Good Practices on Hand Gestures Recognition for the Design of Customized NUI

Damiano Malafronte and Nicoletta Noceti
DIBRIS, Università degli Studi di Genova, via Dodecaneso 35, 16146, Genova, Italy

Keywords: Natural User Interfaces, Gesture Recognition, Hand Pose Classification, Fingers Detection.

Abstract: In this paper we consider the problem of recognizing dynamic human gestures in the context of human-machine interaction. We are particularly interested to the so-called Natural User Interfaces, a new modality based on a more natural and intuitive way of interacting with a digital device. In our work, a user can interact with a system by performing a set of encoded hand gestures in front of a webcam. We designed a method that first classifies hand poses guided by a finger detection procedure, and then recognizes known gestures with a syntactic approach. To this purpose, we collected a sequence of hand poses over time, to build a linguistic gesture description. The known gestures are formalized using a generative grammar. Then, at runtime, a parser allows us to perform gesture recognition leveraging on the production rules of the grammar. As for finger detection, we propose a new method which starts from a distance transform of the hand region and iteratively scans such region according to the distance values moving from a fingertip to the hand palm. We experimentally validated our approach, showing both the hand pose classification and gesture recognition performances.

1 INTRODUCTION

In the last decades, the way people interact with digital devices has changed dramatically. From the era of the Command Line Interfaces, introduced in mid-1960s and used throughout the 1980s, users lived the evolution of Graphical User Interfaces (GUI), that had their key moment in the 1990s with the introduction of the WIMP (Windows, Icons, Menus and Pointer) paradigm.

In the last years, touch-based interaction systems have gained more and more popularity, outshining old-fashioned methods based on physical buttons. Now, digital users are getting closer and closer to a new revolution on human-machine interaction, characterized by a more natural and intuitive matching between an action and the consequent event associated with it. The term *Natural User Interfaces* (NUI) refers to all those modalities of interaction where a user is asked to use his own body to interact with the system. So, a user might be required to tap or slide a finger on a touch-sensitive screen, as well as to move a remote in the air while standing, or again to use arms and hands to perform dynamic gestures in front of a camera so that the system can recognize them (see an example of application in Fig. 1).

Figure 1: An example of photo browsing application which relies on a natural user interface.

Rauterberg gave a really nice definition of NUI (Rauterberg, 1999):

> *A system with a NUI supports the mix of real and virtual objects. As input it recognizes (visually, acoustically or with other sensors) and understands physical objects and humans acting in a natural way (e.g., speech input, and writing, etc.). [...] Since human beings manipulate objects in the physical world most often and most naturally with hands, there is a desire to apply these skills to user-system*

Malafronte D. and Noceti N..
Good Practices on Hand Gestures Recognition for the Design of Customized NUI.
DOI: 10.5220/0005304203600367
In *Proceedings of the 10th International Conference on Computer Vision Theory and Applications* (VISAPP-2015), pages 360-367
ISBN: 978-989-758-090-1

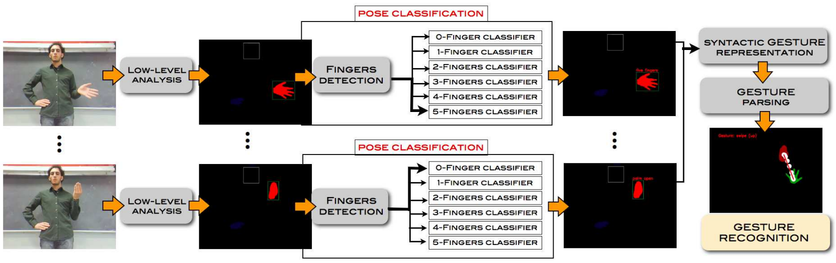


Figure 2: A sketch of our procedure.

> *interaction. In fact, NUIs allow the user to interact with real and virtual objects on the working area in a –literally– direct manipulative way.*

In this context, we propose a dynamic gestures recognition system with specific reference to applications in which users can bind a different effect to each gesture. So, our method may be instantiated to be the basis of a photo browsing application, but it might be also adopted in a context in which the user can not physically interact with a device, as e.g. during surgery to browse the patient clinical records.

Fig. 2 reports a visual representation of our pipeline. We start from a video stream acquired from a webcam and apply a well accepted video analysis procedure, that combines appearance and motion information to localize the user hand and gather trajectories of the hand position over time. At each time instant, we classify the shape of the detected region to understand the hand pose. We propose a new method that couples a distance transform of the shape with its convex hull, and that is based on a iterative procedure aiming at detecting stretched fingers. This specification allows us to organize the actual classification using multiple classifiers, one for each possible hand configuration (one finger, two fingers, ...). A simple KNN classifier is adopted to learn hand shape representations, based on the Hu moments.

Moreover, starting from a sequence of hand poses – or *history* – we propose an efficient modality of gestures recognition which relies on the use of generative grammars combined with the definition of an appropriate syntactic parser to semantically recognize the gesture representation at runtime. Thank to this approach, not only known gestures can be efficiently recognized, but also users can define new customized movements in a very intuitive way. This is done by assigning a user-friendly linguistic signature to the new gesture, which is then rendered with appropriate production rules of the gesture grammar.

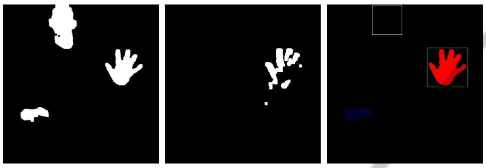
Related Works. In the past years the interest for gesture-based interaction has grown more and more, with a spread out of methods applied to a variety of fields. A rather complete review may be found in (Rautaray and Agrawal, 2012). Here we cite their use for sign language recognition (Paulraj, 2008), virtual and augmented reality (Choi et al., 2011), supporting tools for impaired users (Zariffa and Steeves, 2011) and robotics (Droeschel et al., 2011).

Previous approaches for finger detection (Sato et al., 2000; Fang et al., 2007; Dardas and Georganas, 2011) often rely on the use of appropriate shape descriptors. More related to our approach are previous works in which finger detection acts as a tool for recognizing static gestures, where the notion of gesture is blended in with the concept of hand pose. In (Ghotkar and Kharate, 2013) the authors proposed a method that starts from Kinect data and detects fingers by scanning the hand contour. Classification is fully based on the number of detected fingers, limiting the number of allowed poses. Kinect is also used in (Ren et al., 2013), where a part-based hand gesture recognition is presented, based on the Finger-Earth Mover's Distance. Recently, in (Chen et al., 2014) it is presented a real-time method based on segmenting palm and fingers and adopting a rule classifier.

In our work, we decouple the problem of classifying hand poses (static) from the hand gesture (that may be dynamic) recognition task. Similarly to (Chen et al., 2014), we do not require the final user to wear gloves, nor need the video streams to be acquired with refined sensors (as in (Ghotkar and Kharate, 2013)), so to widen the range of potential users. Our approach to fingers detection significantly differs from previous works in that we fully rely on the distance transform, whereas it is typically adopted as an intermediate step towards the computation of the hand skeleton.

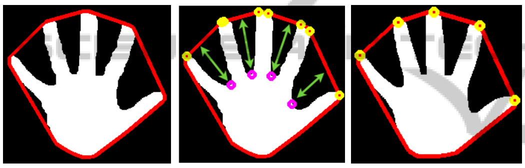
Finally, from the gesture recognition side, our work is related to syntactic approaches. For instance, high-level human activities have been recognized using context-free grammars (see e.g. (Bobick and Wilson, 1997; Ivanov and Bobick, 2000; Joo and Chellappa, 2006)). Compared to them, in our work the idea is that of keeping the model complexity under control, so to allow users to easily personalize the gestures portfolio. This simplicity does not affect the global strength of the system, since even humans memory capability fails to cope with lexicons that are too complex.

Structure of the Paper. The remainder of the paper is organized as follows. In Sec. 2 we provide a detailed description of our method for pose classification, while gesture recognition is the main topic of Sec. 3. We perform an experimental validation of our method in Sec. 4, while Sec. 5 is left to a final discussion.



(a) Skin (b) Foreground (c) Moving blob

Figure 3: Examples of output from the low-level analysis.



(a) Convex hull (b) Defects points (c) Fingertips candidates

Figure 4: Output of the main pre-processing steps.

2 HAND POSE CLASSIFICATION

In this section we present our approach to the classification of hand poses. We first analyse each image of the sequence to understand the presence of a moving hand (see Fig. 3). To this end, we start by segmenting the image with a skin detection step, which according to previous works (Singh et al., 2003) makes use of the YCbCr color space. Simultaneously, we also perform background subtraction, based on an adaptive mixture model (Zivkovic, 2004), to detect moving regions. We intersect the two binary maps to obtain moving regions corresponding to skin, and finally apply a face detection algorithm (Viola and Jones, 2001) to discard skin regions not corresponding to the user hand, which is finally detected.

At each time instant, the hand region undergoes different processing steps:

Finger Detection. We propose a new method to determine the presence of stretched fingers (see Sec. 2.2)

Hand Description. We describe the shape of the hand region – i.e. a *blob* of the binary map – by means of the Hu Moments (Hu, 1962), that are invariant to changes of position, scale or orientation. Although rather simple, they are very appropriate for a use in the considered scenario, where both invariance and computational efficiency are important properties.

Hand Pose Classification. The hand description is classified to determine the hand pose with a K-Nearest Neighbors (KNN) classifier (Dasarathy, 2002) with a distance computed as the sum of squared differences. We consider a portfolio of 6 classifiers, one for each plausible hand configuration (from 0 to 5 stretched fingers). The choice of the appropriate classifier is guided by the output of the finger detection.

2.1 Some Preliminaries

In this section we set the scene for the application of the finger detection method we propose. We start by determining the *convex hull* (Berg et al., 2008) of the points lying in the hand region (Fig. 4(a)). With *convexity defects* we refer to boundary points (purple in Fig. 4(b)) characterized by a distance (green arrows) from the corresponding contour segment (i.e. segments of the convex hull between two yellow points) greater than a threshold ε, which we set proportional to the blob area.

We go one step further with the computation of the *distance transform* (Danielsson, 1980) of the hand region. Starting from a binary map, the method relies on computing for each position inside the region the distance from the nearest boundary point (see Fig. 5(a) and 6, third column) estimated following (Borgefors, 1986). The highest value of the distance provides an estimate of the palm center (Fig. 6, bottom).

Information about convexity defects and distance transform concurs to understand the presence of stretched fingers, as clarified in the next section.

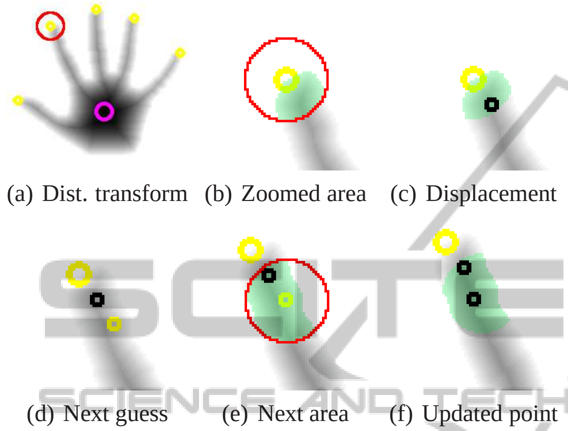
2.2 Fingers Detector

We propose here a new algorithm for fingers detection based on an effective scanning procedure that iteratively reasons on the distance transform.

Let CH be the convex hull of a hand region B, and let D be the set of the defects points detected on B with respect to CH. Each element $DP_i \in D$ is associated with a set of information, i.e. (i) the actual image point $\mathbf{P}_i$ lying on the boundary of B (purple points in Fig. 4(b)), (ii) two points $(\mathbf{P}_i^L, \mathbf{P}_i^R)$ delimitating from left and right the segment of the boundary of CH associated with $\mathbf{P}_i$ (Fig. 4(b) in yellow), and (iii) the distance d_i of $\mathbf{P}_i$ from the above mentioned segment (green arrows in Fig. 4(b)).

All points $\mathbf{P}_i^L$ and $\mathbf{P}_i^R$ are candidates to be recognized as fingertips. In presence of multiple detected locations in a limited spatial range a single candidate point is considered, computed as the mid-point of the segment connecting the two candidates (see Fig. 4(c)).

Starting from each candidate, the distance transform is iteratively analysed as follows (see a sketch in Fig. 5).



(a) Dist. transform (b) Zoomed area (c) Displacement

(d) Next guess (e) Next area (f) Updated point

Figure 5: A visual sketch of our finger detector algorithm.

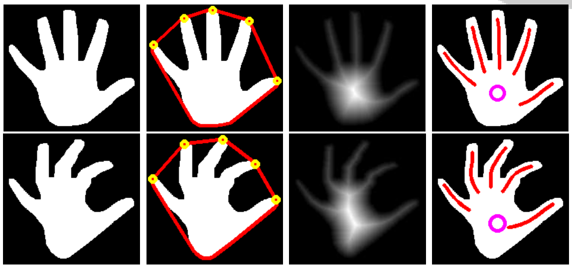


Figure 6: Examples of intermediate output of the finger detection procedure. From the left: hand blob, convex hull and candidates points, distance transform, detected fingers and estimated palm center.

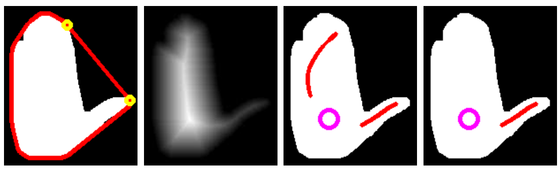


Figure 7: An example of detection failure, that might be corrected by refining our procedure (see text).

We start by considering a spatial range around a candidate of radius τ proportional to the hand palm extent. Within this range we only consider points belonging to the hand blob (in green in Fig. 5(b)) and compute their average position weighted according to the distance transform (*DT*). More formally, if $\mathbf{P}_i^0 = [x_i, y_i]$ is the candidate point and $\mathbf{P} = [x, y]$ is a point lying in the hand blob B within a distance τ from $\mathbf{P}_i^0$ we compute

$$\mathbf{P}_i^1 = [x_i^1, y_i^1] = \frac{1}{N} \sum_{x,y} DT(x,y) \cdot \mathbf{P}. \qquad (1)$$

Fig. 5(c) shows an example where the estimated average point is denoted in black. Intuitively, the weighted average allows us to move from the candidate towards the direction of higher distance, thus following a path which is situated in the middle of the finger. The procedure is then iterated as follows. Let $\mathbf{P}_i^n$ be the last weighted average, then:

1. If $\mathbf{P}_i^n$ is within the palm radius, the procedure is stopped.
2. Otherwise, we consider the displacement between $\mathbf{P}_i^{n-1}$ and $\mathbf{P}_i^n$, apply it to $\mathbf{P}_i^n$ and so obtain an initial guess $\hat{\mathbf{P}}_i^{n+1}$ (Fig. 5(d)).
3. We consider a spatial range around our guess with a radius τ (Fig. 5(e)) and estimate the next point $\mathbf{P}_i^{n+1}$ using Eq. 1 where we replace $\mathbf{P}_i^1$ with $\mathbf{P}_i^{n+1}$ (Fig. 5(f)).
4. Return to point 1, with $n = n + 1$.

Fig. 6 (last column) reports examples of our output. To reduce the amount of detection failures (as the one reported in Fig. 7) we also consider a second threshold τ', initially set to τ and then fixed to twice the current displacement, that we use to define a range in which we estimate the area of the hand blob overlapping with it (i.e. the extent of the green area in Fig. 5(b)). This is to the purpose of estimating the finger size, so to avoid anomalous detections. Consider the example in Fig. 7: without this further step of analysis, the part of the blob corresponding to the four connected fingers would be incorrectly classified as a single finger.

3 GESTURE RECOGNITION

Thanks to the algorithm introduced in the previous section we can predict a label describing the hand pose whenever we observe in the image a moving region with all the appropriate characteristics. Supported by the tracking algorithm, we are also able to segment sequences of temporally adjacent hand shapes, representing the evolution of the hand appearance during the dynamic event.

We start from here to define our method for recognizing gestures. Let us define as *history of a blob* a sequence $H = [h_{t_1} h_{t_2} \ldots h_{t_N}]$ of observations, each one including different descriptions of the blob instance. More specifically $h_{t_i} = (t_i, P_i, A_i, L_i)$, where t_i is the time instant of the observation, P_i is the position of the

blob centroid in the image plane, A_i is the blob area, while L_i is the pose label associated with the blob in case it has been classified as instance of one known hand poses, a special value otherwise.
Starting from the history, each gesture is compactly represented using a linguistic description as

Gesture = { `Start_Pose`, `Final_Pose`, `Area_Var`, `Pos_Var`, `Turns` }

where

- `Start_Pose` and `Final_Pose` are identifiers of known poses, but may be also *Any*, when all poses are allowed, or *Same* (as final pose) when the same hand shape characterizes the whole gesture.
- `Area_Var` $\in$ {*Incr*, *Decr*, *Same*} indicates the variations of the blob extent across the history (it may increase, decrease or remain stable)
- `Pos_Var` $\in$ {*Diff*, *Same*} describes the variation between the initial and final positions of the history, which may change or not.
- `Turns` $\in$ {*Yes*, *No*,} specifies the presence of changes in the motion direction.

We define the known gestures $\mathcal{G}$ as a set of formal representations generated by a grammar G, or, in other words, a subset of the language generated by G: $\mathcal{G} \subset \mathcal{L}(G)$.

A formal generative grammar G, which we define according to (Chomsky, 1956), is determined by a tuple $< N, \Sigma, s, P >$, where N is a set of non-terminal symbols (disjoint from from G), Σ is a set of terminal symbols (disjoint from N), $s \in N$ is the start symbol, while P is a set of production rules.
In order to evaluate the pertinence of a history with one of the known gestures we try and match the annotated characteristics with the observed one by defining a syntactic parser, tailored for a set of gestures known to the system, able to associate a semantic meaning (if any) to a formal representation. Should one annotated gesture match the sequence of the history, the recognition is accomplished.

The customization of the system is favored by this intuitive representation, in which the user is only asked to specify a set of well defined information characterizing the gesture. The parser, then, is updated to enroll the rules associated with the new gestures.

Table 1: Comparison of the classification accuracies without and with finger detector.

Method	Mean Acc. (%)
K=1, without finger detection	92.7
K=1, with finger detection	99.4
K=3, with finger detection	99.8

Table 2: Comparison with other approaches to finger detection.

Method	Mean Acc. (%)
Th. Dec.+FEMD	93.2
Near-Con. Dec.+FEMD	93.9
Our approach	94.5

Table 3: Comparison of pose classification accuracies.

Method	Acc. (%)	Time (s)
Shape context (no bending cost)	83.2	12.346
Shape context	79.1	26.777
Skeleton Matching	78.6	2.4449
Near-convex Dec.+FEMD	93.9	4.0012
Thresholding Dec.+FEMD	93.2	0.075
Our approach	90.5	0.0009

4 EXPERIMENTS

In this section we present the experimental validation of the approach we propose.

Some clarifications on the choice of the method parameters are in order. The thresholds used during the skin detection phase are determined with a brief training phase in which the user is asked to put his fist in front of the camera. As for the background subtraction, we experimentally fixed the size of the temporal window to be of 30 frames (at 25fps).

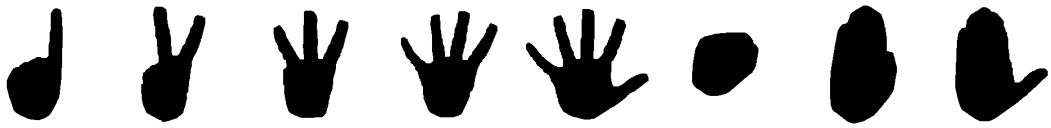
The threshold ε used to detect the convexity defects was set to the 45% of the blob area. The radius τ of the spatial range used for detecting fingers was fixed to 25% of the radius of the hand palm (i.e. the distance between the palm center and the nearest boundary point).

All the experiments were run on laptop computer equipped with a Intel i7-2670QM 2.20 GHz CPU and 8 GB RAM.

4.1 Evaluation of the Finger Detector for Pose Classification

We evaluated the robustness of our method considering a selection of 8 possible hand poses, depicted on Fig. 8 (in brackets we report the identifiers we will use henceforth): One (1), Two (2), Three (3), Four (5), Five (5), Punch (Pu), Palm (Pa), Palm and Thumb (PT). We collected a dataset composed by hand poses gathered from four different subjects. For each subject, around 50-60 samples have been acquired. The performance of our finger detector are very close to be perfect, with an average accuracy of 99.7%.

To perform pose classification, we followed a *Leave-One-Person-Out* approach, replicating the ex-



(a) one (1) (b) two (2) (c) three (3) (d) four (4) (e) five (5) (f) punch (Pu) (g) palm (Pa) (h) palm and thumb (PT)

Figure 8: Samples of the 8 poses known to the system (in the brackets we report abbreviations used in the text).

periment by using observations from 3 subjects to gather the training set (on which we performed model selection), while keeping one subject out for testing.

We compare the accuracies of a multi-class K-NN classifier that considers all the allowed poses, with our proposed method, in which classification is guided by the finger detector. In both cases, we selected the best number K of nearest neighbors on the training set and then reported the performances on the test set, averaging with respect to the 4 possible configurations. The comparison is reported in Tab. 1. Without finger detection, the best result on the training set is achieved with K=1, while in presence of the detection K is best set to 3. For a fair comparison, we also report the result of the combination of finger detector and classifier with K=1. The results clearly speak in favor of our approach.

Even from a computational standpoint we may appreciate the benefit of using the step of finger detection to guide the selection of an appropriate classifier. As shown in Fig.9, as the dimensionality of the training set grows (we simulate the use of growing training sets by replicating the data so to allow more comparisons), it substantially reduces the amount of comparisons among the available data, keeping the temporal complexity under control.

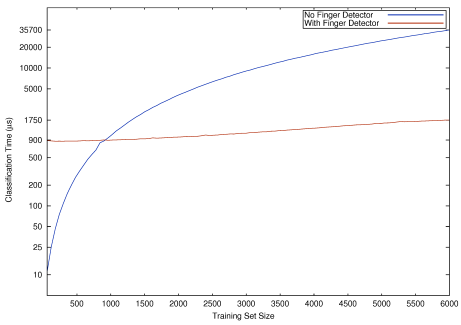


Figure 9: Comparisons of the computational time without and with finger detection as the size of training set grows.

To favor the comparison with related works, we also evaluated our approach on a publicly available dataset (http://eeeweba.ntu.edu.sg/computervision/people/home/renzhou/HandGesture.htm), collected from 10 subjects while performing 10 hand poses. The dataset has been adopted in (Ren et al., 2013), where a part-based hand gesture recognition method has been proposed, based on the use of kinect data to segment the hand region. In such an approach each hand pose is actually interpreted as a static gesture. We started by detecting the hand region combining RGB data with depth information (after having aligned the two) ending up with sometimes very noisy segmentations. We decoupled here the problems of detecting fingers from the actual pose classification, thus we first ran our finger detector and compare in Tab. 2 our performances with what reported in (Ren et al., 2013). Since in the paper only the final results for pose classification have been specified, we simply computed the overall accuracy of groups of poses characterized by the same number of stretched fingers. As noticed, our method performs better. We also consider the full hand pose classification problem, reporting in Tab. 3 the comparisons with other approaches in terms of mean accuracy and mean running time. Results of shape context (Belongie et al., 2002) and skeleton matching (Bai and Latecki, 2008) are extracted from (Ren et al., 2013). As it can be noticed, our results are in line, even thought slightly below, with (Ren et al., 2013). Also our method performs far better in terms of average running time (900μs). Although we report for completeness the full table of comparisons, our approach should be more fairly compared to other methods purely based on instantaneous shape representation and matching (as (Belongie et al., 2002; Bai and Latecki, 2008)). With respect to them, our approach shows considerably higher performances.

4.2 Experiments on Gesture Recognition

We now report the experimental analysis to validate the gesture recognition procedure. Following the nomenclature adopted in Sec. 3, the values allowed for `Start_Pose` ad `Final_Pose` fields in the gesture signature are to be selected from the set of poses included in the dataset we collected in-house, i.e. { *1, 2, 3, 4, 5, Pa, Pu, PT, Any, Same* }.

Fig. 10 shows visual representations of the gestures we consider in this experimental analysis, whose linguistic representations are the following:

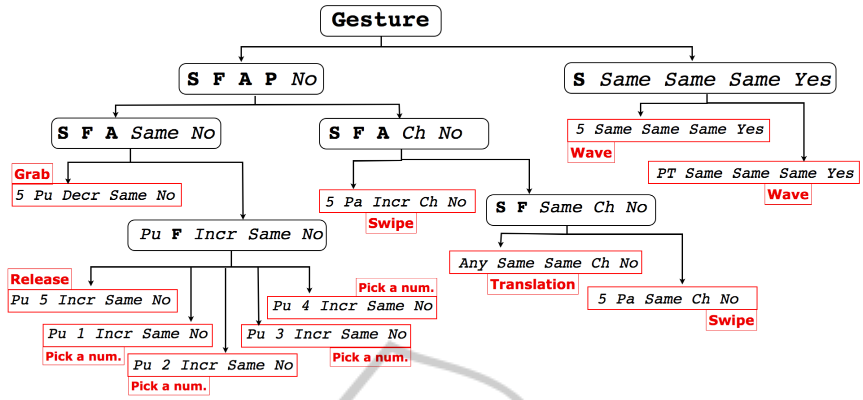
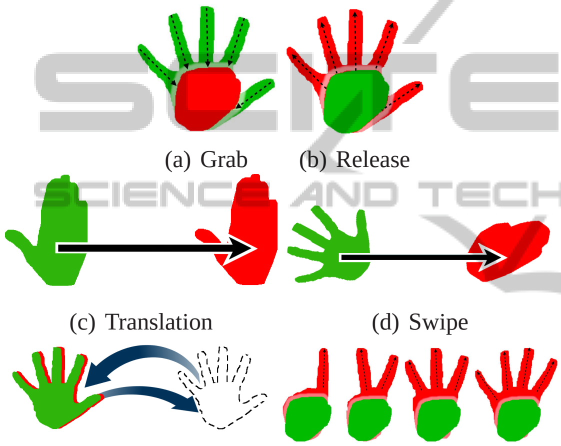


Figure 12: A visual representation of the syntactic parser recognizing the gestures representations we consider.



(a) Grab (b) Release

(c) Translation (d) Swipe

(e) Wave (f) Pick a number

Figure 10: A visual representation of the considered gestures (starting pose is green, final pose in red).

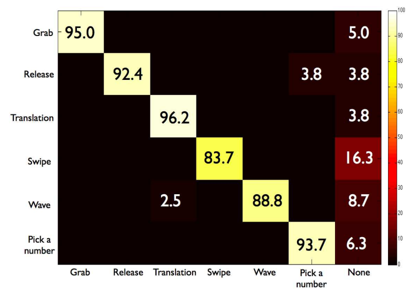


Figure 11: Accuracy (%) of gesture recognition (91.6 on average).

`Grab`	*5 Pu Decr Same No*
`Release`	*Pu 5 Incr Same No*
`Translation`	*Any Same Same Ch No*
`Swipe`	*5 Pa Same*|*Incr Ch No*
`Wave`	*5* | *PT Same Same Same Yes*
`Pick a number`	*Pu 1*|*2*|*3*|*4 Incr Same No*

A formal grammar G generating the gesture representations $\mathcal{G}$ may be the following:

s ::= **Gesture**
N ::= **Gesture** | **SFAP** | **SFA** | **SF** | **S**
Σ ::= *1* | *2* | *3* | *4* | *5* | *Pu* | *Pa*
PT | *Same* | *Any* | *Ch* | *Yes* | *No*
P ::=
Gesture $\rightarrow$ **SFAP** *No* | **S** *Same Same Same Yes*
SFAP $\rightarrow$ **SFA** *Same* | **SFA** *Ch*
SFA $\rightarrow$ *5 Pa Incr* | **SF** *Same* |
5 Pu Decr | *Pu* **F** *Incr*
SF $\rightarrow$ *Any Same* | *5 Pa*
S $\rightarrow$ *5* | *PT*
F $\rightarrow$ *1* | *2* | *3* | *4* | *5*

Now, the gestures $\mathcal{G} \subset \mathcal{L}(G)$ are identified by the parser we report in a graph-like fashion in Fig. 12

We collected videos of 4 subjects performing 20 replicates of each gesture. To take into account the variability of the dynamic events, subjects were asked to apply some variation to the movements, e.g. changing the direction or the adopted hand. Also, we acquired the videos in two different indoor environments (a living room and a classroom), to account for contextual variations (light changes, background variability, ...).

Fig. 11 reports the overall confusion matrix. The method is proved to be very robust to the variability among the users movements, even if a few failures have been experienced – as in particular for *swipe* gestures. Not surprisingly, it is the less constrained gesture, thus it is more likely it deviates with respect to the annotation, especially when the user is not enough familiar with the system.

5 CONCLUSIONS

This paper considered the problem of recognizing static and dynamic human gestures, with particular reference to the use for designing Natural User Interfaces. Moving regions extracted from image sequences acquired with a webcam are first processed to detect the presence of stretched fingers with a new method based on iteratively analysing the distance transform of the hand region. The result guides the classification of a set of known hand poses, which is based on a family of classifiers related to the hand configuration. Gesture recognition is

achieved using a syntactic approach making use of linguistic gestures annotation formalized with a generative grammar.

We experimentally validated our method and showed how it compares favorably with other approaches, while performing significantly better from a computational standpoint.

As a first prototypical application, we developed a picture browsing (see a screenshot in Fig. 1) in which all the available functions are enabled by only the use of hands.

Future improvements will be devoted to attenuate the constraints required by the system (e.g. to overcome problems for detecting hands). A straightforward development refers to extending the system so to enroll two-handed gestures. From the standpoint of the computational tools, the K-NN classifier can be replaced with more refined machine learning methods, that may be beneficial especially as the number of known hand poses increases. Also, users evaluations will be taken into account to judge the ease in the use of the interface. These aspects are object of current investigations.

REFERENCES

Bai, X. and Latecki, L. (2008). Path similarity skeleton graph matching. *Trans. on PAMI*, 30(7):1282–1292.

Belongie, S., Malik, J., and Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *Trans. on PAMI*, 24(4):509–522.

Berg, M. d., Cheong, O., Kreveld, M. v., and Overmars, M. (2008). *Computational Geometry: Algorithms and Applications*.

Bobick, A. F. and Wilson, A. D. (1997). A state-based approach to the representation and recognition of gesture. *Trans. on PAMI*, 19:1325–1337.

Borgefors, G. (1986). Distance transformations in digital images. *Comput. Vision Graph. Image Process.*, 34(3):344–371.

Chen, Z., Kim, J., Liang, J., Zhang, J., and Yuan, Y. (2014). Real-Time Hand Gesture Recognition Using Finger Segmentation. *The Scientific World Journal*.

Choi, J., Park, H., and Park, J. (2011). Hand shape recognition using distance transform and shape decomposition. In *ICIP*, pages 3605–3608.

Chomsky, N. (1956). Three models for the description of language. *Trans. on Information Theory*, 2:113–124.

Danielsson, P. (1980). Euclidean distance mapping. In *Comp. Graph. and Image Proc.*, pages 227–248.

Dardas, N. and Georganas, N. D. (2011). Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques. *Trans. on Instrumentation and Measurement*, 60(11):3592–3607.

Dasarathy, B. V. (2002). Handbook of data mining and knowledge discovery. chapter Data Mining Tasks and Methods: Classification: Nearest-neighbor Approaches, pages 288–298.

Droeschel, D., Stuckler, J., and Behnke, S. (2011). Learning to interpret pointing gestures with a time-of-flight camera. In *Int. Conf. on HRI*, pages 481–488.

Fang, Y., Wang, K., Cheng, J., and Lu, H. (2007). A real-time hand gesture recognition method. In *Int. Conf. on Multimedia and Expo*, pages 995–998.

Ghotkar, A. S. and Kharate, G. K. (2013). Vision based real time hand gesture recognition techniques for human computer interaction. *Int. Jour. of Computer Applications*, 70(16):1–8.

Hu, M.-K. (1962). Visual pattern recognition by moment invariants. *Trans. on Inf. Theory*, 8(2):179–187.

Ivanov, Y. A. and Bobick, A. F. (2000). Recognition of visual activities and interactions by stochastic parsing. *Trans. on PAMI*, 22(8):852–872.

Joo, S.-W. and Chellappa, R. (2006). Attribute grammar-based event recognition and anomaly detection. In *CVPRW*, pages 107–107.

Paulraj, Y. e. a. (2008). Extraction of head and hand gesture features for recognition of sign language. In *Inter. Conf. on Electronic Design*.

Rautaray, S. and Agrawal, A. (2012). Vision based hand gesture recognition for human computer interaction: a survey. *Artificial Intelligence Review*, pages 1–54.

Rauterberg, M. (1999). From gesture to action: Natural user interfaces.

Ren, Z., Yuan, J., Meng, J., and Zhang, Z. (2013). Robust part-based hand gesture recognition using kinect sensor. *Trans. on Multimedia*, 15(5):1110–1120.

Sato, Y., Kobayashi, Y., and Koike, H. (2000). Fast tracking of hands and fingertips in infrared images for augmented desk interface. In *Int. Conf. on Automatic Face and Gesture Recognition*, pages 462–467.

Singh, S. K., Chauhan, D. S., Vatsa, M., and Singh, R. (2003). A robust skin color based face detection algorithm, tamkang. *Jour. of Science and Engineering*, 6:227–234.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *CVPR*, volume 1, pages I–511–I–518.

Zariffa, J. and Steeves, J. (2011). Computer vision-based classification of hand grip variations in neurorehabilitation. In *ICORR*, pages 1–4.

Zivkovic, Z. (2004). Improved adaptive gaussian mixture model for background subtraction. In *ICPR*, volume 2, pages 28–31.