CHAPTER 4

Compositional and Hierarchical Semantic Frameworks for Hand Gesture Recognition

G. Simion, C. David, C. Caleanu and V. Gui

This chapter presents some new approaches in hand gesture recognition from monocular and from 3D images. After introducing the main trends from literature, the chapter addresses the hand gesture recognition problem in a compositional framework. The ability of compositional methods to capture and extract semantic information from selected salient parts of the images is demonstrated for the hand gesture application. Performances on images with static hand gestures executed on uniform backgrounds rival the state of the art, while living a lot of room for further optimization and improvement. The second work reported in this chapter is focused on using multiple cues to overcome difficult problems arising when dynamic gestures are executed in front of heterogeneous backgrounds, with camouflage and sudden illumination changes. Ideas from robust estimation are integrated in the proposed approach. Finally, some preliminary results of hand gesture recognition obtained with images generated by 3D time of flight image sensors, presumed to become prevalent in the near future, are presented.

G. Simion, C. David, C. Caleanu, V. Gui
Faculty of Electronics and Telecommunications, Politehnica University Timisoara
Timisoara, Romania
e-mail: {georgiana.simion,ciprian.david,catalin.caleanu, vasile.gui}@upt.ro

Editor: A. Chaudhary, *Recent Trends in Hand Gesture Recognition*
DOI: 10.15579/gcsr.vol3.ch4, GCSR Vol. 3, pp.75-114, 2015

4.1 Introduction

The interaction between people and the surrounding devices has changed dramatically in the last decades. In the beginning there were knobs, buttons, and keyboards. Nowadays we have touchscreens and multiple sensors interfaces. Lately, the interaction with smart devices became a common thing. We have smart computers, smart phones, smart TVs, smart cameras, smart cars, smart homes etc. Generally, this is the result of the hardware evolution and the development of new software algorithms. The arising question is: how should we interact with all these smart devices? The common sense answer is: as natural as possible. The Natural User Interface (NUI) principle is that machines should interact more like people do using gestures, speech and other means. In this context, the use of hand as a direct input became an appealing interfacing method. The new trend is to develop touchless interfaces, which use no gloves or other sensors connected to the hand.

The hand gesture application domains are varied and new application areas emerged in the last decades. One of these is vehicle telematics. In [1], the authors study the use of hand gestures to control telematics. A secondary task is to reduce the driver distraction and allow him to focus on the primary task of driving. A study result of Virginia Technology Transportation Institute concluded that 80% of all crashes involved driver distraction in the three seconds prior to the incident [19]. Mobile phones and telematics secondary tasks were associated with the highest frequency of distraction-related crashes and near-crashes. The crashes risk may be reduced and substantial safety benefits can be added by using gesture recognition systems. The driver no longer has to take his eyes off the road to operate conventional secondary controls [46, 47].

In desktop and tablets PC applications hand gestures can replace the classical mouse and keyboard [58] or the touchscreens. Pen-based gestures are used in desktop computing tasks, like manipulating graphics and editing documents [14]. Mouse gestures are also used for various applications, including web browsing tasks. Recently eyeSight introduced gesture recognition Technology for Android Tablets, Windows-based Portable Computers and iPads. The latest release allows the iPads users to browse the "*. pdf" documents and eBooks with simple swipes of the hand.

The medical field is one of the newcomers which could potentially rely on hand gesture recognition technologies. For example, the need of sterile interfaces could push forward the development of these computer based technologies. Hand gesture based interfaces, that use no direct touch may be used to replace touchscreens from hospital operating rooms. A system that allows the doctors' hand to remain sterile while manipulating digital images during surgery is proposed by [61]. Another advantage of the system is closely related to the fact that doctors were able to stay in place during the entire intervention. It was no longer needed to move to the main control wall since the commands were performed using hand gestures.

In [24], one of the earliest systems which enables gesture-based interaction between surgeon and operation room equipment is proposed. The developed system is a non-contact mouse and allows surgeons to perform standard mouse functions like pointer movement and button presses with hand gestures.

Hand gestures are used in virtual and augmented reality applications to achieve

natural human computer interaction. By using hand gestures, one can manipulate realistically virtual objects [34] and assembly them [49], or can navigate around the 3D information space such as the nodes of a graph [45], arranging them and choosing the focus nodes. In many augment reality application markers are used. These markers are patterns printed on objects, easily trackable using computer vision. In [10], the authors used markers and hand gestures to select and manipulate objects on an AR display.

In robotics, the hand gestures can control the arm and the hand movement of a robot to pick up and manipulate real world objects and also to guide the robot movement through the real world [23, 37, 62].

Nowadays the hand gestures are used for remote control of TV sets and DVD players [35]. In [32] a report regarding the possibility of using gestures to control domestic appliances is presented. Other application fields for hand gesture are computer games [31] and sign language. Specifically, sign language for the deaf has received significant attention in the gesture literature [39, 56, 60], etc.

Hand gesture recognition approaches have been usually divided into two main categories: model based and view based. Model based approaches [17, 57] use articulated models inspired from computer graphics to estimate the 3D hand pose, while view based approached rely on pattern classification techniques to derive the hand pose information based on features extracted from the image. In this chapter, we present three approaches from the second category ,developed by the authors and discuss the current trends.

This chapter is organized as follows: In Section 4.2 we give a short overview regarding the features used for hand gesture recognition and tracking, while in Section 4.3 we present contributions in hand gesture recognition from monocular cameras. Section 4.4 describes view based approaches using 3D cameras, including some results of the authors and some conclusions are drawn.

4.2 Features for Hand Gesture Recognition

4.2.1 Motion Cues

Motion is an important cue in many computer vision applications. There is a huge amount of work devoted to motion estimation and it is out of the scope of this section to review it. Background subtraction is a widely used and powerful method in video surveillance, which has been also applied successfully in several HCI applications with controlled lighting or indoor environments. In spite of its limitations, background subtraction can be a valuable tool in a multi-cue hand tracking application. The basic assumption in foreground/background segmentation is that the background is more stable than the foreground. While being generally more stable than the foreground, the background is far from being constant. Static background appearance varies with illumination changes and shadows cast by moving objects. Objects removed from the background or new objects placed in the scene also cause changes of the background. Outdoor scenes often contain dynamic backgrounds, such as water or waving trees, but dynamic backgrounds can be also encountered indoors (computer screens, escalators etc.). Due to these problems, after initial estimation, the background model needs

continuous updating during its use. Background updating can be successful in coping with gradual changes, but it fails in responding instantly to sudden changes, such as those produced by moving clouds or switching bulbs. There are several methods to reduce the effect of fast illumination changes. One major idea is to use texture information instead of intensity and/or color. Such solutions were discarded since texture change based segmentation produces less accurate contours of the extracted objects. Another assumption which is currently used in motion detection is that all changes in the image are produced by moving objects, which is generally useful, although not always true.

Background subtraction works by comparing on a pixel by pixel level the current frame with a reference background model. All significant changes of the current frame with respect to the background model are attributed to foreground objects. Within the scope of this work, it is important to note that, fortunately, all the mentioned factors produce false positive rather than false negative foreground detections. In this sense, background subtraction can be viewed as a method to fast discard regions which do not belong to the tracked object. Most of the false positive detections can be eliminated easily by making use of the additional color and shape cues. A more difficult problem is caused by skin colored background areas. Foreground detection in such areas is likely to fail (camouflage problem), thus producing false negative pixels in foreground detection. To deal with this problem, one can use edge detection in all skin colored regions. This makes possible to detect hands, even when they move over skin colored regions, where foreground segmentation may fail or, for example, when the tracked hand moves over a skin colored foreground region, like the face of a person. Relevant and detailed surveys on background extraction methods can be found in [5, 6]. Due to the real time constraint, one should choose a fast approach for extracting motion cues, like the one based on codebooks, proposed by [29]. One important advantage of the sparse modeling over purely probabilistic approaches is the ability to model dynamic backgrounds, thus preventing potentially false positive detections caused by such backgrounds.

4.2.2 Skin Tone Cues

Since the object of this chapter is hand detection a powerful cue can be represented by the skin tone. A skin tone detector having good results in different illumination conditions is desired. It is important that the skin tone detector be invariant to the user, meaning that any type of skin zone is detected regardless of the users' skin color. The real time aspect is of great importance for a human-computer interface, too. So, the choice of skin tone detector has to be made with respect to these aspects. There are a lot of models proposed in the dedicated literature regarding skin tone detection. Many of them are based on training an elaborated model for the skin. Some examples of approaches presenting good results are based on Gaussian mixture models [6], beta mixture models [5], nonparametric models [29], or based on the more recent random forest classifier [28]. The main drawback of these models is that they lack in terms of time efficiency, making them not suitable for a real time constrained framework. A more suitable approach could be the one proposed by [11]. The method uses a reduction of space dimensionality from the classical RGB space to a 1D space. The

skin cluster is obtained by imposing predefined thresholds on an error signal issued from the difference between the luminance and the maximal non-red component. The space dimensionality reduction is implicit with the use of this error signal. The skin cluster thresholds are chosen optimally after extensive testing of the approach on a large image database. One advantage of this approach is that the skin tone samples used to compute the thresholds are chosen from the entire range of races and extreme variations of lighting conditions.

4.2.3 Edge Cues

Edges play an important role in vision, as demonstrated by our ability to interpret line drawings. To a certain degree, edge maps are invariant to illumination changes. They can be extracted at a low computational cost and contain essential shape information. Like other cues, edge detection has its own problems. Edge detection produces many spurious edges due to its sensitivity to noise and it fails to detect blurred edges. This is a potential problem, considering motion blur occurring with fast hand motion and low speed cameras. The reason to introduce edge cues in such a framework is to alleviate problems caused by camouflage.

4.2.4 Invariant Features

In [3, 12, 36] Haar like features are used for the task of hand detection. Haar like features focus on the information within a certain area of the image rather than each single pixel. To improve classification accuracy and achieve real-time performance, AdaBoost learning algorithm that is able to adaptively select the best features in each step and combine them into a strong , can be used. The training algorithm based on AdaBoost learning algorithm takes a set of "positive" samples, which contain the object of interest and a set of "negative" samples, i.e., images that do not contain objects of interest.

In [13] the ARPD descriptor (Appearance and Relative Position Descriptor) is proposed. This descriptor includes color histogram, relative position information, and SURF [4]. The process of constructing ARPD includes two steps: extracting SURF keypoints and color histogram from images, and computing relative-position information of every keypoint within images. The relative-position information is also included as part of ARPD. The ARPD was used in the BoW representation. The BoW was used to detect and recognize hand posture based on sliding-window framework. To meet real-time request, several approaches were proposed to speed up hand posture recognition process.

In [20], Maximally Stable Extremal Region (MSER) detector and color likelihood maps are used for hand tracking. Such a combination allows performing repeated figure/ground segmentation in every frame in an efficient manner. The MSER detector is one of the best interest region detectors in computer vision [41]. MSER detection is mostly applied to single gray scale images, but the method can be easily extended for analysis of color images by defining a suitable ordering relationship on the color pixels. In general the MSER detector finds bright connected regions which consequently have darker, values along their boundaries. The set of MSERs is closed under continuous

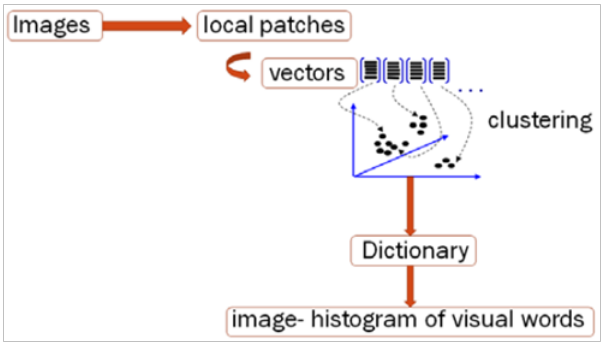


Figure 4.1: BoW representation.

geometric transformations and is invariant to affine intensity changes. Furthermore MSERs are detected at all scales. Therefore, due to these properties, MSER detection is suited for segmentation purposes.

In [16] Bag-of-Words representation (BoW) and SIFT features are used. In a typical BoW representation, "interesting" local patches are first identified from an image, either by densely sampling, or by an interest point detector. These local patches, represented by vectors in a high dimensional space, are often referred to as the key points. The bag-of-words methods main idea is to quantize each extracted key point into one of the visual words, and then represent each image by a histogram of visual words (see Fig.(4.1)). A clustering algorithm is generally used to generate the visual words dictionary. In [16] k-means algorithm has been used for clustering. A multi-class SVM was used to train the classifier model. In the testing stage, the key points were extracted from every image captured from the webcam and fed into the cluster model to map them with one (Bag-of-words) vector, which is finally fed into the multi-class SVM training classifier model to recognize the hand gesture.

4.3 The Proposed Approaches

4.3.1 The Compositional Approach to Hand Gesture Recognition

Bag of Words methods and compositional methods become more and more popular in hand gesture recognition. These techniques have been studied in many diverse fields such as linguistics, logic, and neuroscience, but compositionality is especially evident in the syntax and semantics of language, where a limited number of letters scan form a huge variety of words and sentences. In computer vision these techniques are used in the context of a general problem: categorization. Using these techniques we address also to the semantic gap that exists between the low level features and high level representations. The hand posture is no longer modeled as a whole. These characteristic regions are assembled to form compositions; these compositions at their turn can be grouped in compositions of compositions and so on. The invariant features

allowed us to model the hand as collection of characteristic parts. Key points or characteristic regions are extracted. Using such features the hand gesture is split in simpler parts which are easier to recognize. This approach has major advantages: even if some parts are missing gestures still can be recognized, so there are robust to partials occlusions, changes in view point and considerable deformations.

In our work we used a compositional technique for hand posture recognition. A hand posture representation is based on compositions of parts: descriptors are grouped according to the perceptual laws of grouping to obtain a set of possible candidate compositions. These groups are a sparse representation of the hand posture based on overlapping subregions.

The detected part descriptors are represented as probability distributions over a codebook which is obtained in the learning phase. A composition is a mixture of the part distributions. From all candidate compositions, relevant compositions must be selected. There are two types of relevant compositions: those compositions that occur frequently in all classes and also those which are specific for a class. The category posterior of compositions is learned in the training phase, and it is a measure of relevance. The entropy of the class posterior helps to discriminate between classes. A cost function is obtained by combining the priors of the prototypes and the entropy. The process of recognition is based on bag of composition method, where a discriminative function is defined.

Even if the proposed method is a general one, for different applications it is still important what features are used for the sparse hand posture representation.

4.3.1.1 Feature extraction

The first question according to compositional technique in our case is how the hand can be represented in order to be decided which image locations had to be captured and which to dispose of. The main idea is that each hand posture can be described by: the V shapes between the fingers when these are apart, the curve shapes which correspond to the fingertips and the straight lines for the finger length. Each hand pose can be defined as a combination of these shapes. Based on the relations among them, the hand pose can be recognized. It is important how these shapes are oriented and which their relative position to each other is. The second question is how these relevant image regions can be represented.

The RGB hand posture image is converted to a gray scale image, and then the Canny edge detector is used in order to extract the hand contours. Salient image locations are detected by using Harris interest point detector on hand contours. The Harris interest point detector is used on hand contours in order to have a low computational cost, and edges are able to capture that information which is enough and useful for our brain-view processor to recognize the object. Quadratic patches of size 20×20 pixels are extracted around each Harris interest point to capture discriminative local information. The patch size is chosen so that it captures the fingertip. For each extracted patch, its correspondent in the RGB image is searched and a two bin color histogram (skin-non skin) is extracted. In this work the goal of the 2 bin color histogram is to detect different types of regions around the interest point assuming that the background is extracted.

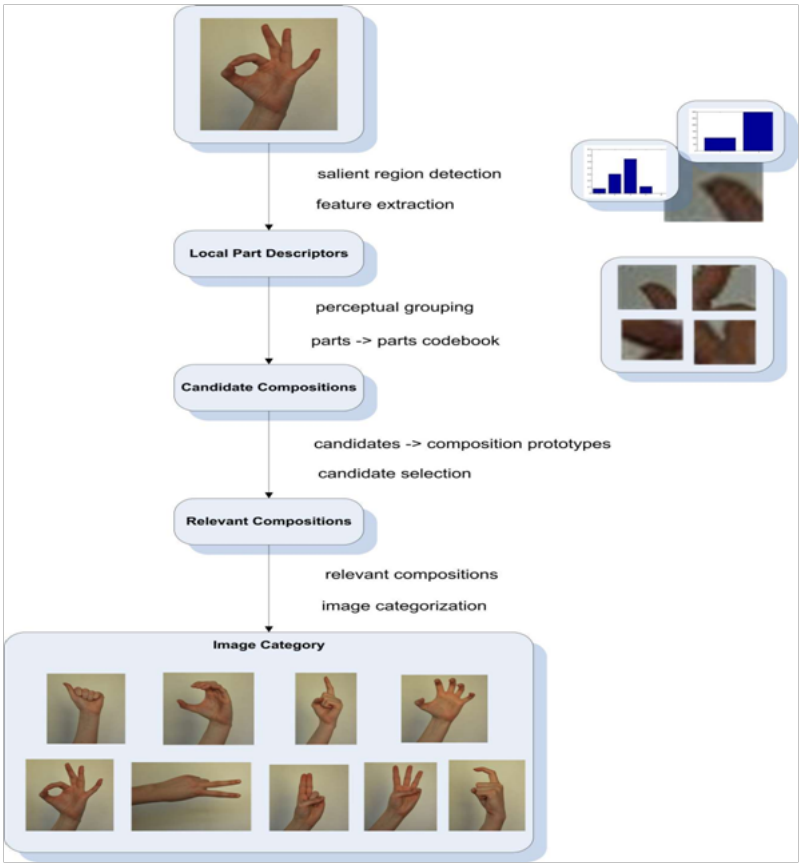
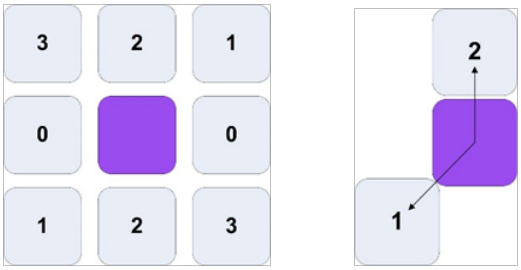


Figure 4.2: Compositional model for statistical pattern recognition.



(a) (b)

Figure 4.3: (a) the 4 orientations considered, (b) contour points contribution to histogram.

The contour orientations histogram with four bins is also extracted. Using Canny edge detector the obtained contours are thin and each contour point contribute to the histogram with one, two (as it can be seen in Fig.(4.3b)) or more local directions.

We define:

$$N(i) = \begin{cases} 1, & EDGE = True \\ 0, & ELSE \end{cases} \tag{4.1}$$

The orientation of a contour point is defined:

$$O_{x,y}(i) = \begin{cases} i, & N(i) = 1 \\ 0, & ELSE \end{cases} \tag{4.2}$$

The histogram is computed according to Eq.(4.3)

$$h(i) = \sum_{x,y \in region} O_{x,y}(i) \tag{4.3}$$

Then the relative direction of the interest point is computed similarly with contour orientation histogram. For the same patch the number of contour points is also extracted.

The resulting eight parameters extracted from a patch are used to form a feature vector, $\mathbf{e}_i$. It is important to remark the small dimension of the feature vector, which is eight.

Hand posture representation is based on compositions of parts, more precisely patches around a Harris interest point, described by a feature vector $\mathbf{e}_i$. Based on the features vectors $\mathbf{e}_i$ from all training images, a codebook with relevant features for all classes is obtained using the k-means clustering algorithm. The codebook is subsequently used in order to assess the similarity of extracted image features to learned classes of relevant features. The feature classes generated by the clustering algorithm are not associated with the hand posture classes. These are used to generate an alternative representation of image parts as presented in the next sections.

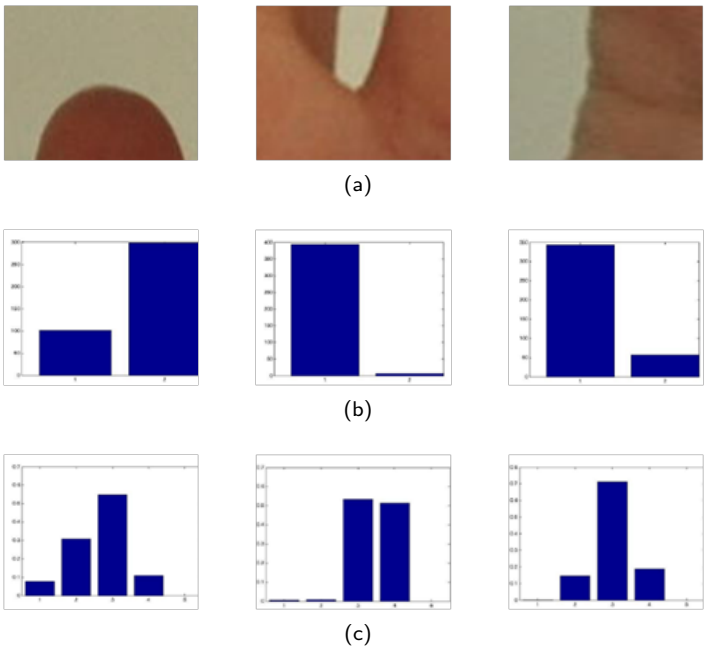


Figure 4.4: Some examples: (a) region type, (b) color histogram with 2 bins and (c) orientation histogram with 4 directions.

One of the parameters of the k-means clustering algorithm is represented by the number of clusters k, which was set to five. The main reason why the number of clusters is set to five is related to the types of patches detected in an image. Actually there are five types of patches: patches that have more skin and less background region, patches with more background and less skin region, some patches have the same percentage of skin and background, there are patches that may have only background, while some patches may have only skin regions.

According to the Gibbs distribution law, the feature assignment random variable, F_i, is given by Eq.(4.4).

$$P\left(F_i = v|\mathbf{e}_i\right) = Z\left(\mathbf{e}_i\right)^{-1} \exp\left(-d_{v,\sigma}\left(\mathbf{e}_i\right)\right) \tag{4.4}$$

$$Z\left(\mathbf{e}_i\right) = \sum_v \exp\left(-d_{v,\sigma}\left(\mathbf{e}_i\right)\right) \tag{4.5}$$

$$d_{v,\sigma}\left(\mathbf{e}_i\right) = \left\|\mathbf{e}_i - \mathbf{a}_v\right\|^2 \tag{4.6}$$

where F_i is a feature assignment random variable, $P\left(F_i = v|\mathbf{e}_i\right)$ is the probability of feature vector $\mathbf{e}_i$ to belong to the class defined by the prototype vector $\mathbf{a}_v$, $d_v\left(\mathbf{e}_i\right)$ is the Euclidian distance of a measured feature $\mathbf{e}_i$ to a centroid $\mathbf{a}_v$ of class and is a normalization factor. Equation4.4 is evaluated for all centroids $\mathbf{a}_v$ and the results for a feature point described by $\mathbf{e}_i$ are grouped in a part distribution vector:

$$\mathbf{d}_i = \left(P\left(F_i = 1|\mathbf{e}_1\right), ..., P\left(F_i = k|\mathbf{e}_i\right)\right)^T \tag{4.7}$$

In order to form a higher level of abstraction, image parts are grouped into compositions. In order to decide which parts should be grouped to form the candidate compositions, the principles of perceptual organization are used. To this end, all detected local parts from an image, represented by their part distribution vector, are grouped with their neighbors that are not farther away than N pixels. This grouping principle follows the principle of perceptual organization from Gestalt laws, more precisely the grouping principle of proximity [9]. In this work the number of pixels N is 25. This number depends on the types of objects and compositions that one wants to form, the number of interest points detected in an image, the number of objects present in an image and also by the image resolution. In [44] this number is between 60-100 pixels.

Candidate compositions are represented as mixtures of the part (feature point) distributions as defined in Eq.(4.6). If $\Gamma_j = \{\mathbf{e}_1, ..., \mathbf{e}_{mj}\}$denotes the grouping of parts represented by $\mathbf{e}_1, ..., \mathbf{e}_{mj}$, and $\mathbf{d}_1, ..., \mathbf{d}_{mj}$, (where m is the number of vectors which generate the candidate composition), compositions are then represented by the vector valued random variable $\mathbf{G}_j$ which is a bag of parts with the particular values given by:

$$\mathbf{g}_j = \frac{1}{m}\sum_{i=1}^{m}\mathbf{d}_i = \frac{1}{m}\sum_{i=1}^{m}\left(P\left(F_i = 1|\mathbf{e}_1\right), ..., P\left(F_i = k|\mathbf{e}_i\right)\right)^T \tag{4.8}$$

In Eq.(4.8), the number of constituents, $m_j = |\Gamma_j|$, is not predefined and can be different for each composition. It depends on how many parts the grouping algorithm can combine into composition in a certain region of an image. Note that the representation of a composition depends on the type of constituent parts and not on the number of parts. A composition is represented by the vector $\mathbf{g}_j$, which can be thought of as the average distribution of its parts over the codebook containing relevant parts for recognition. This model is also robust with respect to variations in the individual parts.

4.3.1.2 Learning compositions

On the set of all compositions that can be formed, a selection of relevant compositions must be performed in order to have the discriminative ones and to discard the clutter. The relevant compositions must reflect a trade-off between generality and singularity. The goal is to learn a small number of compositions so that estimating class statistics on the training data becomes feasible. There are compositions which are present in many classes and there are compositions that help to discriminate sets of classes from one another, not necessarily one class from all the other.

First, compositions which are specific for a large majority of hand posture classes are learned. These compositions should be shared among many classes. In order to do this, in the learning phase, all composition candidates found in all the training images, represented by average distribution vector of parts, $\mathbf{g}_j$, are clustered using once more k-means clustering. Let $\pi_i \in \Pi$ be the composition prototypes found by clustering.

Then the prior assignment of the probabilities of candidate compositions to clusters $P(\pi_i)$are computed by using the Gibbs distribution:

$$P(\pi_i = \Pi|\mathbf{g}_j) = Z(\mathbf{g}_j)^{-1} \exp(-d_{\Pi,\sigma}(\mathbf{g}_j)) \tag{4.9}$$

$$Z(\mathbf{g}_j) = \sum_{\Pi} \exp(-d_{\Pi,\sigma}(\mathbf{g}_j)) \tag{4.10}$$

In the second stage, relevant composition prototypes for specific classes are selected. Those prototypes help to distinguish between classes. To this end, the class posteriors of compositions must be estimated. In order to estimate the class posteriors of compositions a Bayesian approach was used:

$$\begin{gathered} P(c|\Gamma_j) = \frac{P(\Gamma_j|c)P(c)}{P(\Gamma_j)} = \frac{P(\Gamma_j|c)P(c)}{\sum_c P(\Gamma_j|c)P(c)} \\ P(c|\Gamma_j) \approx \frac{P(\Gamma_j|c)}{\sum_c P(\Gamma_j|c)} \end{gathered} \tag{4.11}$$

where $c \in \wp$, $\wp$ is the set of all category hand postures. We assume that $P(c)$ are equal, all classes are used with the same probability. The class posterior is used to calculate the relevance of a composition for discriminating hand postures. In order to find a relevance measure, the class posteriors of compositions are learned from the training data. The relevance of a composition for discriminating hand postures is then estimated by the entropy of its class posterior:

$$H(P_{\Gamma_j}) = -\sum_{c \in \wp} P(c|\Gamma_j) \log(P(c|\Gamma_j)) \tag{4.12}$$

The entropy is used as a measure of discriminative relevance; since entropy measures how uniformly a random variable is distributed the entropy should be minimized.

In order to measure the total relevance of a compositional prototype, a cost function is defined. The cost function combines the prior assignment probabilities of clusters and the entropy, so it combines the reusability criterion with the criterion that measures the ability of compositions to discriminate hand postures from one another. The resulting cost function defined guides the selection of relevant compositions.

In [44] the following cost function was proposed:

$$S(\pi_i) = -\log(P(\pi_i)) + \lambda H(P_{\pi_i}) \tag{4.13}$$

Both constituents of the cost function should be normalized to the same dynamic range, giving rise to an additional additive constant that can be discarded and to the parameter λ. Parameter λ defines the balance between the two conflicting demands: generality and specificity. Its value proved to be very important in practice. Parameter λ reflects the way the generality and specificity combines in order to select the relevant prototypes which determinate further the relevant composition used to describe an image. In this approach, the parameter is estimated by using the inter-quartile range (IQR) which is equal to the difference between the third and first quartiles. The proposed robust method for estimating parameter is presented in Eq.(4.14).

$$\lambda = \frac{IQR\left(P\left(\pi_i\right)\right)}{IQR\left(H\left(P_{\pi_i}\right)\right)} \tag{4.14}$$

From the set of all compositional prototypes a set of relevant composition prototypes is established through minimization of Eq.(4.13). For all composition prototypes π_i, the cost function is computed and a set of r relevant composition prototypes is selected. The distance between all compositions and all relevant composition prototypes and irrelevant compositional prototypes is computed. The image is represented by those candidate compositions which are closer to the relevant prototypes than any irrelevant ones.

4.3.1.3 Training step

For all training images the features vectors $\mathbf{e}_i$ are extracted and k-means is performed in order to generate the feature codebook, which is the first product of the training step. Based on feature vectors and the feature codebook, the candidate compositions are extracted and modeled with their distribution vectors over the feature codebook.

Candidate compositions from all test images are clustered using one more time k-means, and the resulted composition prototypes are used to form the composition codebook. Based on the cost function defined in Eq.(4.13), *relevant* composition prototypes are learned in the next stage. A set of r relevant composition prototypes is established. This set is obtained by selecting the prototypes π_i with minimal cost $S\left(\pi_i\right)$. Only those relevant compositions which are not farther away from the relevant composition prototypes than the irrelevant ones are retained.

Each image from the training set is described by those candidate compositions which are closer to the relevant prototypes than any irrelevant ones (these are the relevant compositions) and also by the relative rescaled position coordinates of the relevant compositions.

The hand position may vary from one image to another, so in order to get invariance to translation the relative coordinates are used. The relative position of the compositions is estimated using the median, not the mean because the median is more robust. These relative positions are rescaled by means of parameter α.

4.3.1.4 Hand posture recognition

The recognition part is done based on the bag of compositions method. For the new image, a set of composition vectors $\mathbf{h}_i$ is computed. These vectors consist of $\mathbf{g}_i$ distributions and relative, rescaled position coordinates of the relevant compositions. In order to get invariance to translation, the relative rescaled coordinates x_i, y_i, are used. Hand position is estimated using the median, not the mean because the median is less influenced by the maximum and minimum values from the set of coordinates and is more robust. Evaluation of the data set using median is good if half of the data are correct. For this application more than half of the data is correct because most of the compositions are generated from interest points located on hand and less from interest points found on background.

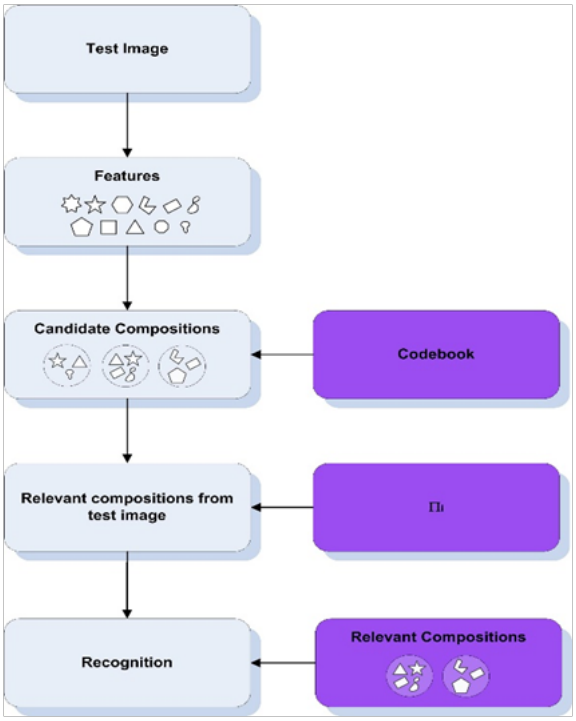


Figure 4.5: Work diagram for hand posture recognition.

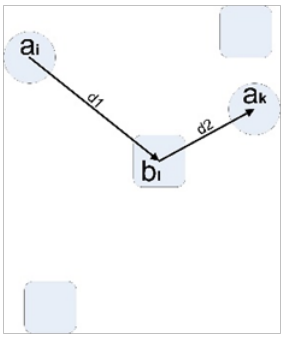


Figure 4.6: Work diagram for hand posture recognition.

The relative position is rescaled using the parameter α. The evaluation of the parameter α is a problem of feature extraction and depends on the data characteristics. Its value influences the space shape.

$$\mathbf{h}_i = \begin{bmatrix} x_i \\ y_i \\ \mathbf{g}_i \end{bmatrix} = \begin{bmatrix} \alpha x_r \\ \alpha y_r \\ \left(P\left(F_i = 1|\mathbf{e}_1\right), ..., P\left(F_i = k|\mathbf{e}_i\right)\right)^T \end{bmatrix} \tag{4.15}$$

where $x_i = \alpha\left(x - x_{median}\right) = \alpha x_r$ and $y_i = \alpha\left(y - y_{median}\right) = \alpha y_r$.

The range for x_r, y_r, is larger than the range of probabilities. Both compositions and their position should have similar importance because the hand posture is recognized based on types of compositions and their relative position one to another. The value of parameter α is learned based on the experimental data.

The classification of a new image which is described by vectors is not straight forward. The number of compositions that describe the testing image differs from the number of compositions which describes the images from the bag (each image from the bag might have different numbers of compositions). All components which describe an image can be seen as a vector; because the length of the vectors is not equal for all images it is not possible to use traditional classification methods.

The proposed classification method is inspired by point matching used in image registration, where two sets of points need to be registered and correspondence of points need to be formed. The two sets of points usually suppose different numbers of points. The minimum distance from a fixed point $a_i \in A$ found in set 1, to points $b_n \in B$ from set 2 (according to Fig.(4.6)) is shown in Eq.(4.16)

$$\min_{\forall n}\left(a_i, b_n\right) = d\left(a_i, b_I\right) \tag{4.16}$$

The minimum distance from point $b_I \in B$ to points from set 1 according to Fig.(4.6) is:

$$\min_{\forall n}\left(b_I, a_n\right) = d\left(b_I, a_k\right) \tag{4.17}$$

In Fig.(4.5) is can be seen that$d_1 \neq d_2$, where $d_1 = d\left(a_i, b_I\right)$ and $d_2 = d\left(b_I, a_k\right)$.

For each new image only the minimum distance from the training images compositions, to the test image composition $\mathbf{h}_i$ is computed $\min_{c_i} \left\| h_v^{k,q_v} - h_i^{c_i} \right\|$, then all these distances are sum and normalized according to Eq.(4.18). In the equation, v is the number of pictures per class, k is the class, q_v is the number of compositions from a class, i is the current image and c_i is the number of composition for the test image.

$$d\left(c, v_k\right) = \frac{1}{\#q_v} \sum_{q_v} \left(\min_{c_i} \left\| h_v^{k,q_v} - h_i^{c_i} \right\| \right) \tag{4.18}$$

$$d\left(c, k\right) = \arg\min_{v_k} \left(d\left(c, v_k\right) \right) \tag{4.19}$$

The reason why the distance from the test image compositions to training images is not computed is related to the fact that the testing image might have some compositions which are not specific for that class; it might have compositions as a result of some interest points detected on background. This is less likely to happen for training images. These distances are computed for all images.

The discriminant function used in the experiments from this work is defined as:

$$k_{opt} = \arg\min_{k} \left(d\left(c, k\right) \right) \tag{4.20}$$

4.3.1.5 Experimental results and conclusions

In order to prove the power of the compositional approach in hand posture recognition, two sets of hand gestures were used. The first one consists of nine classes of hand postures and the second one is represented by six classes, as it can be seen in Fig.(4.7) and Fig.(4.8).

For the first set of hand postures 30 training images per class are used. The first set of training images has as background a white wall. The first training set pictures are taken in natural conditions, no artificial light was added. For the first set the pictures were taken with Nikon D60 and the images have a resolution of 255×171 pixels. The number of composition prototypes is 20 and the number of relevant composition prototypes r, which conduct to the best result is equal to 19. The number of relevant prototypes is 19 because almost all compositions resulted from interest points detected on hand and just a few are the result of some points detected on background.

The second set of hand postures has six classes. This six hand postures are chosen by the considerate that they are easy to perform in front of a webcam by a person while being sited. The pictures from set 2 are taken in different light and illumination conditions. The background is a white paper. The training set has 60 samples per class and the testing sets have other 30 training samples per class. These images are acquired by a Canyon webcam- CN-WCAMNI.

For set 2 the images resolution is 640×480 pixels. The number of composition prototypes is 30 and the number of relevant composition prototypes r, which conduct to the best result is equal to 28.

In this section the experimental results which prove the potential of the compositional tehniques are presented. Our best result for the first set of images which consists

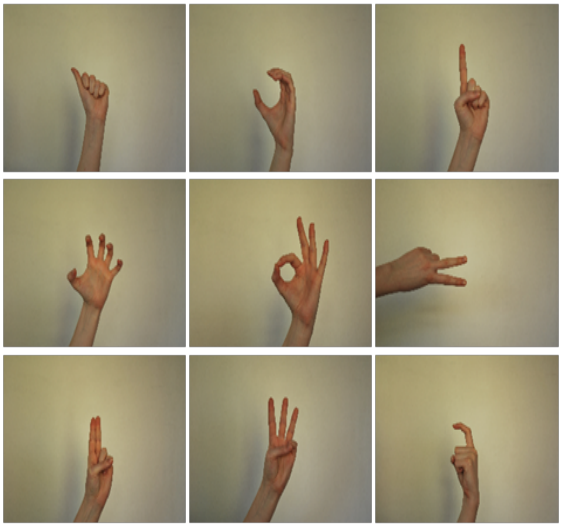


Figure 4.7: The 9 classes from set 1.

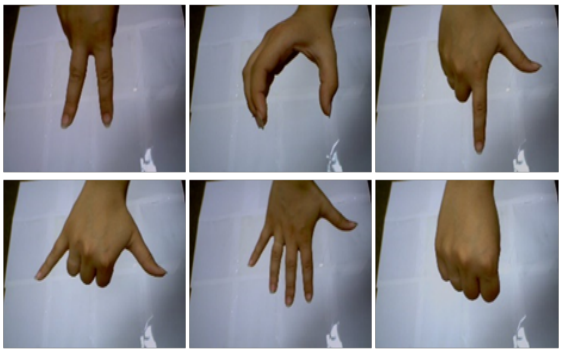


Figure 4.8: The 6 classes from set 2.

of nine classes is 96.29%; the best result for the second set of images, which consists of 6 clases is 99.82%.

The experiments also prove the importance of parameter λ, which makes a trade-off between general and specific in the cost function defined in Eq.(4.13). The robust estimation of parameter λ in order to select the relevant compositions prototypes represents a major asset of this work. Using the robust estimation of parameter λ for set 1 of images, the recognition rate was 96.29% and using the non-robust estimation of the parameter λ, the error rate for the same experiment was 93.033%. For set 2 of images the recognition rate was 96.82% when the "leave one out" method was used. For the same experiment using the non-robust estimation of parameter λ the recogniton rate was 99.59%. The results obtained for a new set of hand postures (different from the training images) are: 97.25% when we used the value of parameter λ estimated with the proposed Eq.(4.14) and 96.15% when its value was estimated using Eq.(4.13)[44].

Based on relevant composition prototypes the relevant compositions are selected. Relevant compositions and their rescaled position is used to describe the image. Both relevant compositions and their positions should have similar importance because the hand posture is recognized based on types of compositions and their relative position one to another. In order to have this, the parameter α is introduced and its value is learned based on the experimental data. The importance of parameter α is shown in experiments. The best recognition rate 96.29% was obtained for $\alpha = 0.02$.

The number of relevant composition prototypes proves to have a great influence in practice. For the first set of hand postures, the best recognition rate, 96.29%, was obtained for 19 relevant composition prototypes. For 14 relevant composition prototypes the recognition rate decreased dramatically to 29.8%; for 16 relevant composition prototypes the recognition rate was 93%, and for 18 relevant composition prototypes it was 95.6%.

The main contribution of this work is the compositional approach used to hand posture recognition. One of the contributions of this work is to carefully select the basic features (contours, interest points, patches, colour histograms, orientation histograms). These basic features generate the primitive features (the V shape, the curves and the lines). The primitive features are like Lego components, they are not extremely diverse, but by combining them it is possible to generate a lot of object shapes. The object representation is based on *compositions* of parts: descriptors are grouped according to the Gestalt law of proximity, to obtain a set of possible candidate compositions. In order to generate the desired primitive features it was important to choose the right distance between the parts which are about to be grouped. *Candidate* compositions from all test images are clustered and the resulted composition prototypes are used to form the composition codebook. Based on the cost function the relevant compositions prototypes are learned in the next stage. The optimization of parameter λ, its robust estimation in order to select the relevant compositions prototypes represents a major asset of this work.

Based on relevant composition prototypes, the relevant compositions are selected. Relevant compositions and their rescaled positions are used to describe the image. Both relevant compositions and their positions should have similar importance, because the hand posture is recognized based on types of compositions and their relative

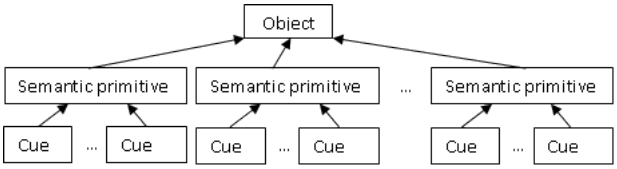


Figure 4.9: General semantic hierarchy.

position one to another. In order to achieve this, the parameter α is introduced and its value is learned based on the experimental data.

The discriminant function for classification, inspired by the point matching used in image registration, represents also a contribution of this work.

4.3.2 Hierarchical Semantic Architecture for Gestural Based Human-Computer Interfaces

In spite of the intensive research activity during the last decades, object tracking in complex environments remain a challenging task. More specifically, a good tracker should be able to perform well in scenes containing variable illumination, background clutter and occlusion. The first task of a good tracker is to avoid target loss. Additionally, in HCI applications based on dynamic gesture recognition, the accuracy of the extracted trajectories has to be considered. Accuracy is particularly important in computer gaming applications, where the players have to interact with virtual environments, or in modern robotic applications, where robots have to learn motion patterns from humans.

A popular approach addressing the problem of detection and tracking robustness is to use multiple cues. Every cue has its own strength and weakness. Methods to combine them in order to maximize system performance are still an open research problem. Among the most widely used cues we name color, motion, shape, edges, and depth information. The last one became an attractive option with the progress in stereo vision and 3D sensing [43]. In this work we rely on the first three cues, although with the use of additional imaging sensors, the depth cue can be incorporated in a straightforward manner in the proposed processing scheme.

A second approach dealing with the robustness problem is to use detection and tracking algorithms which are inherently robust. Our work uses extensively solutions which are theoretically based on robust estimation methods [50].

Our framework adopts a hierarchical semantic architecture. Good results are reported in computer vision tasks by adopting such architectures [21, 22, 38]. A general semantic hierarchy is presented in Fig.(4.9). The object is viewed as a collection of semantic primitives (the semantic layer). We can have one or more semantic layers. Each semantic primitive is determined by a set of specific cues. A general architecture can include one or more semantic layers.

Since we treat the case of hand tracking for HCI, we have extensive prior information about the target. Naturally, a first semantic layer in our approach consists of two primitives in the sense of Fig.(4.9): palm and fingers. Fig.(4.10) presents the flowchart of

our hierarchical semantic framework. We focus our tracker on active finger detection. By active fingers we mean the fingers shown stretched by the user. Our approach is designed to be used in a dynamic gesture based HCI. This is the main reason to focus our detection on the users' active fingers. Also, fingers allow a more robust detection than focusing on the palm. However, a simplified version of palm detection is used in order to impose a spatial correlation between the two semantic primitives. Prior knowledge of the target (i.e. hand) allows imposing some spatial constrains: the active finger zone is placed directly above the palm. Using this spatial correlation, relates our approach to context aware trackers [64]. To simplify the presentation and for a better understanding of our framework, we consider the case of the hand with fingers pointing upwards. Assuring rotation invariance is straightforward.

The finger primitive detection has a hierarchical semantic architecture, also. We start from the hypothesis that the finger is a collection of geometrically constrained line segments. This finger detection concept is already used with success by [54]. Our novelty is the proposal of a constrained structure composed of such finger line segments. We call these structures fingerlets. A fingerlet is represented by a couple of finger line segments under some spatial constraints. Finger primitives are determined by the set of fingerlets. Robust estimation methods on fingerlet candidates are used to segment the finger primitives. The semantic layer consists of three semantic primitives: motion-, color- and edge-based. Each primitive is defined with respect to the situation it will respond stronger and it is the most reliable. The primitives defined above are determined by a set of three cues: foreground, skin and edge cue. In our approach, the detected cues are represented in the form of binary maps. The foreground binary map gives the motion primitive. This is the most reliable primitive, and we conjecture that in non-perturbed tracking situations it will have the strongest and most reliable response, compared to the other two primitives. Color-based primitives are determined jointly by the skin and foreground cues. In cases of non-skin-like foreground objects passing behind the hand, the motion primitive alone is ineffective. In these cases the hand detection is determined by the color-based primitives. Another challenging situation for a hand tracker is the camouflage produced by skin-like objects. Background or foreground objects can present skin tones. When the hand passes over such objects either motion or color-based primitives are likely to fail. To overcome this situation the edge-based primitive is defined using the skin and edge cues.

After semantic primitive extraction stage, depending on the situation, the fingerlet feature space can be corrupted by outliers. A filtering stage is introduced, based on size (scale) constraints. Prior knowledge of the target is used to define a scale parameter which takes into account the finger thickness. A suitable range of the scale parameter is predefined by the prior knowledge of the target combined with specific details of the HCI's application. In our application, users are supposed to stand within a known range of distances from the camera. Moreover, the scale parameter is adaptively estimated after each fingerlet extraction. It is to be noted that the aforementioned challenging situations are surpassed only by coupling different semantic primitives with the shape filtering of the fingerlet feature space. In addition to this filtering, certain outliers of the feature space are eliminated by the context aware character of our approach. Palm and finger primitives must obey the spatial constraints mentioned before. For palm primitive extraction only the foreground and skin cues are used. Also, relative

size constraints are imposed. Only palm primitives that fall in a certain size range are considered. The size range is defined with respect to the finger scale parameter and a priori knowledge of hand geometry.

Fingerlet extraction is performed on the filtered feature space. Valid fingerlet structures are obtained at this stage. Robust estimation techniques are used further on in order to segment the finger primitives from the set of fingerlets. At this stage we also adaptively tune the scale parameter and the Region Of Interest (ROI).

All cues are obtained from low level processing methods applied on a ROI. The use of a ROI is based on the object persistency assumption, meaning that, if the target disappears, the most likely it will reappear in the same place. Specific parameters of the hand model and palm-finger zone spatial correlation are used during tracking to optimize ROI position and size. We begin by setting the ROI to include the entire image and then as the hand is detected the size and position is adapted accordingly in the next frames. If the target disappears, some inertia is attached to the ROI. It will keep a reduced size for a period of time and then will gradually increase to the entire image, if the target is not detected meanwhile. A straightforward advantage of this strategy is the computational time reduction, much needed for real time applications. Following the idea of computational time reduction, all cues used in this framework are represented by binary maps. Handling only binary maps in the semantic layer will decrease the computational complexity. Furthermore, the finger segmentation block operates on a collection of fingerlets stored in a list. Fingerlet parameters are initialized with values optimized offline and tuned to new values, based on extracted hand parameters throughout the hand tracking session.

The system can be in one of the following two states: hand search state and hand tracking state. The first state occurs before detecting a valid hand. Once a hand was first detected, the system enters the tracking state. We use a part based tracking-by detection approach in this work, which is a new trend in the object tracking literature [51]. One of the major advantages of the tracking by detection approach is the ability of the system to recover from target losses. This state is not instantly changed if the hand is temporarily lost. To switch back to the hand search state, the hand has to be lost for a specified period of time.

There is a major difference between our approach and traditional multiple cue approaches. In our work, the cues are organized in an intermediary semantic layer and not directly determining the object detection. A major advantage arises from this architecture. The semantic primitives complete each other and will act differently in various challenging situations. It is to be noticed that the proposed framework allows further improvement and generalization. Based on new sets of cues, one can define other fingerlet related semantic primitives.

An important aspect of our approach is the cascaded design. We construct the feature space by detecting fingerlet candidates from different cues. The feature space outliers are then filtered by scale constraints and by imposing the spatial correlation between the palm and finger primitives. It is of utmost importance that in the feature space we detect all the valid fingerlet candidates. This is done by relaxing the parameters of the corresponding detectors, even if along with valid fingerlets we detect some false positives. False positives will be rejected in the filtering stages, but false negatives cannot be recovered later. Also, the robust estimation of the parameters

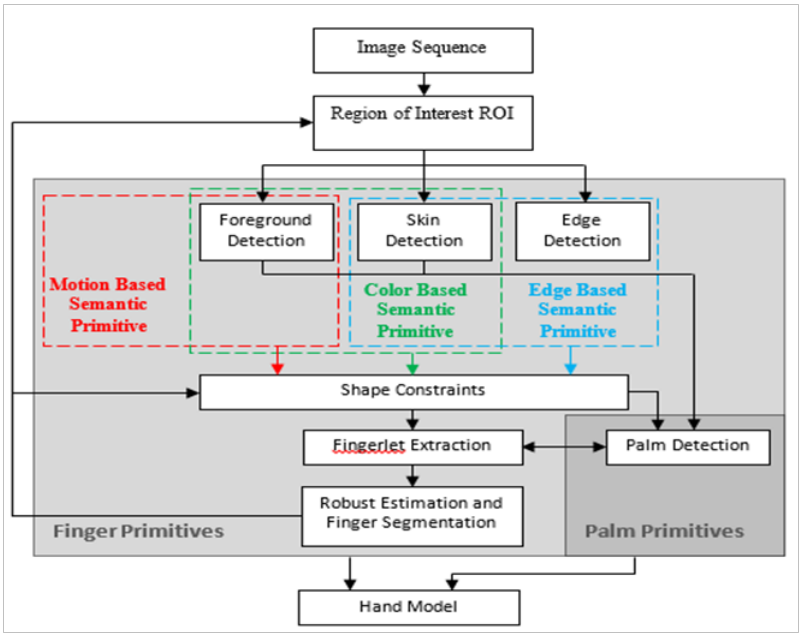


Figure 4.10: Hierarchical semantic framework for hand tracking.

and the segmentation will eliminate some of the false positives.

Cue extraction is based on some classical detection approaches presented earlier. One can experiment with different such extraction methods, our choice for the following experiments consists in using [29] for background subtraction, [11] for skin tone detection and Canny edge detector. In the following we will detail the rest of the blocks from our framework.

4.3.2.1 Fingerlets features

Motivated by the recent success of part based representations in object categorization [65] and in object tracking [2], a simple yet effective feature called fingerlet, is proposed within our framework. It is less general than SIFT, SURF, MSER or FERNS features, but more focused on the task of hand tracking. Specifically, it is designed to detect and localize accurately open hand fingers. Moreover, it can be computed effectively both from binary hand masks and from hand edge maps. Fingerlets are invariant to translation, rotation and scale. The last property is obtained by means of the feedback loop of the system, providing an estimate of the finger thickness.

A fingerlet (Fig.(4.11)) is a defined by a set of six points, grouped in two similar pixel triplets, A, B, C and $A^{'}, B^{'}, C^{'}$. Triplets can be extracted effectively in a single image scan, either from the binary foreground/background segmentation map or from the combination of the edge and skin binary maps. In the following, only the horizontal scan is described. To form a fingerlet, the pixels need to possess the properties given bellow.

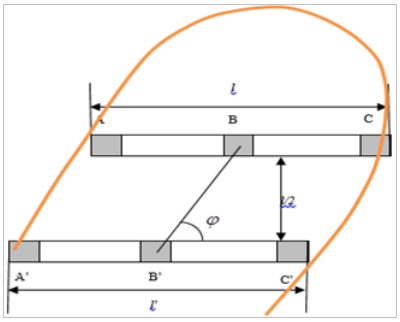


Figure 4.11: Fingerlet definition.

P1. Pixels of a triplet belong to the same scan line, the second point being the median of the line segment spanned by the triplet, Eq.(4.21).

$$\begin{aligned} y_A = y_B = y_C, x_A < x_C, \quad & x_B = (x_A + x_C)/2 \\ y_{A'} = y_{B'} = y_{C'}, x_{A'} < x_{C'}, \quad & x_{B'} = (x_{A'} + x_{C'})/2 \end{aligned} \tag{4.21}$$

P2. This property concerns the relative positions of the pixel triplets. A pair of pixel triplets has to be spatially close, since it is supposed to belong to the same finger. On the other hand, the relative position of the pair of triplets has to enable accurate estimation of the finger's local direction (angle φ) in the discrete image. To this end, the vertical displacement of the triplets is defined to be half the length, l, of the line segment AB and the horizontal displacement of the first points of the triplets is restricted to l, Eq.(4.22).

$$\begin{aligned} |y_A - y_{A'}| &= l/2, \\ |x_A - x_{A'}| &= l. \end{aligned} \tag{4.22}$$

P3. In tracking mode, the size property requires the lengths of the line segments AC and $A'C'$ to be close to the current estimate, Eq.(4.23).

$$\begin{aligned} \varepsilon < l/\hat{l} < 1/\varepsilon, \\ \varepsilon < l'/\hat{l} < 1/\varepsilon, \end{aligned} \tag{4.23}$$

where $0 < \varepsilon < 1$ is the maximum ratio between a valid finger length and the currently estimated model length,$\hat{t}$ given Eq.(4.24).

$$\hat{l} = \hat{t}/\sin(\varphi) \tag{4.24}$$

The local finger direction is given Eq.(4.25).

$$\varphi = \arg(x_A - x_{A'}, l/2) \tag{4.25}$$

To account for anatomical finger thickness variations and effects of motion blur, the maximum normalized difference of an accepted triplet is set in this work to a relatively large value: 50%. This large value is used because it is more important

Figure 4.12: From left to right: original image, palm primitives, motion primitives, color primitives and edge primitives.

to have low percentage of false negative than false positive detections. The reason behind this option is twofold. On one hand, most distracting objects have already been filtered out by motion and skin cues. On the other hand, a fake triplet is only half of a fingerlet. Another one has to be found, at the required distance. Moreover, to generate a false finger, the corresponding fingerlets need to form valid clusters, as explained further on, which is very unlikely (although not impossible) to happen.

Before initial hand detection, the value of $\hat{t}$ is highly uncertain, because of the unknown distance to camera and hand size variation between users. In this stage, $\hat{t}$ is set to a fixed fraction of the image line length, L, and ε is set at a higher value. In this work, the values $\hat{t} = L/25$ and $\varepsilon = 33\%$ have been used.

Fingerlets extracted from a video frame are saved in a scan ordered list. Typically, the list contains only a few hundreds of fingerlets, which is convenient for fast processing. All operations described in this section are operated on the fingerlet list or histograms extracted from the list. Fingerlets are extracted in a two-step process. The first step creates a list of triplets with appropriate length. In the second step, in one scan, the triplets without a match are discarded, while the triplets with match, as illustrated in Fig.(4.11), will form a fingerlet structure, defined by spatial coordinates, length and orientation, Eq.(4.26).

$$\begin{aligned} \mathbf{f}_i &= [x_{B_i}, y_{B_i}, t_i, \varphi_i] \\ t_i &= l_i / \sin(\varphi) \end{aligned} \tag{4.26}$$

These vectors are saved on a fingerlet list. Also, 1D histograms of vector components h_x, h_y, h_t and h_φ are obtained from the same scan.

4.3.2.2 Primitives

Primitives are represented by fingerlets, hence, the method of primitive extraction can be implemented with a simple horizontal scan and some simple detection logic, allowing thus an increased computation speed. Motion primitives (Fig.(4.10)) are obtained from the foreground binary map. Color primitives are determined in the binary map obtained by intersecting the skin tone map and foreground map. Finally, edge primitives involve the skin tone map and the foreground edge map. As mentioned before, these primitives were introduce in order to prevent the tracker to fail in various challenging situations. Depending on the application, one can add more primitives related to other situations not treated in this work. An example of extracted primitives, after imposing the spatial constraint between palm and finger related primitives, is given in Fig.(4.12).

One can argue on some detection faults in this example. The idea of this framework is to be applied mainly in dynamic gestural cases, and as we will see in the following these detection faults are coped with by other processing blocks of the framework.

4.3.2.3 Robust Estimation of ROI and Scale Parameters

a) Scale Adaptation In spite of the filtering steps making use of motion, skin, edge and size cues, some of the detected fingerlets may not belong to real hand fingers. Within the framework of parameter estimation, such samples, not actually belonging to the object of interest, are named outliers. In the presence of outliers, the simplest parameter estimator, which is the sample mean finger thickness extracted from fingerlets, may be biased, if outliers are not symmetrically distributed around the sample mean. The possibility of having "conspiring" outliers cannot be ruled out, as fake fingerlets can be generated by (elongated skin colored, moving) objects.

There are many robust estimation methods used in computer vision. In this work, we use nonparametric probability density estimation to find a maximum likelihood (ML) estimate of the finger thickness. The approach can be thought of as a particular case of an M estimator, as pointed out by [15]. Our estimator is based on the assumption that no other object generates more valid fingerlets than the user's hand. To make this assumption as realistic as possible, the tracker is supposed to be initialized with a frontal view of the hand with stretched fingers. The ML estimate of the finger thickness is defined in Eq.(4.27).

$$\hat{t} = \arg\max_{t} \left(p\left(t \right) \right) \tag{4.27}$$

where $p(t)$ is the probability density of finger thickness t. A widely used method to find maxima of the probability density function is the mean shift algorithm [15], which is a gradient ascent optimization method. Starting from any point in the feature space, it converges to a local maximum of the probability density, estimated from a finite number of data samples. Probability density estimation from a finite number of samples is based on a kind of interpolation carried out by means of a kernel function. For 1D data, it can be written in the form of Eq.(4.28):

$$p\left(x \right) = \frac{1}{N} \sum_{n=1}^{N} K\left[\left(x - x_n \right) / s \right] \tag{4.28}$$

where $K\left(\right)$ is the kernel function and s is a scale parameter, controlling the degree of smoothing. The mean shift finds the location of a maximum by iterating until convergence, Eqs. (4.29) (4.30).

$$\hat{x}^{(j+1)} = f\left(\hat{x}^{(j)} \right) \tag{4.29}$$

with

$$f\left(\hat{x} \right) = \sum_{n=1}^{N} \frac{K^{'}\left[\left(\hat{x} - x_n \right) / s \right]}{\sum_{m=1}^{N} K^{'}\left[\left(\hat{x} - x_m \right) / s \right]} x_n \tag{4.30}$$

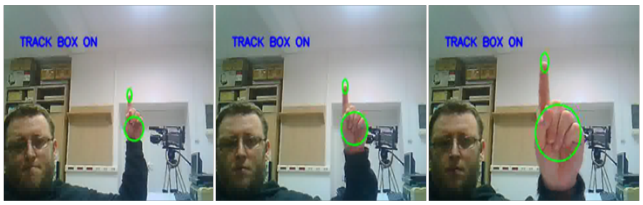


Figure 4.13: Scale adaptation example.

where $K^{'}(t) = dK/dt$ is the derivative of the kernel. Depending on the scale parameter, the estimated density may have more local maxima. To find location of the global maximum, a multi-scale generalization of the mean shift [25] is used, coupled with the use of the 1D Epanechnikov. In this work, iterations for finger thickness estimation start with infinite scale s_0 and then continue with:

$$s_{j+1} = \hat{t}^{(j)}/2 \tag{4.31}$$

To speed up computation, we use the finger thickness histogram to estimate the mode of the finger thickness distribution, and it follows that:

$$f^{(j)}\left(\hat{t}\right) = \frac{\sum_{t=\hat{t}-s_j}^{\hat{t}+s_j} th\left(t\right)}{\sum_{t'=\hat{t}-s_j}^{\hat{t}+s_j} h\left(t'\right)} \tag{4.32}$$

Figure 4.13 presents an example of scale parameter adaptation. As mentioned before, the palm scale parameter is defined with respect to the finger scale parameter, thus becoming adaptive, too. Based on prior knowledge of hand geometry, in our experiments we considered that the palm scale parameter is 4-6 times the finger scale parameter.

b) ROI Adaptation Based on the same theory, we find the estimated ML of the ROI. The ROI anchor, C, is defined by the modes of the fingerlet spatial coordinate histograms, h_x and h_y. The scale parameter used is constant and is set to be twice the estimated finger thickness. The ROI is a rectangular region. The width and height parameters are set as a function of the finger thickness and the current hand speed, as illustrated in Fig.(4.14). Notice that the dynamic ROI is asymmetrically extended from the static size of $10\hat{t} \times 10\hat{t}$ with the frame speed components, v_x and v_y. At the beginning of the session, the ROI is represented by the entire image. If the hand is detected the system switches into hand tracking mode and the ROI is adapted accordingly. It is to be noted that the ROI size is consistent with the scale. When the scale parameter value is low the ROI size is reduced up to approximately 5% of the entire image. At the end of the session, when the users' hand exits the scene ROI size is gradually increased to the entire image, for the last frames the tracker being switched again to search mode.

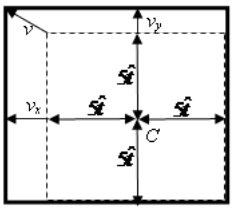


Figure 4.14: ROI definition.

4.3.2.4 Hand segmentation

Finger counting is used in the proposed framework for two purposes. The first one is to provide a safe initialization of the system. Safety means avoiding false starts, while not failing to detect real hands. In this work, the system switches from hand search to hand tracking state when a predefined number of fingers, n_f, is detected. We found convenient to work with $n_f = 3$. A second reason why we count fingers is to provide a fast way to switch from one dynamic gesture to the next one. Alternately, the user could make a pause between two gestures, allowing the system to detect still periods.

Fingers generate connected chains of fingerlet coordinates, (x_{B_i}, y_{B_i}). The length of such a chain depends on the distance of the hand to the camera and the finger orientation. To gain scale independence, the required finger length is expressed in terms of the finger scale parameter. Their ratio is normally confined to a ratio of 4 to 6. However, to allow for robust detection under partial occlusions, we found convenient to accept chains with length exceeding only one finger thickness. Again, the optimal threshold is a tradeoff between false positive and false negative detections.

Fingerlet coordinate chains are segmented in this chapter by means of a morphological tool, reconstruction by geodesic dilation. In the fingerlet list the markers are first detected. Fingertip coordinates represent the markers. Considering the vertical orientation of the hand the uppermost fingertip is detected in a simple horizontal scan. Connected neighbors are searched then downwards by geodesic dilation. The set of connected fingerlets represents the segmented finger. Previous fingerlet set is removed from the list and the same procedure is iterated to find the remaining fingers. The algorithm ends when the list is empty or a small number of fingerlets is left. At the same location multiple fingerlets can be present in the list. The right fingerlet for the segmentation is chosen with respect to angle conservation relative to the previous fingerlet or starting marker.

Figure 4.15 shows some examples of finger and palm detection for different hand poses. We can see that the hand tracker has some degree of freedom; slight tilted or rotated poses are correctly detected.

4.3.2.5 Experiments and discussions

In this paragraph we present some extensive tests that validate our framework. First we prove that the real time constraint is met. Second, a series of tests demonstrate the validity of our model in different challenging situations: camouflage, varying illumination and occlusion. Accuracy, needed in a dynamic gestural based interface, is

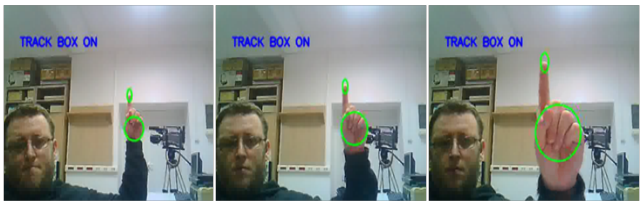


Figure 4.15: Detected fingers for different hand poses.

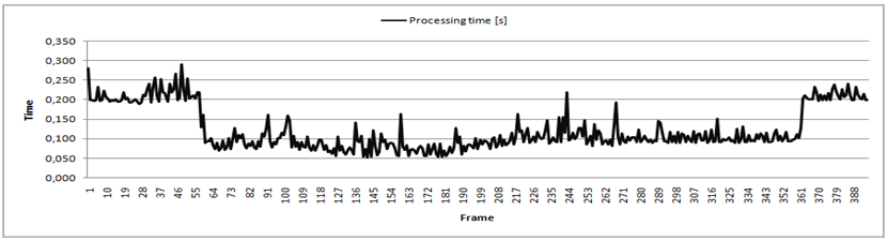


Figure 4.16: Processing time per frame.

proven at the end.

a) Real Time Constraint Figure 4.16 shows the computation speed of our hand detection framework. In hand searching mode we obtain an average of 5 fps, but in hand tracking mode we achieved an average computation speed of 11 fps, and depending on ROI size this speed can increase up to 20 fps. One should note that poor lighting of the scene increases the computational time. This experiment, and the ones that follow, were carried out with a laptop webcam (Intel Core2 Duo CPU at 2,5GHz, 2GB RAM, NVIDIA GeForce 8600M GT and 1GB VRAM). Since a processing speed of 5 fps is not exactly real time we recommend that the framework be used with some initialization (e.g. a dedicated hand posture to enter the hand tracking mode).

b) Camouflage Motion cue camouflage denotes the situation when a non-skin tone foreground object passes behind the hand. In this case, in the foreground map a camouflage situation arises. The motion primitives are unreliable and the problem is surpassed by the color and edge primitives (Fig.(4.17) first row). Classic camouflage situation refers to the case of background objects having skin tones. This is a highly probable situation for HCI applications. Furniture or some other objects most often are skin like colored. Skin and possibly motion cues are most likely to fail in this situation. So, it is the edge based primitives that are the most reliable with respect to classic camouflage (Fig.(4.17) second row).

c) Varying illumination Indoor use of our framework presents the advantage of being able to control the scene lighting. Nevertheless, depending on the application, human or some other factors can influence the scene. One straightforward situation

Figure 4.17: Camouflage tests: motion camouflage - first row; color camouflage - second row.

Figure 4.18: Experiment concerning illumination changes.

is represented by the dynamic shadows casted by moving neighboring objects. The illumination change is simulated by tuning the brightness parameter of the camera (Fig.(4.18)).

d) Occlusions Another type of challenging situation for a tracking system is the occlusion. Accidentally or not, foreground objects can momentarily hide the target from the camera. If partial occlusion of the hand occurs it is desirable that the tracker does not lose at least the uncovered primitives (Fig.(4.19) first row). Figure 4.19 second row illustrates an example of full occlusion of the target by a skin tone object. It is obvious that this situation will cause target loss by our approach. What we consider important in such a case is rapid recovery and target detection. In this example the tracking framework only needs a few frames to redetect the target, independently of the obstructing object shape or targeted hand pose. This advantage arises from the tracking by detection character of our approach.

Our approach is different from the model based approaches proposed in tracking related literature. We do not impose a predefined model and fit it within the tracking session. However, with some limitations our framework constructs a model of the hand during the tracking session (e.g. separate detection of palm and fingers). Also, as already pointed out, our framework addresses mainly to HCI involving dynamic gestures. As possible application we mention designing communication dictionaries for remote control.

Another type of tracking algorithms are box based trackers. These approaches

Figure 4.19: Occlusion experiment: partial occlusion - first row; full occlusion - second row.

present the drawback of being sensible to attractors with same characteristics as the target. Instead, our semantic hierarchical framework allows surpassing these situations by using multiple cues that integrate in different semantic primitives. Other hand tracking approaches are focused on detecting the hand as a whole and can be thought as a particular case of box based trackers.

For the sake of simplicity the case of the hand with fingers pointed upwards was treated here. Hence the horizontal scan of the ROI was considered. One can argue that the framework lacks in terms of rotation invariance. Fingerlet structures are characterized also by an angle, which approximates the general hand orientation. The entire fingerlet set gives a set of angles. Same robust estimation techniques, presented before, can be employed to estimate a single finger angle value from the fingerlet set. Considering that between consecutive frames the hand pose does not change significantly, the previous estimated angle value can be used to rotate the ROI. The horizontal scan is maintained and the rotation invariance is achieved. If multiple fingers are detected, a mean estimate of finger angles can be considered as the general hand orientation.

4.4 New Directions: ToF Based 3D Gestures Recognition System

Increasing interest for 3D based gesture recognition systems, driven by the potential applications (medicine, entertainment, automotive, etc.), is justified twofold, at least. On one hand there are the inherent weaknesses of the 2D systems, e.g., the low reliability of the segmentation stage – see the problem which appears when background and skin have similar colors; on the other hand we have the advancements in the 3D range sensor technologies which currently offer a higher accuracy at constantly decreasing costs.

The modern 3D approaches use either the structured light principle, e.g., [7] or the Time-of-Flight (ToF) technique [26], although other acquisition methods were also

reported [40]. In the following, we will briefly discuss ToF based related work, as we consider it the most promising alternative.

One of the earliest approaches is represented by the work in [8] where they presented a 3D hand gesture recognition system based on Swissranger SR2, an infra-red ToF camera. Some key operations are camera calibration and noise reduction using a median filter on the range data. The segmentation step implies defining a region of interest (ROI) on the depth information. Next, the resulting cloud of points is fitted using an ellipsoid in order to obtain a raw estimation of the hand. To determine the principal axes of the ellipsoid a PCA is performed. A more elaborate approach is to fit a seven degrees of freedom hand model on the cloud of points with a frame rate of about 3Hz. No classification accuracy information is provided by the authors. Obviously, the response does not meet the demands of a real-time functioning.

The goal of the system proposed by [30] is to recognize 12 different static hand gestures using a 3D ToF sensor. Using a simple nearest neighbor classifier, they reported a classification time around 30ms on a standard PC and a recognition rate of 94.61%. The chosen evaluation procedure was "Leave-One-Out" using a set of 408 images (12 gesture $\times$ 34 persons). The 3D information helps mainly in the iterative seed fill segmentation algorithm. Additionally, depth features are included to distinguish certain gestures when the 2D projections are identical. Following similar principles (coarse to fine segmentation, PCA on the feature vectors, nearest neighbor, k-d Tree Based k-Means Clustering, and the Bayesian Plug-In as classifiers) [55] developed a touchless user-interface for medical intra-operative applications.

A combination of RGB and ToF cameras for real-time 3D hand gesture interaction is described in [18]. In this situation, the hand detection is achieved by using a novel algorithm which uses both depth and color. Hybrid Gaussian mixture model and histogram-based skin color segmentation are performed in the first stage. Then, the face is detected and the distance from face to camera is estimated. This way, the search region is drastically reduced and the hand is detected based on skin color. Inspired by the work of [63], the classification stage uses a dimensionality reduction technique based on Average Neighborhood Margin Maximization. A Haarlet-based hand gesture recognition algorithm is implemented for the case of color, depth and combined data.

Molina et al. [42], present a system for hand gesture recognition, based on a ToF camera (SR4000 developed by Mesa Imaging) devoted to control Windows applications. For foreground segmentation, a mask is generated including at least 20 gray levels below the brightest one (20 cm from the nearest detected point). Then the geodesic center of the hand, the length and orientation of the axes of the ellipse fitted to the hand silhouette and the minimum depth point are used both for extracting additional silhouette features and for the gesture classification. The chosen modality to describe the shape consists in modeling the skeleton of the silhouette although there are many other possibilities (Fourier descriptors, Zernike or Hu moments etc.). The system shows remarkable performance, the user independent gesture detection rate being around 94%.

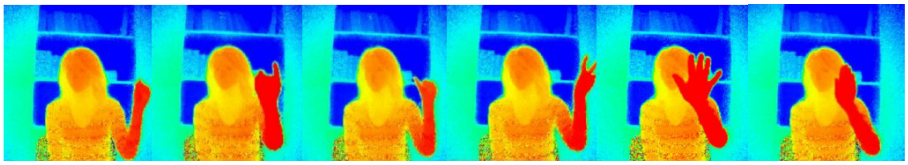


Figure 4.20: Selected frames from subject no. 7 - left hand, performing 6 different hand gestures. UPT-ToF3D-HGDB hand gesture database.

4.4.1 Method for 3d Hand Gestures Recognition

Gesture recognition approaches can be classified according to [27] in at least two categories. One takes into account just the hand and the fingers and it detects hands, fists, palms, and fingers. The other category considers the full body or body part moving gestures, thus, concentrating on the detection of the median axis of the body part. In the following we will present our ToF based solution for a gesture recognition system, which can be included in the first category.

4.4.2 The Hardware

Our experiments were made using a leading-edge PMD[vision]® CamCube 3.0 ToF 3D camera [48]. A modulated optical signal sent out by a transmitter illuminates the scene to be measured. The sensor, PhotonICs®PMD 41k-S2, detects the reflected light, and determines the phase difference between incident and reflected optical signal per every single pixel. This enables computing the distance to the target:

$$d = \frac{c \times \varphi}{4\pi f_{mod}} \tag{4.33}$$

where c is the speed of light and φ the phase shift and f_{mod} is typically 20KHz [33].

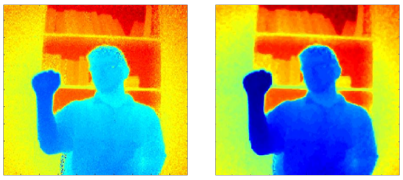
PMD[vision]® CamCube 3.0 is the highest resolution all solid-state TOF 3D camera worldwide, enabling the real-time capture of distance and grayscale information (200×200 pixels) at high frame rate (up to 40 fps at full resolution) with superior ambient light suppression.

4.4.3 3D Data Acquisition

Considering the scarcity of the publicly available 3D ToF databases and the very high costs for a ToF video camera acquisition, we developed a ToF 3D hand gesture database, called UPT-ToF3D-HGDB, to support researchers who are willing to test their algorithms [59]. Current release, described in details in our previous work [52], contain multiple subjects expressing six static hand poses and four dynamic hand gestures, as exemplified in Fig.(4.20).

4.4.4 3D Filtering

The first step of our algorithm aims at suppressing the speckle noise contained by range data. This is done with the help of a 3×3 median filter and it is crucial for



(a) (b)

Figure 4.21: (a) Original and (b) filtered images.

the success of following steps. The effect of the filter is presented in Fig.(4.21a) and Fig.(4.21b).

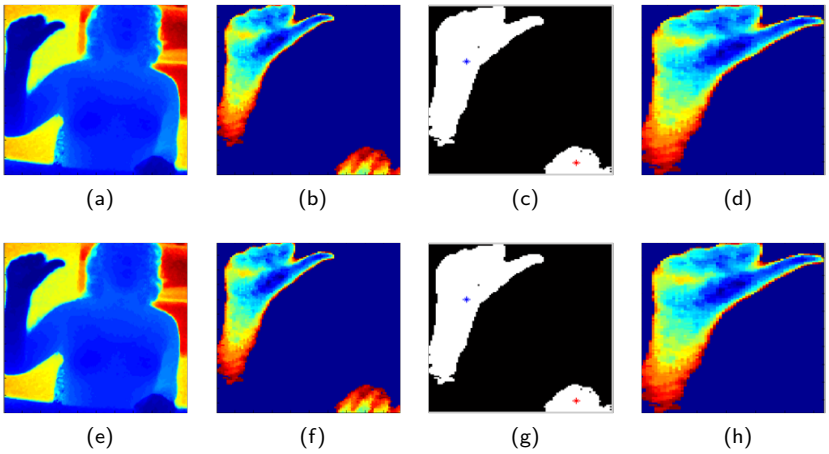
4.4.5 Depth Segmentation

Assuming that the hand is the closest object to the camera, the segregation from the background is done by defining a region of interest in depth and discard any measurements which do not fall within the predefined range of depth, in our case 7 cm from the closest point. To eliminate possible outliers, the number of points from the hand volume is counted. If the number of potential candidates from the hand volume is smaller than a threshold (80 in our experiments), the next closest point is selected and the operation is repeated.

There are some situations (see Fig.(4.22a) and Fig.(4.22e)) where this procedure would not provide satisfactory results, more exactly when the extracted cloud of points do not belong to the same cluster (Fig.(4.22b), Fig.(4.22f)). As one could observe from Fig.(4.22c) and Fig.(4.22g) other parts of the body (abdomen, the other hand, head, etc.) could be initially segmented as hand. To solve this, we project 3D points onto a 2D space. The resulting image is binarized and a set of properties (area, centroid, orientation, pixel list) are extracted. The number of clusters found in the hand plane is counted. For each cluster, the number of points is estimated. If the number of points in a cluster is smaller than a threshold, the cluster is discarded. If we have more than one remaining clusters, a new analysis is performed in order to decide which cluster represents the hand. The coordinates of the centroids (which are the centers of mass of each cluster) are inspected. If these coordinates are not in the upper half of the hand plane, the clusters are discarded. The final results of the segmentation step are depicted in Fig.(4.22d) and Fig.(4.22h). The details regarding the proposed segmenting algorithm are presented in [53].

4.4.6 Delaunay Triangulation

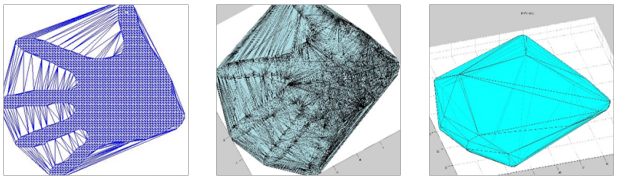
The vast majority of the techniques employed for gestures recognition from 3D data use depth information, mainly for the segmentation purpose, the feature extraction step being performed in 2D space. Our approach is different because it relies on the information provided by the cloud of points itself. In order to obtain the 3D shape of the hand we perform a Delaunay triangulation (DT) on the available set of



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 4.22: The segmentation steps results: (a),(e) filtered images; (b),(f) the result of the first segmentation stage; (c),(g) binarized 2D projections - the center of the clusters are marked in blue and red, the red cluster will be discarded; (d),(h) the segmented hand.



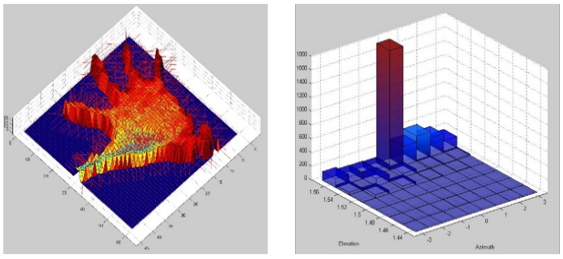
(a) (b) (c)

Figure 4.23: Delaunay triangulation in (a) 2D, (b) 3D and (c) the free boundary facets of triangulation.

points. The DT is one of the most popular methods used to the generate meshes and it satisfies the empty sphere property, meaning that a 3D Delaunay triangulation does not have any points in the interior of the circumsphere associated with each tetrahedron (Fig.(4.23)).

4.4.7 Feature Extraction

Some possibilities of geometric features are: surface curvature or normal vectors to the 3D surface, all of them being considered as local features. We compute the normal vectors to the surface (Fig.(4.24a)), then map the orientations into spherical coordinates in order to obtain a three-dimensional histogram of bivariate data (Fig. (4.24b)).



(a) (b)

Figure 4.24: (a) Compute and display 3D surface normals as radiating vectors and (b) Three-dimensional histogram of bivariate data (azimuth and elevation) binned into a 9-by-7 grid.

4.4.8 Classification

In order to evaluate the proposed approach, we considered 10 persons performing 6 different gestures with 20 frames per gesture, making a total of 1200 3D images selected randomly from the above mentioned UPT-ToF3D-HGDB database. A simple distance-based classifier calculates the pairwise distance between two sets of observations followed by a leave-one-out cross-validation. Experiments with various possibilities in implementing the distance computation, showed that the best results are obtained by using 'cosine' distance. Good results were also obtained with 'correlation', 'cityblock', 'chebychev' and 'minkowski' distances.

Although the preliminary results show lower recognition rates than the state of the art solutions reported in similar conditions, we are confident that there are a number of possibilities to increase the system performances, e.g. adding supplementary 3D features, fine tuning of the ensemble parameters, and, probably the most important aspect implementation of a more elaborate classification scheme. The SVM, Random Trees and Localist Attractor Networks are expected to provide better classification results.

4.4.9 Conclusions

Hand gesture recognition is an active research field, aiming to generate real world applications within the context of the emerging technologies like pervasive computing, mobile computing, ambient intelligence etc. Some of these applications are highlighted in this study. There are two dominant approaches to hand gesture recognition: appearance based and model based. The work described in this chapter belongs to the first category, which is more general and computationally more efficient.

Reliable extraction of semantic information from images, in particular the recognition of hand gestures, have to find consistent ways to deal with inaccuracies of the existing segmentation and feature extraction algorithms. One possible answer investigated here is to use a part based representations of the hand. Part based representations have been used in compositional methods for object categorization, leading to the huge

popularity of the bag of words and related classifiers. Our pioneering work revealed the potential of this approach to hand gesture recognition and we strongly believe that there still is a lot of room for future research and improvement within the compositional framework to hand gesture recognition.

Another key concept addressed in this work is data fusion. Multiple cues, such as color, edges, motion, and a custom designed feature for hand detection and tracking, called "fingerlets" are combined in a hierarchical semantic architecture. Fingerlets are defined and used with ideas from robust estimation in mind and play an important role in the success of our method.

A brief presentation of hand gesture recognition solutions based on data acquired with 3D sensors and some preliminary results of the authors using ToF sensors is the subject of the last section of this chapter. While advances in computational capabilities and sensors suggest that solutions based on 3D data are likely to prevail in the near future, the authors believe that several concepts described in this chapter, demonstrated on monocular video, are going to be of genuine interest and can be naturally integrated in the next generations of human computer interaction systems, including hand gesture recognition.

References

[1] M. Alpern and K. Minardo. Developing a car gesture interface for use as a secondary task. In *Extended Abstracts on Human Factors in Computing Systems (CHI)*, pages 932–933, 2003.

[2] S. Avidan. Ensemble tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):261–271, 2007.

[3] A.L.C. Barczak and F. Dadgostar. Real-time hand tracking using a set of cooperative classifiers based on Haar-like features. *Research Letters in the Information and Mathematical Sciences*, 7:29–42, 2005.

[4] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[5] T. Bouwmans. Recent advanced statistical background modeling for foreground detection: a systematic survey. *Recent Patents on Computer Science*, 4(3):147–176, 2011.

[6] T. Bouwmans. Traditional and recent approaches in background modeling for foreground detection: an overview. *Computer Science Review*, 11-12:31–66, 2014.

[7] M. Bray, E. Koller-Meier, and L. Van Gool. Smart particle filtering for 3D hand tracking. In *6th IEEE International Conference on Automatic Face and Gesture Recognition (AFGR)*, pages 675–680, 2004.

[8] P. Breuer, C. Eckes, and S. Müller. Hand gesture recognition with a novel IR time-of-flight range camera – a pilot study. In *3rd International Conference MIRAGE*, volume 4418 of *LNCS*, pages 247–260, 2007.

[9] V. Bruce, P.R. Green, and M.A. Georgeson. *Visual Perception: Physiology, Psychology and Ecology*. Psychology Press, East Sussex, UK, 3rd edition, 1996.

[10] V. Buchmann, S. Violich, M. Billinghurst, and A. Cockburn. FingARtips: gesture

based direct manipulation in augmented reality. In *2nd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia (GRAPHITE)*, pages 212–221, 2004.

[11] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt. A skin tone detection algorithm for an adaptive approach to steganography. *Signal Processing*, 89(12):2465–2478, 2009.

[12] Q. Chen, N.D. Georganas, and E.M. Petriu. Real-time vision-based hand gesture recognition using Haar-like features. In *Instrumentation and Measurement Technology Conference (IMTC)*, pages 1–6, 2007.

[13] Y. Chuang, L. Chen, G. Zhao, and G. Chen. Hand posture recognition and tracking based on Bag-of-Words for human robot interaction. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 538–543, 2011.

[14] P.R. Cohen, M. Johnston, D. McGee, S. Oviatt, J. Pittman, I. Smith, L. Chen, and J. Clow. Quickset: multimodal interaction for distributed applications. In *5th ACM international conference on Multimedia (MULTIMEDIA)*, pages 31–40, 1997.

[15] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[16] N. Dardas, Q. Chen, N.D. Georganas, and E.M. Petriu. Hand gesture recognition using Bag-of-features and multi-class support vector machine. In *IEEE International Symposium on Haptic Audio-Visual Environments and Games (HAVE)*, pages 1–5, 2010.

[17] M. de La Gorce, D.J. Fleet, and N. Paragios. Model-based 3D hand pose estimation from monocular video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1793–1805, 2011.

[18] M. Van den Bergh and L. Van Gool. Combining RGB and ToF cameras for real-time 3D hand gesture interaction. In *IEEE Workshop on Applications of Computer Vision (WACV)*, pages 66–72, 2011.

[19] T.A. Dingus, S.G. Klauser, V.L. Neale, A. Petersen, S.E. Lee, J. Sudweeks, M.A. Perez, J. Hawkey, D. Ramsey, S. Gupta, C. Bucher, Z.R. Daersaph, J. Jermeland, and P.R. Knipling. 100-car naturalistic driving study - phase ii - results of the 100-car field experiment. Technical Report Report No. DOTHS810593, Virginia Tech - Transportation Institute -Sponsored by National Highway Traffic Safety Administration, 2006.

[20] M. Donoser and H. Bischof. Real time appearance based hand tracking. In *19th International Conference on Pattern Recognition (ICPR)*, pages 1–4, 2008.

[21] B. Epshtein and S. Ullman. Semantic hierarchies for recognizing objects and parts. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[22] L. Fei-Fei and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 524–531, 2005.

[23] S.M. Goza, R.O. Ambrose, M.A. Diftler, and I.M. Spain. Telepresence control of the nasa/darpa robonaut on a mobility platform. In *SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 623–629, 2004.

[24] C. Grätzel, T. Fong, S. Grange, and C. Baur. A non-contact mouse for surgeon-computer interaction. *Technology and Health Care*, 12(3):245–257, 2004.

[25] V. Gui. Edge preserving smoothing by multiscale mode filtering. In *16th European Signal Processing Conference (EUSIPCO)*, pages 25–29, 2008.

[26] M. Hansard, S. Lee, O. Choi, and R.P. Horaud. *Time-of-Flight Cameras: Principles, Methods and Applications*. Springer Briefs in Computer Science. Springer, 2013.

[27] D. Ionescu, V. Suse, C. Gadea, B. Solomon, B. Ionescu, S. Islam, and M. Cordea. A 3D NIR camera for gesture control of video game consoles. In *IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, pages 1–5, 2014.

[28] R. Khan, A. Hanbury, and J. Stoettinger. Skin detection: a random forest approach. In *17th IEEE International Conference on Image Processing (ICIP)*, pages 4613–4616, 2010.

[29] K. Kim, T.H. Chalidabhongse, D. Harwood, and L. Davis. Real-time foreground-background segmentation using codebook model. *Real-Time Imaging*, 11(3):172–185, 2005.

[30] E. Kollorz, J. Penne, J. Hornegger, and A. Barke. Gesture recognition with a Time-Of-Flight camera. *International Journal of Intelligent Systems Technologies and Applications*, 5(3-4), 2008.

[31] T. Konrad, D. Demirdjian, and T. Darrell. Gesture + play: full-body interaction for virtual environments. In *Extended Abstracts on Human Factors in Computing Systems (CHI)*, pages 620–621, 2003.

[32] J. Koster. Applying gestures in domestic appliances. `http://hmi.ewi.utwente.nl/verslagen/capita-selecta/CS-Koster-Jacobjob.pdf`, 2006.

[33] R. Lange. *3D time-of-flight distance measurement with custom solid-state imagevsensors*. PhD thesis, Dept. of Electrical Engineering and Computer Science, University of Siegen, 2000.

[34] M.E. Latoschik. A gesture processing framework for multimodal interaction in virtual reality. In *1st International Conference on Computer Graphics, Virtual Reality and Visualisation (AFRIGRAPH)*, pages 95–100, 2001.

[35] S. Lenman, L. Bretzner, and B. Thuresson. Using marking menus to develop command sets for computer vision based hand gesture interfaces. In *2nd Nordic conference on Human-computer interaction (NordiCHI)*, pages 239–242, 2002.

[36] R. Lienhart and J. Maydt. An extended set of Haar-like features for rapid object detection. In *International Conference on Image Processing (ICIP)*, volume 1, pages 900–903, 2002.

[37] A. Malima, E. Ozgur, and M. Cetin. A fast algorithm for vision-based hand gesture recognition for robot control. In *14th IEEE Signal Processing and Communications Applications (SIU)*, pages 1–4, 2006.

[38] M. Marszalek and C. Schmid. Semantic hierarchies for visual object recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–7, 2007.

[39] A.M. Martinez, R.B. Wilbur, and R. Shay A.C. Kak. Purdue RVL-SLLL ASL database for automatic recognition of American sign language. In *4th IEEE International Conference on Multimodal Interfaces (ICMI)*, pages 167–172, 2002.

[40] Microchip. http://www.microchip.com/pagehandler/en-us/press-release/microchips-new-gestic-technolo.html, 2012.

[41] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1-2):43–72, 2005.

[42] J. Molina, M. Escudero-Viñolo, A. Signoriello, M. Pardàs, C. Ferrán, J. Bescós, F. Marqués, and J.M. Martínez. Real-time user independent hand gesture recognition from time-of-flight camera video using static and dynamic models. *Machine Vision and Applications*, 24(1):187–204, 2013.

[43] I. Oikonomidis, N. Kyriazis, and A.A. Argyros. Efficient model-based 3D tracking of hand articulations using Kinect. In *22nd British Machine Vision Conference (BMVC)*, pages 101.1–101.11, 2011.

[44] B. Ommer and J.M. Buhmann. Learning compositional categorization models. In *9th European Conference on Computer Vision (ECCV)*, volume III, pages 316–329, 2006.

[45] N. Osawa, K. Asai, and Y.Y. Sugimoto. Immersive graph navigation using direct manipulation and gestures. In *ACM Symposium on Virtual Reality Software and Technology (VRST)*, pages 147–152, 2000.

[46] C.A. Pickering. Gesture recognition driver controls. *Computing & Control Engineering Journal*, 16(1):26–27, 2005.

[47] C.A. Pickering, K.J. Burnham, and M.J. Richardson. A research study of hand gesture recognition technologies and applications for human vehicle interaction. In *3rd Institution of Engineering and Technology Conference on Automotive Electronics*, pages 1–15, 2007.

[48] PMD[vision]®CamCube-3.0. http://www.pmdtec.com/products-services/pmdvisionr-cameras/pmdvisionr-camcube-30.

[49] R. Radkowski and C. Stritzke. Interactive hand gesture-based assembly for augmented reality applications. In *5th International Conference on Advances in Computer-Human Interactions*, pages 303–308, 2012.

[50] A.C. Rencher. *Methods of Multivariate Analysis*. John Wiley & Sons, 2002.

[51] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof. PROST: parallel robust online simple tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 723–730, 2010.

[52] G. Simion and C.D. Caleanu. A ToF 3D database for hand gesture recognition. In *10th International Symposium on Electronics and Telecommunications (ISETC)*, pages 363–366, 2012.

[53] G. Simion and C.D. Caleanu. Multi-stage 3D segmentation for ToF based gesture recognition system. In *11th International Symposium on Electronics and Telecommunications (ISETC)*, 2014. in press.

[54] P. Song, S. Winkler, S.O. Gilani, and Z. Zhou. Vision-based projected tabletop interface for finger interactions. In *IEEE International Workshop on Human–Computer Interaction (HCI)*, volume 4796 of *LNCS*, pages 49–58, 2007.

[55] S. Soutschek, J. Penne, J. Hornegger, and J. Kornhuber. 3-D gesture-based scene navigation in medical imaging applications using Time-of-Flight cameras. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 1-6, 2008.

[56] T. Starner, J. Weaver, and A. Pentland. Real-time American sign language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1371–1375, 1998.

[57] B. Stenger. *Model-Based Hand Tracking Using A Hierarchical Bayesian Filter*. PhD thesis, University of Cambridge, Cambridge, UK, 2004.

[58] D. Stotts, J. McC. Smith, and K. Gyllstrom. Facespace: endo- and exo-spatial hypermedia in the transparent video face top. In *15th ACM Conference on Hypertext and Hypermedia (HYPERTEXT)*, pages 48–57, 2004.

[59] UPT ToF 3D Hand Gesture Database. `http://www.ea.etc.upt.ro/UPT-ToF3D-HGDB.html`, 2012.

[60] C. Vogler and D. Metaxas. A framework for recognizing the simultaneous aspects of American sign language. *Computer Vision and Image Understanding*, 81(3):358–384, 2001.

[61] J.P. Wachs, H.I. Stern, Y. Edan, M. Gillam, J. Handler, C. Feied, and M. Smith. A gesture-based tool for sterile browsing of radiology images. *Journal of the American Medical Informatics Association*, 15(3):321–323, 2008.

[62] C.C. Wang and K.C. Wang. *Recent Progress in Robotics: Viable Robotic Service to Human*, volume 370 of *LNCIS*, chapter Hand posture recognition using adaboost with SIFT for human robot interaction, pages 317–329. Springer Berlin, 2008.

[63] F. Wang and C. Zhang. Feature extraction by maximizing the average neighborhood margin. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[64] M. Yang, Y. Wu, and G. Hua. Context-aware visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(7):1195–1209, 2009.

[65] B. Yao and L. Fei-Fei. Grouplet: a structured image representation for recognizing human and object interactions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9–16, 2010.