

Development of motion trackers for space debris research

Kush Kumar Sharma

Internship Report submitted to the International Space University in partial fulfillment of the requirements of the M.Sc. Degree in Space Studies

November, 2020

Internship Mentor: **Mrs. Denise Keil**

Host Institution: **DLR, Institute of Technical Physics, Stuttgart, Germany**

ISU Academic Advisor: **Dr. Virginia Wotring**

Abstract

Space debris are dysfunctional artificial objects that are orbiting around the earth. Miniaturization and advancement in space technology have encouraged the increase in the number of small-satellite constellations. Over the years, in-orbit catastrophic events have resulted in an exponential increase in space pollution with the ever-expanding coverage area of space debris. An international consortium of private institutions and space agencies works together to address the concern by extensive research and development related to active debris tracking and removal methods. On the same grounds, the Institute of Technical Physics of German Aerospace Center is developing ground based high energy laser facility and optical instruments to track and remove space debris from Low Earth Orbit. The internship project aims to develop a motion tracker software to track the sample in a technology demonstration experiment of impulse generation through laser-matter interaction. Several object detection and motion tracking algorithms in computer vision were reviewed and analyzed to accomplish it. For object detection, Harris Corner Detector and Scale Invariant Feature Transform algorithms exhibit a decent success rate. Optical flow point based tracking was most promising to obtain a 3-Dimensional sample track specifically in a multi-view camera configuration. The reference data files used for software development are the high-speed videos originally obtained during the laser-matter interaction experiment throughout the project.



Acknowledgements

I would like to thank Mrs. Denise Keil, for her support, guidance, and encouragement throughout the internship.

Also, I would like to thank Mr. Wolfgang Riede for providing administrative support during the visa application process.

Special thanks to Dr. Virginia Wotring and Kristell Bars for their constant support throughout the academic year and securing internship during the pandemic.

To the DLR Institute of Technical Physics for providing me an opportunity to work on the subject that I am passionate about.

To the MSS20 class and friends for their support and beautiful memories throughout the year.

To my family for giving me the chance to follow my dreams.

Table of Contents

Chapter 1: Introduction	1
1.1 Host institution.....	1
1.1.1. Institute of Technical Physics, German Aerospace Center (DLR).....	1
1.1.2. Research interest	2
1.2 Introduction to the project	3
1.2.1. Project Background.....	3
1.2.2. Aim and Objectives	3
1.2.3. Project Deliverables	5
1.3 Structure of the report.....	5
CHAPTER 2: Space Debris	6
2.1. Problem of space debris	6
2.2. Active debris removal methods.....	7
2.2.1. Contact based Removal	7
2.2.2. Contactless Removal	8
2.2.3. Laser based space debris removal.....	8
2.3.Laser-matter interaction experiment	11
2.3.1 Experimental setup	11
2.3.2. Observations.....	11
Chapter 3: Fundamentals of Computer Vision	13
3.1 Background	13
3.2 Image Preprocessing.....	14
3.2.1. Gray Level concept	14
3.3. Object and Feature detection.....	17
3.3.1. Types of features	17
3.3.2. Edge Detection	18
3.3.3. Corner detection.....	22
3.3.4. Feature Extraction.....	23
3.3.5. Feature matching:	25
3.4. Object Tracking	26
3.4.1. Point based tracking.....	27
3.4.2 Kernel based tracking	27

3.4.3. Silhouette based tracking	28
3.5. Camera Calibration:.....	29
3.6. Triangulation.....	31
Chapter 4: Computer Vision Software	32
4.1 Software Configuration	32
4.2 Feature detection and matching using python	32
4.2.1. Harris Corner Detection.....	32
4.2.2. SIFT Detection	33
4.2.3.Feature Matching	34
4.3 Software for laser-matter interaction experiment.....	35
4.3.1. Experimental Setup.....	36
4.3.2. Sample detection	37
4.3.3 2D Motion Tracker.....	40
4.3.4 3D Multi-view camera calibration.....	44
4.3.5. 3D Motion Tracker.....	47
4.4. Source of error.....	49
4.5 Estimation of Interest parameters	49
Chapter 5: Internship experience.....	50
Chapter 6: Conclusion.....	52
References	54

List of Tables

Table 1.1: Tasks conducted during internship	4
Table 6.1: Project activities with status	52

List of figures

Figure 1.1: Image from a) active optical systems, b) atmospheric propagation and effect, c) solid state lasers and non-linear optics.....	1
Figure 2.1: Growth of number of space debris as of September, 2020.	6
Figure 2.2: Photoablation on the surface of target space debris	8
Figure 2.3: Removal of debris using space-based lasers	9
Figure 2.4: Ground-based laser removal method.....	10
Figure 2.5: Experimental setup. (a) Top view of the vacuum chamber. (b) Sample dropping mechanism setup and (c) Sample trajectory during the laser-irradiation	11
Figure 3.1: a) Single object classification and localization b) Multiple object detection in a scene c) Tracking multiple objects in a scene	13
Figure 3.2: Different types of noises	15
Figure 3.3: a) Original image, b) Image after Gaussian filtering.....	16
Figure 3.4: Gaussian lobe.....	16
Figure 3.5: a) Flat region b) Edge c) Corner (credits: A. Efros) d) Blob e) Ridge	17
Figure 3.6: Image intensity variation for edge detection	18
Figure 3.7: a) Original chessboard image b) Chessboard corners detected as blue dots	22
Figure 3.8: Harris measure.....	23
Figure 3.9: SIFT feature detection	24
Figure 3.10: Computation of keypoint descriptor	25
Figure 3.11: Feature matching between two images of same scene.....	25
Figure 3.12: Locating pixel using optical-flow method.....	26
Figure 3.13: Categorization of object tracking.....	27
Figure 3.14: Kernel based tracking (red bounding box)	28
Figure 3.15: Pinhole camera model	29
Figure 3.16: Point correspondence geometry: Two cameras C and C' , projection points x and x' and 3D world point X	31
Figure 4.1: Algorithm for Harris Corner Detector.....	33
Figure 4.2: Successful detection of corners (red dots) using Harris Corner detector	33
Figure 4.3: Algorithm for SIFT detector.....	34
Figure 4.4: Feature detection using SIFT	34
Figure 4.5: Algorithm for feature matching using SIFT.....	35
Figure 4.6: Feature matching of two scale variant candle scenes.....	35
Figure 4.7: Experimental setup for high-speed sample video data.....	36

Figure 4.8: Target sample with corner markers	37
Figure 4.9: Algorithm for sample detection using sift	37
Figure 4.10: SIFT detection of a) moving objects b) dropping sample	38
Figure 4.11: Algorithm for contour-based Harris corner detector	39
Figure 4.12: Contour based detection and tracking	39
Figure 4.13: Algorithm for kernelized correlation filter tracker	40
Figure 4.14: a) Target object selection b) Two corners detected before laser-irradiation c) All four corners detected after laser-irradiation	41
Figure 4.15: Algorithm for simple optical flow point based tracker.....	42
Figure 4.16 Algorithm for robust optical flow point based tracker.....	42
Figure 4.17: a) Simple optical flow on sample videos b) Robust optical flow tracker on laser-matter experiment videos (red circle – user point selection; green line – automatic 2D trajectory)	43
Figure 4.18: a) Calibration pattern, b) Frame captured from OS7 camera, c) Frame captured from M3 camera, d) Pattern detected in OS7 frame, e) Pattern detected in M3 frame, f) 3D Pose estimation of OS7 and M3 cameras.....	46
Figure 4.19 Algorithm for 3-dimensional motion tracker.....	47
Figure 4.20: 3D trajectory of sample a) bottom- left corner (red), b) top-left corner (blue), (c)top-right corner (green).....	48

Nomenclature

Cm	Impulse Coupling
P	Ablation Pressure
I	Intensity
τ	Laser pulse duration
Φ	Laser fluence
cm	Centimeter
ms	Millisecond
mbar	Millibar

Abbreviations

AOS	Active Optical Systems
UAV	Unmanned Aerial Vehicles
DLR	Deutsches Zentrum für Luft- und Raumfahrt
EXPEDIT	Examination Program for Irregularly Shaped Debris Targets
ESA	European Space Agency
IADC	Inter-Agency Space Debris Coordination Committee
D.S.M.R.	Dynamically Scanned Magnetic Retarder
NHELIX	Nanosecond High Energy Laser for Heavy Ion Experiments
AR	Anti-Reflective
MIT	Massachusetts Institute of Technology
ISS	International Space Station
SIFT	Scale Invariant Feature Transform
DOG	Difference of Gaussians
FLANN	Fast Library for Approximate Nearest Neighbors
KCF	Kernelized Correlation Filter

Chapter 1: Introduction

This chapter presents a brief introduction to the report. Section 1.1 describes the research and activities conducted by the internship host institution. Section 1.2 provides an overview of the project along with aim, objectives, tasks and deliverables. Section 1.3 summarizes the structure of the report.

1.1 Host institution

1.1.1. Institute of Technical Physics, German Aerospace Center (DLR)

The internship project was hosted by the Institute of Technical Physics in Stuttgart, Germany, an integral part of the German Aerospace Center (DLR). Research and development in laser and optics are conducted for applications in aerospace, defense and security. Team of engineer, scientist and technician work on projects addressing interdisciplinary issues in the following key areas (DLR, nd):

1. Detection and Removal of Space debris
2. Laser-based flight instruments
3. Long-range laser effectors
4. Laser-based remote detection of harmful explosive substances

Departments under Institute of Technical Physics:

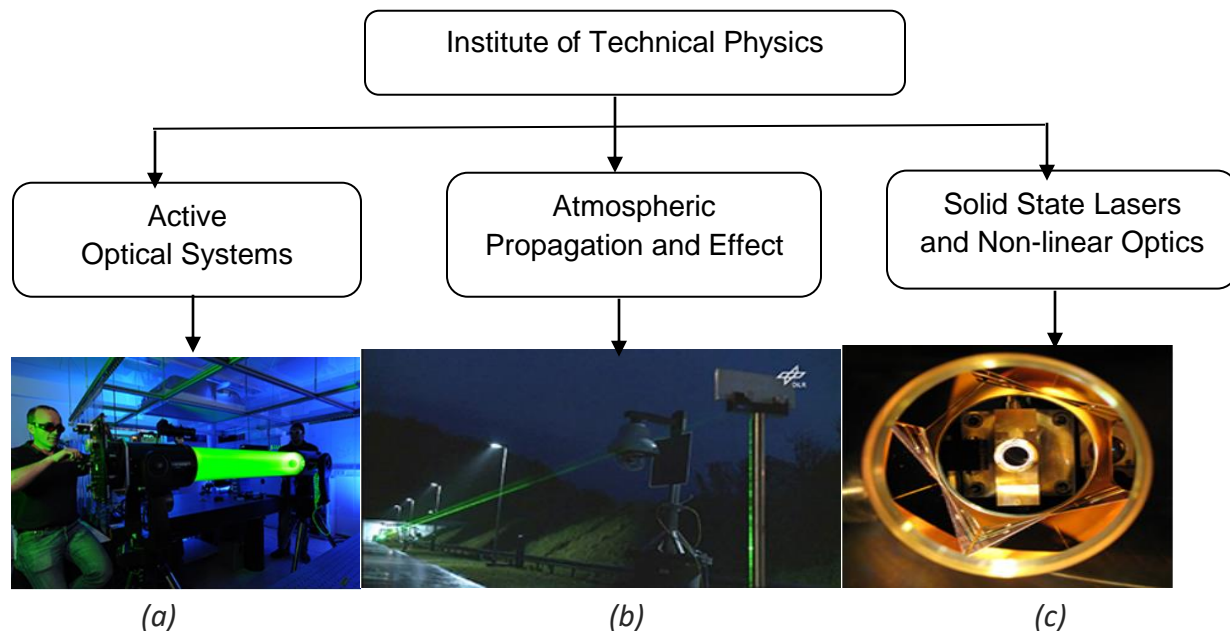


Figure 1.1: Image from a) active optical systems, b) atmospheric propagation and effect, c) solid state lasers and non-linear optics (credits: DLR)

1.1.2. Research interest

Active Optical Systems (AOS): Real-time systems required for high precision detection and tracking that enables beam towards fast-flying remote objects such as space debris and unmanned aerial vehicles (UAV) with a tracking accuracy of sub-micro radian and a bandwidth of up to several kilohertz are developed in the department (DLR, n.d).

Specific Research topics are as follows:

1. Atmospheric real-time laser tracking
2. Atmospheric compensated imaging
3. Damage threshold tests on laser optics
4. Laser-induced contamination
5. Quantification of frequency conversion crystals

Atmospheric Propagation and Effect: Studies related to the transmission of lasers over long distances and the detection of pollutants and hazardous substances are the main focus of research. Atmospheric factors such as air turbulence and aerosols on deflection and deterioration of laser beam is studied and numerically analyzed. Also, tests related to evaluation of laser effects on moving objects are carried out in the optical test range, which can be later used for orbital data acquisition and laser-based removal of space debris (DLR, n.d).

Specific Research topics are as follows:

1. Optical test range
2. Optical turbulence and micrometeorology
3. Atmospheric propagation
4. Effect
5. Laser-based standoff detection.

Solid-State Lasers and Nonlinear Optics: Optimization of solid-state lasers' beam quality in continuous wave or pulsed operations for transmission of laser over large distances is carried out for applications such as detection and removal of space debris, optical transmission etc. Laser measurement techniques for atmospheric research using non-linear optics methods are also developed (DLR,n.d).

Specific research topics are as follows:

1. Pulsed lasers and non-linear optics
2. Coherent coupling
3. High power lasers

1.2 Introduction to the project

Internship project titled “Development of motion tracker for space debris research” is carried out under the supervision of Mrs. Denise Keil, a Ph.D. student and a research scientist at the institute. The project serves as an extension and improvisation of the work “3D-Motion-Tracking zur Erfassung von Freifallereignissen im ms-Bereich”¹ conducted by the former bachelor thesis student Michael Zwilich at DLR, Institute of Technical Physics (Zwilich, 2017).

1.2.1. Project Background

The growing population of space debris causes danger to the space assets currently orbiting around the earth. At the Institute of Technical Physics of German Aerospace Center, research and development related to ground-based laser orbital debris removal are carried out. To prove the feasibility of the laser-based removal, technical concerns related to laser-matter interaction were addressed by modeling and simulation and experimental research (Eckel et al., 2016). The study was first focused on determining the relationship between impulse coupling and laser fluence for different space debris targets exhibiting simple geometric shapes such as a cube, sphere, plate etc. The drawback was the negligence of effects caused by complex geometry and random orientation of target in the calculation of laser-induced momentum. Later, Scharring et al. (2017) develop a simulation code EXPEDIT - Examination Program for Irregularly Shaped Debris Targets to investigate laser-induced thrust generation over a wide range of parameters, specifically allowing consideration of arbitrarily shaped objects. It shows the relationship between generated impulse against parameters such as materials, pressure and angle of incidence. The numerical method presented in the literature develops the basis for experimental verification and feasibility demonstration of laser-generated impulse with realistic constraints (Lorbeer et al., 2018). From the experiment conducted at GSI, Darmstadt using the NHELIX laser system explained in section 2.3, displacement in the trajectory of the dropping sample was observed after the laser hit, exhibiting impulse generation after laser-matter interaction. Lorbeer et al. (2018) mention that interest parameters, including momentum and angular momentum, were calculated using EXPEDIT and parameters such as angular velocity ($\vec{\omega}$) and velocity (\vec{v}) before and after the laser hit using python scripts. The dropping sample tracking was performed manually using ImageJ, followed by stereoscopic triangulation to determine 3D world points (Lorbeer et al., 2018).

1.2.2 Aim and Objectives

The internship project aims to develop a motion tracker software using python to automatically track the sample during laser-matter interaction experiment and validate the impulse

¹ Bachelor thesis title in English – 3D motion tracking to capture free-fall events in the ms range

generation and trajectory displacement after laser irradiation. During the internship, original test videos from the experiment were obtained using two high-speed cameras, MotionPro X3 – IDT, Inc. and RedlakeMotionScope M-3, in addition to the sample videos captured in the Institute of Technical Physics, Stuttgart, were processed for debugging computer vision algorithms.

The primary objectives of the internship are as follows:

1. Introduction to laser-based orbital debris removal
2. Introduction to object detection and motion tracking
3. Automatic object detection algorithm
4. Automatic 3D motion tracker algorithm
5. Deduce interest parameters from the obtained 3D trajectory

The objectives mentioned above are divided into four core packages. The description of the task and activities conducted to achieve the objects are described in Table 1.1. As the date of submission of the internship report does not align with the actual end of the internship. Task 3.4 is currently in progress. Task 3.5 and 4.3 are scheduled for the later part of the internship and will be included in the final internship documentation.

Table 1.1: Tasks conducted during internship

Activities / Tasks		
1.	Literature Review	
	1.1	Laser baser detection and removal of space debris
	1.2	Laser-matter interaction experiment
	1.3	Introduction to computer vision
	1.4	Identification of object detectors and motion trackers (2D)
	1.5	Beginner course on Python, Pycharm and Anaconda
2.	Computer Vision Software development in python	
	2.1	Feature Detection
	2.2	Object Detection
	2.3	Motion tracker (2 - Dimensional)

3.	Multi-view 3D tracking	
	3.1	Introduction to camera calibration and triangulation
	3.2	Understanding existing camera calibration code
	3.3	Development of motion tracker (3 - Dimensional)
	3.4	Debug 3D motion tracker
	3.5	Extract interest parameters
4.	Internship final stage	
	4.1	Prepare internship report
	4.2	Prepare presentation
	4.3	Finalize project, assessment and software package

1.2.3. Project Deliverables

1. Software:
 - 1.1 Object detector/ 2D & 3D motion tracker
 - 1.2 Interest parameter calculator
2. Documentation:
 - 2.1 Internship report
 - 2.2 Internship presentation
 - 2.3 Software user manual

1.3 Structure of the report

The structure of the report from the next chapter is as follows:

Chapter 2 describes the problem of space debris and proposed active debris removal methods. Emphasis is given on the laser-based removal method, especially using ground-based lasers and the description of technology demonstration laser-matter interaction experiment. Chapter 3 provides an introduction to the computer vision, image pre-processing, object detection and motion tracking. Chapter 4 shows the implementation of computer vision algorithms and obtaining a 3-dimensional trajectory plot. Thus, representing the actual work of the internship. Chapter 5 summarizes the overall internship experience. Chapter 6 provides the conclusion of the scientific work conducted during the internship.

CHAPTER 2: Space Debris

2.1. Problem of space debris

To date, about 2787 active satellites (UCS, 2020) and more than 34000 trackable non-functional objects with size more than 10 cm are orbiting around the earth (ESA, 2020). Space debris is all artificial objects orbiting the earth or reentering the atmosphere, including elements and fragments no longer in the active state (IADC, 2002). Though scientists were well aware of the risk posed by space debris, no effective measures were taken to control the growing population. Space debris can be broadly classified into two categories: identified or traceable and unidentified. Identified debris can be either payload, payload mission-related objects, payload fragmentation object, rocket body, rocket mission-related objects, and rocket fragmentation objects (ESA, 2020).

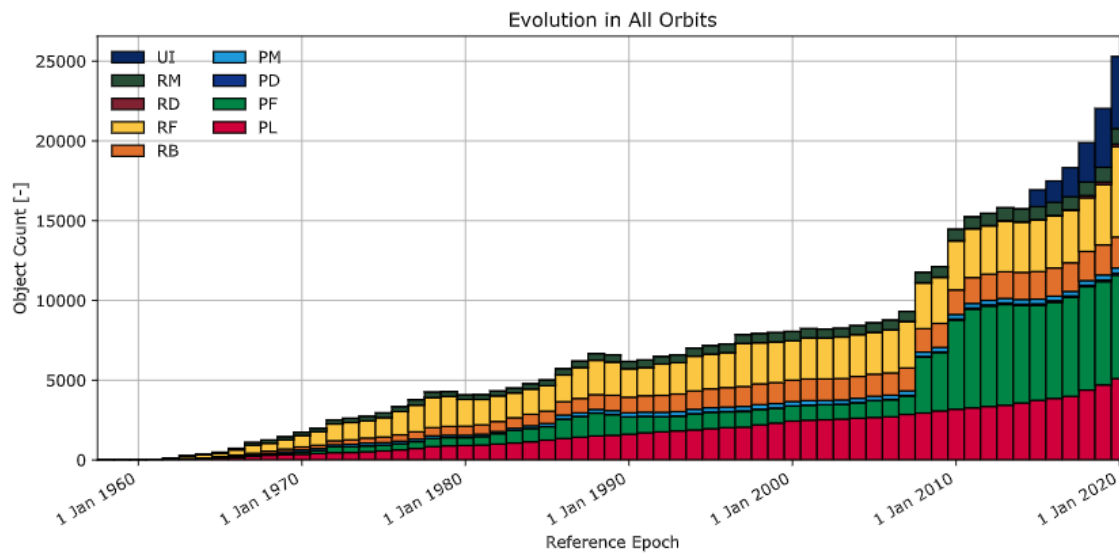


Figure 2.1: Growth of number of space debris as of September, 2020 (ESA, 2020).

The space debris problem's most common origin is the breakup events such as explosions and collisions in orbit. From the 1960s self-destruction and anti-satellite tests, fragmentation debris has become the majority amongst the debris population (Sylvestre and Parama, 2017). Notably, self-destruction of Chinese Fengyun-1C which was a defunct weather satellite and the accidental collision of Iridium 33 with Cosmos 2251 were considered as the major events resulting in origin of space debris in orbit as they added 5500 fragments to the catalog of tracked objects along with several unidentified small sized elements (Muelhaupt, 2015). As per the 2020s ESA report, there are about 900,000 objects with a size of 1 to 10cm and over 128

million objects with a size less than 1 cm (ESA, 2020). A piece of debris, traveling at high orbital velocities of about 7 - 14 km/s, can create catastrophic collision resulting in a chain reaction or Kessler syndrome. It can also result in a loss of satellite, spacecraft, or even life of astronauts. The growing population of unidentified debris elements in-orbit is a significant threat to the current and future space missions. The subject of space situational awareness and active debris removal have gained much popularity in recent years followed by some research proposals and technology demonstrations to eliminate the growing concern, as explained in the following section.

2.2. Active debris removal methods

Shan et al. (2016) mention the spectrum of the proposed concepts for capturing and removing space debris from highly dense orbits. The methods are generally classified into two categories based on the type of interaction between chaser and target as follows (Shan et al., 2016):

2.2.1. Contact based Removal

Concepts depending upon the physical interaction between the chaser and a target through an external mechanism are considered contact capturing and removal techniques (Shan et al., 2016). Some of the examples are as follows:

Dynamically Scanned Magnetic Retarder (D.S.M.R.): Sharma et al. (2018) propose a unique method to use eddy current drag force generated by a solenoid electromagnet to initially retard small and medium-sized space debris followed by a net collection mechanism. High magnitude of drag force results in damping of impact parameters of medium-sized particles and the burning of small-sized particles.

Tentacles: A target can be captured using tentacles with or without a supporting robotic arm (Shan et al., 2016). The target's point is selected by a robotic arm that enables the clamping mechanism to grasp the target. A slight increment in velocity can deorbit the target along with the chaser satellite (Biesbroek, 2012). Whereas, in the absence of a robotic arm, it ideally captures the target without any physical contact. To avoid bouncing, the attitude control system of the chaser satellite is on hold. After capturing, the chaser and target's tandem turn stiff as a single unit (Wormnes, 2013).

Net Capturing: The net capture mechanism has four weights attached at the corner points, shot by a net gun (Shan et al, 2016). The weights, also known as bullets, expand the net, and wrap up the target (Bischof et al., 2004). Shan et al. (2016) analyze that the technique provides a sufficient distance between the chaser and target such that it avoids rendezvous and docking.

Harpoon mechanism: Barbs on the tip of a harpoon mechanism are shot from a chaser satellite to penetrate the target space debris. Thus, large space debris can be pulled by a chaser satellite to re-enter the earth's atmosphere. Using it, the risk of creating new debris is high and cannot capture targets with a high tumbling rate (Shan et al., 2016).

2.2.2. Contactless Removal

Concepts that involve no direct contact with the target and attempts to reduce the velocity and altitude by momentum transfer (Shan et al., 2016). The contactless removal techniques are as follows:

Artificial atmosphere influence: To decelerate the debris, atmospheric particles such as gaseous plume or vortex are propelled in the path of it (Gregory, 2014). Kofford and Dunn (2011) proposed an artificial atmosphere delivery system that can create a transient gaseous cloud with sufficient density to deorbit space debris.

Ion Beam Shepherd: The concept involves targeting a highly collimated plasma beam on the space debris particle. Bombardelli (2011) developed an optimized system of a shepherd. It is equipped with a secondary propulsion system to overcome the reaction force (Shan et al., 2016). The momentum transfer achieved by ion beams is higher than from material ablation.

Laser System: Pulsed laser beam systems are available to shoot onto large and small space debris to decrease its velocity and altitude (Shan et al., 2016). Philips et al. (1996) proposed a system called ORION to remove all objects less than 1 cm size and 500 kg mass below 1000 km altitude in the timeframe of 4 years. The basic principle and scientific concepts involved in laser-based removal are presented in the next section.

2.2.3. Laser based space debris removal

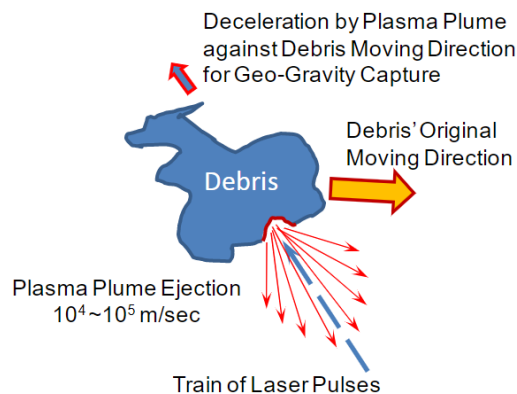


Figure 2.2: Photoablation on the surface of target space debris (Choi and Pappa, 2011)

Laser-based removal is a promising concept which uses optical methods and laser physics (Eckel et al., 2016). High-powered lasers are beamed on space debris particles either from ground or space resulting in a fraction of surface material ablation. Thus, providing an impulse to the target particle. Impulse vector produced deflects trajectory of the target such that it de-orbits. Phipps et al. (2012) mention that laser-based methods can be categorized based on their aims and technical parameters as follows:

Lower laser intensity: Laser's with intensity below a threshold value is used to divert debris using photon momentum transfer. Momentum transfer efficiency is less, and it is highly sensitive to space weather alteration (Mason et al., 2011).

Higher laser intensity: Continuous lasers can be used for the slow heating to ablation. In the case of tumbling debris, an ablation jet due to slow heating will cancel its momentum contribution as it creates messy melt ejection rather than a clean jet (Phipps et al.,2012).

Pulsed laser orbital debris removal (LODR): Pulsed ground-based lasers are used to change the orbit of debris. The target does not melt or get fragmented, but it vaporizes a few nanometer surfaces, and small debris can slow down by 10cm/s for each pulse (Phipps, 2012).

2.2.3.1. Space based laser system

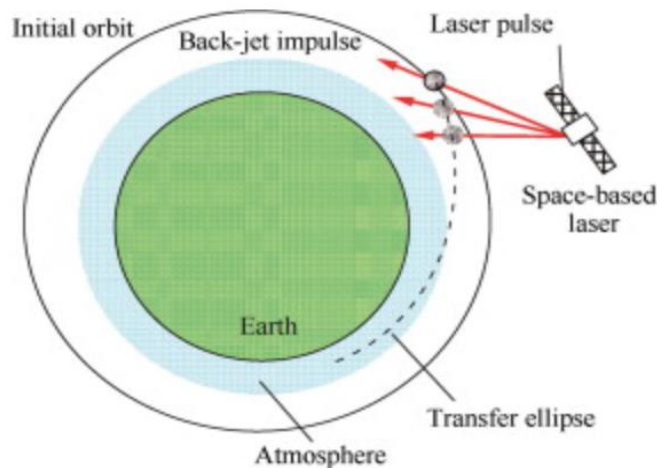


Figure 2.3: Removal of debris using space-based lasers (Shuangyan et al., 2014)

Shuangyan (2014) mentions that laser satellites can eliminate debris by ablation and ablation back-jet mode. Ablation mode aims at removal of small sized debris using laser beam energy is to burn the target. Ablation back-jet mode targets large space debris, and laser energy results in alteration of the target's orbit. The beamed laser energy is converted into thermal during the contact with the target, and the irradiated region's temperature rises to the breakdown point

of the target. During ablation, debris will experience a reaction force that will change its trajectory and reduce the perigee altitude after beaming multiple laser pulses. Finally, the target will burn by the aerodynamic heating in the higher atmospheric region (Avdeev, 2011). For the application, lasers must have high power, pulse energy, beam quality and technical maturity (Shuangyan et al., 2014). Solid-state laser with 1.06μm wavelength operating in burst mode is a potential candidate for a space-based laser system. It has no mechanical driven component, and no refueling is required (Shuangyan et al., 2014).

2.2.3.2. Ground based laser system

The European consortium has investigated and validated the concept of a ground-based laser system under the CLEANSPACE project framework (Esmiller et al., 2014). It uses a closely directed laser energy beam from the ground station to alter space debris' trajectory in low earth orbit (Eckel et al., 2016). An incident laser pulse produces a mechanical impulse because of the pressure created by photoablation on the surface, as shown in figure 2.4.

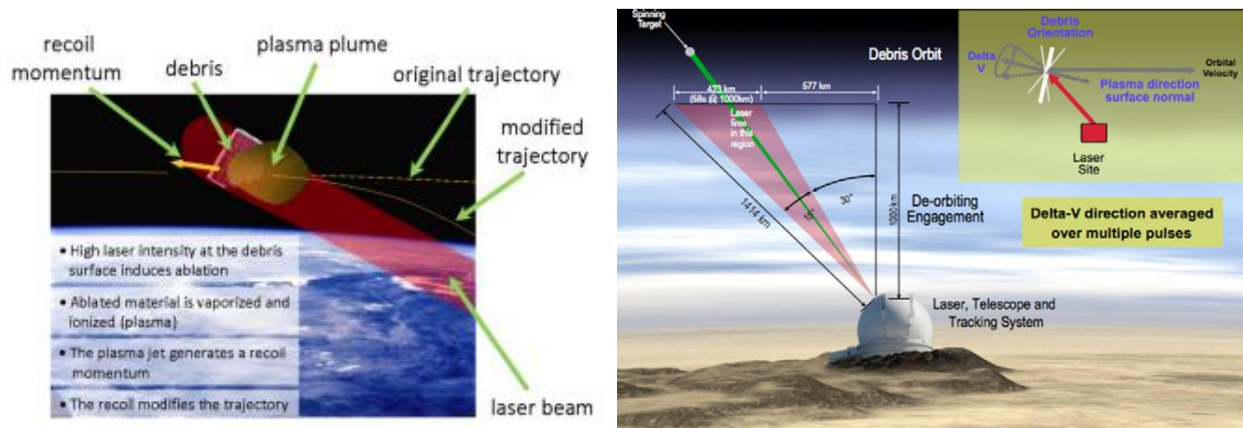


Figure 2.4: Ground-based laser removal method(credits: DLR)

The concept of LDR is based upon the laser-matter interaction process. The impulse coupling, which is the ratio of exerted thrust to the average laser power, is the prominent figure of merit (Eckel et al., 2016). It can be calculated by equation (Phipps et al., 2012):

$$C_m = \frac{p}{I} = p \left(\frac{\tau}{\Phi} \right) \quad (1)$$

where p is ablation pressure, I is the intensity, τ is the laser pulse duration and Φ is the laser fluence (Phipps et al., 2012). From equation 1, it is seen that Cm depends upon the laser pulse fluence. Cm is inversely proportional to intensity. Initially, it rises to the maximum with the intensity increase, and then it decreases at high-intensity values. The maximum value of momentum coupling occurs at vapor-plasma transition (Phipps et al., 2012)

2.3. Laser-matter interaction experiment

2.3.1 Experimental setup

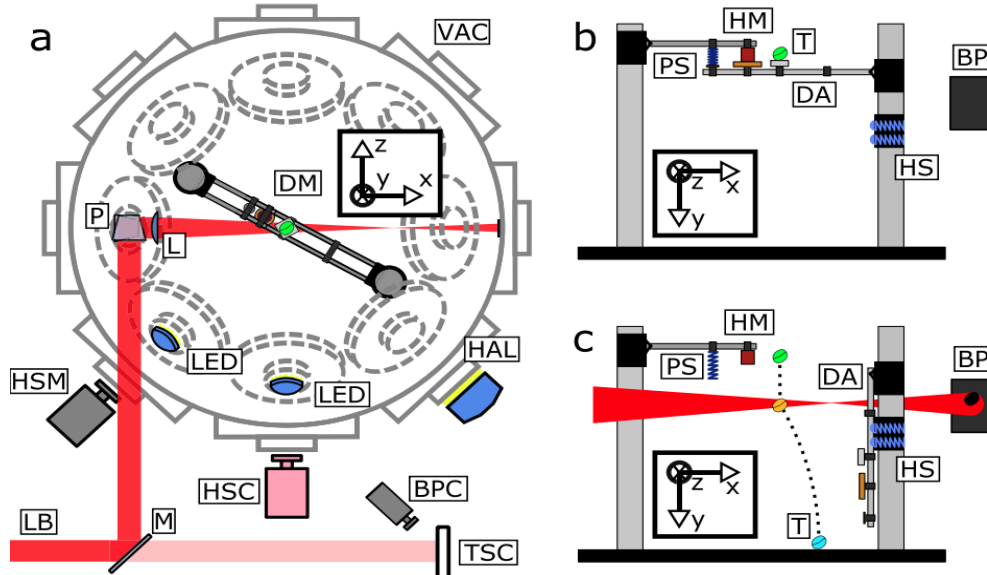


Figure 2.5: Experimental setup. (a) Top view of the vacuum chamber. (b) Sample dropping mechanism setup and (c) Sample trajectory during the laser-irradiation (Lorbeer et al., 2018)

To achieve microgravity, the target was dropped from 0.5 meter of height and was illuminated by high energy laser pulse after approximately 150ms of free fall (Lorbeer et al., 2018). The experiment used NHELIX laser system (Schaumann et al, 2005) with ns pulsed laser head generating ten seed pulses per second. The laser beam was directed to a vacuum chamber using an anti-reflected coated window and redirected to the center of the chamber using highly reflecting mirrors. Beam diameter was reduced using an AR coated lens and the remaining beam was dumped on anodized aluminum foil (Lorbeer et al., 2018). A dropping arm as shown in figure 2.5 (b) was constructed to release the debris sample. Target moment was captured using a pair of high-speed cameras MotionPro X3 and MotionScope M3 which were placed between incoming laser and at a 45° angle to each other (Lorbeer et al., 2018).

2.3.2. Observations

Lorbeer et al. (2018) mention that stereoscopic imaging was used to evaluate kinetic effects and about 27 tests were conducted exhibiting the generation of impulse. Motion capture from high-speed cameras shows that all targets were deflected from their original trajectory. Parameters such as surrounding pressure and variation of material show a significant difference during the ablation process. With an increase in surrounding pressure, the ablation process

expands its duration from one frame at 0.003 mbar to up to two frames at 0.1; 0.2 mbar (Lorbeer et al., 2018). Amongst other materials, a sample formed with a transparent biopolymer, also known as gummy bear, shows high intensities saturation of camera pictures front and back. Other significant observations are as follows (Lorbeer et al., 2018):

1. Lighter objects show high velocities and dominant acceleration in the direction of laser radiation.
2. Velocities of objects are strongly influenced by their weights.
3. Orientation of L-corner targets influences the amount of generated impulse.
4. Rotation rates achieved were higher for lighter materials such as copper and aluminum than heavy materials.

The trajectory obtained after the experiment was plotted manually using ImageJ software. Computational techniques such as software for automatic object detection and motion tracking can ease the task. The obtained 3-dimensional trajectory can validate the deflection in the trajectory observed during the experiment. It can also be used to calculate several parameters of interest, such as linear and rotational velocity, the center of gravity, moment, etc. The next chapter briefs about the fundamentals of computer vision algorithm, which are implemented for the application.

Chapter 3: Fundamentals of Computer Vision

3.1 Background

In the late 1960s, computer vision was initially started as a summer project at MIT to mimic human vision in robots to exhibit intelligent behavior (Papert, 1966). Since then, it has expanded its application into multidisciplinary fields, performing tasks ranging from data recording to information interpretation.

Computer vision is a building algorithm that allows a computer system to extract, analyze and understand the vital information from an input, which can be in the form of a single image, a sequence of images, or a video (Wiley and Lucas, 2018). As an analogy to the human visual system, there is a light source that illuminates the objects in the scene, a camera that captures the scene and a computer vision algorithm that acts as a brain, resulting in the identification and segmentation of objects in the scene.

Typical tasks associated with computer vision are as follows:

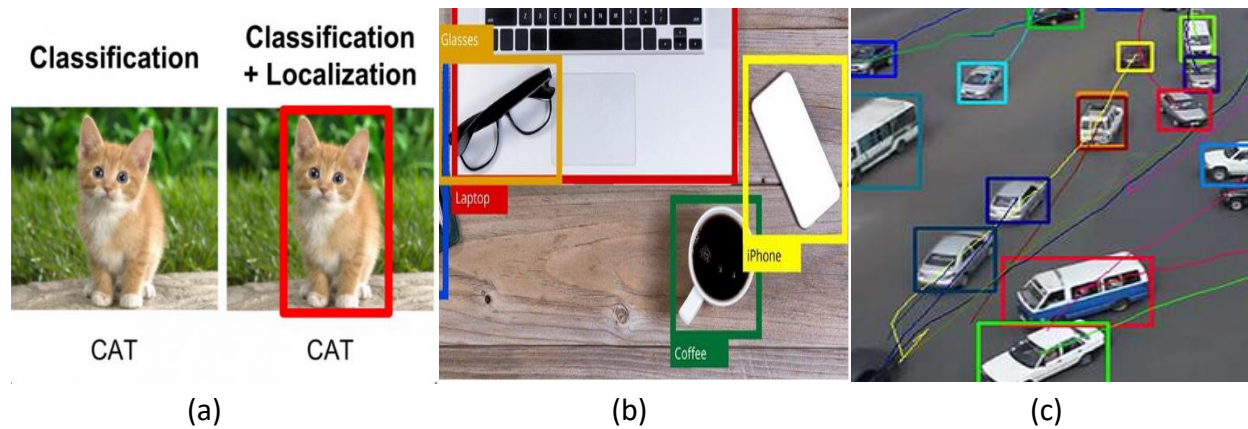


Figure 3.1: a) Single object classification and localization b) Multiple object detection in a scene (Remanan, 2019) c) Tracking multiple objects in a scene

1. Image classification and localization: Ability to detect and locate a specific object using pre-defined classifiers in the image frame. Classifiers are generally a set of data that describes the characteristics of an object.

2. Object detection: Ability to identify multiple objects in a scene and determine their location based on the types or classes of the objects.

3. Motion analysis: Tracking specific/multiple objects through the sequence of images or a video by estimating the target's current position based on the information obtained from the previous scene.

Computer vision has a broad spectrum of application, mainly in the field of:

1. Medicine: Diagnosing a patient through medical image processing.

2. Autonomous vehicle: Automatic driving control, navigation, surveillance through unmanned aerial vehicles in case of disasters

3. Defense and security: Surveillance through security cameras, detection of enemy soldiers, and guided missiles or weapons.

4. Agriculture: grain production, monitoring soil quality and insects/pests on crops sorting data of fruits, vegetables, and flowers based on their size, texture, quality etc.

5. Space: Space robotics for ISS assembly and servicing, on-orbit servicing and manufacturing, active debris removal, planetary exploration, and mining.

3.2 Image Preprocessing

A digital image is a two-dimensional data matrix or an array with a finite set of values known as elements or pixels. Each discrete value in the matrix has an associated integer, which represents the brightness value of it. This brightness value is known as a grayscale or gray level concept.

3.2.1. Gray Level concept

An image is formed on the sensor as a 2D function suppose $f(x,y)$ where x and y are the spatial coordinates. The amplitude of the function estimates the intensity or the grey level at (x,y) at that point in the image. For a basic, 8-bit image, the pixels' value or amplitude is between 0 to 255. Pixel with a 0 intensity value is black and a 255 is white. For a true-color image, also known as an RGB image, each pixel represents a combination of three values corresponding to red, blue, and green intensity components. But for image processing algorithms, dealing with a true-color image is a bit more complicated than a grayscale image due to factors such as signal to noise ratio or difficulty of visualization of the features or edges in the image. Thus, in an RGB image, a 3D array is converted into a grayscale 2D array by averaging red, green, and blue pixel values to get a corresponding value in the grayscale (Saravanan, 2010).

3.2.2. Image Filtering

In most of the images, there is some unknown variation in the data that is of no interest and is regarded as noise. For a computer vision algorithm, processing becomes more straightforward if the noise is filtered out from the input image. Apart from the corrupted image data, the noise can result from instrumentation error such as variation in sensitivity of the detector, data-transmission errors or image-quantization errors etc.

Noise in the image as a function:

$$\vec{f}'(x,y) = \underbrace{\vec{f}}_{image}(x,y) + \underbrace{\vec{\eta}}_{noise}(x,y) \quad (2)$$



Figure 3.2: Different types of noises (credits: S. Seitz)

Common types of noises (Jain et al., 1995):

1. Salt and pepper noise: Random occurring black and white pixels
2. Impulse noise: Random occurring white pixels
3. Gaussian noise: Change in image intensity as a result of Gaussian distribution function

Thus, applying image filters can provide an improved visual representation of the image data. A kernel or a mask is a small matrix applied to the image for the filtering process. Mathematically, it can be integrated into the image function by two methods:

Correlation Filtering: Correlation is the measure of similarity between the two pixels and is given by equation 3:

$$\text{Image (f) x masks/kernel (h)} = \sum_k \sum_l f(k,l)h(i+k,j+l) \quad (3)$$

Convolution Filtering: Convolution is the measurement of the difference created by masking the image such that the derivative of which results in detection of edges and is given by equation 4:

$$f * h = \sum_k \sum_l f(k, l)h(i - k, j - l) \quad (4)$$

3.2.2.1. Gaussian Filter



Figure 3.3: a) Original image, b) Image after Gaussian filtering (Guerra, 2014)

The most common filter used in image preprocessing is the Gaussian Filter. Filter blurs the image and removes the noise using a kernel-based on 2-Dimensional gaussian function given by:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (5)$$

The gaussian filter works by using convolution of the 2-Dimensional gaussian distribution function with the input. The properties of gaussian filter are (Jain et al., 1995):

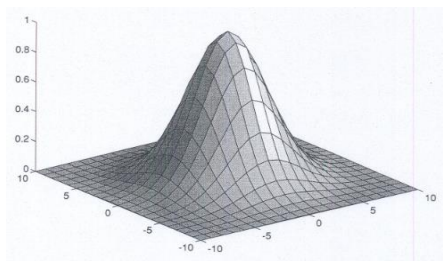


Figure 3.4: Gaussian lobe (Jain et al., 1995)

1. The smoothing magnitude is equivalent in all directions
2. A single lobe function, replaces each image pixel with a weighted average of neighboring pixels with a substantial decrease in weight with a distance from center pixel. Thus, it does not disturb the local features in an image.
3. Standard deviation of the function determines the extent of smoothing and width of gaussian filter.

3.3. Object and Feature detection

Feature detection is the method to search for relevant information by analyzing every pixel throughout the image. A feature is typically known as an interesting part of an image, and its extraction is also a primary task in object detection. Hassaballah et al. (2016) mention that image features can be broadly classified into global and local features. Global features such as color and texture are single feature vectors which generally represent the content of the entire image (Hassaballah et al., 2016). On the other hand, local features are a set of multiple features points such as shape, color, orientation and texture etc. representing small patches of an image that can be used to determine that image's information (Zhang, 2014). For feature extraction, the first step is to detect the set of interest regions or keypoints and then, for each keypoint, determine an invariant feature vector (descriptor) for image data. It has been found that local descriptors are relatively fast to detect and compute exhibiting good accuracy and less memory consumption. For applications where image matching or match moving, camera calibration, and object tracking are involved, local feature vectors are more robust to scale variance such as brightness, viewpoint, distortions etc.

3.3.1. Types of features

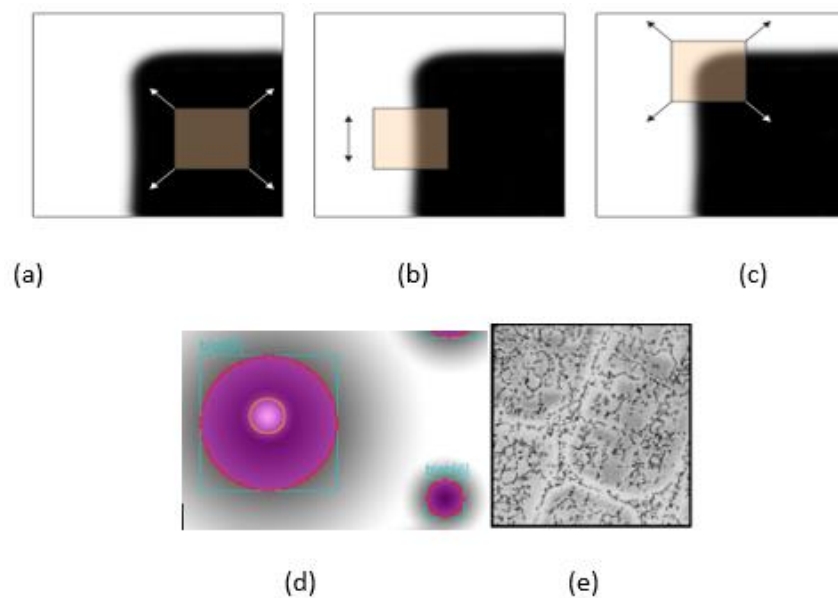


Figure 3.5: a) Flat region b) Edge c) Corner (credits: A. Efros) d) Blob e) Ridge (Lindberg, 1998)

Edge: A significant local change in the image intensity typically occurs at the boundary of two distinct regions of an image, as shown in figure 3.5 b). Edge is also referred to as the discontinuity in the first derivative of image intensity.

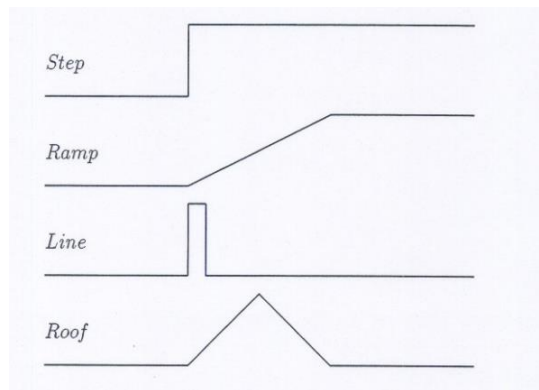


Figure 3.6: Image intensity variation for edge detection (Jain et al., 1995)

Generally, four types of edges, which are step, ramp, line, and roof, are detected in an image. Step and Line are rare occurrence scenarios for the variation in image intensities as there is an abrupt change in value from one phase to the other within the same region. After image preprocessing, sharp discontinuities in image intensity are observed quite rarely. Thus, step edges often become ramp edges due to gradual intensity change, and line edges become roof edges. For detecting an object with a rounded surface corner, a combination of both step and line profile is also possible (Jain et al., 1995).

Corner: Corners are mainly the intersection of two edges and can be determined by the variation in image intensity of a window around a specified pixel in all directions.

Blob: A blob is a region or pixel group that exhibits common properties such as image intensity or grayscale value. Also, they can deliver essential hypotheses on the existence of a particular object based on its size.

Ridge: In addition to the blob descriptor, the ridge feature determines the object's axis of symmetry. For Example, in topography, a ridge acts as a separator between the water channels depending upon flow direction (Lindeberg, 1998).

Properties of interest points detector:

1. Detect all true interest points and no false points.
2. Well localized.
3. Robust with respect to noise and provides efficient detection.

3.3.2. Edge Detection

Methods adopted for edge detection are broadly classified as (Ahmed, 2005):

1. **Gradient based edge detection:** The method detects maxima and minima points in the first derivative of the image. A pixel is considered as an edge if the gradient value exceeds a certain threshold value.
2. **Laplacian based edge detection:** Obtaining second derivative at maxima results in a zero value. It identifies zero crossings in the second derivative image to identify the location of an edge.

3.3.2.1. Steps for edge detection

The process of edge detection in an image is as follows (Ahmed, 2005):

1. **Smoothing:** To reduce the noise and increase the performance of the edge detector.
2. **Enhancement:** Improving digital quality and visualization of edges in an image.
3. **Detection:** The edge points with the finest quality are selected and the one with noisy pixels are discarded.
4. **Localization:** Confirming the location of the detected edge.

3.3.2.2. Edge Detection Techniques

Sobel Edge Detection: Sobel operator performs a two - dimensional gradient measure as it transfers a 2D pixel array to uncorrelated data to eliminate the repetition of the data set. A gradient is a vector whose components measure the change in the pixel value with respect to the distance in both x and y directions. The vectors are obtained using the equation (Alazzawi et al., 2015):

$$\frac{\partial f(x,y)}{\partial x} = \Delta x = \frac{f(x+dx,y)-f(x,y)}{dx} \quad (6)$$

$$\frac{\partial f(x,y)}{\partial y} = \Delta y = \frac{f(x,y+dy)-f(x,y)}{dy} \quad (7)$$

Where dx and dy measures distance along the respective directions. The presence of gradient discontinuity can be calculated by the measure of magnitude and the gradient direction (θ) from the following equation:

$$\theta = \tan^{-1} \frac{\Delta y}{\Delta x} \quad (8)$$

An approximation of the intensity gradient function is computed by the Sobel operator. To reduce the effects of noise and to have a better estimate based on the central pixel, a larger mask of odd size is applied. An example of the mask used in the Sobel edge operator is as follows:

$$\Delta x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad \Delta y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

The gradient of image intensity is calculated by the operator at each point based on the largest magnitude increase in light conditions and considering the rate of change in a particular direction. The obtained result exhibits the likelihood of finding an edge in the particular region of the image along with its orientation. At the edge point, the Sobel operator will show a gradient vector whose magnitude corresponds to the rate of change and points towards the largest possible intensity increase direction.

Prewitt Edge Detection: Prewitt operator is similar to the Sobel but with the different masking conditions. In the case of Prewitt, the derivatives must have the following properties (Adlakha et al., 2016):

1. Values in a mask must contain opposite signs.
2. Sum of the mask must be zero
3. Detects more edges due to more weight as compared to the Sobel mask.

1	1	1
0	0	0
-1	-1	-1

-1	0	1
-1	0	1
-1	0	1

These masks are convolved together with the original value. It results in two different pixel values P1 and P2 corresponding to both the row and column mask. These values can be used to determine the edge magnitude and orientation by using the following equations (Adlakha et al., 2016):

$$\text{Edge magnitude} = \sqrt{P1^2 + P2^2} \tag{9}$$

$$\text{Edge direction} = \tan^{-1} \frac{P1}{P2} \tag{10}$$

Laplacian or Zero-crossing Edge Detector: Laplacian is used to determine the 2D magnitude of the second derivative of an image. It highlights the zone where there is a rapid intensity variation to search for edges marked by the zero-crossing. Laplacian is applied to an image after reducing noise by smoothing using a gaussian filter. The operator takes a grayscale image and converts it into a binary image (Alazzawi et al., 2015).

Zero crossing detector searches for the point where the image intensity is zero, or the Laplacian changes its sign. The Laplacian derivative operator can be obtained from equation 11 (Alazzawi et al., 2015):

$$\Delta^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (11)$$

For X-direction: $\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$

For Y-direction: $\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$

Canny edge detector: Among other edge detection algorithms, canny is excellent and reliable mainly due to its simplicity for implementation and satisfies the three criteria to be an optimal edge detector as follows (Nisha, 2015):

1. Good detection – High real edge detection success rate.
2. Good localization - Marked edges lies in the vicinity of the original edges in an image.
3. Minimal response - No false edges were created due to noise.

Algorithm for canny edge detector (Nosrati, 2013):

1. **Smoothing** – An application of a Gaussian smooths the image to reduce the camera's noise and the detection of the false point.
2. **Compute intensity gradients** - The gradient magnitude and direction are calculated using finite-difference approximations. Sobel operator is used to determining the gradient intensity of each pixel. Masks implemented for Sobel operation approximates the gradient in both x and y directions.
3. **Non-maximum suppression** – Non-Maxima suppression is applied to the gradient magnitude. This is done to convert the blurred image edges into sharpened edges. The algorithm to perform this suppression is as follows:
 1. Rotate gradient orientation by the angle of 45 degree.
 2. Determine and compare the pixel's edge intensity in each possible gradient orientation.
 3. Keeping the pixels with maximum edge intensity value and eliminating the rest. The pixels with the maximum value will then be marked white in color (Nisha, 2015).

1. **Double thresholding:** To eliminate the possibility of false detections, the canny algorithm also implements double thresholding by detecting robust edge pixels with power more than a high threshold limit. Pixels value ranging out from threshold are discarded, and the one in between the high and low threshold is not considered weak edges.

2. Hysteresis Edge Tracking: The group of weak edges marked after double thresholding can be either because of noise/color variation or real edge. Thus, it separates the weak pixels into two parts to obtain accurate results and reconsider the weak pixels due to the real edge.

3.3.3. Corner detection

Harris Corner Detector

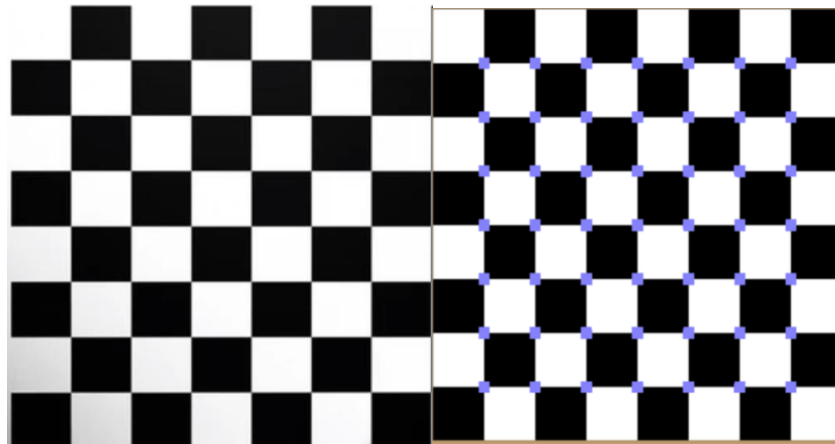


Figure 3.7: a) Original chessboard image b) Chessboard corners detected as blue dots (Madhav, 2019)

Harris detector is a standard technique for detecting interest points, especially corners in an image (Harris and Stephens, 1988). It is based on Moravec's detector (Moravec, 1980), which determines the autocorrelation function to measure the intensity variation between a patch or a window shift in all possible directions. The change in intensity for a small shift $[u,v]$ can be determined by equation 12 (Le et al., 2011):

$$E(u,v) = \sum_{x,y} w(x,y) [I(x+u, y+v) - I(x,y)]^2 \quad (12)$$

where I is the image scalar function, (x,y) are the point coordinates, $w(x,y)$ defines the window and is typically represented by a rectangle or gaussian function. If there is a small shift $[u,v]$ then bilinear approximation is applied to the intensity function as follows (Le et al., 2011):

$$E(u,v) = [u,v] M \begin{bmatrix} u \\ v \end{bmatrix} \quad (13)$$

where M is a 2x2 matrix and is calculated from image derivatives using equation 14 (Le et al., 2011):

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (14)$$

The two eigenvalues obtained from matrix M are $\lambda_1 \lambda_2$ and these are used to determine the measure of corner response by using equation (Le et al., 2011):

$$R = \det M - k (\text{trace } M)^2 \quad (15)$$

where $\det M = \lambda_1 \lambda_2$, $\text{trace } M = \lambda_1 + \lambda_2$ and the value of k (empirical constant) is between 0.04 - 0.06.

Interpretation of region from degree of cornerness(R):

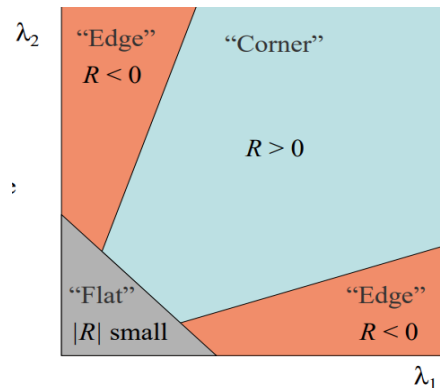


Figure 3.8: Harris measure (Sanchez et al., 2018)

1. It is an edge, if the value of R is large and negative
2. It is a corner, if R is large and positive
3. It is a flat region, if the magnitude of R is small.

3.3.4. Feature Extraction

Scale Invariant Feature Transform: To identify an object in the image or video sequence, the process is to deduce the object's properties and then match it with the following sequence of images. SIFT or scale-invariant feature transform is an image descriptor algorithm which localizes and characterizes properties of an image. It offers a precise object recognition algorithm with the lowest possibility of mismatching. As the name suggests, the features extracted using this algorithm are invariant to scaling and rotation, change in light conditions, 3-dimensional change in viewpoint and noise. Identified properties include information about (Gbash and Saleh, 2017):

1. Pixel location in an image.

2. Scale, which is being characterized by the standard deviation.
3. Orientation of the image structure.
4. Image's local structure characterized as per the gradient histograms.



Figure 3.9: SIFT feature detection (Lowe, 2004)

Computational steps for application of the SIFT algorithm are as follows:

1. Detect local extrema of the scale-space: A Gaussian kernel/ mask of a certain standard deviation value is applied to the input image by iteratively using convolution to obtain a pyramid of the difference of Gaussians (DOG). The position of the property is determined by the local extrema of the DOG pyramid. The final convolving image undergoes downsampling, which means scale reduction by factor of 2 until the point after which no down sampling is possible. All octaves obtain form the Gaussian pyramid which is a 3D function $L(x,y)$. DOG function is calculated by the difference of two neighboring images in the gaussian pyramid. The comparison of each pixel finds the value of the local extrema of DOG with its adjacent pixels in the scale space (same scale, above scale and below scale) (Gbash and Saleh, 2017).
2. Localization of key points: Most of the detected local extremas are the potential key points and must be precisely localized. Thus, a three-dimensional quadratic function is applied to the local sampling point. Low contrast and high noise sensitivity are the parameters to separate the weak key points corresponding to the edges (Hoon et al., 2016).
3. Assigning the orientation: As per the local image gradients at each point, a direction is assigned to the feature (Shweta et al., 2015). The gradient magnitude and direction of each pixel surrounding the property position can be calculated by the following equation (Gbash and Saleh, 2017):

$$M(X,Y) = \sqrt{(L(X + 1, Y, \partial))x^2 + (L(X, Y + 1, \partial)) - (L(X, Y - 1, \partial))x^2} \quad (16)$$

$$\theta(X,Y) = \arctan((l(x,y+1, \partial) - l(x,y-1, \partial)) / (l(x-1,y, \partial) - l(x+1,y, \partial))) \quad (17)$$

Gradient magnitude is weighted by using a gaussian window and it is used to create a direction histogram that covers 360° direction range. The amplitude of more than 80% of the histogram's highest peak is used to create a key point with a particular direction (Gbash and Saleh, 2017).

4. Descriptor of the keypoint: The area surrounding a key point is divided into 4 x 4 boxes, as shown in the figure below. Weighted gradient magnitude and direction are calculated for each box and gradient direction undergoes rotation based on the key points directions.

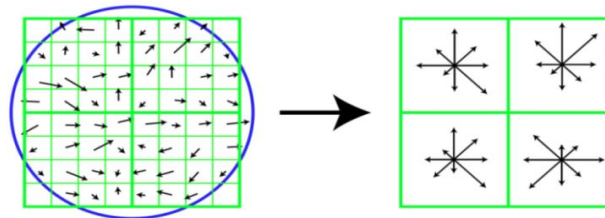


Figure 3.10: Computation of keypoint descriptor (Sunhyoung and Nuvo, 2011)

A histogram of 8 is created representing a box and from 16 obtained histograms, a 128D vector or a SIFT descriptor is created. The descriptor is then normalized to unit length and is along the main direction to be scale-invariant (Sunhyoung and Nuvo, 2011).

3.3.5. Feature matching:

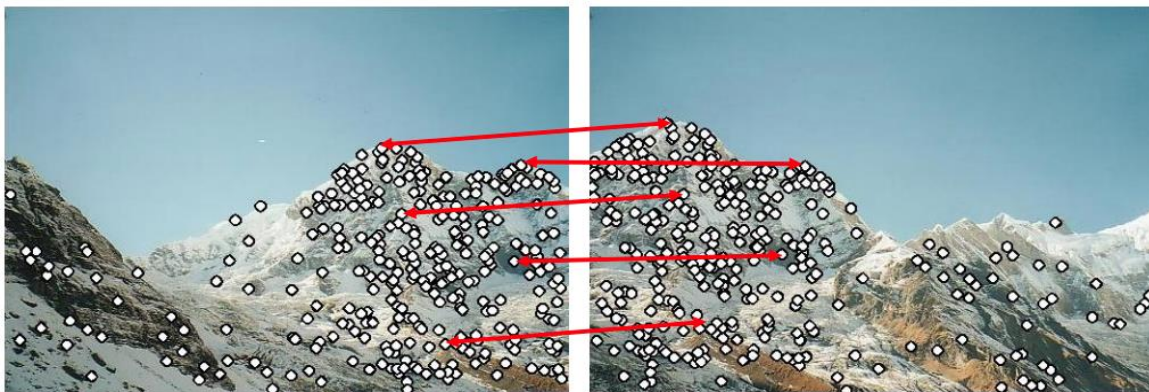


Figure 3.11: Feature matching between two images of same scene (credits: M. Brown and D.G. Lowe)

For identifying objects in a sequence of images or a video, feature matching is a method adopted by the computer vision algorithm. As explained above, features in any given image can be identified using detection techniques such as SIFT and Harris. The SIFT algorithm's application to an image will generate a description of the interest point, such as its location, orientation, scale, illumination, etc. These properties will be stored in a descriptor, a data set for the feature detection in the subsequent images. Algorithms such as Brute-Force matcher or Fast Library for Approximate Nearest Neighbors (FLANN) can be used for matching purposes (Jakubovic and Velagic, 2018). Brute force matcher calculates the Euclidean distance, which is a straight-line distance between two keypoints in different images and is more robust. Thus, as shown in figure 3.11, it creates a straight line between the interest points with the same descriptors.

3.4. Object Tracking

Tracking an object in a video means to find the position or the motion patch of the object in every subsequent frame. Even though trackers for surveillance and facial recognitions exist, object tracking is considered a challenging problem for tracking generic objects due to variation in scale, light intensity, or noise. Most of the motion tracking algorithms assume that the object being tracked follows a smooth path, and there is no change in its velocity and acceleration. In a scene with multiple objects, amongst all the detected interest points in an image, the critical part is to select the features which will make the object distinguishable. It is a manual-based feature selection. Some of the standard feature selection methods are as follows (Agren, 2017):

1. EDGES: It is known that the edges of an object can easily distinguish from the background as they produce a high image intensity variation. Thus, object tracking algorithms can track its boundaries by creating a bounding box around an object or selecting its edges.
2. OPTICAL FLOW: Apparent motion of the brightness pattern is calculated in the visual input scene by identifying each pixel movement between the frames. The brightness of the selected pixel is used as a reference characteristic with an assumption that the brightness will remain invariant throughout the entire video.

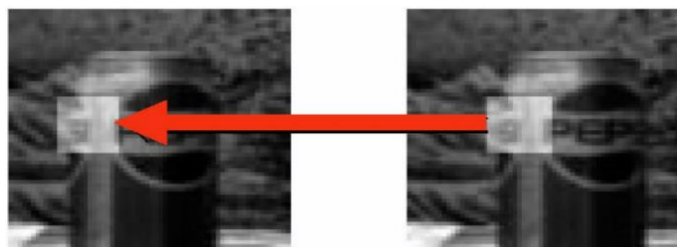


Figure 3.12: Locating pixel using optical-flow method (Sun and Srdjan, 2016)

3. COLOR: The color in an image is a combination of RGB (red, green, blue) but sometimes they are also represented by color spaces such as HSV. The drawback in using color is that it is sensitive to the change in light conditions and sometimes it appears to be different from the original (Yilmaz et al., 2006).

Object tracking algorithms can be broadly categorized into three groups, which are point based, kernel-based, and silhouette-based tracking, as shown in figure 3.13 (Agren, 2017).

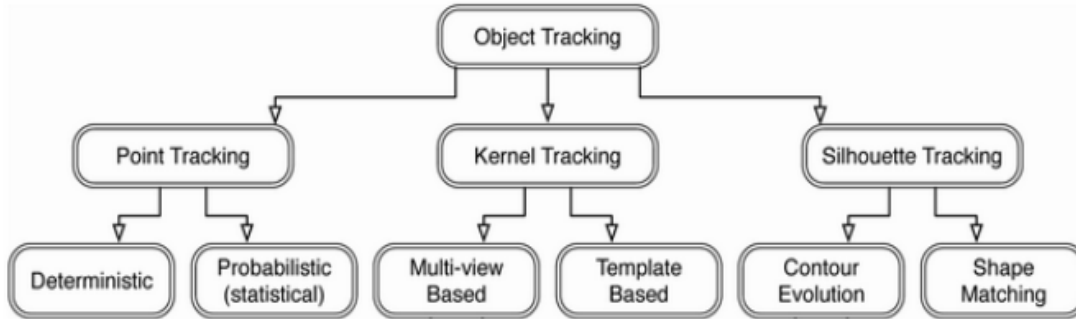


Figure 3.13: Categorization of object tracking (Agren, 2017)

3.4.1. Point based tracking

A point-based tracking approach is adopted when an object is represented as a point (Agren, 2017). It evaluates the position and motion of the object. It searches for associating points throughout the sequence of frames (Agren, 2017). Association between the two points is conducted based on the state of the object in the previous frame. This association process or matching each object point is referred to as the corresponding cost and must be minimized. The minimum corresponding cost can be achieved using deterministic and statistical approach. Several constraints can be used in different combinations to minimize the cost (Yilmaz et al., 2006). The algorithm assumes that the object's location and velocity do not drastically change between the frames. Also, the object is assumed to be rigid in nature. In the case of a deterministic method, no randomness in the object's state is expected, which means that it is supposed to follow some sample trajectory. Whereas the statistical approach considers uncertainty factors such as noise and random perturbations, making it more robust (Yilmaz et al., 2006).

3.4.2 Kernel based tracking

The kernel-based tracking approach is based on the object representation as a primitive region enclosing the object. Kernel refers to the shape and appearance of the object (Agren, 2017). To represent an object, shapes such as rectangle or ellipse are used as a template (Patil and Bhagat, 2015).



Figure 3.14: Kernel based tracking (red bounding box) (Mahadevan and Vasconcelos, 2017)

Kernel tracking is further divided into multi-view and template-based appearance models (Agren, 2017). Template matching is the most common approach while tracking single objects. Features such as image intensity or color or an object in the previous frame can be used as its template (Yilmaz et al., 2006). Also, certain algorithms allow users to select the target object using a bounding box and track them using fast and robust kernelized correlation filters (KCF) (Henriques et al., 2014). When the object is subjected to a drastic change in appearance or motion between two frames, there is a high probability that the tracker loses the object. Thus, a multi-view based algorithm uses a subspace-based approach in which an eigenspace computes the target object's transformation and reconstructs its image using eigenvectors (Agren, 2017).

3.4.3. Silhouette based tracking

Representing objects with simple geometric shapes can sometimes be an inadequate description for more complex objects such as a human hand. Using a silhouette tracking method, it is possible to create an accurate shape description (Rosenhahn, 2005). It is also known as a region-based tracking approach. Object models such as color histograms, object edges, or contours are created in the previous frames, then matched with the current frame models. This can be achieved by either using shape matching or contour evolution algorithm (Yilmaz et al., 2006). Shape matching is similar to template matching, where it looks for corresponding silhouettes in two consecutive frames. But it matches the entire region of the object rather than a feature keypoint (Veltkamp, 2001). On the other hand, contour tracking, also known as boundary tracking, uses the edge-based features to create a contour around the object along its edges. It is more robust as it is less sensitive to illumination change and is faster than shape matching (Jacob and Anitha, 2012).

3.5. Camera Calibration:

The pinhole model, often known as the classical model, is adopted for camera calibration. It assumes that all camera rays pass through a single point called an optical center. Pinhole perspective projection is the simplest model, and it creates an inverted image on the image plane, as shown in figure 3.15. The film is considered to be the image or retina plane. Aperture is the pinhole or center of the camera. The focal length is the distance between the image plane and the pinhole (Hata and Savarese, 2017).

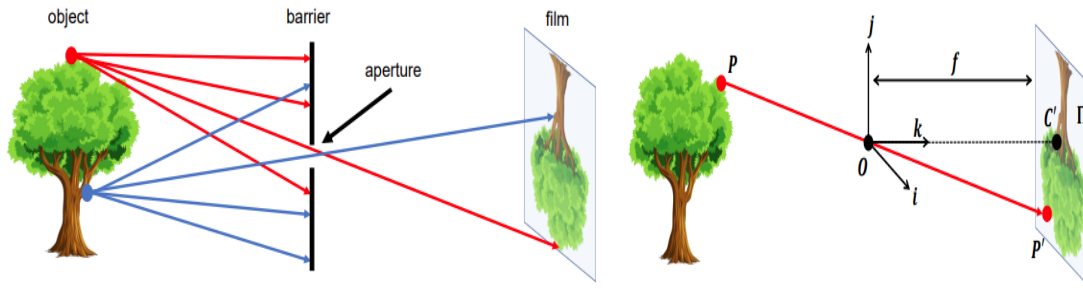


Figure 3.15: Pinhole camera model (Hata and Savarese, 2017)

A 3D point $P [x \ y \ z]^T$ is projected as the image plane as $P' [x' \ y']^T$ and the pinhole of the camera is projected as C' (Hata and Savarese, 2017). The coordinate system is defined as $[i, j, k]$ centered at the pinhole O such that the k -axis is perpendicular to the image plane. Using, the law of similar triangles, we can find the coordinates of the projected point P' as (Hata and Savarese, 2017):

$$f/z = x'/x = y'/y, \text{ which implies } x' = f(x/z); y' = f(y/z) \quad (18)$$

Mapping of a 3D point into a 2D point is often referred to as projective transformation. In case of digital images, images are divided into discrete pixels rather than being continuous as in the image plane. Thus, the sensors attached to digital imaging can create some distortion to the mapping. There is an offset between the 2D points in the image plane and 2D points in the image by a translation vector $[c_x, c_y]^T$. The mapping now becomes (Hata and Savarese, 2017):

$$P' = \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} f \frac{x}{z} + c_x \\ f \frac{y}{z} + c_y \end{pmatrix} \quad (19)$$

After introducing a homogeneous coordinate system: P becomes $(x, y, z, 1)$ and P' becomes $(x', y', 1)$. Thus, it can be written as (Hata and Savarese, 2017):

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} f & 0 & t_u \\ 0 & f & t_v \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad (20)$$

Accounting the resolution of the camera and modifying the above equation, assuming rectangular pixels with resolution m_x and m_y in x and y direction respectively. P' can be expressed as:

$$u = m_u \frac{fX}{Z} + m_u t_u \quad (21)$$

$$v = m_v \frac{fY}{Z} + m_v t_v \quad (22)$$

In the matrix form, this can be represented as follows:

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} m_u f & 0 & m_u t_u \\ 0 & m_v f & m_v t_v \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} \alpha_x & 0 & u_o \\ 0 & \alpha_y & v_o \\ 0 & 0 & 1 \end{pmatrix} P = KP \quad (23)$$

From the above question, K depends upon the properties such as focal length and principal axis of the camera which are intrinsic in nature and are known as intrinsic parameters of the camera. Sometimes the camera matrix K also includes parameters such as skewness and distortion. Camera coordinates are skewed only when the angle between the 2 axes is slightly greater or smaller than 90 degrees. Often some amount of skewness occurs due to sensor manufacturing errors, Thus the camera matrix K can be modified as (Hata and Savarese, 2017):

$$K = \begin{pmatrix} \alpha_x & s & u_o \\ 0 & \alpha_y & v_o \\ 0 & 0 & 1 \end{pmatrix} \quad (24)$$

where s is the skewness parameter.

Assuming that the center of projection of a camera is not at the origin and is located at some arbitrary position. There is a need to calculate the rotation and translation in order to make the camera's coordinate system coincide with the original axis configuration. Let the translation vector be given by (T_x, T_y, T_z) and the rotation applied is given by 3 X 3 matrix R. The matrix formed by application of first translation followed by rotation is (Hata and Savarese, 2017):

$$E = (R | RT) \quad (25)$$

and is known as an extrinsic parameter matrix.

Computer camera transformation can be expressed as $K (R \mid RT) = (KR \mid KRT) = KR (I \mid T)$. Thus, the projected point P' can be written as: $P' = KR (I \mid T) P = CP$, where C is the complete calibration matrix (Hata and Savarese, 2017). Thus, camera calibration estimates the intrinsic and extrinsic camera parameters based on given input images and camera properties. In order to reconstruct a 3D world model, good camera calibration is important (Hata and Savarese, 2017).

3.6. Triangulation

Triangulation or reconstruction is a method which is used to determine a 3D point in space by using its given projection on two or more images. For a 3D world point X , the projected point x can be calculated using equation: $x = P X$, P representing the camera matrix

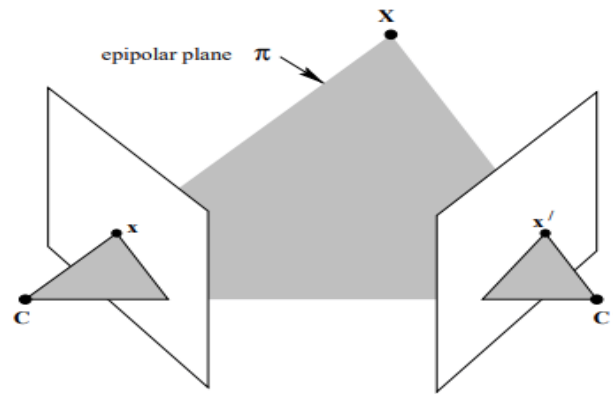


Figure 3.16: Point correspondence geometry: Two cameras C and C' , projection points x and x' and 3D world point X (Hartley and Zisserman, 2004)

For triangulation, x the image or projected point and the camera matrix P , obtained from camera calibration are known parameters. As shown in figure 3.16, to triangulate a function, two lines must intersect at a point. According to Chen et al. (2017), the simplest method to achieve multi-view triangulation is the linear or midpoint methods. At present Bundle Adjustment method is most commonly used (Chen et al., 2017). In the case of Multiview geometry, where a scene is captured from two different cameras, there will be two projection points (x and x') on the cameras' image plane. Hence, the triangulation equation can be used to obtain the intersection point X . In reality, due to noise, these two lines have a degree of error and never intersect (Chen et al., 2017).

Chapter 4: Computer Vision Software

This chapter describes the software implementation and the results of the object detection and motion tracking methods that are explained in the previous chapter.

4.1 Software Configuration

The software, tools, libraries and applications that were used in successful implementation of algorithms throughout the internship period are listed below:

- **Programming language:** Python 3.8
- **Integrated Development Environment:** Pycharm Community Edition 2020.2.2
- **Project Interpreter:** Anaconda
- **Libraries Installed:**
 1. Matplotlib
 2. Numpy
 3. Opencv-contrib-python
 4. Pandas

4.2 Feature detection and matching using python

To get into the computer vision software development, the first step was to code feature detection and matching algorithm. From the literature review, the Harris and SIFT detectors seem promising and are the most popular algorithms for feature detection. This section describes the algorithm used to implement both Harris Corner and SIFT detector on sample images.

4.2.1. Harris Corner Detection

For implementation of Harris Corner Detection:

Input: a) DLR logo and b) Random building scene.

Output: Corners marked

The steps followed by the python script to obtain corners in an image scene is shown in figure 4.1.

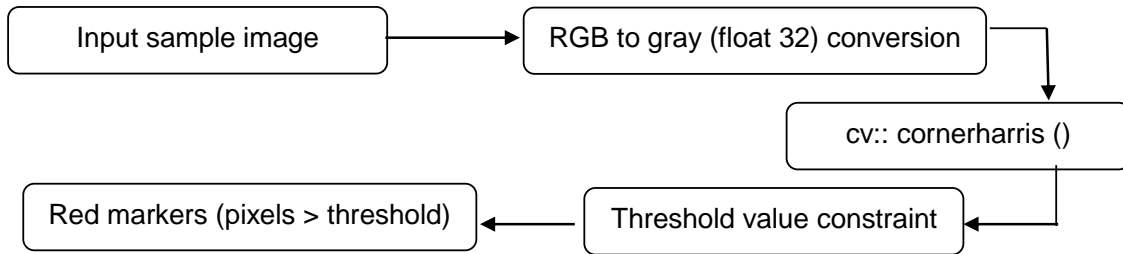


Figure 4.1: Algorithm for Harris Corner Detector

The threshold value constraint based on the harris detector response (dst) obtained using function `cv::cornerHarris()`. Red markers are implemented to the pixel with intensity value of:

$$x(\text{pixelvalue}) = \text{dst} > 0.01 * \text{dst.max()} \quad (26)$$

The output of the code, as shown in figure 4.2 shows that the corners are marked as red dots in the image. The Harris corner detector shows a more than 95% success rate in detecting the feature corner points.



Figure 4.2: Successful detection of corners (red dots) using Harris Corner detector

4.2.2. SIFT Detection

For implementation of SIFT Detector:

Input: a) Book & pen and b) Random building image

Output: Features marked in an image scene

The steps followed by the python script to obtain features using sift detector is shown in figure4.3.

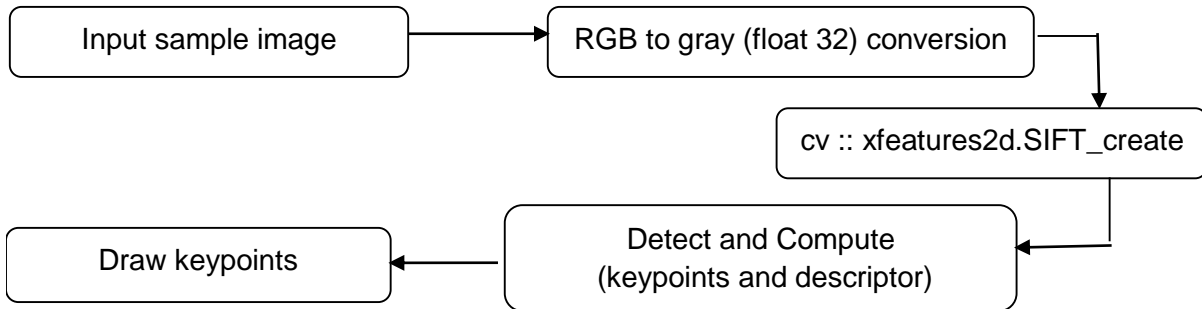


Figure 4.3: Algorithm for SIFT Detector

The results obtained from the code are shown in figure 4.4. Features mainly corners and edges of the book, pen, and the building's windows are marked with arbitrary colored circles. SIFT detection exhibits insensitivity towards the variation in properties, mainly scale and light conditions. Thus, it can detect more key points in a given image scene. In addition to the high detection rate of key points, it describes the given set of features using a unique descriptor that shows each key point's characteristics. This descriptor can be used to achieve feature matching in the sequence of images representing a single scene. The feature matching is shown in section 4.2.3. The disadvantage of using SIFT is its requirement of high computational time during the detection process (Le et al., 2011).

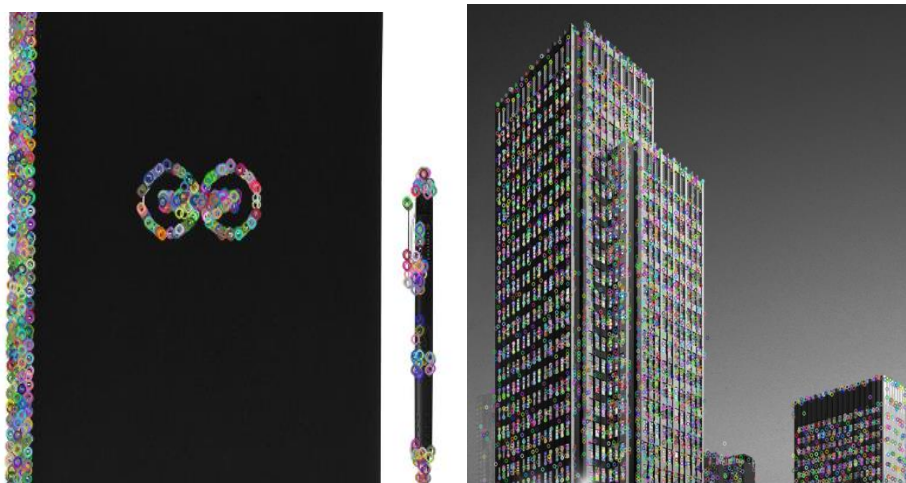


Figure 4.4: Feature detection using SIFT

4.2.3. Feature Matching

For implementation of Feature Matching:

Input: Two candle images captured in different orientation and light condition.

Output: Feature marking and matching

The steps followed by the python script for feature matching using sift detector is shown in figure 4.5

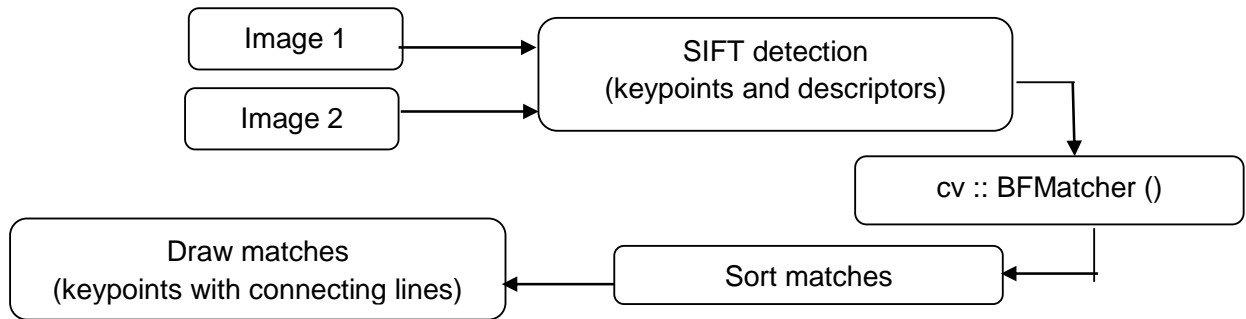


Figure 4.5: Algorithm for feature matching using SIFT

The output of the feature matching code is shown in figure 4.6. The difference in the light conditions and orientation of the object (candle) is visible. SIFT detector detects features of the candle and hand. It then compares the descriptors using a brute force matcher. The best matches are considered based on the minimum Euclidean straight-line distance between the two images' features. The algorithm shows a 70% success rate. The reason for 30% false detection is the indistinguishable pixel intensity values exhibiting the same orientation vectors.

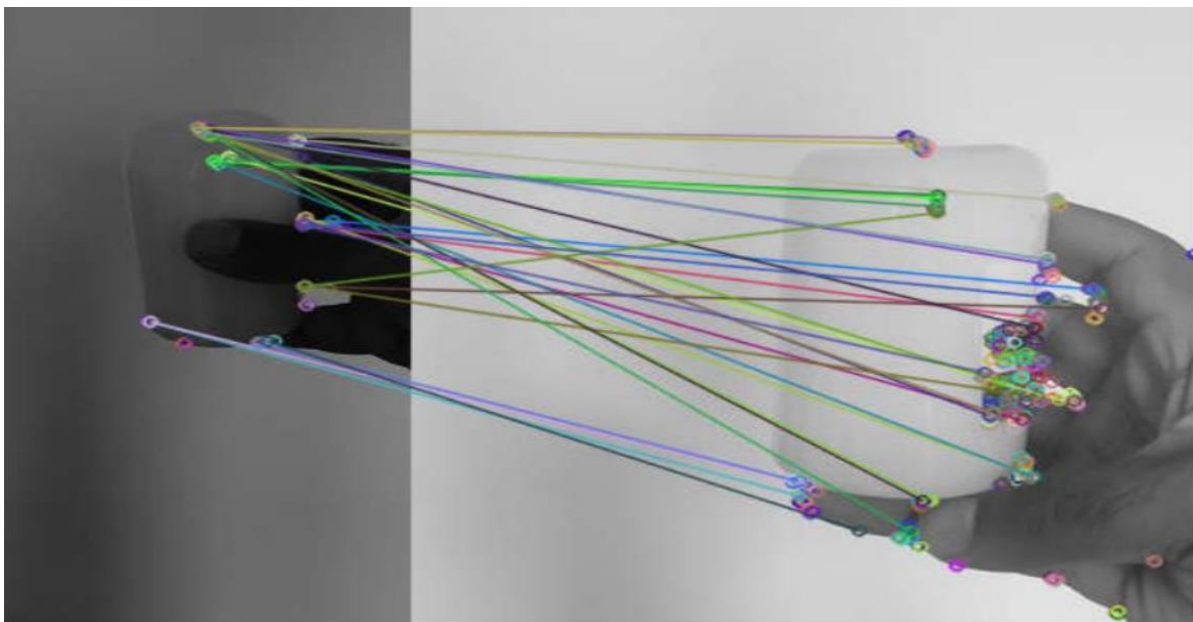


Figure 4.6: Feature matching of two scale variant candle scenes

4.3 Software for laser-matter interaction experiment

The next step was developing object detection and motion tracker software to detect and track the sample before and after the laser irradiation in the laser-matter interaction experiment.

4.3.1. Experimental Setup

To capture high speed videos at Stuttgart TP for sample detection and tracking, the experimental setup as shown in figure 4.7.

Hardware configuration:

- 2 Computer system
- High speed cameras: 1) IDT OS7 @ 4200 frames per second and 2) MotionScope M3 upto 8000 frames per second.
- 2 Led Lights for sample illumination: Veritas Constellation 120 E28
- Test sample
- Calibration plates with circular grid pattern
- Dropping arm setup

Software configuration:

- Motion Studio software for OS7 camera
- Imaging Studio software for M3 camera
- Arduino

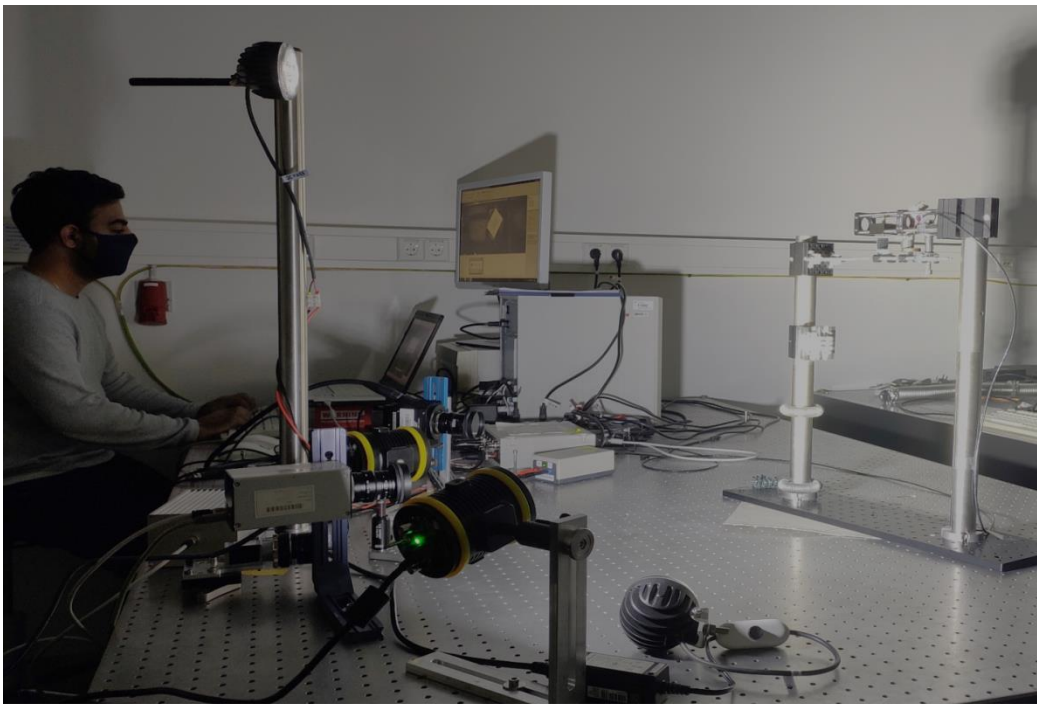


Figure 4.7: Experimental setup for high-speed sample video data.

4.3.2. Sample detection

This section describes about the algorithm used to write python scripts for detection of sample using a) SIFT and b) Contour based Harris Corner detector.

Input: High-speed videos @ 1000 fps in .avi and .mp4 format

Output: Sample detection

The sample shown in figure 4.8 contains corners highlighted using chess-board markers. The markers allows the detection of sample corners in the input video captured in less intensity light conditions.

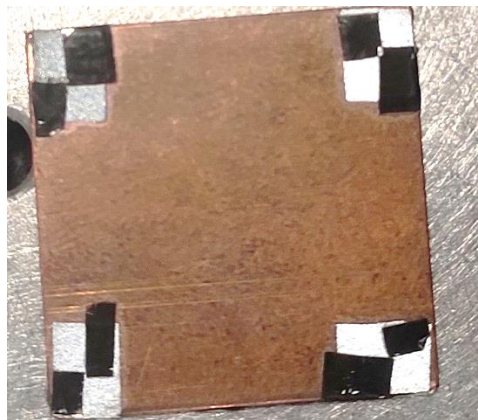


Figure 4.8: Target sample with corner markers

4.3.2.1. SIFT Detector

The steps followed by the python script to detect sample using sift detector is shown in figure 4.9.

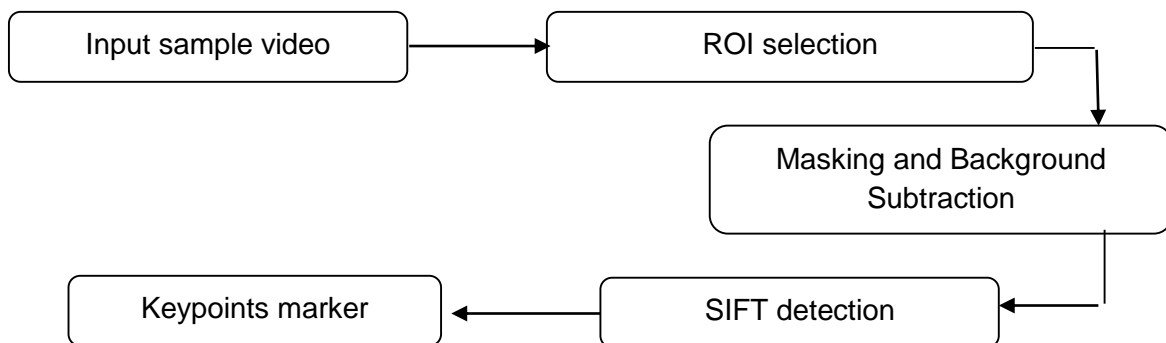


Figure 4.9: Algorithm for sample detection using sift

Explanation:

To eliminate the complexity of true key point detection and reduce computational processing time. The region of interest was sliced out from the original frame, resulting in the frame shown in figure 4.10. The method of background subtraction was used to identify the moving object in the scene. In this case, the moving objects detected were the sample during free-fall and the dropping arm, as shown in figure 4.10a. Due to foreground objects' motion, change in shadow was detected by the background subtraction algorithm as the features. However, after ROI selection, the shadow features were later regarded as noise. Sample and corner marker detection can be observed in figure 4.10b.

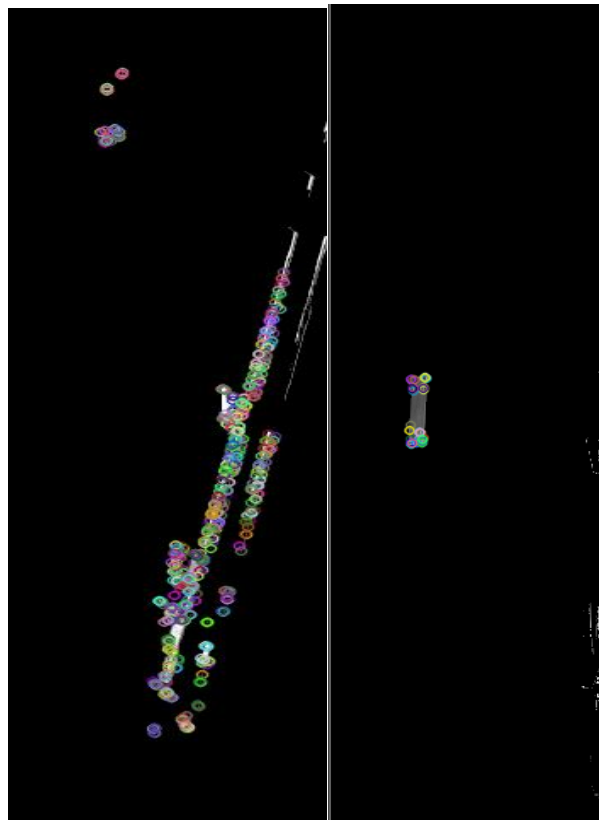


Figure 4.10: SIFT detection of a) moving objects b) dropping sample

4.3.2.2. Contour based Harris Detector

The steps followed by the python script to detect sample using sift detector is shown in figure 4.11.

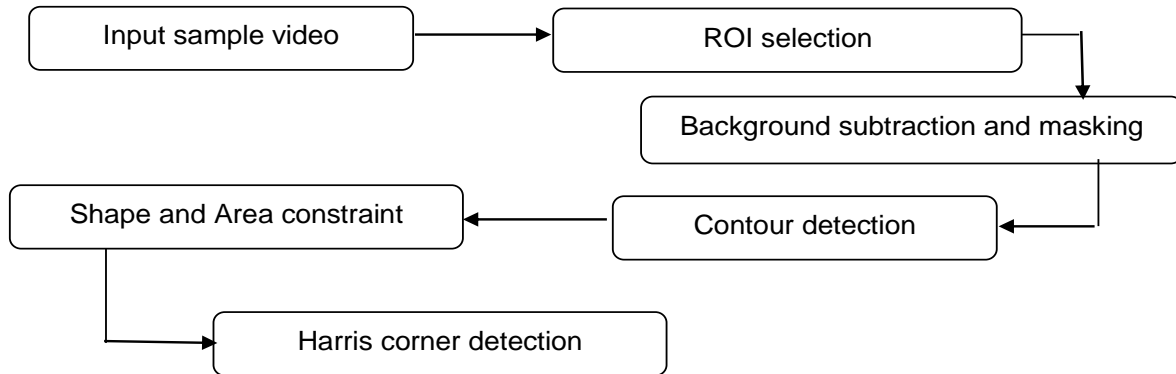


Figure 4.11: Algorithm for contour-based Harris corner detector

Explanation:

Silhouette/ Contour-based tracking identifies the object within a defined pixel area and shape of the contour. The contour's shape was a quadrilateral along with a pixel area between the range of 800- 1100 pixel². The defined contour limits the Harris Corner detector's detection area to reduce the complexity in the detection and increase the computational process time. The code's output is shown in figure 4.12, where the sample contour was represented by a green color box and corners were marked with red dots.



Figure 4.12: Contour based detection and tracking

4.3.3 2D Motion Tracker

Kernelized Correlation Filter tracker and Point-based Optical flow tracker were used to detect the dropping sample's motion and trajectory. This section describes the algorithm and the results obtained from the implementation of the tracker on sample videos.

Input: Laser- matter interaction experiment (Lorbeer et al., 2018) and sample testing videos

Output: Sample tracking

4.3.3.1 KCF Tracker

The steps followed by the python script to track the sample using in-built KCF tracker is shown in figure 4.13.

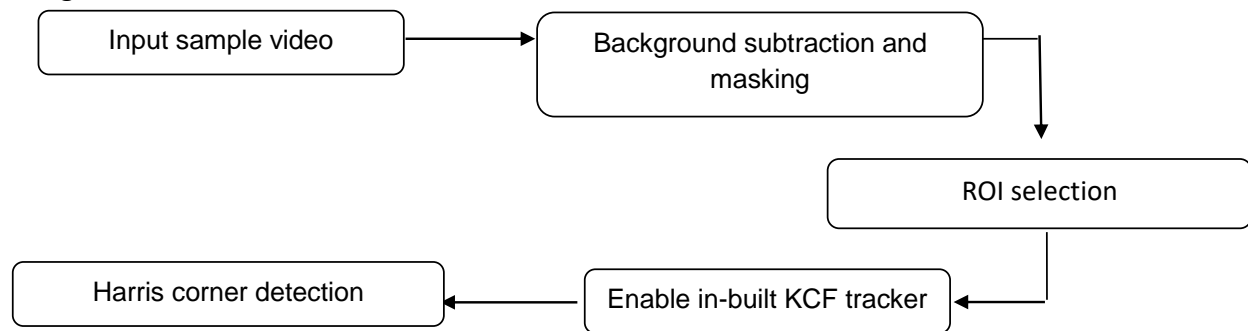


Figure 4.13: Algorithm for kernelized correlation filter tracker

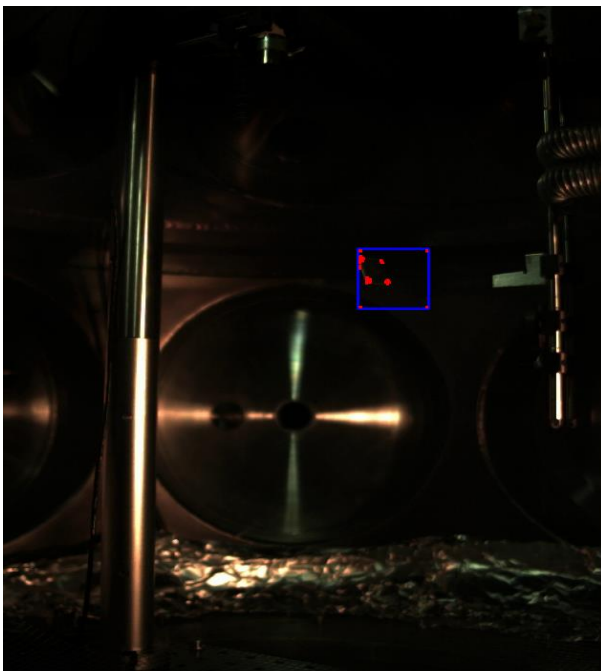
Explanation:

Initially, the video was set to begin from the 100th frame, such that the user can select the object by drawing a bounding box around it, as shown in figure 4.14 a. After selecting ROI, the pixel confined in the area was stored in the OpenCV tracker function `cv2.TrackerKCF_create()`. The target gets locked, and the tracker position was updated with the sequence of frames. In addition to the tracker function, the Harris corner detector algorithm was used within the region of interest or the bounding box. Harris function detects the corners and marks them with red dots. Since the sample's side view was visible during the freefall, the Harris detector only detects two corner points, as shown in figure 4.14 b. But after laser irradiation, the orientation of the object changes resulting in the detection of all four corners of the sample, as shown in figure 4.14 c.



(a)

(b)



(c)

Figure 4.14: a) Target object selection b) Two corners detected before laser-irradiation c) All four corners detected after laser-irradiation

4.3.3.2 Optical Flow Tracker

The steps followed by the python script to track the sample using simple optical flow point based tracker is shown in figure 4.15.

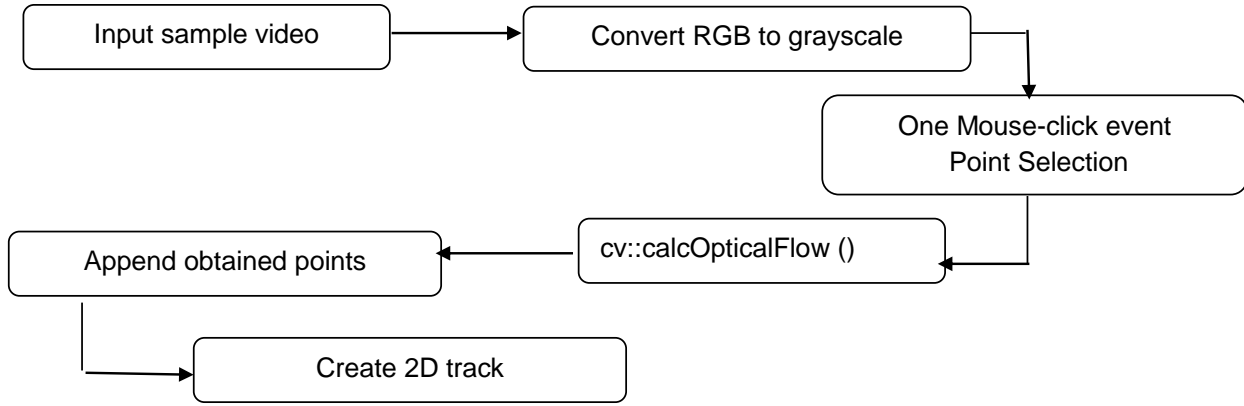


Figure 4.15: Algorithm for simple optical flow point based tracker

The simple optical flow point based tracker begins reading the video frame and tells user to select a point using mouse click event such that the tracking can be initiated, while the frame count is increasing. This results in a number of errors such as:

1. Selection of wrong target point.
2. In case optical flow function loses the point, the user has to end the program and start it all over again.

Thus, to reduce the complexity in target point or feature selection and maintain the accuracy in tracking. A more robust script was developed which allow user to

1. Select the interest frame/target point.
2. Manual autocorrection of target pixel track.

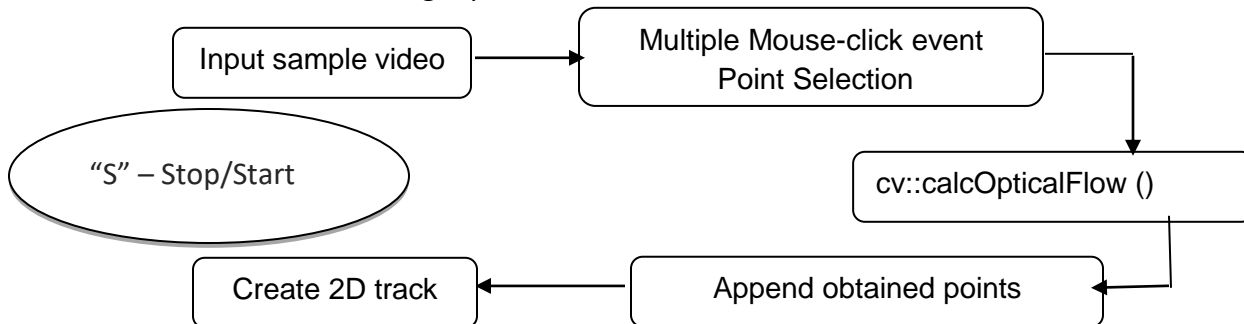
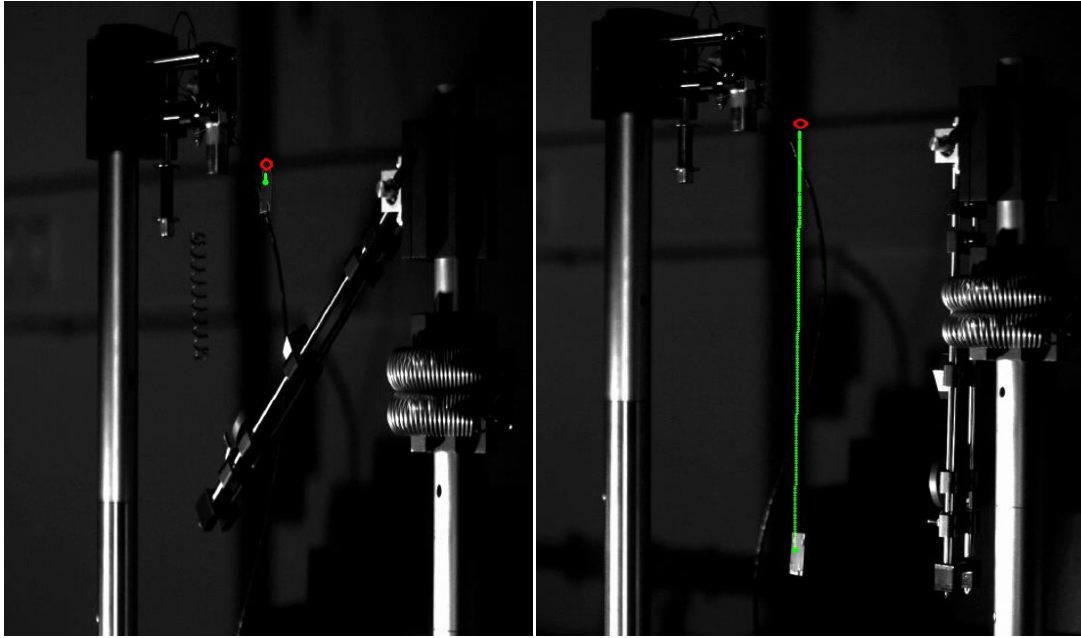
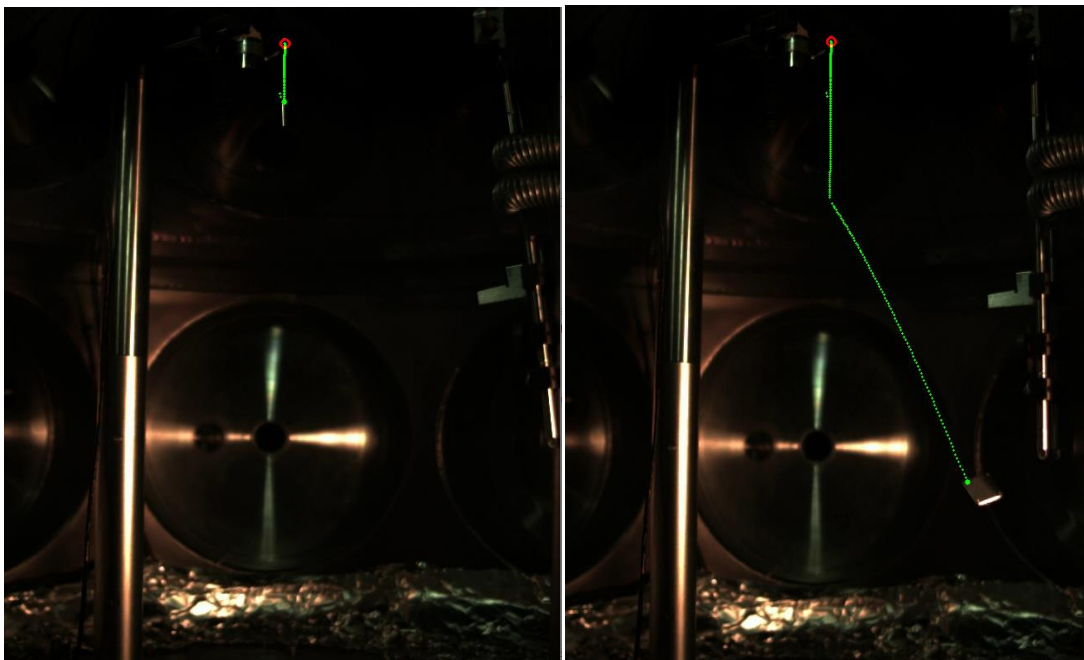


Figure 4.16 Algorithm for robust optical flow point based tracker



(a)



(b)

Figure 4.17: a) Simple optical flow on sample videos b) Robust optical flow tracker on laser-matter experiment videos (red circle – user point selection; green line – automatic 2D trajectory)

4.3.4 3D Multi-view camera calibration

Calibration and pose estimation:

Michael Zwillich originally developed the code for camera calibration during his bachelor thesis. It was a necessary step, and it determines the input parameters required for the development of a 3D tracker.

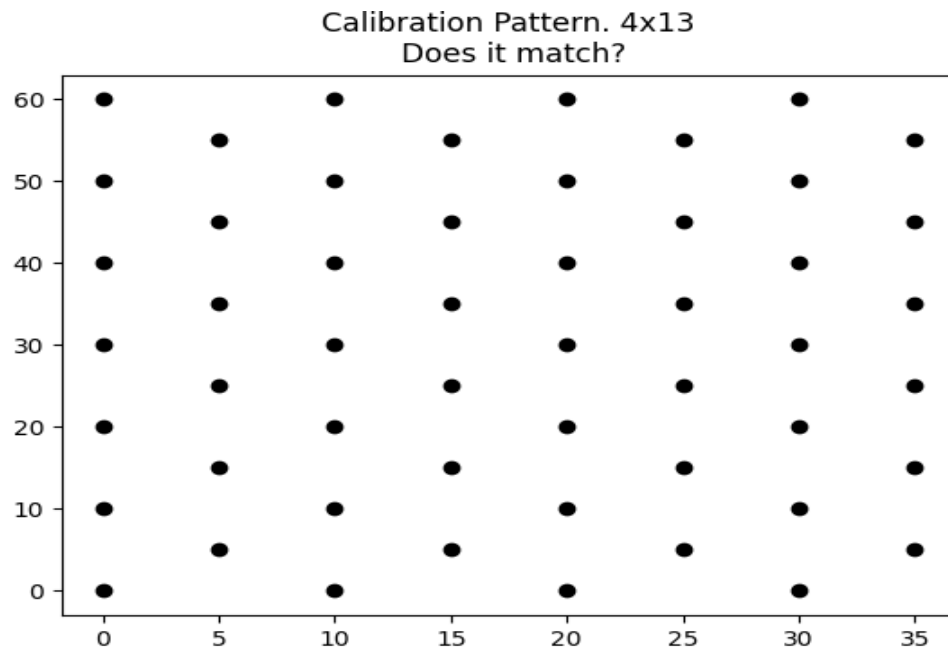
Input: Calibration images folder in .tif or .tiff format @ 150 frames/sec

Output: Camera intrinsic parameters: Camera matrix and distortion matrix

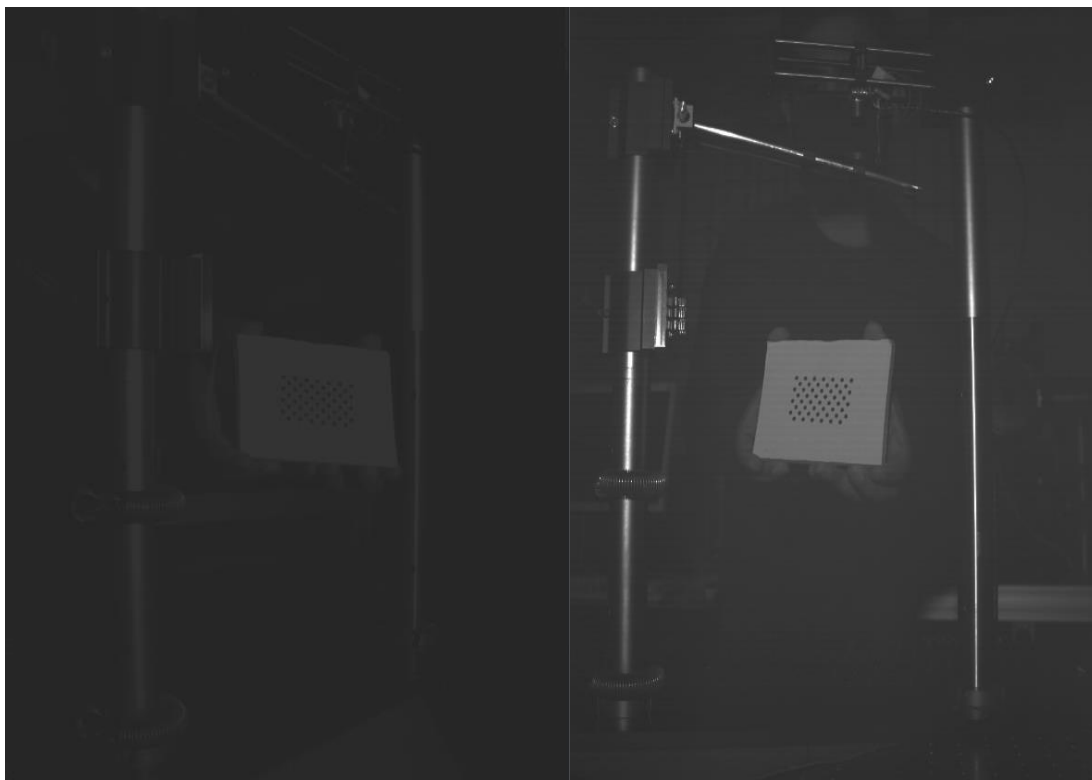
Camera extrinsic parameters: Rotation, Translation, Fundamental matrix

The steps for obtaining 3D calibration are as follows:

1. Print circular-grid pattern as shown in figure 4.18 (a) on rigid plate.
2. Capture calibration videos using high speed cameras and save them as .tif image files as shown in figure 4.18 (b) and (c).
3. Load camera properties and calibration images of both OS7 and M3 camera.
4. Detect calibration pattern in images using `cv::drawChessboardCorners()` for each camera.
5. Sort the images in found and not found folder for each camera.
6. Draw epipolar lines on the images with detected pattern in found folder for each camera as shown in figure 4.18 (b).
7. Calculate intrinsic parameters for each camera using `cv::calibrateCamera()`.
8. Calculate reprojection error for each camera.
9. Calculate extrinsic parameter for each camera using `cv::stereoCalibrate()`.
10. Pose estimation of both cameras as shown in figure 4.18 (c).
11. Save intrinsic and extrinsic parameters.

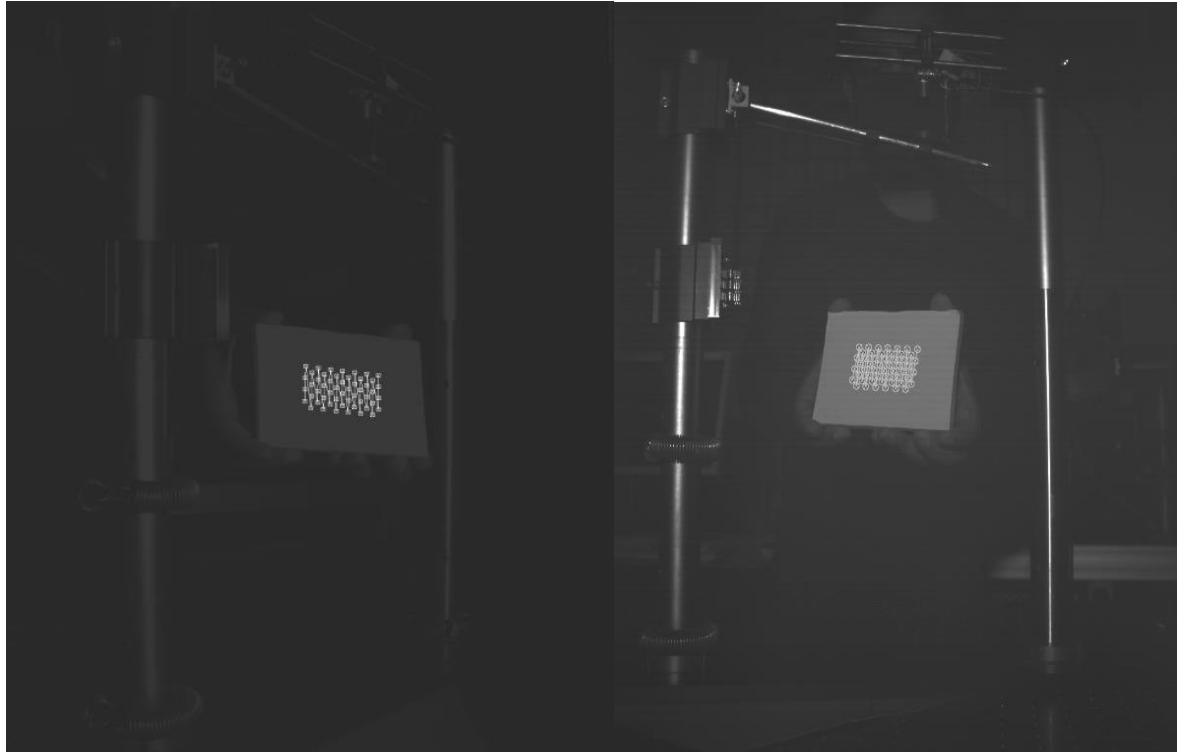


(a)



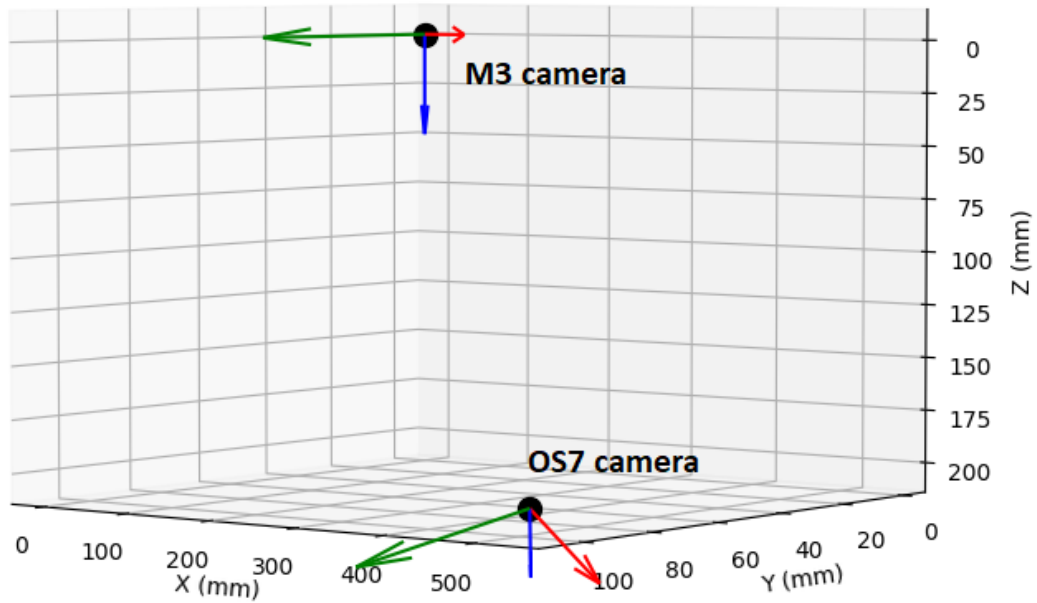
(b)

(c)



(d)

(e)



(f)

Figure 4.18: a) Calibration pattern, b) Frame captured from OS7 camera, c) Frame captured from M3 camera, d) Pattern detected in OS7 frame, e) Pattern detected in M3 frame, f) 3D Pose estimation of OS7 and M3 cameras

4.3.5. 3D Motion Tracker

This section describes the method used to obtain a 3-Dimensional track from two view camera calibration setup. 2D tracks obtained from robust optical flow point-based trackers were triangulated to obtain their corresponding 3D points, as explained in section 3.6.

Input: 2D trajectory coordinates (.txt file) and camera calibration parameters

Output: 3D trajectory plot

Homogeneity in data: Coordinates of 2D track obtained using optical flow were saved in the data file with .txt format. The file contains the frame number and the corresponding x and y coordinate of the target pixel. To maintain homogeneity in data and match the corresponding frame number and coordinates from both camera, the manual modification was implemented to sort the data files. The points stored at frame number 1 show the coordinate of the same pixel in both the cameras.

The steps followed by the python script to obtain the 3-dimensional trajectory of the dropping sample is shown in figure 4.19.

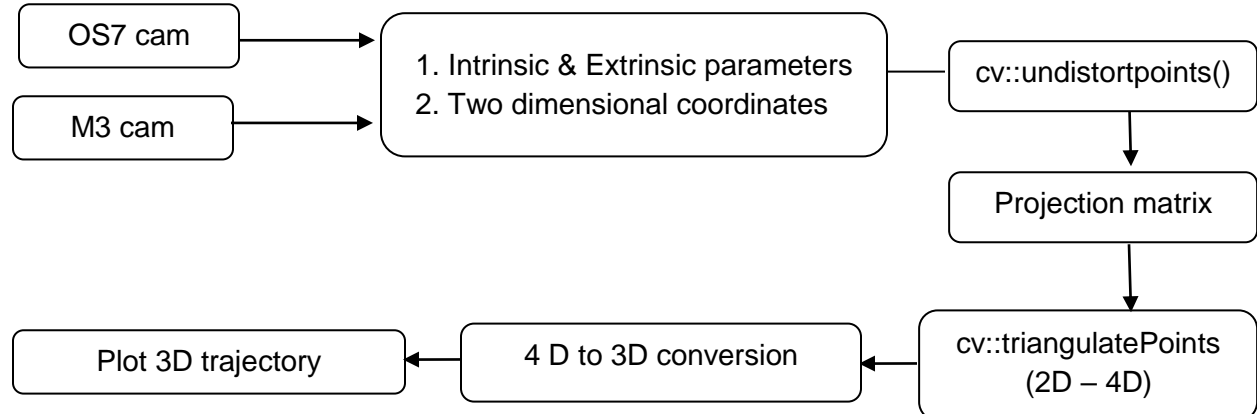


Figure 4.19 Algorithm for 3-dimensional motion tracker

Summary:

1. Load intrinsic camera parameters, extrinsic camera parameters and 2D target/corner points.
2. Remove distortion from the camera parameters to make points homogenous using `cv::undistortPoints()`.
3. Conversion a) 2D to 4D points using triangulation function `cv::triangulatePoints` and then b) 4D to 3D points.
4. 3-Dimensional trajectory

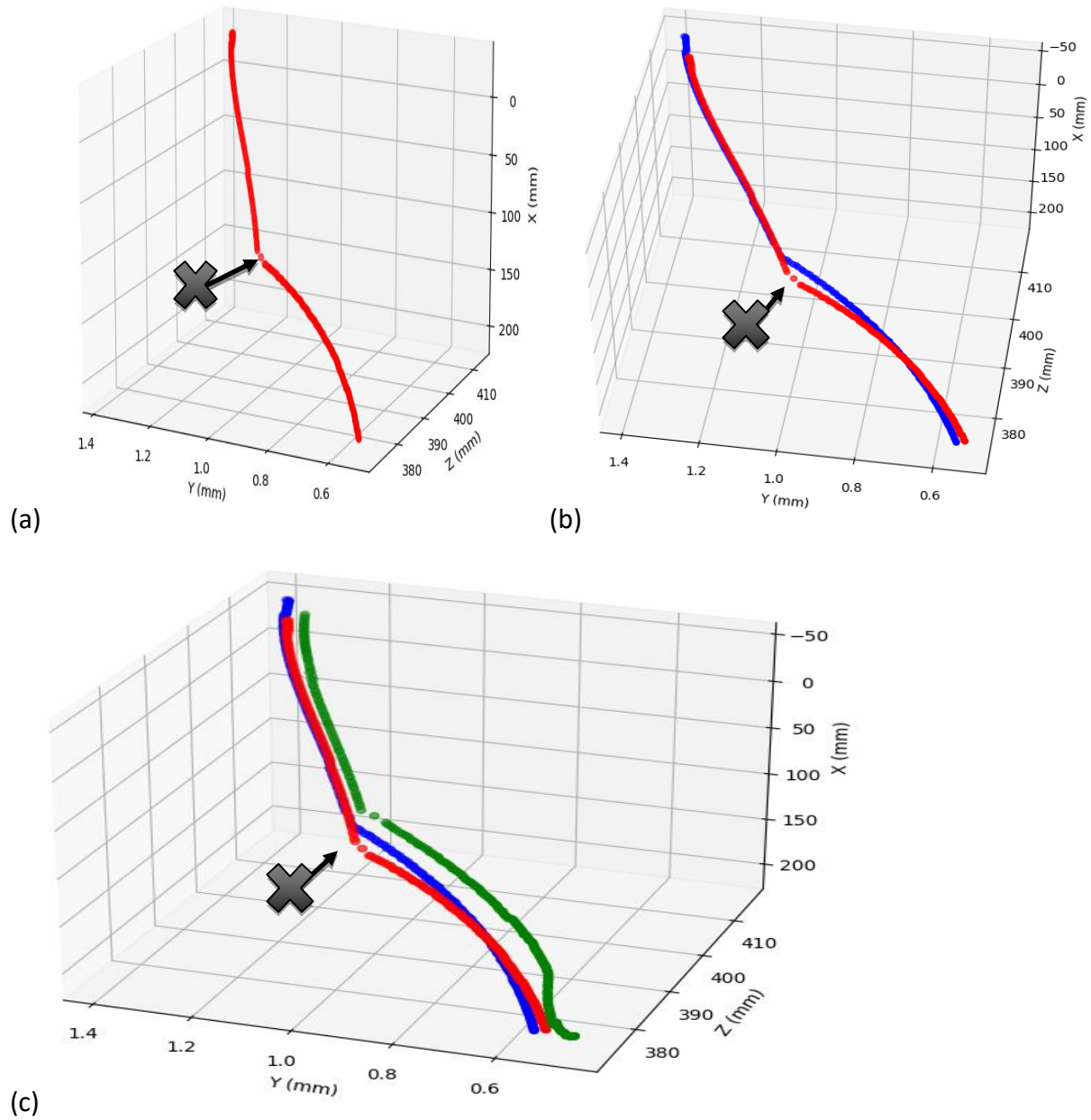


Figure 4.20: 3D trajectory of sample a) bottom- left corner (red), b) top-left corner (blue), (c) top-right corner (green)

Discussion of results:

From the 3D trajectory plot shown in figure 4.20, the deflection after a point marked X is observed. It means that an impulse was generated after the laser-irradiance, which deflected the sample from its original free-fall vertical trajectory into a parabolic path. Thus, it nearly verifies the underlying principle of impulse generation from laser-matter interaction. Also, from figure 4.20 (b), it is seen that the position of the lower-left edge of the sample marked with the

red trajectory crosses the position of the top-left edge marked with blue trajectory while reaching the end of the flight path. It signifies that the impulse generation also introduces the rotation of the sample. The parameters related to this will be quantified in the later part of the internship

4.4. Source of error

In the process of development of object detection and motion tracker software, the errors can be produced in three stages:

1. Data collection (Capturing high-speed videos): During the imaging and videography process, the error can be caused by physical damage to the camera such as a scratch in the lens, low camera resolution, unfocused lens, low light conditions, delay in the trigger of the camera and the dropping setup, the position of the cameras etc.
2. Software development: Image preprocessing parameters such as brightness and image filters, background noise, defining the range of threshold parameters for OpenCV functions, a heavy computational process sometimes crash the software.
3. Data postprocessing: Input files with different/unreadable format, a user-induced error such as selecting the wrong pixel point while tracking, visibility concerns in the gray-scale image due to low light conditions.

4.5 Estimation of Interest parameters

Lorbeer et al. (2018) calculate the interest parameter using the EXPEDIT simulation tool. The list of interest parameters are as follows:

1. Velocity
2. Angular Velocity
3. Momentum
4. Impulse
5. Center of Gravity of the sample

For the remaining part of the internship, the task is to obtain these parameters from the 3D trajectory plot achieved from the 3D tracker code, as explained in section 4.5.6. These parameters will then be validating the values obtained using the EXPEDIT simulation tool in the experiment.

Chapter 5: Internship experience

From the beginning of the MSS program, I wanted to pursue my career and conduct research on space debris removal. The opportunity to work at one of the prestigious space research institutes involved in research and development related to active space debris removal was a dream turning into reality. The project work at the Institute of Technical Physics of German Aerospace Center was a part of the technology demonstration for laser based orbital debris removal. Even though there were restrictions due to the globally faced COVID-19 pandemic, I had a chance to do the project on-site.

The internship project was my first-time exposure to the professional working culture in an elite organization, and thus it brings a lot of leanings. On the first day, there was a campus tour and briefing about the project. The entire plan was discussed, along with a description of the tasks required to complete the project. Initially, it took me some time to understand all the protocols and guidelines to be followed during the internship. But later, I was more comfortable with it. The project focused on software development. Due to a lack of experience and background in the field, it was a bit struggle to learn the coding language. To get used to software and programming, I took help from the online tutorials and coding platforms from time to time to proceed further and accomplish the given tasks. After the first month, I was entirely into the topic and was progressing faster than expected. Getting introduced to computer vision, especially object detection and motion tracking, was advantageous due to its broad spectrum of applications even in the space industry. It feels incredibly proud that at the end of the project, I will be delivering my first software package related to motion tracker that can be used in space debris research. To enhance decision making and analytical skills, I had to justify selecting an algorithm based on comparative analysis and choosing the best fit for the application.

From the team, Mrs. Keil's support and guidance related to professional and personal development made things flow well during the internship period. At times when I asked things repetitively, she helped me in understanding the concept with patience. She was always willing to help whenever I felt stuck to accomplish a task and encouraged me for independent decision-making and problem-solving, which is an important skill required for a scientific career. The experience working under her guidance was amazing, and I have learned a lot from her. As the only non-European student in the institute, I received friendly gestures and a comfortable working environment from the team.

During the internship, the most challenging task was to figure out the procedures and paperwork to obtain the required visa, particularly during the COVID-19 pandemic. There were

many unanswered questions related to the visa and the extension of the contract. However, with the team's help at the institute and university, things did work out in the end.

The summary of benefits gained from the internship is as follows:

1. I had a chance to work on the subject that I am passionate about, which is space debris removal
2. I had a chance to learn a python programming language and develop my first software package
3. I learned a lot about computer vision, object detection, and motion tracking.
4. I learned how to manage the project, tasks, and time independently.

My recommendation to future students interested in research in optics and laser physics is to apply without hesitation to the Institute of Technical Physics. The institute is working on a large number of projects related to the development of cutting-edge technology in optics and lasers. The working environment and the team here are welcoming and supportive. Thus, working at the German national research institute will be a boost for your professional career. Having necessary German skills will be an advantage in daily and professional life.

The project covers some of the ISU disciplines, as explained below:

1. **3Is:** The team comprised of scientists and engineers. During team meetings, lunch breaks and extra outing activities, everyone shared their personal and professional experiences.
2. **Engineering:** The project involved programming and software development.
3. **Science:** The project involves a basic understanding of scientific principles behind laser-based debris removal in the context of the ablation of surface material and laser-matter interaction.
4. **Management:** Project and task management were essential topics of discussion during the internship.

Chapter 6: Conclusion

The internship project contributes to the development of motion trackers using computer vision techniques resulting in the verification of impulse generation through laser-matter interaction experiments. Different object dejection and motion tracking algorithms were reviewed to decide the best fit for the application. The code for object detection and motion tracker software was developed using the python programming language. The input data consists of video files from DLR's laser-matter interaction experiment conducted at GSI, Darmstadt, and some sample videos captured at the Institute of Technical Physics, Stuttgart, during the internship.

Contour-based Harris Corner and Scale Invariant Feature Transform (SIFT) Detector seems promising for identifying features and corners, resulting in the detection of objects in the moving frame. A specific region of interest was defined in the frame to reduce the computation process's complexity and background noise. Contour based detection depends on the area of visibility of the object in the masked frame. For motion tracking, an optical flow point-based tracker seems a more robust option. It allows the user to manually select the target pixel point and manually autocorrect the errors generated. At last, using multi-view 3D camera calibration and triangulating 2D tracks from the optical method, the sample trajectory in a 3-dimensional world coordinate system was obtained. Initially, the aim was to develop a fully automated 3D motion tracker. Later, it was decided that motion trackers based on a hybrid method involving user interaction and automatic algorithms generate relatively fewer errors. The trajectory clearly shows the change in flight path from vertical to parabolic and rotation of the sample after the point of laser-irradiation.

As the internship is not completed yet, few tasks are pending and will be updated in the remaining part of the internship. The status of the tasks listed in section 1.2.2. is shown in Table 6.1.

Table 6.1: Project activities with status

Activities / Tasks			Status
1.	Literature Review		
	1.1	Laser baser detection and removal of space debris	Complete
	1.2	Laser-matter interaction experiment	Complete
	1.3	Introduction to computer vision	Complete

	1.4	Identification of object detectors and motion trackers (2D)	Complete
	1.5	Beginner course on Python, Pycharm and Anaconda	Complete
2.	Computer Vision Software development in python		
	2.1	Feature Detection	Complete
	2.2	Object Detection	Complete
	2.3	Motion tracker (2 - Dimensional)	Complete
3.	Multi-view 3D tracking		
	3.1	Introduction to camera calibration and triangulation	Complete
	3.2	Understanding existing camera calibration code	Complete
	3.3	Development of motion tracker (3 - Dimensional)	Complete
	3.4	Debug 3D motion tracker	In progress (20 %)
	3.5	Extract interest parameters	Not started
4.	Internship final stage		
	4.1	Prepare internship report	Complete
	4.2	Prepare presentation	Not started
	4.3	Finalize project, assessment and software package	Not started

During the internship, I had a chance to develop my first software package related to real-life applications. The overall experience has been outstanding, enriching, and challenging with learnings related to personal and professional development. Also, I had the opportunity to work on my interest, which is space debris removal. It feels great and honored to be a part of a technical demonstration project for the betterment and safety of space technology and humanity.

References

- Adlakha, D., 2016. Analytical Comparison between Sobel and Prewitt Edge Detection Techniques. Ågren, S., 2017. *Object tracking methods and their areas of application: A meta-analysis*. [online] Available at: <<http://www8.cs.umu.se/education/examina/Rapporter/SannaAgrenFinal.pdf>> [Accessed 21 Oct. 2020].
- Ahmed, A.S., 2005. COMPARATIVE STUDY AMONG SOBEL, PREWITT AND CANNY EDGE DETECTION OPERATORS USED IN IMAGE PROCESSING.
- Alazzawi, A., Alsaadi, H., Shallal, A., OF, S.A.-D.J. and 2015, undefined, n.d. Edge detection-application of (first and second) order derivative in image processing. *iasj.net*. [online] Available at: <<https://www.iasj.net/iasj?func=article&ald=108975>> [Accessed 21 Oct. 2020].
- Avdeev, A. V, Bashkin, A.S., Katorgin, B.I. and Parfen'ev, M. V, 2011. About possibilities of clearing near-Earth space from dangerous debris by a spaceborne laser system with an autonomous cw chemical HF laser. *Quantum Electronics*, 41(7), pp.669–674.
- Biesbroek, R., 2013. *The e.Deorbit CDF Study*. [online] *iaassconference2013.space-safety ...*. Available at: <www.esa.int/cdf> [Accessed 23 Oct. 2020].
- Bischof, B., 2003. Roger - Robotic Geostationary Orbit Restorer. In: *54th International Astronautical Congress of the International Astronautical Federation, the International Academy of Astronautics, and the International Institute of Space Law*. [online] Reston, Virginia: American Institute of Aeronautics and Astronautics. Available at: <<http://arc.aiaa.org/doi/10.2514/6.IAC-03-IAA.5.2.08>> [Accessed 23 Oct. 2020].
- Brown, M. and Lowe, D.G., 2003. Recognising panoramas. In: *Proceedings of the IEEE International Conference on Computer Vision*. Institute of Electrical and Electronics Engineers Inc. pp.1218–1225.
- Chen, J., Wu, D., Song, P., Deng, F., He, Y. and Pang, S., 2020. Multi-View Triangulation: Systematic Comparison and an Improved Method. *IEEE Access*, 8, pp.21017–21027.
- Chen, Z., Tang, X., Hu, Z. and Yang, Y., 2014. Investigations into sensing characteristics of circular thin-plate electrostatic sensors for gas path monitoring. *Chinese Journal of Aeronautics*, 27(4), pp.812–820.

DLR, 2020. *DLR - Institute of Technical Physics - Active Optical Systems*. [online] Available at: <<https://www.dlr.de/tp/en/desktopdefault.aspx/tabid-2787/>> [Accessed 21 Oct. 2020].

DLR, 2020. *DLR - Institute of Technical Physics - Institute of Technical Physics*. [online] Available at: <https://www.dlr.de/tp/en/desktopdefault.aspx/tabid-2773/4211_read-6272/> [Accessed 21 Oct. 2020].

Eckel, H.-A., Göge, D. and Zimper, D., 2016. Laser-Based Space Debris Removal An Approach for Protecting the Critical Infrastructure Space. *undefined*.

Eckel, H.-A., Göge, D. and Zimper, D., 2016. Laser-Based Space Debris Removal An Approach for Protecting the Critical Infrastructure Space. *undefined*.

ESA, 2020. *ESA - Space debris by the numbers*. [online] Available at: <https://www.esa.int/Safety_Security/Space_Debris/Space_debris_by_the_numbers> [Accessed 20 Oct. 2020].

ESA, 2020. *ESA - Latest report on space junk*. [online] Available at: <https://www.esa.int/Safety_Security/Space_Debris/Latest_report_on_space_junk> [Accessed 23 Oct. 2020].

ESA, 2020. *ESA's Annual Space Environment Report*. Esmiller, B., Jacqueland, C., Eckel, H.-A. and Wnuk, E., 2014. Space debris removal by ground-based lasers: main conclusions of the European project CLEANSPACE. *Applied Optics*, [online] 53(31), p.145. Available at: <<https://www.osapublishing.org/viewmedia.cfm?uri=ao-53-31-145&seq=0&html=true>> [Accessed 23 Oct. 2020].

Esmiller, B., Jacqueland, C., Eckel, H.-A. and Wnuk, E., 2014. Space debris removal by ground-based lasers: main conclusions of the European project CLEANSPACE. *Applied Optics*, 53(31), p.145.

Gbash, E.K. and Saleh, S., 2017. Scale-Invariant Feature Transform Algorithm with Fast Approximate Nearest Neighbor.

Gregory, D.A. and Mergen, J.F., 2011. *Space debris removal using upper atmosphere and vortex generator*. [online] Available at: <<http://online.wsj.com/article?>> [Accessed 23 Oct. 2020].

Guerra, C., 2020. *Downsampling and Gaussian blur*. [online] Available at: <<http://chemaguerra.com/downsampling-and-gaussian-filters/>> [Accessed 21 Oct. 2020].

H. Choi, S. and S. Pappa, R., 2013. Assessment Study of Small Space Debris Removal by Laser Satellites. *Recent Patents on Space Technology*, 2(2), pp.116–122.

HABIMANA, S. and PARAMA, R.V.R., 2018. Space debris: Reasons, types, impacts and management. *Indian Journal of Radio & Space Physics (IJRSP)*, [online] 46(1), pp.20–26. Available at: <<http://op.niscair.res.in/index.php/IJRSP/article/view/15316>> [Accessed 23 Oct. 2020].

Han, S. and Vasconcelos, N., 2008. Object-based regions of interest for image compression. In: *Data Compression Conference Proceedings*. pp.132–141.

Harris, C. and Stephens, M., 2013. A Combined Corner and Edge Detector. British Machine Vision Association and Society for Pattern Recognition.pp.23.1-23.6.

Hartley, R. and Zisserman, A., 2004. Epipolar Geometry and the Fundamental Matrix. In: *Multiple View Geometry in Computer Vision*. [online] Cambridge University Press. Available at: </core/books/multiple-view-geometry-in-computer-vision/0B6F289C78B2B23F596CAA76D3D43F7A> [Accessed 23 Oct. 2020].

Hassaballah, M., Abdelmgeid, A.A. and Alshazly, H.A., 2016. Image features detection, description and matching. *Studies in Computational Intelligence*, 630, pp.11–45.

Hata, K. and Savarese, S., n.d. CS231A Course Notes 1: Camera Models. *www-inst.eecs.berkeley.edu*. [online] Available at: <<http://www-inst.eecs.berkeley.edu/~ee290t/fa18/readings/01-camera-models.pdf>> [Accessed 21 Oct. 2020].

Henriques, J.F., Caseiro, R., Martins, P. and Batista, J., 2014. High-Speed Tracking with Kernelized Correlation Filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, [online] 37(3), pp.583–596. Available at: <<http://arxiv.org/abs/1404.7584>> [Accessed 21 Oct. 2020].

Jacob, A.M. and Anitha, J., 2013. Inspection of Various Object Tracking Techniques. Jain, R., Kasturi, R. and Schunck, B.G., 2020. *Machine vision*. [online] Available at: <<https://eresources.nlb.gov.sg/printheritage/detail/c072b64d-7180-458a-b5d8-2ca057dfbd68.aspx>> [Accessed 21 Oct. 2020].

Jakubović, A. and Velagić, J., 2018. Image feature matching and object detection using brute-force matchers. In: *Proceedings Elmar - International Symposium Electronics in Marine*. Croatian Society Electronics in Marine - ELMAR.pp.83–86.

Kofford, A.S., 2013. *Patent Application Publication*.

Le, M.H., Woo, B.S. and Jo, K.H., 2011. A Comparison of SIFT and Harris corner features for correspondence points matching. In: *2011 17th Korea-Japan Joint Workshop on Frontiers of Computer Vision, FCV 2011*.

Lee, D.-H., Lee, D.-W. and Han, B.-S., 2016. Possibility Study of Scale Invariant Feature Transform (SIFT) Algorithm Application to Spine Magnetic Resonance Imaging. *PLOS ONE*, [online] 11(4), p.e0153043. Available at: <<https://dx.plos.org/10.1371/journal.pone.0153043>> [Accessed 24 Oct. 2020].

Lindeberg, T., 1998. Edge Detection and Ridge Detection with Automatic Scale Selection. *International Journal of Computer Vision*, [online] 30(2), pp.117–156. Available at: <<https://link.springer.com/article/10.1023/A:1008097225773>> [Accessed 21 Oct. 2020].

Lorbeer, R.A., Zwilich, M., Zabic, M., Scharring, S., Eisert, L., Wilken, J., Schumacher, D., Roth, M. and Eckel, H.A., 2018. Experimental verification of high energy laser-generated impulse for remote laser control of space debris. *Scientific Reports*, [online] 8(1), pp.1–9. Available at: <www.nature.com/scientificreports/> [Accessed 20 Oct. 2020].

Madhav, 2019. *Object Detection using Python OpenCV*. [online] Available at: <<https://circuitdigest.com/tutorial/object-detection-using-python-opencv>> [Accessed 21 Oct. 2020].

Mahadevan, V. and Vasconcelos, N., 2010. Saliency-based discriminant tracking. Institute of Electrical and Electronics Engineers (IEEE).pp.1007–1013.

Maini, R., 2004. Study and Comparison of Various Image Edge Detection Techniques.

Mason, J., Stupl, J., Marshall, W. and Levit, C., 2011. Orbital Debris-Debris Collision Avoidance. *Advances in Space Research*, [online] 48(10), pp.1643–1655. Available at: <<http://arxiv.org/abs/1103.1690>> [Accessed 23 Oct. 2020].

Merino, M., Ahedo, E., Bombardelli, C., Urrutxua, H., Peláez, J. and Summerer, L., 2011. Space debris removal with an Ion Beam Shepherd satellite: Target-plasma interaction. In: *47th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit 2011*.

Moravec, H., 1980. Obstacle avoidance and navigation in the real world by a seeing robot rover. *undefined*. Muelhaupt, T., 2020. *Crosslink Fall 2015 | The Aerospace Corporation*. [online] Available at: <<https://aerospace.org/paper/crosslink-fall-2015>> [Accessed 24 Oct. 2020].

Nisha, M., Mehra, R. and Sharma, L., 2015. Comparative Analysis of Canny and Prewitt Edge Detection Techniques used in Image Processing. *International Journal of Engineering Trends and Technology*, 28(1), pp.48–53.

Nosrati, M., Karimi, R., Hariri, M. and Malekian, K., 2013. Edge Detection Techniques in Processing Digital Images: Investigation of Canny Algorithm and Gabor Method. *undefined*.

Papert, S., 1966. The summer vision project. [online] Available at: <<https://dspace.mit.edu/handle/1721.1/6125>> [Accessed 23 Oct. 2020].

Patil, H. and Bhagat, K.S., 2015. Detection and Tracking of Moving Object: A Survey. *undefined*.

Phipps, C.R., Albrecht, G., Friedman, H., Gavel, D., George, E. V., Murray, J., Ho, C., Priedhorsky, W., Michaelis, M.M. and Reilly, J.P., 1996. ORION: Clearing near-Earth space debris using a 20-kW, 530-nm, Earth-based, repetitively pulsed laser. *Laser and Particle Beams*, 14(1), pp.1–44.

Phipps, C.R., Baker, K.L., Bradford, B., George, E.V., Libby, S.B., Liedahl, D.A., Marcovici, B., Olivier, S.S., Pleasance, L.D., Reilly, J.P., Rubenchik, A., Strafford, D.N. and Valley, M.T., 2011. Removing Orbital Debris with Lasers. *Advances in Space Research*, [online] 49(9), pp.1283–1300. Available at: <<http://arxiv.org/abs/1110.3835>> [Accessed 23 Oct. 2020].

Phipps, C.R., Baker, K.L., Libby, S.B., Liedahl, D.A., Olivier, S.S., Pleasance, L.D., Rubenchik, A., Trebes, J.E., George, E.V., Marcovici, B., Reilly, J.P. and Valley, M.T., 2012. Removing orbital debris with pulsed lasers. In: *AIP Conference Proceedings*. pp.468–480.

Remanan, S., 2020. *Beginner's Guide to Object Detection Algorithms | Analytics Vidhya | Medium*. [online] Available at: <<https://medium.com/analytics-vidhya/beginners-guide-to-object-detection-algorithms-6620fb31c375>> [Accessed 21 Oct. 2020].

Rosenhahn, B., Kersting, U., Andrew, S., Brox, T., Klette, R. and Seidel, H., 2005. A Silhouette Based Human Motion Tracking System. *undefined*.

Saravanan, C., 2010. Color Image to Grayscale Image Conversion. In: *2010 Second International Conference on Computer Engineering and Applications*. [online] IEEE. pp.196–199. Available at: <<http://ieeexplore.ieee.org/document/5445596/>> [Accessed 21 Oct. 2020].

Scharring, S., Wilken, J. and Eckel, H.-A., 2016. Laser-based removal of irregularly shaped space debris. *Optical Engineering*, 56(1), p.011007.

Schaumann, G., Schollmeier, M.S., Rodriguez-Prieto, G., Blazevic, A., Brambrink, E., Geissel, M., Korostiy, S., Pirzadeh, P., Roth, M., Rosmej, F.B., Faenov, A.Y., Pikuz, T.A., Tsigitkin, K., Maron,

Y., Tahir, N.A. and Hoffmann, D.H.H., 2005. High energy heavy ion jets emerging from laser plasma generated by long pulse laser beams from the NHELIX laser system at GSI. In: *Laser and Particle Beams*. [online] Cambridge University Press. pp.503–512. Available at: <https://doi.org/10.1017/S0022244805000000> [Accessed 20 Oct. 2020].

Shan, M., Guo, J. and Gill, E., 2016. *Review and comparison of active space debris capturing and removal methods. Progress in Aerospace Sciences*, .

Sharma, K.K., Ghazali, S., Dalai, A., Saraswat, K., Mondal, S., Devalla, V., Joshi, S. and Rawat, P., 2018. Space debris reduction using eddy currents. In: *2018 Atmospheric Flight Mechanics Conference*. [online] American Institute of Aeronautics and Astronautics Inc, AIAA. Available at: <https://arc.aiaa.org/doi/abs/10.2514/6.2018-3161> [Accessed 23 Oct. 2020].

Shen, S., Jin, X. and Hao, C., 2014. Cleaning space debris with a space-based laser system. *Chinese Journal of Aeronautics*, 27(4), pp.805–811. State University of New York at Buffalo, n.d. *Feature extraction Corners and blobs*.

Sun, M. and Srdjan, K., n.d. *Optical flow*. [online] Available at: <https://www.cs.princeton.edu/courses/archive/fall08/cos429/optiflow.pdf>. UCS, 2020. *Satellite Database | Union of Concerned Scientists*. [online] Available at: <https://www.ucsusa.org/resources/satellite-database> [Accessed 23 Oct. 2020].

Veltkamp, R.C., 2001. Shape matching: Similarity measures and algorithms. In: *Proceedings - International Conference on Shape Modeling and Applications, SMI 2001*. pp.188–197.

Vivian and Amy, 2020. *Tutorial 1: Image Filtering*. [online] Available at: <http://ai.stanford.edu/~syueung/cvweb/tutorial1.html> [Accessed 21 Oct. 2020].

Wiley, V. and Lucas, T., 2018. Computer Vision and Image Processing: A Paper Review. *International Journal of Artificial Intelligence Research*, 2(1), p.22.

Wormnes, K., Le Letty, R., Summerer, L., Schonenborg, R., Dubois-Matra, O., Luraschi, E., Cropp, A., Krag, H. and Delaval, J., n.d. *ESA TECHNOLOGIES FOR SPACE DEBRIS REMEDIATION*. [online] *esa.int*. Available at: <https://www.esa.int/gsp/ACT/doc/MAD/pub/ACT-RPR-MAD-2013-04-KW-CleanSpace-ADR.pdf> [Accessed 23 Oct. 2020].

Yakkali, S., Nara, V., Tikone, N. and Ingle, D., 2015. Robust Object Detection and Tracking Using Sift Algorithm. *undefined*.

Yilmaz, A., Javed, O. and Shah, M., 2006. *Object tracking: A survey*. *ACM Computing Surveys*, .

Zhang, S., Tian, Q., Huang, Q., Gao, W. and Rui, Y., 2014. USB: Ultrashort binary descriptor for fast visual matching and retrieval. *IEEE Transactions on Image Processing*, 23(8), pp.3671–3683.