun, X. Yuan, X. Lu, and X. Sun, "Advances in *c. elegans* behavior research with microfluidic devices and its future prospects in the evaluation of exogenous pollutants," *TrAC Trends in Analytical Chemistry*, p. 116195, Jan. 2021.
- [239] R. M. Basnet, D. Zizioli, S. Taweedet, D. Finazzi, and M. Memo, "Zebrafish larvae as a behavioral model in neuropharmacology," *Biomedicines*, vol. 7, no. 1, p. 23, 2019.
- [240] A. Kithcart and C. A. MacRae, "Using zebrafish for high-throughput screening of novel cardiovascular drugs," *JACC: Basic to Translational Science*, vol. 2, pp. 1–12, Feb. 2017.

- [241] T. C. Farrell, C. L. Cario, C. Milanese, A. Vogt, J.-H. Jeong, and E. A. Burton, "Evaluation of spontaneous propulsive movement as a screening tool to detect rescue of parkinsonism phenotypes in zebrafish models," *Neurobiology of Disease*, vol. 44, pp. 9–18, Oct. 2011.
- [242] M. Itoh and K. Hatta, "Munch's scream: A spontaneous movement by zebrafish larvae featuring strong abduction of both pectoral fins often associated with a sudden bend," *Neuroscience Research*, vol. 94, pp. 17–27, May 2015.
- [243] M. B. Orger and G. G. de Polavieja, "Zebrafish behavior: opportunities and challenges," *Annual Review of Neuroscience*, vol. 40, pp. 125–147, July 2017.
- [244] K. Mani, T.-C. C. Chien, and C.-Y. Chen, "Orientation control of zebrafish embryos using artificial cilia in a 3d flow-through microchannel," in *10th IEEE International Conference on Nano/Micro Engineered and Molecular Systems*, pp. 15–19, 2015.
- [245] S. Zhuang, W. Lin, J. Zhong, G. Zhang, L. Li, J. Qiu, and H. Gao, "Visual servoed three-dimensional rotation control in zebrafish larva heart microinjection system," *IEEE Transactions on Biomedical Engineering*, vol. 65, pp. 64–73, Jan. 2018.
- [246] Z. Zhu, Y. Geng, Z. Yuan, S. Ren, M. Liu, Z. Meng, and D. Pan, "A bubble-free and low-shear-stress microfluidic device for high-quality monitoring of zebrafish embryonic development," in *2019 20th International Conference on Solid-State Sensors, Actuators and Microsystems Eurosensors XXXIII (Transducers & Eurosensors XXXIII)*, pp. 921–924, 2019.
- [247] G. Zhang, M. Tong, S. Zhuang, X. Yu, W. Sun, W. Lin, and H. Gao, "Zebrafish larva orientation and smooth aspiration control for microinjection," *IEEE Transactions on Biomedical Engineering*, vol. 68, pp. 47–55, Jan. 2021.
- [248] R. Vaz, W. Hofmeister, and A. Lindstrand, "Zebrafish models of neurodevelopmental disorders: limitations and benefits of current tools and techniques," *International Journal of Molecular Sciences*, vol. 20, p. 1296, Jan. 2019.
- [249] A. Ogungbemi, D. Leuthold, S. Scholz, and E. Küster, "Hypo-or hyperactivity of zebrafish embryos provoked by neuroactive substances: A review on how experimental parameters impact the predictability of behavior changes," *Environmental Sciences Europe*, vol. 31, pp. 1–26, Dec. 2019.
- [250] J. Braam, "In touch: plant responses to mechanical stimuli," *New Phytologist*, vol. 165, no. 2, pp. 373–389, 2005.
- [251] S. Belin, K. L. Zuloaga, and Y. Poitelon, "Influence of mechanical stimuli on schwann cell biology," *Frontiers in Cellular Neuroscience*, vol. 11, p. 347, 2017.

- [252] H. Kim, H. Arakawa, T. Osada, and A. Ikai, "Quantification of cell adhesion force with afm: distribution of vitronectin receptors on a living mc3t3-e1 cell," *Ultramicroscopy*, vol. 97, no. 1-4, pp. 359–363, 2003.
- [253] C. Obbink-Huizer, C. W. Oomens, S. Loerakker, J. Foolen, C. V. Bouten, and F. P. Baaijens, "Computational model predicts cell orientation in response to a range of mechanical stimuli," *Biomechanics and Modeling in Mechanobiology*, vol. 13, no. 1, pp. 227–236, 2014.
- [254] D.-H. Kim, S. Yun, and B. Kim, "Mechanical force response of single living cells using a microrobotic system," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 5, pp. 5013–5018, IEEE, 2004.
- [255] P.-H. Wu, C. M. Hale, W.-C. Chen, J. S. Lee, Y. Tseng, and D. Wirtz, "High-throughput ballistic injection nanorheology to measure cell mechanics," *Nature Protocols*, vol. 7, no. 1, pp. 155–170, 2012.
- [256] P. D. Antonio, M. Lasalvia, G. Perna, and V. Capozzi, "Scale-independent roughness value of cell membranes studied by means of afm technique," *Biochimica et Biophysica Acta (BBA)-Biomembranes*, vol. 1818, no. 12, pp. 3141–3148, 2012.
- [257] M. Sato and T. Ohashi, "Biorheological views of endothelial cell responses to mechanical stimuli," *Biorheology*, vol. 42, no. 6, pp. 421–441, 2005.
- [258] M. T. A. Saif, C. R. Sager, and S. Coyer, "Functionalized biomicroelectromechanical systems sensors for force response study at local adhesion sites of single living cells on substrates," *Annals of Biomedical Engineering*, vol. 31, no. 8, pp. 950–961, 2003.
- [259] M. E. Goldschmidt, K. J. McLeod, and W. R. Taylor, "Integrin-mediated mechanotransduction in vascular smooth muscle cells: frequency and force response characteristics," *Circulation Research*, vol. 88, no. 7, pp. 674–680, 2001.
- [260] C. Rianna and M. Radmacher, "Cell mechanics as a marker for diseases: biomedical applications of afm," in *AIP Conference Proceedings*, vol. 1760, p. 020057, AIP Publishing LLC, 2016.
- [261] K. M. Yamada and M. Sixt, "Mechanisms of 3d cell migration," *Nature Reviews Molecular Cell Biology*, vol. 20, no. 12, pp. 738–752, 2019.
- [262] M. Bao, J. Xie, N. Katoele, X. Hu, B. Wang, A. Piruska, and W. T. Huck, "Cellular volume and matrix stiffness direct stem cell behavior in a 3d microniche," *ACS Applied Materials & Interfaces*, vol. 11, no. 2, pp. 1754–1759, 2018.

- [263] C. S. Chen, “Mechanotransduction—a field pulling together?,” *Journal of Cell Science*, vol. 121, no. 20, pp. 3285–3292, 2008.
- [264] J. F. Cornhill, M. J. Levesque, E. E. Herderick, R. M. Nerem, J. W. Kilman, and J. S. Vasko, “Quantitative study of the rabbit aortic endothelium using vascular casts,” *Atherosclerosis*, vol. 35, no. 3, pp. 321–337, 1980.
- [265] C. S. Pinto, M. Mishima, and K. Sampath, “Tools of the trade: studying actin in zebrafish,” *Histochemistry and Cell Biology*, pp. 1–13, 2020.
- [266] B. Bonneau, N. Popgeorgiev, J. Prudent, and G. Gillet, “Cytoskeleton dynamics in early zebrafish development: a matter of phosphorylation?,” *Bioarchitecture*, vol. 1, no. 5, pp. 216–220, 2011.
- [267] M. Krause, J. Te Riet, and K. Wolf, “Probing the compressibility of tumor cell nuclei by combined atomic force–confocal microscopy,” *Physical Biology*, vol. 10, no. 6, p. 065002, 2013.
- [268] S. Soloman, *Sensors handbook*. McGraw-Hill Education, 2010.
- [269] M. Schutera, T. Dickmeis, M. Mione, R. Peravali, D. Marcato, M. Reischl, R. Mikut, and C. Pylatiuk, “Automated phenotype pattern recognition of zebrafish for high-throughput screening,” *Bioengineered*, vol. 7, no. 4, pp. 261–265, 2016.
- [270] A. A. Popova, D. Marcato, R. Peravali, I. Wehl, U. Schepers, and P. A. Levkin, “Fish-microarray: A miniaturized platform for single-embryo high-throughput screenings,” *Advanced Functional Materials*, vol. 28, no. 3, p. 1703486, 2018.
- [271] M. de Oliveira Barreiros, D. de Oliveira Dantas, L. C. de Oliveira Silva, S. Ribeiro, and A. K. Barros, “Zebrafish tracking using yolov2 and kalman filter,” *Scientific Reports*, vol. 11, no. 1, pp. 1–14, 2021.
- [272] X. Wang, E. Cheng, I. S. Burnett, Y. Huang, and D. Wlodkovic, “Crowdsourced generation of annotated video datasets: a zebrafish larvae dataset for video segmentation and tracking evaluation,” in *2017 IEEE Life Sciences Conference (LSC)*, pp. 274–277, IEEE, 2017.
- [273] X. Wang, E. Cheng, I. S. Burnett, R. Wilkinson, and M. Lech, “Automatic tracking of multiple zebrafish larvae with resilience against segmentation errors,” in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pp. 1157–1160, IEEE, 2018.
- [274] Y.-X. Bai, S.-H. Zhang, Z. Fan, X.-Y. Liu, X. Zhao, X.-Z. Feng, and M.-Z. Sun, “Automatic multiple zebrafish tracking based on improved hog features,” *Scientific Reports*, vol. 8, no. 1, pp. 1–14, 2018.

- [275] S. Usui and I. Amidror, “Digital low-pass differentiation for biological signal processing,” *IEEE Transactions on Biomedical Engineering*, no. 10, pp. 686–693, 1982.
- [276] T. K. Kim, “T test as a parametric statistic,” *Korean Journal of Anesthesiology*, vol. 68, no. 6, pp. 540–546, 2015.
- [277] H. A. Patel and D. G. Thakore, “Moving object tracking using kalman filter,” *International Journal of Computer Science and Mobile Computing*, vol. 2, no. 4, pp. 326–332, 2013.
- [278] L. Jing, L.-Y. Hsiao, S. Li, H. Yang, P. L. P. Ng, M. Ding, T. Van Truong, S.-P. Gao, K. Li, Y.-X. Guo, *et al.*, “2d-material-integrated hydrogels as multifunctional protective skins for soft robots,” *Materials Horizons*, vol. 8, no. 7, pp. 2065–2078, 2021.
- [279] P. De Lellis, E. Cadolini, A. Croce, Y. Yang, M. Di Bernardo, and M. Porfiri, “Model-based feedback control of live zebrafish behavior via interaction with a robotic replica,” *IEEE Transactions on Robotics*, vol. 36, no. 1, pp. 28–41, 2019.
- [280] M. D. Abràmoff, P. J. Magalhães, and S. J. Ram, “Image processing with imagej,” *Biophotonics International*, vol. 11, no. 7, pp. 36–42, 2004.
- [281] A. Pérez-Escudero, J. Vicente-Page, R. C. Hinz, S. Arganda, and G. G. De Polavieja, “idtracker: tracking individuals in a group by automatic identification of unmarked animals,” *Nature Methods*, vol. 11, no. 7, pp. 743–748, 2014.
- [282] R. Bruch, P. M. Scheikl, R. Mikut, F. Loosli, and M. Reischl, “epitracker: A framework for highly reliable particle tracking for the quantitative analysis of fish movements in tanks,” *SLAS TECHNOLOGY: Translating Life Sciences Innovation*, vol. 26, no. 4, pp. 367–376, 2021.
- [283] N. Vue and Vuetify, “Scisharp,” 2019. Available: <http://www.scisharpstack.org>. Accessed on: May 24, 2023.
- [284] H. Renner, M. Grabos, K. J. Becker, T. E. Kagermeier, J. Wu, M. Otto, S. Peischard, D. Zeuschner, Y. TsyTsyura, P. Disse, *et al.*, “A fully automated high-throughput workflow for 3d-based chemical screening in human midbrain organoids,” *Elife*, vol. 9, p. e52904, 2020.
- [285] W. Yin, D. Brittain, J. Borseth, M. E. Scott, D. Williams, J. Perkins, C. S. Own, M. Murfitt, R. M. Torres, D. Kapner, *et al.*, “A petascale automated imaging pipeline for mapping neuronal circuits with high-throughput transmission electron microscopy,” *Nature Communications*, vol. 11, no. 1, p. 4949, 2020.

- [286] E. Ackerman, “A robot for the worst job in the warehouse: Boston dynamics’ stretch can move 800 heavy boxes per hour,” *IEEE Spectrum*, vol. 59, no. 1, pp. 50–51, 2022.
- [287] Merck KGaA, “Corning® costar® tc-treated multiple well plates,” 2023. Available: <https://www.sigmaaldrich.com/DE/en/product/sigma/cls3527>. Accessed on: May 24, 2023.