Universidad Autónoma de Madrid

Escuela politécnica superior

Proyecto fin de carrera

Estudio de descriptores visuales aplicados a la interpretación de gestos manuales

Laura de las Heras
Diciembre de 2011

Estudio de descriptores visuales aplicados a la interpretación de gestos manuales

AUTOR: Laura de las Heras
TUTOR: Javier Molina Vela
PONENTE: José María Martínez Sánchez

Video Processing and Understanding Lab
Dpto. de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Diciembre de 2011

Resumen

Resumen

En nuestra vida cotidiana estamos acostumbrados a experimentar el fenómeno, fruto del constante avance tecnológico, de la sustitución por obsolescencia de muchos dispositivos por otros tecnológicamente más avanzados, de menor coste, o con mejores características de uso o manejabilidad. De esta forma, controladores físicos (mandos por cable, teclados, ratones, interruptores, etc.) normalmente empleados en la comunicación hombre-máquina, tanto videoconsolas, como ordenadores e incluso nuevos dispositivos domóticos están siendo paulatinamente sustituidos, primero por dispositivos inalámbricos y, recientemente, por aplicaciones basadas en el reconocimiento de imágenes recogidas por cámaras 3D. El convencimiento de que esta nueva tecnología se ha convertido, de facto, en el nuevo estándar para el manejo de cualquier dispositivo inteligente es la que me ha impulsado a acometer este proyecto, el estudio de descriptores gestuales estáticos sobre capturas de imágenes 3D.

A lo largo de este proyecto se han implementado dos extractores de descripciones: Tanibata [1]y Roussos [2]. El primero de estos descriptores se basa en la descripción de imagenes mediante la extracción de caracteríticas de una elipse alineada con la palma de la mano, mientras que el segundo describe las imágenes con los pesos obtenidos al ser proyectadas en un base construida a partir una subselección de imágenes. Tras realizar algunas modificaciones sobre los descriptores propuestos, éstos han sido implementados y evaluados con el fin de encontrar un modelo de manos que permita reconocer posturas estáticas, con independencia a la variación que presenten las imágenes capturadas en su iluminación, escala u orientación. La evaluación de la capacidad de separación de estos descriptores se ha hecho sobre una colección compuesta de capturas sintéticas y de usuarios reales, previamente compilada en el VPU-Lab.

Conforme al estudio hecho en este proyecto, se incluye un repaso detallado sobre todos los procesos seguidos durante la extracción de las descripciones para cada uno de los descriptores implementados. Acompañado por resultados cuantificados tras las simulaciones, este estudio recorre desde la captura de imagénes con cámaras de profundidad TOF (Time Of Flight), hasta la descripción de las imágenes mediante los parámetros definidos por cada descriptor, pasando por la segmentacion y alineación afín requeridas para el preprocesado de las imágenes. De esta manera, cada descriptor ha sido evaluado sobre una colección de gestos manuales grabados por 11 usuarios diferentes con 3 diccionarios distintos. El entrenamiento se ha realizado con imágenes sintéticas. Este es un marco de evaluación muy exigente y ambicioso que justifica en gran medida la diferencia con los resultados del estado del arte, que rondan el 90% de acierto.

Palabras Clave

Interfaces de usuario, Reconocimiento de gestos manuales, Transformación Afín, Tanibata, PCA, eigenhands.

Abstract

Nowadays the use of wired controls in Human-Computer interaction (HCI) is being superseded in game consoles, personal computers or domotic devices. Wireless controls, such as 3D cameras, are getting more and more popular for controlling devices. Assuming this fact, the main objective of this project has been the study of descriptors for recognizing static hand pose in depth images. Along this project two descriptors have been considered: Tanibata [1] and Roussos [2]. The first is based on the use of the characteristics of the ellipse of inertia of the captured hand, while the second describes images with the weigths resulting from their projection to a previously generated hands base. After introducing some modifications to these descriptors they have been implemented in order to evaluate them with a collection of images with synthetically generated and real users hand poses, compiled by VPU-Lab.

According to the study performed along this project, a detailed description of all the followed processes for the descriptors extraction is presented. Apart from the numeric results obtained on the basis of the extracted descriptors, the project also introduces the capture, segmentation and aligment stages. Each descriptor has been evaluated with the videos captured by 11 users, for 3 different dictionaries. The training was performed with synthetically generated images. This is a very exigent and ambitious evaluation framework that justifies the difference with the State Of Art results, that are around 90% of accuracy.

Key words

User interfaces, Object recognition, Pattern recognition.

Agradecimientos

En primer lugar me gustaría agradecer a las cuatro personas más importantes de mi vida su apoyo durante estos largos 6 años y pico de carrera. Ellos son mi madre Conchi, mi padre Rafa, mi querida abuela Marita y por supuesto, mi abuelo Isidoro.

Mamá, gracias a ti soy la persona en la que me he convertido. Gracias por tu constante apoyo, por tu cariño, tus cuidados, tus consejos, tu gran paciencia y sobre todo gracias por ser como eres y haber intentado hacerme el camino lo más más corto posible. Sin ti no habría llegado hasta aquí, ni sería quién soy ahora. Gracias.

Papá muchísimas gracias por toda la ayuda que me has dado y me das siempre, por tu paciencia y por tu atención cada vez que la he necesitado. Quiero que sepas que cada día intento hacer mejor las cosas para que puedas estar orgulloso de mi tanto como lo estoy yo por tenerte como padre.

Mi abuela, mi segunda madre, otro de los grandes pilares de mi vida. Sabes que crecí contigo, que me dedicaste gran parte de mi vida y que lo mínimo que puedo hacer ahora es dedicarte el primer fruto de todo mi esfuerzo. Te quiero muchísimo. Ahora me toca recompensarte un poco toda la dedicación que invertiste en mi.

Abuelo, con sólo pensar en ti se me llena el corazón de agradecimientos. Cada examen que he superado, cada traba que he tenido que afrontar a lo largo de la carrera he conseguido superarla por ti, por la fuerza que me has dado desde ahí arriba. Lo que más feliz me haría es poder verte sonreír una vez más, orgulloso de tu nieta, lo cual siempre ha sido la fuente de mis fuerzas, el remolque de mis energías para no ceder ante los baches. Ojala pueda agradecértelo algún día.

También me gustaría agradecerle a mi tutor, Javier, toda su ayuda durante el tiempo que he estado en el laboratorio. Quiero recalcar que a pesar de su serio semblante, encierra una gran persona, siempre dispuesta ayudar cuando lo necesitas. Gracias por haber acudido siempre cuando tenía dificultades.

Quiero agradecer también el apoyo de todos mis amigos, en concreto a Silvia, Isa, Claudia, Lore y Bader que me han acompañado todo este tiempo tanto en la Universidad como fuera de ella, animándome para no decaer y escuchando mis problemas. Gracias. También quiero hacer mención especial a Julius, el cual ha sido durante mucho tiempo compañero de laboratorio y de eternas peleas con el ordenador. !Conseguimos acabar antes del 2025!

Otra de las personas a las que debo el haber podido acabar esta carrera es a Rafa, quién supo hacer de las duras épocas de exámenes y de prácticas un poco más llevaderas y que me dio en la mayoría de las ocasiones el impulso que necesitaba para seguir esforzándome.

Gracias también a Chema por haberme ayudado en este último tramo tan duro del proyecto, así como a Jesús y todos mis profesores a lo largo de la carrera.

Por supuesto también debo mi esfuerzo al resto de mi familia, como mi hermana Lara, Mª Jesús y Miguel, mis abuelos Ana y Alfonso, así como a todos mis tíos y primos.

Gracias a todos.

Índice general

Índice de figuras

Índice de cuadros

Chapter 1

Introduction

1.1 Motivation.

During last years, gesture interfaces have become an important technique with many applications in our daily life. The manufacturers of the most popular gaming consoles, such as Microsoft Xbox-360[1], SONY PlayStation3[2] or Nintendo Wii[3] are working in developing recognition gesture systems for using them in gaming, leisure and multimedia environments. The most widely known recognizing gesture project applied in consoles comes from Microsoft, that has replaced in the Kinect (see Figure 1.1) a game console already on the market, which has replaced the usual controllers by the movements of the user by means of a 3D camera based on the Time Of Flight (TOF) technology (a camera that produces images with an intensity level inversely proportional to the depth of the objects observed).

Capture systems of yesteryear had prices too high prices to find their place in the mass market. However, the lowering of prices in current systems, such as 3D cameras, have however boosted the development of the necessary software for implementing new applications which can work with 3D capture , exploiting the depth information provided by these kinds of cameras. Several applications have been developed during last year allowing users to interact without the needing of a physical controller, devices such as computers,

[1] http://www.xbox.com/kinect

[2] http://playstation.com/psmove

[3] http://wii.com

Figure 1.1: Kinect camera (image taken from http://kinectforwindows.org)

consoles, home automation systems and any kind of multimedia environment. Moreover, the development of new technologies has always been focused on reducing dimensions in electronic devices as well as on making them easier to use. In fact the evolution of the technology began with the replacement of wired by wireless devices and now it is still putting lots of efforts on reducing the most of the elements required in users-machine communication. Some examples can be found in the HCI (Human Computer Interaction) methods tendency in the last decade, when most of the direct communication was made using remote controllers, such as TV remotes or console remotes. More examples of this wireless set of sensors are gloves or markers, which allows the mapping of hand or body in a 3D virtual space. Another research line in HCI is the recognition of voice commands. Nevertheless, the best obtained results by the researchers focused on the reduction of interaction devices are associated with 3D captures of the scene. Thus 3D video processing has become popular for hand/body tracking and later gesture recognition.

This project is focused on the study of hand modelling via volumetric surface descriptions which allow an efficient recognition of static hand poses. In advance, these hand models will be used in the implementation of real time applications which will allow users to communicate with their computers.

However the main problem when designing a hand descriptor is the achievement of a robust model against rotation, scale and shift of the hand. For this reason it is highly valuable the capacity to identify the wrist region with the least possible error, since these will permit the elimination of the forearm from the image and will be useful to perform rotations of the input 3D surface.

In fact there are multiple descriptors, but not all of them, entirely satisfy previous invariance requirements. Therefore it is needed to find out a descriptor which fulfils the most of the above mentioned characteristics to model hand gestures. In addition most of the descriptors studied during this project make use of certain restrictions during image capture to limit the problems of rotation, scale and shift.

Some of the advantages of working with 3D images, which will be extensively explained later, are the following:

- Improvement of hand segmentation.
- More information of interest gathered from the pixels of each image due to the depth information provided by the 3D capture.
- Improvements in hand features representation and recognition of 3D data.

A still unsolved problem of this kind of technology consists on the occlusion of most of the real volume associated to the hand, since these cameras only capture a volumetric surface of the volume. In fact, the occluded points from which no information is obtained during the capture cannot be modelled. A solution to this problem could be the use of a multicamera system with could be able to capture the whole volume rather than a volumetric surface.

After explaining some of the advantages provided by the 3D captures systems, some HCI applications based on the hand description can be pointed out:

- Computer managing without keyboard [7].
- Drawing in the PC without a mouse or a board [4].
- Home Automation equipment managing without switches [8].
- Translation of sign languages [3].

Summarizing, we can say that this technology is useful for the development of whatever generic sign recognition system applicable to different fields as:

Communication languages, including sing languages [9] or games or leisure platforms, where hands can replace remote devices in the control of consoles or/and computers.

1.2 Objectives.

The main goal of this project is the study of different hand descriptors, as well as implementing their extractors. This study will consist on the evaluation of the descriptors in terms of separation in poses capacity. These separations will be performed over different collections of captures: synthetic and real, with different range variations in scale, positions of the hands and rotation of the point of view (POV).

The descriptors for this study will be selected from the State Of Art, considering the different approaches to hand description. Then descriptors will be analysed for its usefulness for the characterization of hand poses, and the chosen ones will be implemented and tested. Some implementation concerns were taken into account during the evolution of this study, and different action lines were defined to achieve a robust detection system designed for being used in real time applications:

- Application of segmentation techniques based on depth information, to remove forearm regions from images and for subsequent alignment.
- Definition of the thresholds required by the descriptors implementations.
- Description extraction and later selection of a subset of features invariant to scale, rotation and shift.
- Training of the detection system with machine learning techniques to later get predictions of image descriptions.
- Evaluation of pose detection for each considered description and over dictionaries with different characteristics and with different configuration parameters.

Previous tasks will be followed by an exhaustive study of the effectiveness of each implemented descriptor, which will lead to the introduction of possible improvements.

1.3 Structure of the document.

The content of this document is organized to attend the objectives were described in the previous Section

Therefore the first part of the document, Chapter 2 is focused on the description of existing solutions related to topics considered in this project: this is bibliography related with capture technologies, hand gestures data sets, image processing techniques and descriptors to model static hand poses. Moreover, a whole Chapter 3 is addressed to the data set involved in the study of the hand descriptors of this project. This will be followed by Chapter 4, with the explanation of the segmentation and alignment used to prepare images for the extraction of their descriptions. One of the most important parts of the project is included in Chapter 5 where Tanibata and Roussos descriptors are described in detail, including an overview of the papers in which they were firstly introduced as well as a description of the decisions made during their implementation. The last Chapter of the project, 6, contains the results obtained in the evaluation stage, using different setups. Besides the conclusions obtained from this study, some possible improvements are proposed as future work lines.

Chapter 2

Related Work

2.1 Introduction.

The task of this project has been studied in many applications, where the data set as well as the technologies and processes involved in the implementation are multiple and different. Therefore, the main goal of this Chapter is to make a brief introduction to the Hand Descriptors and all related topic to them. In addition, next paragraphs will be focused in the description of the existing data sets, captured technologies and segmentation techniques as well as the detailing of some proposed descriptions to model images.

2.2 Data Sets

The set of dictionaries or kinds of Sign Language which have been used along the related works over the recognition of hand postures is quite wide. Furthermore, it includes existing and well known classes, such as the deaf and dumb Sign Language [10], as well as new models of languages created to be used within the context of machine-human interfaces, such as architectural hand signs (AHS) [4].

Some examples of sign languages referenced to communication languages of Deaf community are American Sign Language[11][12] [13][14][15][16][17], Arabic Sign Language [18], Australian Sign Language [19], Signing Exact English [20], Chinese Sign Language [21], Japanese Sign Language [1], Dutch

Sign Language [22], French Sign Language[23], German Sign Language [24], Italian Sign Notation System [25], Irish Sign Language [26] , Polish Sign Language [27] and Korean Sign Language [28].

On the other hand, we can find some kinds of languages related to the computer design applications as the above mentioned AHS orientated to the generation of 3D architectural model[4].

2.3 Capture and Segmentation

There are two principal bullets differentiated between the tasks of Hand descriptors:

- TheCapture and preparation of input images to be described
- The extraction of the descriptor related to each input image.

In this section, the related works about the first set of the tasks is described. Several existing solutions will be covered,begining with the current capture technologies which extract images to be described from the videos performed by real users, and ending with thesegmentation and images processing of the captured set requiredto arrange and prepareimages to be described.

2.3.1 Capture Technologies

In this first phase belonging to the hand recognition system of the captured images, there are different kinds of technologies which classification usually depends on the hardware of the sensor and the colour of the filters used during the acquisition of the images. Therefore, as a whole view of capture systems, their classification includes the following differentiated technologies:

- Full colour image Capture, which are known as traditional cameras:
    - These cameras generally use two different types of sensors: CCD or CMOS. The CCD or Charge-Coupled Device is composed of several joined arrays of photosensible devices with the capability of storing charge.

  - Moreover, the kind of filters they use, are Bayer filters or independent image sensors (one for each colour).
  - The colour capture could be obtained using CCDs depending on the quality wanted:
    - The maximum colour quality can be obtained at a higher cost by dividing the light into its three components, the primary colours (RGB), and the using of a CCD for each component.
    - In cases where the capture time is high, the maximum colour quality can be kept by the use of a rotatory colour filter.
    - Colour filter arrays is a technique easier to be used which consist on setting a colour filter in front of each pixel, with the counterpart of obtaining less resolution and quality colour than with the CCD device.
  - This kind of capture process have been included in several systems: J. Van Despielberg describes in its “Analog VLSI implementation of neural systems” [29] the features, the design and the implementation of a foveated retina-like sensor performed with CCD technology, and also the results from the study of the performance of this sensor for the 2D pattern recognition and object tracking. Moreover, CFA or colour filter arrays are combined with a single sensor to give measurable features of the captured images due to the use of interpolation CFA algorithms applied by most of the digital cameras. Traces of digital tampering in colour images can be detected attending to their specific correlations introduced by interpolation of the colours [30]. Gijs Molenaar proposed a real time method for estimating hand poses in video by the use of a current RGB camera [31].

- Captures made with *Time-Of-Flight* (TOF) cameras. These kinds of cameras obtain a depth image from the capture. The process used consists on sending an infrared signal and timing how long the reflection takes to arrive. So this information let the camera make an estimation of a depth map.

  - Several developments for hand recognition systems have used time of flight cameras: Pia Breuer measured 3D surfaces points from the user´s hand using an infra-red time of flight range cameras in order to implemented a natural man-machine interface [32]. Using a standard video camera and a DLP projector, researches from Stanford University developed a real time structured light range scanning based on coding the boundaries between projected stripes to determine depths [33].

Figure 2.1: ToF camera:SR4000

- In the University of Tokyo have been developed a scanner for 3D human-machine interface which uses a laser diode combined with steering mirrors and a non-imaging detector to generate an active tracking system [34]. This laser scanner can acquire three dimensional coordinates in real time without the need of image processing at all.

- A Position Sensitive Device and/or Position Sensitive Detector (PSD) is an optical position sensor, that measures the position of a light spot in one or two-dimensions on a sensor surface. These devices are used in CCD and CMOS cameras as discrete sensors. Moreover, PSD sensors besides SOKUIKI sensors haven been combined with fuzzy algorithms to construct an operation assists system which prevents collision with obstacles for wheelchairs users [35]. Furthermore, PSD camera have been used in combination of neural networks and trapezoidal motion planning method to implement a real time visual servo tracking system for an industrial robot [36].

- Other authors have prefered combining the two capture techniques RGB and ToF cameras for 3D Hand Gesture recognition in a real time interface in [37]. The RGB module is used to determine the face region, then the depth information gathered by the ToF camera is projected to discriminate that region from the background and detecting finally the hands from remaining pixels by the use of colour restrictions.

2.3.2 Hand Segmentation

The segmentation process is applied after the capture stage, so the method used will be different depending on the followed capture technique. thus, two types of segmentation can be distinguished from current techniques:

- Segmentation based on colour information of images which have pixels over the same plane of the image. One of the characteristics of this kind of segmentation is the working in a well-controlled area where the involved currently processes include background extraction, skin-colour region search, etc. Nevertheless, the colour and the luminance is not a reliable measure for segmenting skin pixels due to theirs variation depending on the light source during the capture. Some techniques used based on colour segmentation are the next.

  - One of the current method used in this kind of segmentation consists on generating a skin model which allows differentiate and classify face and hand regions from the image. There are multiple chromatic colour spaces, such as LAB, HSV or normalized RGB. Several Hand Poses or Face Detectors creates the skin colour distribution after detecting the location of the face in the image using Haar classifiers, like in [31] and in [38]. This method obtains the average of pixel intensities within adjacent rectangular regions at a specific location in a detection window and calculates the difference between regions enclosed in that window. This difference can be used to classify subsections of the image as well as to create a generic skin colour model. Furthermore, in Viola-Jones object detection framework this difference is compared to predefined threshold in order to detect skin regions for each window set over the image. Moreover, the more number of Haar-like features are used to describe an object, the higher accuracy the face location will have. In the case of Hand detectors, after the face location and the generation of a skin colour distribution, pixels from hands can be located and extracted from background region.

  - Other researches have used a different transformation to extract the colour map of the image, although the method followed is similar to the previous one. This is the case of Yining Deng, who proposed a pixel transformation based on colour class labels after the quantification of these pixels [39]. This colour quantization is included in the called method JSEG, which goal is to segment images and video sequences. The second step followed to the quantization of the pixels is the spatial segmentation which consists on given high values to possible boundaries meanwhile low values are given to texture coloured pixels from each local window of the image. For video sequences segmentation, previous processes are combined with additional region tracking scheme. Another publication suggested spatial temporal segmentation to recognize gestures in video sequences in [13]. In fact, new spatial temporal algorithm matches are used here to find candidates in the hand detection process and also the combination of a classifier-based pruning framework and a subgesture reasonable algorithm are defined in this work to allow reflecting false candidates. In addition, Bayes decision theory is used to the creation of a skin model colour in [40]. Nevertheless, this method generated two models for hand and background colour model from each analysed image based on the Gaussian mixture model combined with the restricted expectation–maximization (EM) algorithm. Therefore, each pixel from the image can be classified as a hand or background pixel.
  - There are authors as R. Kjeldsen and J. Kender [41] who use histogram structures to identify target colours trained in real-time captures in order to separate hand from cluttered background.

- Segmentation based on depth or 3D information. Techniques included in this class use distance of pixels to the camera in order to segment undesired regions like the background. There are different lines of work used to make the segmentation depending on the capture method.

  - The segmentation based on the combination of N images at the pixel level or with 3D information belongs to multicamera techniques or stereoscopic vision. Sometimes it requires the use of gloves or markers during the capture which represents marked regions in white colour, making easier their extraction. Several researches have applied these kinds of techniques. In the study of lips and hand movements recognition for Cued Speech applications, blue marks are placed in both lips and fingers to be captured by cameras and to obtain distance between both references points [3]. In the research [42, 4] mentioned in the previous Section 2.3.1 where hands were captured using markers and multiple cameras to generate a 3D architectural model, the segmentation of initial sketch data is made by finding some key points where the curve changes noticeably its path direction. Moreover, the use of information gathered by glove-based sensors allows difference easier hand regions from the background, like in [5], although it requires a long calibration as well as complicated set up and it is also difficult for users to interact with the controlled computer. NASA is currently developing a virtual training environment called Virtual GloveboX (VGX), which has been used by several researchers like the authors of “Global Hand Pose estimation by multiple camera ellipse tracking” in [43]. This article describes a new algorithm for the hand tracking and 3D global pose estimation which uses an elliptical marker (glove) placed in the dorsal part of the hand besides an active camera selection to track user´s hand inside the VGX.

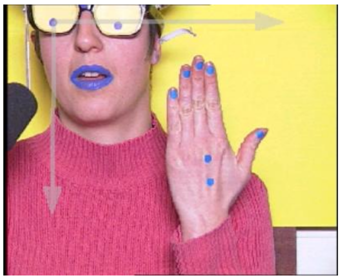


Figure 2.2: Picture from the segmentation in [3]

  - The segmentation of the images captured with TOF cameras, which could be considered a kind of 3D information capture, consist on extracting the nearest points to the camera using depth information and removing the farthest points from the image. An example of this type of segmentation is[44].

- The same research which combined RGB technology with ToF cameras mentioned in the previous Section 2.3.1 in [37], uses depth information associated to pixels belonging to face region in order to remove background from the image. This combined technique not only improves detection rates, but also allows the hand to overlap with the face or with hands from other persons in the background.

2.4 Hand Descriptors

There are multiple kinds of hand descriptors aimed to describe hand poses attending on the features and characteristic that differentiate better each possible pose of the hand image. The chosen descriptions will depend on the environment of the capture, such as real time images or static hand gestures; as well as the source of the data set, if analysed images belong to a deaf-mute language or by the contrary they belong to an architectural model. Moreover, it has to be considered that some gestural interfaces gives the user a higher degree of freedom during the conditions of the initial data capture, like in orientation of hands or the distance during the capture. The more freedom during the capture process, the more complex the descriptor is to achieve independence of scale, rotation and distance of hand captured. Moreover, the dependence on the kind of data means that the more variability and differences of the hand poses is included in a specific dictionary, the easier the distinction is.

Although descriptors studied in this project are focused in static gesture models, related works of dynamic gestures modules are also included in this Chapter. In addition to the lack and existence of movement in gestures, there are lots of kinds of descriptions depending on the parameters used to model hand poses:

- Shape descriptor: the stored information to describe hands are associated the geometry of the hand over the image. Features as the distance of the points of the image to the centroid of the image, the angle of the image with the horizontal axis as well as the approaching of the image to a specific geometric shape, can be used as the parameters of the hand description. One of the descriptors studied in this project answers to this kind of model [1], using parameters as the flatness, the direction of the hand or the number of protrusions as features of the description.

Another kind of model to pattern recognition is the decision-theory approach based on distance classifiers such as kinematic features used to recognize and to represent movements from human hand gestures extracted from a monocular temporal sequence of images in [11]; a real-time hand gesture recognition system to simplify the interaction with in-car devices in [45]; and a detector of static and dynamic gestures using depth information from fingers to apply the distance classifier and static models in[46].

- 3D descriptors: The recognition of sign gestures from isolated 3-D hand trajectories can also be based on the combination of classifiers for hand shape, hand movement and hand location, like in the Fisher's linear discriminant model. This model has been used to classify SEE hand shapes acquired by the CyberGlove and magnetic trackers in [20].This kind of descriptor based on acquired data using gloves or multiple cameras for hand position and finger configuration is combined with hidden Markov models to mitigate the time and gestural variations among different articulations of the signs [10]. Furthermore, 3D architectural models based on hand motion and gesture are developed by a motion capture system based on markers set on the left hand. In addition, two skeleton templates are generated from this 3D design information and after applying hand gesture detection, 3D motion sketches associated with a Marker-Pen on the right hand are used to generate 3D models of buildings[4].

- Template matching: Other descriptors use appearance-based features as well as tangential distance measures to recognize gestures within

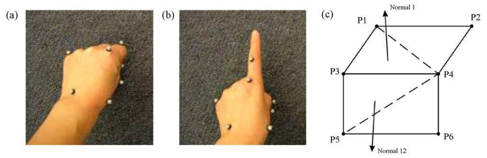


Figure 2.3: Gesture Recognition [4]

the framework of template matching classifiers like in[24]. The hand shape, movement, and location of the hand, can also be used as 3D features to describe different signs motion of images [47]. Furthermore, these three hand features are used in [14] to recognizing hand signs. In this method the recognition of the motion is tightly coupled with the spatial recognition (i.s the hand shape). The proposed system uses multiclass and multidimensional discriminant analysis to automatically select the most discriminating linear features for gesture classification.

- The Dynamic Time Warping Model (DTW): This algorithm measures the similarity between sequences which are delayed in time or speed. Moreoverit finds an optimal match between these sequences regarding to specified restrictions. Therefore this sequences alignment method, which is often used in the context of hidden Markov models, can be used for speech recognition modelling by merging segmenting subunits within the sign language [26]. In the recognition of human movement patterns within the framework of classification problem, a variation of the dynamic time wrapping model has been used to match movements patterns using 3D jointly angles as features [48].Besides Markov models, Bayesian Networks allows construct a multilevel architecture based on the semantic context to analyse the correctness of a sentence given in a sequence of recognized signs like in [25].

- Time-delayed neural networks (TDNN): This algorithm allows to extract and to classify two-dimensional motion in an image sequence based on motion trajectories [12]. Basically, it finds pixels which match along different frames of a sequence and concatenates them to obtain pixel-level motion trajectories. Finally, different trajectories are learned by these time-delayed neural networks.

- Hybrid models that combines previous ones: The use of K Nearest

Neighbour combined with Bayesian classifier in[49] allows to recognize isolated sign language gestures. The proposed method extracts temporal features through predictions, then the motion is represented into one image using threshold prediction errors and therefore, spatial-domain features are extracted from it to represent the whole video by a few coefficients. The linear separability of the extracted features is assessed and complemented by these simply classification techniques K nearest neighbour (KNN) and Bayesian.

Another set of mixture algorithm is the combination of Least-Squares Estimator with Adaptive Neuro Fuzzy Inference System network (ANFIS) which has a learning capability to approximate non linear functions [18]. This descriptor uses extracted features such as rotation, scale, and translation invariant of hand images to describe gestures. Furthermore, the subtractive clustering algorithm and the least-squares estimator are used to identify the fuzzy inference system, using the hybrid learning algorithm for the training stage, allowing to recognize the 30 Arabic manual alphabets with an accuracy of 93.55%.

In addition, Independent Component Analysis is combined with Markov chains in a 2 stage classification in the research [50], meanwhile in [22] hybrid statical classifier (DFFM) is combined with the Dynamic Time Warping Model (DTW) to demonstrate that time warping and classification should be separated to achieve better results in modelling 3D hand motion features.

The combination of the self-organizing feature maps (SOFM) which extracts different signers' feature and transform input signs into low-dimensional representation, with continuous hidden Markov models (HMM), which models the transformed image by the emission probabilities is used in [21].

Fourier Descriptors: This method obtains the Fourier coefficients from a chain-encoded contour. Elliptic properties of the Fourier coefficients are used to normalize the Fourier contour representation in[51].

- Moreover, multi-scale colour image features are used to describe hand postures at different scales, positions and orientations. By the use of a particular kind of filtering hands are tracked after the detection of multi-scale colour features for each image, based on the hierarchical layered sampling. This algorithm for hand posture recognition has been developed in [52].

- Support Vector Machine (SVM) is also used in systems to recognize multiple-angle hand gestures by their training using images of hand gestures which present different angles.[53].
- Principal Component Analysis: This method is used in several sign recognition systems which works with time of flight cameras, like in [32], where a hand is wanted to be reconstructed from data stemming using a model based on fine-matching. PCA is used in this method to obtain a crude estimation on the location and orientation of the hand associated to the first 7 Degrees of Freedom of the reconstructed hand. In the other hand, PCA is also used in Roussos method [2]to extract the minimum number of vectors required to describe hand gesture images. In addition, this descriptor is the second implemented model of the project, so it will be explained in detail in Chapter 5. The main idea is to describe images with the eigenvalues required to project images in a new hand base which principal elements are chosen from the application of PCA analysis.
- Euclidean space: Binary edge images are transformed into a high dimensional Euclidean space by the calculation of their chamfer distance from the cluttered image. Then, the problem of hand pose estimation is turned into an image database indexing problem, where the input image is compared to a large database of synthetic hand images to find the closest matches between them. This descriptor uses a probabilistic line method which identifies those line segment correspondences as the least likely to have occurred by chance[54].

Chapter 3

Data Set

In [55], an experiment with real users was conducted to define a gestural dictionary that allows users to interact with a system in a natural way. The main goal of that work is to perform a study of the the most suitable gestures attending to their usability in users gesture performance.

In that experiment, users are firstly trained by the system using real gestures captures and graphic models or references called metaphors. These models allow the system to recreate in the natural interaction space those gestures wanted to be executed by the user during the capture stage. Moreover, the use of this real/virtual and horizontal/vertical references avoid the capture of users´ gestures without any correspondence associated. Examples of common interactions movements in computer systems with real metaphors are rotation, grabbing or catching as well as examples of non-real metaphors are 'cancel' or "undo" actions. Finally, recorded data combined with the use of different types of metaphors and rotations of the interaction screen is analysed.

The amount of interaction possibilities with the metaphors is very high, but the experiment identifies the most frequent ones and its associated hand gesture. Those gestures and interactions that can improve the user experience without fatiguing him are selected for the recognition stage. Therefore, among the complete set of gestures obtained from the experiment a subset was selected to define the dictionary used in [56] based on a trade-off between usability and recognition. Apart from this dictionary, some other Static Hand Poses (SHPs) collection were compiled in the dataset proposed in our project by the VPU-Lab [1]. These collections are shown in Figure 3.7.

Apart from the real users data compilation (see Section 3.1) these dataset also includes a synthetic depth images generation method (see Section 3.2), which allows to create useful information without the need of users participation. In addition, synthetic images were used to train the machine learning software responsible of assessing studied descriptors. Thus, implemented hand models ensure the independence of the hand descriptors on users capture.

The work presented in this Section is in a review process and its previous to the work proposed in this document. Its use is limited to training and evaluation purposed, being its design previous to the elaboration of this project.

3.1 Real Users Collection

The collection of images used for the evaluation stage of descriptor implementations have been extracted from the set of videos generated in [56] paper. Two different classes of hand poses recording were extracted: Static Hand Poses (SHPs) and Dynamic Hand Gestures (DHGs). The last collection of postures are combined with motion in order to obtain a semantically richer dictionary of gestures, but they are not involved in the study of this project.

For recording the videos a TOF camera (SR4000 developed by Mesa Imaging [2]) was placed 1.5 meters above the floor, with an horizontal orientation orthogonal to the user. This camera captures depth images with QCIF resolution (176x144 pixels) and a depth precision of ±1cm. It was configured to capture 30 fps, and to operate in a 3 m depth range (0.3m-3.3m) in order to remove background objects. For this purpose, 11 users with different heights were asked to perform the recording sessions, making the collection certainly representative to show the potential system performance in dealing with different users. Moreover, the recorded users were not asked to keep a certain distance to the camera neither to perform the gestures with any speed restriction.

[1] www-vpu.eps.uam.es

[2] http://www.mesa-imaging.ch/

The principal features of the two kinds of hand poses recorders, SHP and DHG are the next ones:

- SHP: Static Hand Poses.
  - One static pose video captured from each user.
  - Each video contains 252 frames from the same hand pose.
  - Each user performs gestures of 6 different dictionaries defined in the section 3.3.
- DHG: Dynamic Hand Gestures
  - Five execution videos performed by each user.
  - Each gesture can be composed of a single or multiple static poses sequence.

Nevertheless, only SHP were used to evaluate hand descriptors after the extraction and segmentation of each video frame included in each dictionary.

3.2 Synthetic Collection

Following the structure of the data set gathered from real images, the same collection of Static Hand Poses were generated synthetically to be used during the training stage of each implemented model.

These images were compiled by the VPU-Lab [3] using the kinematic model and the definition of 27 Degrees of Freedom (DOF) implemented in [5]. The kinematic hand model cannot extract the correlation between joints in the hand but, in contrast, it can represent the motion of the hand skeleton. Moreover, this model defines the bones of the skeleton as rigid bodies joined each other by joints represented by one or more degrees of freedom in dealing with rotation configurations. Hence, the only drawback founded in this model is the need of a initial setting of the hand parameters in accordance with the user´s features. In the Figure 3.1 the skeleton hand structure and the kinematic model can be shown.

[3] www-vpu.eps.uam.es

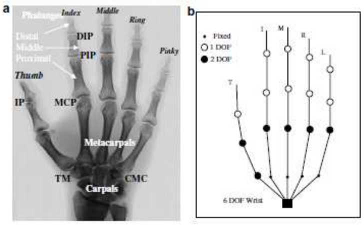


Figure 3.1: Kinematic Model [5]

Therefore, the fingers are modelled as planar kinematic chains attached in serial distribution to the palm at 2 DOF joints, meanwhile metacarpal bones of the palm are connected to the wrist fixedly as a rigid body. In addition to the specifications defined to model hand poses as a collection of rigid and flexible bodies joined together, some restrictions about the inheritance motion and shape of hand have to be considered before generating these synthetic hands:

- *Hand pose or motion constrains related to motion models.* Based on biomechanics hand motion properties, two kinds of restrictions were specified in hand motion models. The first one includes the static constrains where the range of each parameter is defined. The second one involves dynamic constraints about joint angle dependencies. Restrictions covered here allow to generate hand appearances in arbitrary configurations with independence of the user.

- *Calibration procedure related to shape models and based on user dependence of measurement parameters.* Despite the huge configuration freedom given by this model, limits related to the computational efficiency do not allow to use complex shape models for pose estimation. Since the multiple projections of the model into input images are required to extract hand features and the multiple occlusion problems arisen in the model, the use of geometric primitives was increased in the generation of synthetic hands. Therefore, cylinders, spheres and ellipsoids are usually defined to generate the skeleton of the hand.

Once the model parameters have been defined and the shape and motion constrictions have been applied following this Kinematic model [5], images of our data set were generated by a volumetric hand via dilation process using

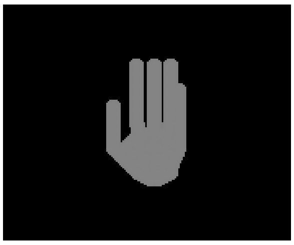


Figure 3.2: Generated Synthetic image based on 27 DOF kinematic model

a 3D morphological library [4], capturing later the range data image similar to the ones captured by Time Of Flight (TOF) technology. An example of a synthetic hand image resulting from the 27 DOF kinematic model is shown in the Figure 3.2.

Therefore, the final synthetic collection of images generated has similar characteristics to the set of real images previously explained. In addition, several sets of images were created for the same dictionaries defined later (see section 3.3) attending to different configurations of specific parameters. These parameters are the number of random samples taken for each gesture from the whole generated collection (from now on called number of points of view) and the increment between the rotation angles of that whole set of images (θ^x;θ^y;θ^z). After the generation of multiple sets of images based on different configurations of these parameters, each hand descriptor will use the collection which performs better in the training stage of the descriptor. The configurations used for the generation of the synthetic set of images are the next ones:

- Number of Points of view:
  - Synthetic images generated with 1 Point Of View. Each hand posture included in the dictionary has been chosen randomly from the whole collection of synthetic images. In Roussos descriptor this set of images is used to generate the base of the hand images required to model hand gestures (see section 5.3.3)..
  - Synthetic images generated with 9 POV. This collection includes 9 random samples of each hand pose from the whole set generated

[4] http://www.mmorph.com/

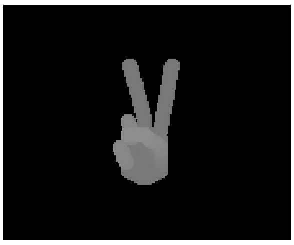


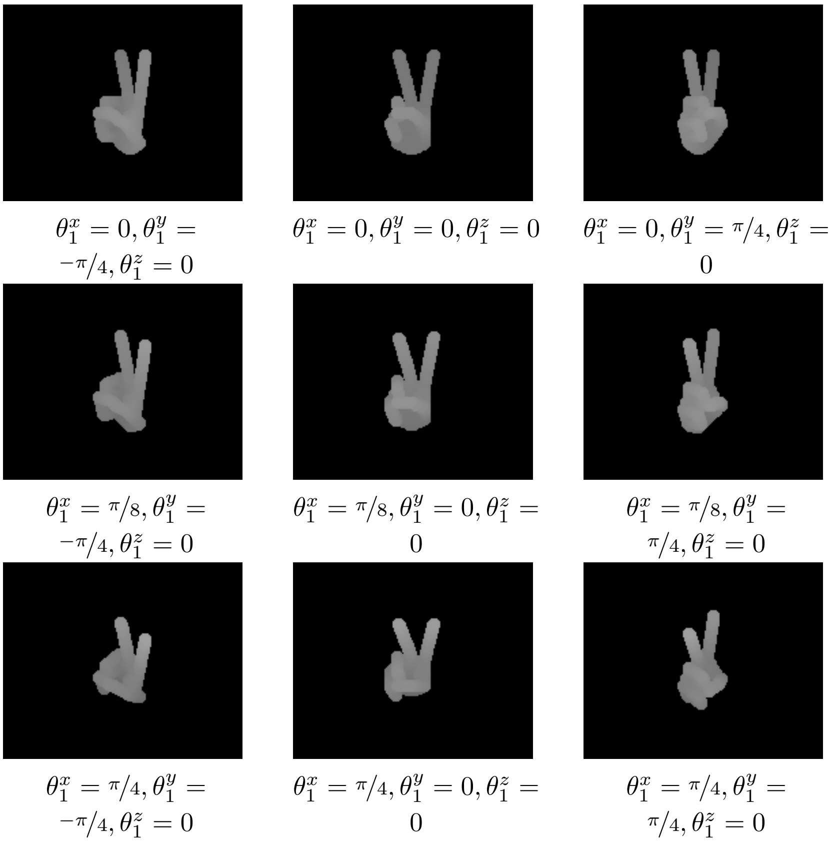
Figure 3.3: Enum2 gesture from the collection of Synthetic Hand Pose with 1 POV

with different rotations of the hand made over the three axis of the 3D space, x, y and z. This set of images is only used in Roussos descriptor to the generation of hand images belonged to the base of the description (see section 5.3.3).

  - Synthetic images generated with 200 POV. Images included in this set present 200 different configurations for each pose, attending to the rotations of the hand in each coordinate of the Cartesian system x, y and z. This configuration was combined with the following sets of images during the training stage of the description process of the two models.

- Noise $\pi/4$:
  - Synthetic images generated with 200 POV and intervals of $\pi/4$. This set of images is used to train models of both Tanibata and Roussos descriptors during the evaluation process, making them the most independent of the user that it is possible. Moreover, this collection of synthetic images is included within an angle range separated by intervals of $\pi/4$ covering different rotations made in each dimension of the virtual space. In the Figure 3.5, 15 random images of the enum 2 posture are shown from the collection of 200 hand poses stored for the data set of this project.

- Noise $\pi/8$:
  - Synthetic images generated with 200 POV and $\pi/8$. The goal of using this set of images is the same than the previous one, this is



$\theta_1^x = 0, \theta_1^y = -\pi/4, \theta_1^z = 0$

$\theta_1^x = 0, \theta_1^y = 0, \theta_1^z = 0$

$\theta_1^x = 0, \theta_1^y = \pi/4, \theta_1^z = 0$

$\theta_1^x = \pi/8, \theta_1^y = -\pi/4, \theta_1^z = 0$

$\theta_1^x = \pi/8, \theta_1^y = 0, \theta_1^z = 0$

$\theta_1^x = \pi/8, \theta_1^y = \pi/4, \theta_1^z = 0$

$\theta_1^x = \pi/4, \theta_1^y = -\pi/4, \theta_1^z = 0$

$\theta_1^x = \pi/4, \theta_1^y = 0, \theta_1^z = 0$

$\theta_1^x = \pi/4, \theta_1^y = \pi/4, \theta_1^z = 0$

Figure 3.4: Enum2 gestures from the collection of Synthetic Hand Pose with 9 POV.

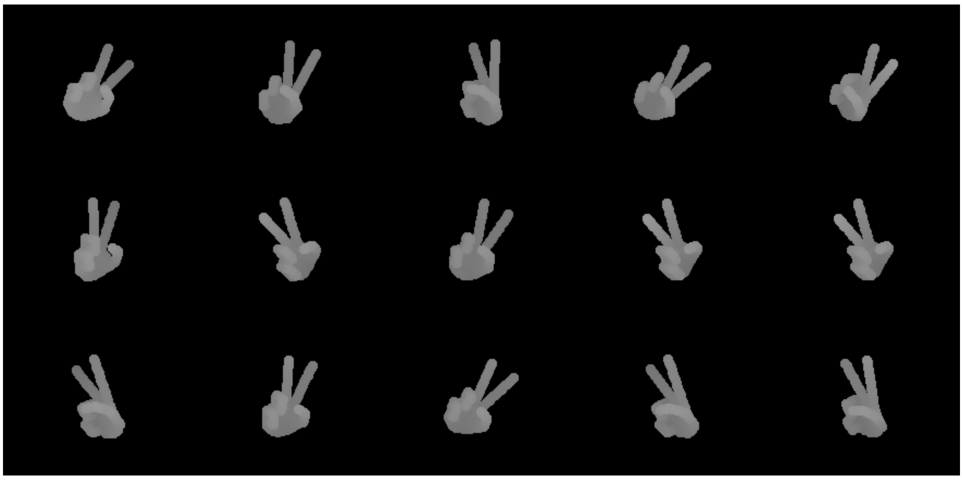


Figure 3.5: Gesture Enum2 from the collection of Synthetic Hand Pose with 200 POV with $\theta_1^x \in [0, \pi/4], \theta_1^y \in [-\pi/4, \pi/4], \theta_1^z \in [-\pi/4, \pi/4]$

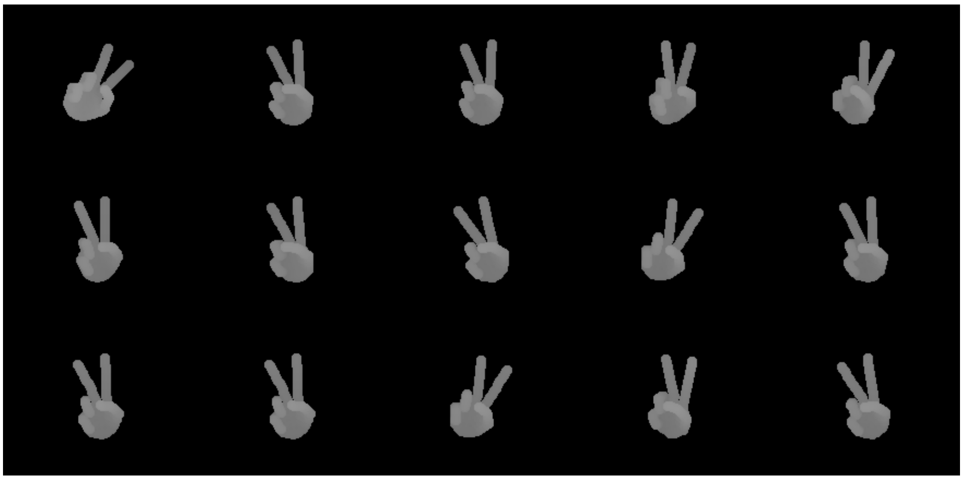


Figure 3.6: Gesture Enum2 from the collection of Synthetic Hand Pose with 200 POV with $\theta_1^x \in [0, \pi/8], \theta_1^y \in [-\pi/8, \pi/8], \theta_1^z \in [-\pi/8, \pi/8]$

training descriptors during the evaluation process. This collection of images also includes 200 random images from the previous hand pose collection generated in separated intervals of $\pi/8$ given by each rotation applied in all the possible directions. In the Figure 3.6 it can be shown 15 images from the collection of 200 poses from the enum 2 hand gesture.

3.3 Dictionaries

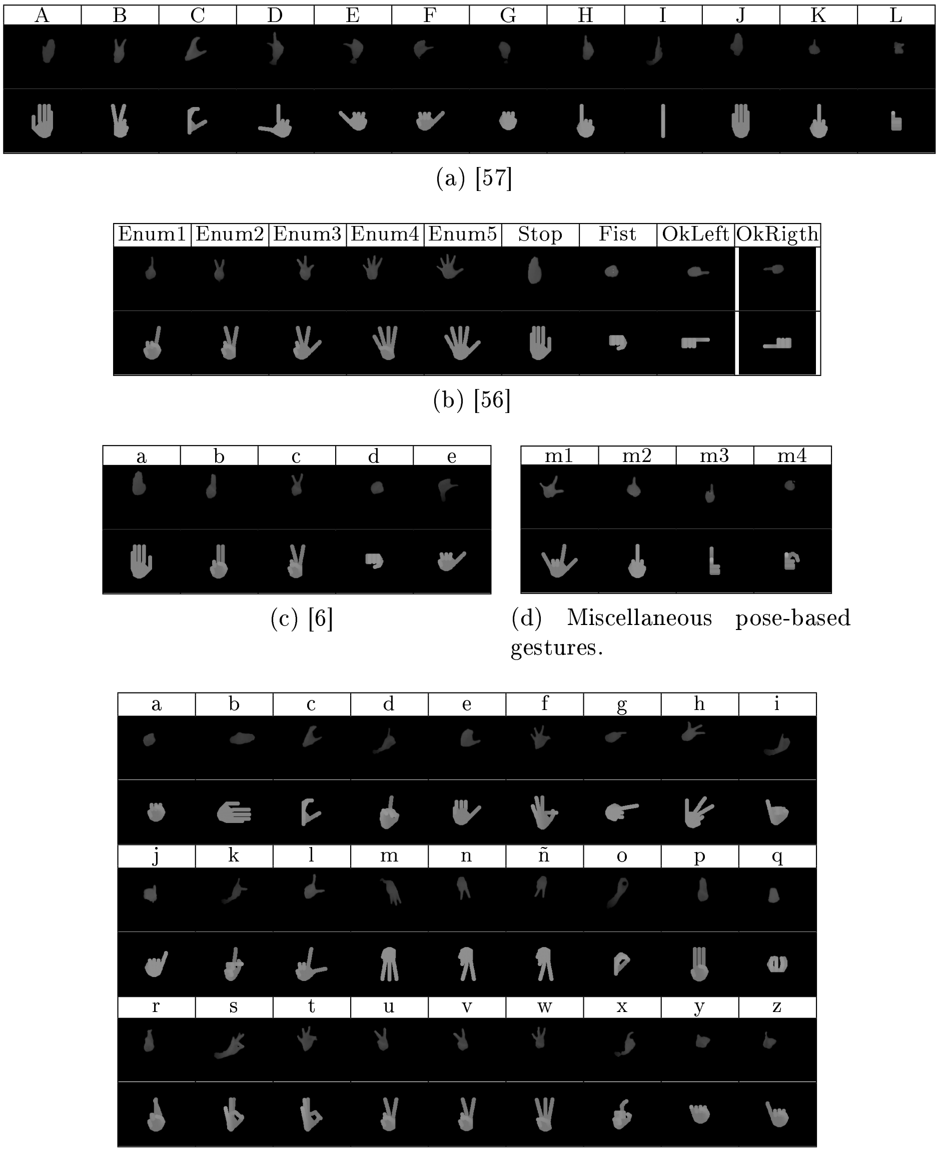
The Dictionaries generated from the poses included in our data set are the next ones:

- Kollorz: The collection of images included in this first dictionary shown in Figure 3.7a belongs to a subset of 8 images extracted from the set of 12 static hand gestures created in the article [57]. This selection of hand gestures was used by authors of this article due to the good quality of depth features present in captured images, which allowed them to be classified rapidly by a simple nearest neighbour classifier.

- Molina: This set of images contains gestures representative enough for a dictionary addressed to the human-computer interaction. In fact, this collection includes five numeric hand gestures as well as three semantic poses which combination provides a high amount of interaction possibilities. This dictionary is shown in the Figure 3.7b.

- Soutschek: The collection of five hand poses chosen for the execution of the experiment in the document [6] was aimed to medical intra-operative applications and it has been also included in the study of this project. The hand poses which compound this dictionary are represented in Figure 3.7c.
- Miscellanius:A set of four hand gestures generated in different orientations was included in this collection of hand images shown in Figure 3.7d.
- Spanish Sign Language: This dictionary contains 24 gestures from the set of 27 gestures defined in the deaf-mute Spanish sign language, which can be observed in Figure 3.7e.

In Figure 3.7 we can find captures for the hand pose-based gestures compiled in the data set. The first rows contain real images, while the second synthetically generated captures.



(a) [57]

(b) [56]

(c) [6]

(d) Miscellaneous pose-based gestures.

(e) Spanish sign language alphabet.

Figure 3.7: Captures from compiled dictionaries. First row of real images from static pose videos. Second row of synthetic images.

Chapter 4

Preprocessing: Hand Segmentation and Alignment

Every image processing technique requires of a segmentation and preprocessing phase before start working with acquired images. The preparation of the image for the later extraction of its description is the main goal of this stage. The segmentation of hand images depends on the capture process as well as the kind of images to be treated.

Regarding to the capture technology, the Time-Of-Flight (TOF) camera SR4000, developed by Mesa Imaging[1], was used to acquire the depth images of our real users data set (used in the evaluation stage in this project). Furthermore, the capture of these kind of images was performed by Video Processing and Understanding Laboratory (VPU-Lab), which also provided with the synthetic images collection of dictionaries shown in Figure 3.7 used as training images in this project. This dataset is explained in more detail in Chapter 3. Two different kind of videos were recorded: Static Hand Postures (SHP) and Dynamic Hand Gestures (DHG). Nevertheless, in this project we only use SHPs recording for evaluation purposes, since no temporal coherence is taken into account in this work.

The main objective of the segmentation is to remove noise and non-desired regions from the hand depth images to prepare them for the signal processes involved during the description extraction stage which will be properly explained in Chapter 5.

[1] http://www.mesa-imaging.ch

The preprocessing of images was performed just before the extraction of the descriptions in order to make descriptors independent from distance to the camera and orientation of the hands. Taking this into account, two kinds of preprocessing methods have been implemented in the project, one which requires the calculation of the wrist point in the image, and a second one which requires three control points of the processed image and the reference one. Nevertheless, both processes are based on the same method for the extraction of characteristic points.

Two descriptors are introduced in Chapter 5, they are the ones proposed in [1] and in [2], that from now will be called Tanibata and Roussos respectively. For both of them the segmentation and alignment techniques will be the same.

4.1 Hand Segmentation

Due to the fact that the captures performed with the TOF camera were made from real users, with their particular ways of executing the gestures, their forearms appeared in almost all the images. Therefore, in this first stage of the project the hand and forearm regions had to be separated.

4.1.1 Simple Depth-based Approach

The method used consists on applying a noise filter to the image, followed by a depth segmentation where the pixels from the image too far from the camera will be removed (see [56]). The value specified as the depth limit of the hand was chosen in order to keep in the segmented image only the hand and wrist regions.

The value of the pixels of the grey images captured by TOF cameras represent the distance to the camera of each point. In this project we work with possible values from 0 (black), which represents the farthest distance, to 255 (white), which belongs to the points of the image closest to the TOF camera.

Therefore, the depth segmentation applied to images is based on the distance to the camera presented by its pixels. Taking in consideration that the descriptors described in this project only focus in the hand region, the

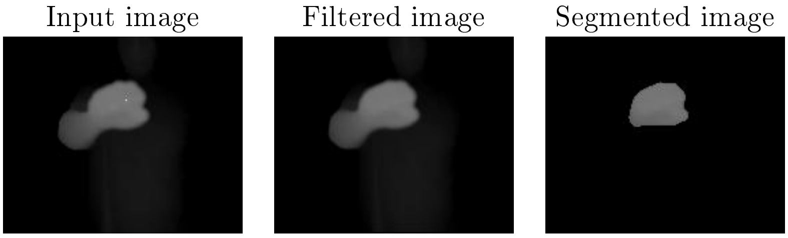


Figure 4.1: Segmentation of depth images:

remainder pixels, which belong to the user body or to the background, need to be rejected in the resulting image. This results difficult when the hand and the forearm are contained in the same plane, and it is parallel to the plane of the camera.

Therefore, when there is a straight angle between the hand and the wrist, the segmentation is easier due to the difference established in distance to the camera between each region. In fact, the threshold defined as the limit of the points which are considered within the hand region was obtained with the minimum distance reached by a point of the hand plane plus a fixed distance. Due to the convention followed by depth images, the pixel closest to the camera is the pixel with the highest value in the image. The threshold chosen to be added to this value was 20 , and it was defined considering a reasonable length of the hand, 20 cm. This threshold presented good performance separating hand pixels from background (see [56]).

In the Figure 4.1 the first picture shows the captured frame of a fist, the second one is the result of filtering that image and the last picture shows the hand pose resulting from the segmentation.Nevertheless, when the forearm is in the same plane as the hand, points of each region have similar values of distance. In this case, the worst one, points of the forearm would remain with the points of the hand in the segmented image.

The described segmentation technique is common to the descriptors considered in this project, explained in Chapter 5.

Tanibata descriptor is a protuberance based descriptor, (see Section 5.1.1) in which the number of protrusions is estimated besides other features extracted from the image. Thus the existence of the forearm region in the segmented image considerably affects to the estimation of this parameter. As well, in Roussos descriptor the appearance of the forearm results in problems in the extraction, since the Principal Component Analysis (PCA) in which it is based is very sensitive to translations.

4.1.2 Other approaches

During this project another approach for hand segmentation was tested, obtaining unsuccessful results. The idea was to, instead of using the direction of vector $\hat{z}$, perform the segmentation estimating the wrist point in the direction of the principal plane computed over the cloud of points associated to the hand.

For this purpose, the volumetric surface was transformed to a voxels set. On the basis of this volumetric representation, two approaches for the estimation of the palm plane were tested:

- The first one consisted on the computation of the principal vectors. , They were estimated by the calculation of the eigenvectors of the covariance matrix of the volume.Nevertheless, the resulting directions did not provide a proper plane.
- The second approach consisted on the estimation of the palm plane via Mean Squared Error (MSE) optimization.

Unfortunately none of the previous approaches presented enough performance, specially when the palm were occluded by the fingers of the point of view was too much sided.

Once the palm plane were estimated, the cloud of points was projected to it, for later estimating the wrist point as it was proposed in [58]. Finally, the segmentation was made on the basis of the depth information associated to the estimated wrist pixel.

The development of this approach required of a lot of efforts and permitted to conclude that it is not possible to estimate the palm plane with plane-based optimizations.

4.2 Hand Alignment

This second stage, previous to the extraction of descriptions is common to the descriptors presented in next Chapter, and it is mainly based on the alignment of these images to obtain independence to scale, rotation and shift. An affine alignment is proposed and the sets of images commonly aligned depend on the concrete descriptor, as it is explained in next Chapter.

4.2.1 Affine Alignment

The main goal of this alignment consists on mapping the points of images into the affine space defined by a reference set of points. This is done in a 2 dimensions spaces, considering the input images as grey images rather than as volumetric surfaces. Moreover, in geometry, an affine transform between two vector spaces is defined as a linear transformation followed by a translation. A linear transform can be composed of a scaling, a rotation and a translation due to the two properties of the affine transform in an euclidean space:

- The collinearity relation between points. This means that points from a line of the input space continue to be collinear in the space after the transformation
- The ratio of distances along a line. In the ellipse of inertia calculated over the input image, the ratio between its axis is preserved in the transformed space.

Therefore, the objective of this alignment process is to find the parameters required to the transformation of the image.

The followed method is based on the computation of three characteristics points from both, the hand image to be aligned and a reference hand image. On the basis of the correspondence between these two sets of points, the parameters of the affine transform are calculated. The reference image, A_{ref}, and the set of images used to perform the alignment depend on the extracted descriptors, as will be explained in next chapter.

On the other hand, the selected three points required from the images are: the centre of the image (x_c, y_c), the estimated wrist point (x_w, y_w) and the 3^{rd} point, (x_{3p}, y_{3p}), which is orthogonal to the line formed by the two previous points. The detailed description of how these three points are extracted can be found in Section 4.2.1.1.

The first point (x_c, y_c), was used to define the centre of the aligned image. This point allowed to estimate the shift parameter for the affine transform. The second and third points, (x_w, y_w) and (x_{3p}, y_{3p}) respectively, in combination with the first one, define the scale and orientation of the image relative to the reference image. The angle between the line crossing (x_w, y_w) and (x_c, y_c) and the horizontal axis define the rotation to be applied in these input images in order to achieve the same orientation than A_{ref} image.

On the basis of these three pairs of control points the correspondent affine transformation matrix is defined. In fact, in a finite-dimensional space, the affine transform can be defined as a matrix multiplication T, which represents the linear transformation, and a vector addition $\vec{s}$, which represents the translation. Next lines define this transformation for any point $\overrightarrow{p}$ of the initial image into the new one $\overrightarrow{p}'$ using a single matrix multiplication:

$$\overrightarrow{p}' = T \cdot \overrightarrow{p} + \overrightarrow{s} \tag{4.1}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} r_x \cdot \cos\theta & -\sin\theta & s_x \\ \sin\theta & r_y \cos\theta & s_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{4.2}$$

The previous equations define the operations required to calculate each point of the transformed image. In addition, at least a set of three points from both transformed image and A_{ref} is required to obtain the parameters involved in the definition of the transformation matrix:

- Scaling, r_x and r_y: These parameters define the relation between the size of the image to be aligned and the size of A_{ref}.

- Rotation θ: The difference between the angle formed by the line defined by the wrist point and the centre point and the horizontal axis from both images.

- Translation s_x and s_y: These parameters define the translation for each direction to be applied to the image in order to achieve the same origin as the reference image.

In the implementation of the alignment stage, the transformation matrix and the resulting aligned images were computed by the use of the following Matlab functions:

- A function to calculate the six parameters of a 2D spatial affine transformation [2] on the basis of pairs of control points: three principal points from the image to be transformed and from the image reference of the transformed space.
- A function which generates the image resulting from the affine transformation [3] of the input image using the transformation matrix previously calculated.

Once the foundations of an affine transformation are explained, the process used for the estimation of the three control points required for the alignment of images is described now.

4.2.1.1 Calculation of alignment triangle

As it was mentioned before, three points of the hand are used for performing an affine alignment. Several methods were followed to achieve an algorithm which yields to a proper estimation of the wrist point and the other three points. Unfortunately, the unavoidable persistence of forearm regions in some of the segmented images makes necessary the introduction of some restrictions to the algorithm, such as the assumption of the area in which the location of the wrist can be, concretely, at the bottom of the image.

These three points, as already mentioned, are the centre of the image, r_c, the estimated wrist point r_w and the 3^{rd} point, r_{3p}, which forms a line with r_c orthogonal to the line formed by the two previous points, with a 90^o clockwise rotation.

[2]

– T=cp2tform(input_points,base_points,'affine')

[3] transfromed_image = imtransform(image,TFORM,'XData',[1 w], 'YData',[1 h]);

The estimation of the wrist point is the most difficult task of the aligning image process.

The first point: centroid of the image In geometry, the centroid is also called geometric centre or barycentre of a plane figure (or in any two dimensional shape), and it is defined as the intersection of all straight lines which divide the figure into two sections of equal moment about the same line. The simplified concept is the average of all points belonged to the plane of the figure. Moreover, the definition could be extended to any object in n-dimensional space as the intersection of all hyperplanes that divide the object into two parts of equal moments.

In physics, the centroid is also called geometric centre of an object´s shape, but its barycentre definition is addressed to the physic centre of mass or centre of gravity, depending on the context. The centre of the mass is defined as the average of all points, weighted by the local density or specific weight. In the special case when the object has uniform density its centre of mass is the same as the centroid of its shape.

This definition were extracted from `http://en.wikipedia.org/wiki/Centroid`.

So, we can say that the **centroid** of a subset X of a n-dimensional space $\mathbb{R}$ can be calculated by the following equation:

$$C = \frac{\int x.g(x).dx}{\int x.dx} \tag{4.3}$$

where the function $g(x)$ is the characteristic function of the subset, which can take only two different values: 1 in the case of the analysed point within the space $\mathbb{R}$ also belongs to the set X and 0 when it does not. The denominator is the area of the shape or simplifying, the number of points of the set X inside the space $\mathbb{R}$.

Moreover, other procedure to this estimation is the assumption of the density function of pixels belonged to a binary image as an uniform function (0 if they pixel is black, 1 if it is white) and hence, previous integrals can be approached with the raw moments in order to obtain the centroid of the image.

Since the moment of a continuous 2D function to scalar (grey scale) image can be adapted with pixel intensities $I(x, y)$, the raw moment of the image can be defined as the Equation 4.4:

$$M_{ij} = \sum\sum x^i y^j I(x, y) \tag{4.4}$$

Furthermore, the theorem written by Papoulis[4] states that if $f(x, y)$ is piecewise continuous and has non-zero values only in a finite part of xy plane, moments of all orders exists, and the moment sequence M_{pq} is uniquely determined by $f(x, y)$. So, in practice, the centroid of an image can be obtained as:

$$C = \{\overline{x, y}\} = \{M_{10}/M_{00}; M_{01}/M_{00}\} \tag{4.5}$$

The second and third point (wrist point and orthogonal point). Different processes were followed to calculate the wrist point of images, although they are different, they share the ellipse of inertia of images as the basis of their calculations, which is described in detail in the Chapter 5. So, in concern to the topic aimed here, it is enough knowing the global idea of its concept: an ellipse centred on the same point than the centroid of the hand image and with the same direction and scale than this image. This is a simplified vision of the second central moment which determines the length in pixels and orientation of the ellipse region to be the same than in the hand image.

The first idea for the estimation of the point of the wrist was defining it as the point of the hand farthest to the centroid of the image, considering both halves of the image, the top and the bottom, depending on the orientation of the hand. Nevertheless, if the hand was positioned horizontally, the wrist point would be detected erroneously in the palm of the hand.

The combined use of the ellipse besides the image allowed to find the orientation of the hand for the most of the cases (all hand poses excepting fist pose

[4] http://en.wikipedia.org/wiki/Athanasios_Papoulis

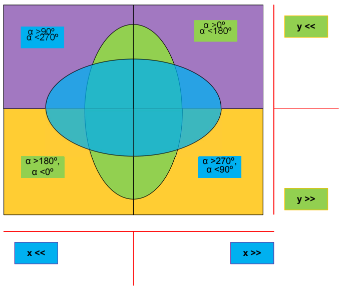


Figure 4.2: Possible orientations of the images to estimate the point of the wrist.

images). Therefore, if the orientation of the hand was known, the wrist was defined as the intersection with the highest/lowest value of x/y (depending on the orientation of the hand) between outline points of the hand and the line crossing the centroid and the extremes of the ellipse. However, knowing the orientation of the hand to generate the line crossing the ellipse implied having the angle of the image, understanding it as the angle of the ellipse. Moreover, the knowledge of this angle also requires knowing the wrist point, or failing that, one finger point, which was the initial objective of this matter. So the angle of the ellipse is required to this method.

Due to the simulation and implementation of this algorithm were carried out in Matlab, one of its functions was used to obtain the angle of the ellipse based on the central moments of the image and its ellipse of inertia. Nevertheless, the function does not understand about fingers or wrists, thus sometimes the angle yielded was obtained from the finger and some other times from the wrist region, depending on which region had the greatest weight. This would not had happened if the segmentation perfectly removed the forearm region from images, because in such case the palm of the hand would be always the heaviest region.

However, if the obtained angle was always right, the estimation of the wrist point could be obtained from the combination of the orientation of the image besides the points extracted from the line crossing centre and the extremes of the ellipse.

The Figure 4.2 shows possible orientations of the image (represented with ellipse regions) and how the wrist point can be chosen from the outline points of the hand which match with the line crossing centre and extremes of the ellipse. Using the fingers as the reference for the definition of the angle of the image as the angle formed with fingers and horizontal axis. Moreover, depending on the location of the fingers the wrist was defined in a different region of the image accomplishing the following conditions:

- When hand is vertical (ellipse in green colour) and the resulting angle has values between $0^{\underline{o}}$ and $180^{\underline{o}}$, the wrist point is defined in the extreme of the ellipse with the higher values of y, which belongs to the bottom half of the image (yellow region). In the other side, when the angle is included between $180^{\underline{o}}$ and $360^{\underline{o}}$, the wanted point is defined in the extreme of the image with the lower values of y, this is the top half of the image (purple region).
- Nevertheless, in the case of the hand in horizontal position (ellipse in blue colour) and the angle between $90^{\underline{o}}$ and $270^{\underline{o}}$, the wrist point is defined within the extreme of the ellipse with higher values of x (the right half of the image), meanwhile for the rest of angles, the wrist was chosen between the points of the extremes with lower values of x (the left half of the image).

But as it was explained before, this method did not work as well as desired because of the lack of accuracy of the hand angle given by the Matlab function, due to the problems appeared in the segmentation stage and the existence of forearm regions in some real processed images.

A new method, similar to the previous one was implemented to calculate the point of the wrist, but improving the estimation of the orientation of the hand.Firstly, a line which crosses the geodesic centre and the extremes of ellipse is calculated like in the previous method. Then, the points of the outline (i.e. the contour) of the hand region is obtained. In most of the images the wrist region is enclosed inside the area of the ellipse meanwhile fingers regions usually cross the line of the ellipse contour. Therefore, there might be only one point within the edge of the hand which matched with the ellipse and the straight line. This point is defined as a finger point. At the same time, the intersection of the same line with the outline points of the hand was calculated too. As a result, there might be at least two separated

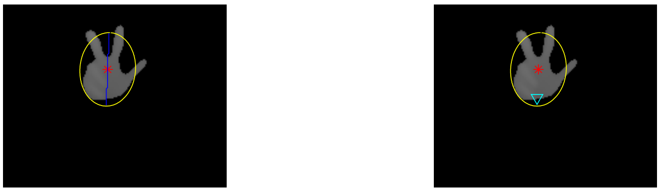


Figure 4.3: Wrist Point defined as intersection of line and boundary line of the hand

points, located in opposite sides of the hand. Finally, the point of the wrist is defined as the point from the set of points previously calculated which is farthest from the finger point obtained in the first intersection.

Furthermore, in both matching processes, when no point is found from the intersection, the line crossing the hand is dilated until one matching point is found or until number of dilations iterations is exceeded.

In Figure 4.3 the calculated line crossing the centroid and extremes of the ellipse, and the wrist point result from the implementation of this method are shown:

However, this method was not the definitive one due to the following problems:

- When the dictionary used includes a fist image, there is no finger point found during the intersection of the straight line with the boundary line of the ellipse and the outline points of the image. Moreover, if the whole forearm region was not previously segmented, it could match with the boundary line of the ellipse, obtaining a point from the forearm region instead of the corresponding finger point. In the opposite case, when the forearm region is entirely segmented, not only no point was found as finger point, but also ellipse extremes of the major axis can take the same orientation than the thumb of the hand. This means that the outcomes of the matching process only give points from the palm of the hand (or a point from the thumb) instead from fingers or wrist.

- The second problem is related to the previous segmentation process and affects to the rest of hand poses. Like in the other cases, when the forearm region is not completely segmented, the forearm can be erroneously considered a finger, resulting in more than one matches of potential finger points.

To conclude, the final implemented method defines some restrictions to input images:

Firstly, the points of the ellipse are defined by the next equations:

$$x(t) = x_c + a\cos(t)\cos(\alpha) - b\sin(t)\sin(\alpha)$$

$$y(t) = y_c + a\cos(t)\sin(\alpha) - b\sin(t)\cos(\alpha)$$

These equations belongs to the parametric representation of the points of an ellipse, centred in the point $r_c = (x_c, y_c)$ and with major and minor semi axis with a length, in pixels, of a and b respectively. The parameter t varies from 0 to 2π and α is the angle between the X-axis and the major axis of the ellipse.

Besides the computation of the points composing the ellipse, there are four principal points which were separately stored: the points that matched with the axis of the ellipse:

$$r_1 = x(0), y(0) \tag{4.6}$$

$$r_2 = x(\pi), y(\pi) \tag{4.7}$$

$$r_3 = x(\pi/2), y(\pi/2) \tag{4.8}$$

$$r_4 = x(-\pi/2), y(-\pi/2) \tag{4.9}$$

The next step consisted on defining the possible combinations of points to be extracted from the image (both wrist and 3^{rd} points are estimated at the same time). The initial collection is formed by the previous principal points of the ellipse ($r_{1,}r_2, r_3, r_4$) and their possible combinations, the ones which satisfy the orthogonality between points. Then, the points belonging to the extremes of the ellipse are replace by points of the hand. Each of these points is obtained as the intersection of the binary hand image with the boundary line of the ellipse and the straight line defined from the centroid of the hand to each possible point of the extremes of the ellipse. From among matching points, the farthest to the centroid of the hand is a candidate to be the wrist point. Therefore all previous points ($r_{1,}r_2, r_3, r_4$) had one possible candidate to be wrist point associated

$$r_1 \Longrightarrow r_{h1} \tag{4.10}$$

$$r_2 \Longrightarrow r_{h2} \tag{4.11}$$

$$r_3 \Longrightarrow r_{h3} \tag{4.12}$$

$$r_4 \Longrightarrow r_{h4} \tag{4.13}$$

Finally, applying the same matching process to each possible point ($r_{1,}r_2, r_3, r_4$) the third possible point is defined with the suitable combination of points. In the next lines, the possible combinations of principal points (this is centroid, wrist point and third orthogonal point) are enumerated:

$$X_1 \equiv \{r_c; r_{h1}; r_{h4}\} \tag{4.14}$$

$$X_2 \equiv \{r_c; r_{h2}; r_{h3}\} \tag{4.15}$$

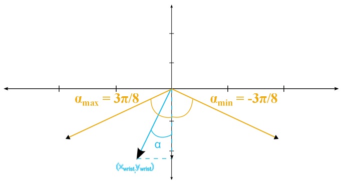


Figure 4.4: Limits of the Angle between Real Wrist and Ideal Wrist

$$X_3 \equiv \{r_c; r_{h3}; r_{h1}\} \tag{4.16}$$

$$X_4 \equiv \{r_c; r_{h4}; r_{h2}\} \tag{4.17}$$

The next step consists on the selection of the best point combination from all the possible ones (X_1;X_2;X_3;X_4). As it was mentioned before, each set of points form orthogonal lines.

The restrictions applied to the triangles in order to estimate the right combination of points are the next:

- Restriction to the location of the writs point. Taking as reference the estimated centre of the hand (x_c, y_c), the cathetus of the wrist is assumed to be located in the lower half of the image, more concretely, considering the vertical as 0 *rad* in the range $\alpha \in ({}^{-3\pi}/{}_{8}, {}^{3\pi}/{}_{8})$ (see Figure 4.4). Therefore, analysing the wrist component r_h (represented by the blue line in the Figure) of the first subset of points resulting from this restriction, its angle α is included inside the limits defined by the α_{min} and α_{max} (represented by the yellow lines of the Figure).

- The second restriction is related with the depth of the final selected wrist point. This point is defined as the deepest among all the candidates, survival from the first restriction application.

  Thus, the first set of points chosen as principal points of the hand will be named from now on:

$$\{(x_c, y_c)\,;(x_w, y_w)\,;(x_{3d}, y_{3d})\} \tag{4.18}$$

- The third restriction is related with the length of the triangle we formed by final selected candidates. This triangle, sometimes, presents very different lengths in its sides. This produces a very distorted affine transformation. This is why this last restriction was introduced. It consist on finding an isosceles right-angled triangle with equal area to the original one, reaming the right-angled corner (i.e. (x_c, y_c)) in the same location.

  Firstly, the area of the initial triangle has to be obtained , for later extracting the length of each cathetus of the isosceles triangle from that value:

$$Area = \frac{b \cdot h}{2} \tag{4.19}$$

$$Area = \frac{b_{t.isos}^2}{2} \tag{4.20}$$

$$b_{t.isos} = \sqrt{2 \cdot Area_{initi}} \tag{4.21}$$

  These new points can be different from the old ones, but they also belong to the hand. From now on they will be called:

$$\{(x_c, y_c)\,;(x_{w-re}, y_{w-re})\,;(x_{3d-re}, y_{3d-re})\}$$

At the end of this stage, the three principal orthogonal points are provided to be used in the alignment process. Moreover, when the point of the wrist is found, the yielded reference points of each image are enough separated to generate an acceptable and no distorted transformed image after the alignment.

Chapter 5

Hand Descriptors

5.1 Related Work

We can find several hand descriptors in the literature, some of them have been selected for their implementation and testing in the evaluation frame work that will be described in Section 6.1. The descriptors under study are two:

- A protuberances based descriptor, [1] (from now we will refer to this work as Tanibata).
- Principal Component Analysis (PCA) based descriptor: an approach based on the generation of an eigenvectors base that from now on we will name eigenhands [2] (from now we will refer to this work as Roussos).

5.1.1 Tanibata, a protuberance based descriptor

Authors of this paper propose a method to recognize words of the Japanese Sign Language (JSL) performed by a user. For this purpose they extract features from each frame of the capture.

In the first half of the paper they explain an approach for finding the position of the wrist. First of all, they propose an algorithm for separating person region from the background, then they use templates in order to find the face and the hands on the basis of the colour these areas present.

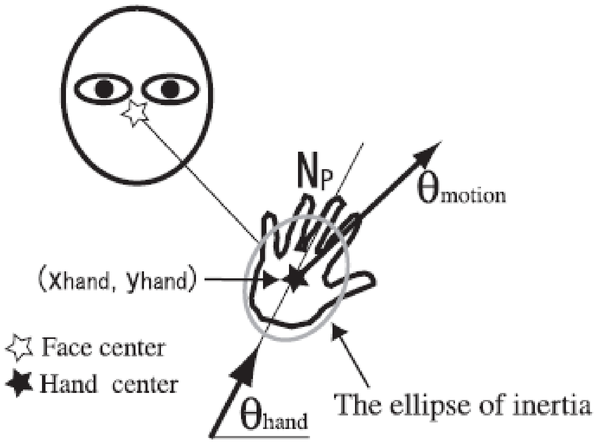


Figure 5.1: Image taken from [1].

The second half of the paper results more interesting for our work, since the hand features we want to use are described.

Once the hand region is detected, the following features are calculated:

1. The flatness of hand region, r.
2. The gravity centre position of the hand region relative to that of face region, $(x_{hand}; y_{hand})$.
3. The area of the hand region, A.
4. The direction of hand motion in the image coordinate,θ_{motion} .
5. The direction of hand region in the image coordinate,θ_{hand}.
6. .The number of protrusions , Np.
7. The ellipse of inertia.$(\vec{x}_{ellipse}, \vec{y}_{ellipse})$.

The first three features of the hand, flatness r, the gravity centre position $(x_{hand}; y_{hand})$ and the area of the hand A can be easily obtained from hand regions.

The ellipse of the hand, shown in Figure 5.1, is defined as the ellipse of inertia of the hand region. In the paper, feature r is the ratio of the major axis to the minor axis and describes the ellipse, and feature θ_{hand} is defined as the angle between the major axis of the ellipse with the horizontal coordinate axis.

Next feature θ_{motion} represents the trajectory of the hand, this is, the angle between two consecutive centre points and the horizontal coordinate axis. The difference between this angle and the previous one is the intervention of the time in the estimation of the motion direction θ_{motion} . For example, during the the computation of this angle, θ_{motion} in instant $t = 1$, both points $(x_{hand}; y_{hand})$ for $t = 0$ and for $t = 1$ are needed in order to obtain the required direction vector, in contrast to θ_{hand}, where the required value of the major axis would not depend on different instants of time. This feature is specially significant to differentiate words sequences, where each word can be followed only by a limited collection of words.

The last feature Np is defined as the number of local maxima of the distance between the wrist and outline points of the hand region, as it is shown in Figure 5.1. In the method proposed in this paper, the wrist position is defined as the hand region point nearest to the elbow position,.Finally, in this paper a Hidden Markov Model in order to recognize the word of the dictionary with the highest probability is proposed, considering that they calculate the kind of gesture operating computing a sequence of poses. Moreover, this document ends presenting the figures for the experiments carried out to evaluate the performance in recognizing 65 JSL words.

5.1.2 Roussos, a PCA based descriptor.

This paper proposes a new model to describe and represent hand configurations via a PCA based descriptor to represent the shapes an the appearance of the hand. As it was mentioned in Chapter 2, some descriptors based on the recreation of hand shapes use landmarks during the capture with 3D cameras, nevertheless, the descriptor proposed in this paper does not. In addition, this model allows the reconstruction of hand poses by the lineal combination of images that are previously aligned and calculated to conform an orthogonal base. These hand images from the base, from now on eigenhands, are yielded from an iterative alignment of a training set of hand poses followed by a PCA analysis. Finally, the weights of the eigenhands derived from the model fitting will be used as hand shape features (i.e. as the descriptions of the hand poses).

The whole process followed to achieve a successful outcome is explained in the following lines:

- The first step proposed in the work consist on **segmenting and tracking frames** from the videos, where the whole body of the user is captured by the camera. Once the hand region is identified, the segmentation of hand poses is made using Geodesic Active Regions (GAR) method, which separates skin-colored regions from the background, minimizing an energy function. Nevertheless a little modification of this GAR method is applied for making the final segmentation of input images. This process is based on fitting an enveloping curve to the edge of the image which separates skin from background using the ratio between the probabilities of a pixel belongs to a skin region and background region. Moreover, linear forward-backward prediction and template matching are techniques used to avoid occlusion effects of the skin colour regions. Finally, the hand region is cropped using a skin colour detector before getting the final colour segmented image of the hand.

- The next step consist on **modelling hand Shape Appearance images** from cropped images. These kind of images are grey-scale images extracted from the coloured cropped images. Therefore each pixel belonging to the hand region is transformed to the YC_rC_b colour space, in which only the texture and shadowing of images is represented. The rest of the pixels are considered as the background of the final image. Due to the fact that the main goal of this project is the description of the images, these preprocessing techniques are not included in the scope of this project.

- Once the images to be described are prepared, they can be modelled by means of a linear combination of a base formed by the mean image, A_0, and a set of computated images, A_i, after applying an affine transformation. These base images are generated from a subset of the training set, combining an iterative affine alignment with PCA optimization. Finally, the image will be approach with equation 5.1.2, where $W_p(x)$ is the affine transformation of the Shape Appearance images and λ_i are the weights of the computed base images A_i for each regenerated image.

$$f(W_p(x)) \approx A_0(x) + \sum \lambda_i{\cdot}A_i(x)$$

Authors of the paper use 200 randomly selected frames from a video to generate the initial training set. Then, these images are recursively aligned to compute the mean image A_0. Followed steps of the proposed iterative method are the following:

1. Selection of the first image from the training set as A_0 image.
2. Alignment of images from the training set with current A_0, estimating the parameters of the transformation $P = (p_1, p_2, p_3, p_4, p_5, p_6)$.
3. Computation of the new mean image, $A_0^{'}$, over the aligned images.
4. Comparison of the new $A_0^{'}$ with the previous A_0.
5. Repetition of the second step changing A_0 by $A_0^{'}$ until there is no difference between both of them.

When the average image A_0 is calculated, it is used as the reference image for the alignment of the training set.

The covariance matrix of these aligned images has to be estimated in order to get the new hands base. In fact, N_c eigenvectors from the largest eigenvalues of this covariance matrix are the base, from now on so-called Eigenhands or Eigenimages. Moreover, the number of eigenvectors will depend on the outcome of the PCA analysis, which allow to reduce the principal components used to model images. Hence, the higher N_c the lower difference between modelled images and the original ones, meanwhile the run time and complexity of the alignment will increase too.

Once the eigenhands have been computed, the weights for each image of the training set , λ_i, are calculated, minimizing the energy of the reconstruction error, this is:

$$\sum_x \left\{ A_0(x) + \sum_{i=1}^{N_c} \lambda_i A_i(x) - f(W_p(x)) \right\}^2$$

- The implementation proposed for extracting the weights is made by the Simultaneous Inverse Compositional algorithm which is based on the Gaussian-Newton gradient descendant optimization.

- Once the eigenhands and the reference mean image A_0 are calculated, handshapes from images are extracted, finding the weights and alignment parameters which fit the model.

- Finally, data used in the paper for the evaluation experiments belong to the continuous American Sign Language Corpus BU400. The classification is made using mixture Gaussian mixture models (GMMs), and maximum likelihood to the selection of the best matching model. At the end of the paper the Affine Shape Appearance Model are compared to other hand shape models and the conclusions from the experiments are described.

To sum up, this model tries to describe hand images by the combination of grey-scale images which represents hands shapes of images. There are two possible points of view to interpret this transformation and, thus, to define the characteristic description, λ_i, of Roussos descriptor:

- From Algebra viewpoint: This transformation is assumed as a new representation space which coordinate axis are defined by base images. In addition, their base elements can be defined by the covariance matrix, eigenvalues and eigenvectors from an aleatory set of images. The parameters of the description λ_i define the coordinates of images in the new space, i.e. the projections of images into the new base.

- From Image viewpoint: This transformation is assumed as a new representation of images by the lineal combination of redefined hand images, eigenhands, to represent any image with the minimum number of elements. Input images has to be aligned with images of the base using A_0 as a reference to increase their resemblances and obtain them with the minimum reconstruction error that is possible. The weights of the hand images used for the reconstruction are the parameters of the description λ_i.

5.2 Tanibata Implementation Concerns

5.2.1 Introduction

The Tanibata descriptor was implemented in Matlab following the steps enumerated in [1], but making some changes during the process.

Previous to feature extraction for describing images, they were segmented. As it was explained in detail in Section 4, the segmentation of the forearm consists on removing the points of the image which are behind the estimated wrist point. This means that by fixing the maximum distance to the camera of hand points (desirably points of the hand), the remaining points belonging to the arm are removed.

The main difference between the original paper and the implementation in this project has relation with the features selected to describe image hands. In fact, only a subset of them were selected, as long as we are focused in static images rather than in image sequences that could contain dynamic gestures. This selection was done in order to be robust to variations in position, distance and rotation of the hand:

- r, flatness of the ellipse.
- $(x_{hand}; y_{hand})$, instead of the centre of the hand region as in the original paper, the estimated wrist point is used.
- N_p, number of protrusions.

In the implementation of the descriptor the flatness of the ellipse is the ratio between the minor and maximum axis of the ellipse, like in the definition of the paper. Nevertheless, the second feature, the gravity centre of the hand region is only used to calculate the last of the features, the number of protrusions. In fact, it would have been more suitable defining the point of the wrist as an important feature of hand image instead the centre of the hand region. Moreover, this last point was used to estimate the position of the wrist point, but applying a different reference system than in the paper, where the centre of the hand was relative to the head region. Due to images used in the implementation only contained the hand region, the reference system used in images had the origin of points in (0,0), at the left-up corner.

From now on, when this point is mentioned it will be referenced as the point of the wrist, instead of the centre of the hand region.

Some features, originally used in [1], but not in this project are: orientation of the hand, θ_{hand}, because we want to be independent to rotations; the trajectory of the hand, θ_{motion}, because we are not working with dynamic gestures; and the area of the ellipse, A, because we want to be independent to scale.

The process followed during the extraction of descriptions can be separated into two different stages:

- Computation of characteristic points of the hand image: flatness of the ellipse, r, and the ellipse of inertia. See Section 5.2.2.
- Calculation of the number of protrusions of the hand, N_p,on the basis of the information previously extracted. See Section 5.2.4.

5.2.2 Estimation of ellipse of inertia and wrist point

In this stage the main features of the hand and ellipse are extracted: the point of the wrist, the flatness of the ellipse and the points of the ellipse aligned with the hand.

The only parameters we use to describe images are the flatness of the ellipse and the number of protrusions. The point of the wrist and the points of the ellipse are only needed to calculate the last parameter N_p.

Therefore, the first parameter of the descriptor, **the flatness of the ellipse** r, is the outcome of this first stage of the computation. The ellipse used here is the ellipse of inertia (see Figure 5.2), referenced at the beginning of this section 5.1.1, a function of Matlab[1], "regionprops", was used to perform this task.

This function operates with binary images, and measures several features of the represented region. In this case, the Major and Minor axis lengths of the ellipse, as well as the orientation of these axis (is the same as the hand orientation θ_{hand}), were calculated with this function by the use of the second normalized central moments of hand region. This function works with

[1] http://www.mathworks.es/index.html

2-D input label matrices, obtained from binary images, giving to each pixel a label to differentiate connected regions. A problem came up when there were isolated pixels in binary images, generating more than one connected region. This was solved changing the value of the labels from pixels belonging to small regions by the label of the widest connected region of the image. This makes that each pixel from the matrix has the same value than the rest and thus, all existing regions can be considered as connected regions.

Once this problem was solved, the lengths of ellipse axis were used to define the first parameter of the description r as the ratio between the Major and Minor semi axis lengths :

$$r = \frac{MajorAxisLength}{MinorAxisLength}$$

The second parameter extracted by this descriptor is the **Number of protrusions** N_p. As it will be explain in next Section 5.2.4, the point of the wrist and the outline points of the hand are involved in the computation of this parameter. Nevertheless, the estimation of **the wrist point** is made following the same process that the one described in Section 4.2.1.1 to extract the principal hand points. These points are: the centre of the hand, the wrist point and a third point which forms a line orthogonal to the one formed by the first two points. Moreover, operations made during the extraction of these hand points required the points of the ellipse, hence the second step followed, after extracting the flatness of the ellipse, is the estimation of the points of its line.

Using the orientation of the ellipse, **the points of ellipse** are estimated using the parametric equations of the ellipse:

$$x(t) = x_c + a \cdot \cos t \cdot \cos\varphi - b \cdot \sin t \cdot \sin\varphi$$

$$Y(t) = y_c + a \cdot \cos t \cdot \sin\varphi + b \cdot \sin t \cdot \cos\varphi$$

where φ was defined as the angle of the hand, θ_{hand}, in radians using images axis reference ($\varphi = -\theta_{hand}$) where angles are covered contrary to clockwise; t is defined from 0 to 2π; a and b are the length of the semiaxis of the ellipse.

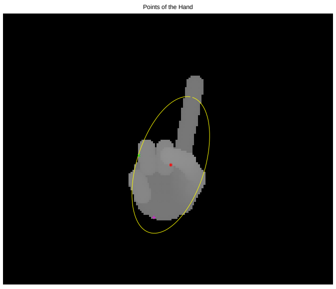


Figure 5.2: Ellipse of Inertia and wrist point estimation in pink.

The coordinates of ellipse points need to be integers, since the results are represented in a pixels image. So, resulting coordinates from equations 5.2.2 and 5.2.2 are rounded. The ellipse aligned with the hand, resulting from this process, is shown in Figure 5.2.

. On the basis of this ellipse, a line crossing the centre of the hand region, in the direction of the major axis, is used for estimating the wrist point as explained in Section 4.2.1.1. The wrist is defined as the furthest point of the hand which belongs to the straight line which crosses both extremes (in the Major axis) of the ellipse and the centre of the hand.

5.2.3 Preprocessing Alignment

Two different alignments are performed, both on the basis of the affine alignment explained in Section 4.2:

- *Implicit alignment with the wrist point.* As it was mentioned before, all the features used as description in Tanibata are independent from the orientation or spatial situation of the hand in the image. In fact, the number of protrusions of the hand, which depends on the distances of the outline hand points to the wrist point, is a relative measure which allows comparing shape appearance between images, no matter where the wrist is placed.

- *Explicit image Alignment :* In Section 4.1 it was mentioned that the alignment of real images with a reference image required to standardize them before extracting their descriptions. It was commented that this alignment consisted on applying the *affine transform* to the three reference points of the real images in order to make them matching with the other three points of a reference image. In this descriptor, the image used as the reference one is the fist image due to the comparison made over their distance function to estimate the number of protrusions in the image. The outcome of this process particularly affects the later extraction of image´s description. This is because, in contrast with other features such as the ratio between the axis of ellipse, the estimation of the number of protrusions of the hand needs the input image and the fist image in the same scale. As it will be explained in Section 5.2.4the establishment of the number of peaks in images requires the computation of the distance between the outline points of the hand and the point of the wrist. This is the before mentioned distance function of the image. So, in order to difference the peak of the distance function of a hand with one extended finger (or a knuckle) from one of a fist image, a fixed threshold is used. The value of the threshold is defined as the maximum value of the distance function for the fist image. Thus, the found maxima with a lower value than that threshold is rejected. If the image to be described does not have the same scale than the fist image, the distance of its knuckles cannot be used as the reference value.

5.2.4 Calculation of number of protrusions of the hand

In this second stage of the descriptor extraction the flatness of the ellipse was already calculated, r, as well as the three principal points of the hand required to the extraction of the **number of protrusions of the hand**, N_p, which is the main goal described in this Section. This parameter is basically defined by the distances between the outline points of the hand and the point of the wrist.

5.2.4.1 Computation of distance function

To begin, a family of straight lines from the wrist to the limits of the image is generated. This set of lines are parametrized with the inclination of the lines, α, as a parameter, covering angles from $-\pi/2$ radians to $3\pi/2$, taking the major axis as reference. The step of the angle depends on the number of samples of the hand contour we want to use. Here it is important to know that a low number of samples could imply the loosing of fingers of the hand, whereas a number of samples too high could give more number of local maxima or minimums on the graphic of the distances. So, a balanced number of samples has to be found in order to generate the best final distance function to calculate the number of extended fingers. Principal vector of the line in the image was defined as the cosine (as coordinate x) and the minus sine (as coordinate y) of these angles. Defining the independent term as the point of the wrist, each obtained straight was a line from wrist to limits of the image:

$$NumSamples = 360$$

$$\alpha = \in \left[-\frac{\pi}{2}; \frac{2\pi}{NumSample}; \frac{3\pi}{2}\right]$$

$$t = \in [0, 100]$$

$$v_x = cos(\alpha)$$

$$v_y = -sin(\alpha)$$

$$r = t.v + b$$

For each line, obtained for a value of α, an image of its points is generated as shown in Figure 5.3.

The next step consists on making a logical AND operation between the binarized image of the hand and the image of the line for each value of α to obtain the points of overlapping. In Figure 5.4 the described AND operation is illustrated. From the resulting set of points the furthest from the wrist point is the one selected, assuming it belongs to the contour of the

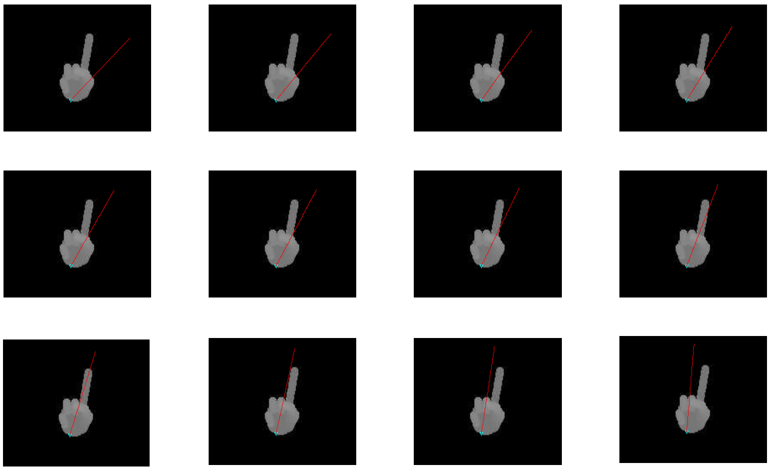


Figure 5.3: Subset of images of the points of the lines which covers the contour of the hand. In this examples $\alpha = \in [\pi/4, 3\pi/2]$.

hand. When this point (x_{PointH}, y_{PointH}) is found, the distance to the wrist $(dist_{Wmax})$ for the considered value of α is stored. This way we obtain an estimation of the distance from the wrist point to the contour of the hand for each of the values of α under consideration (see Figure 5.5).

The next step is the calculation of the number of maxima present in the stored distance function. This is explained in Section 5.2.4.2.

5.2.4.2 Calculation of number of local maxima in distance function

Beginning from the distance values shown in Figure 5.5, it seems easy to determine the number of maxima at first sight. Nevertheless, it was not such an easy task at all, indeed multiple methods for the extraction have been tried.

Basic Local Calculation The first approach was based on the use of a Matlab function, findpeaks[2], to find local peaks in an input data vector. This works in a local way, comparing each element of data to the values of its neighbours. The used function allows the definition of different conditions to detect peaks, however in this project a local peak was defined as an element

[2] http://www.mathworks.es/help/toolbox/signal/ref/findpeaks.html

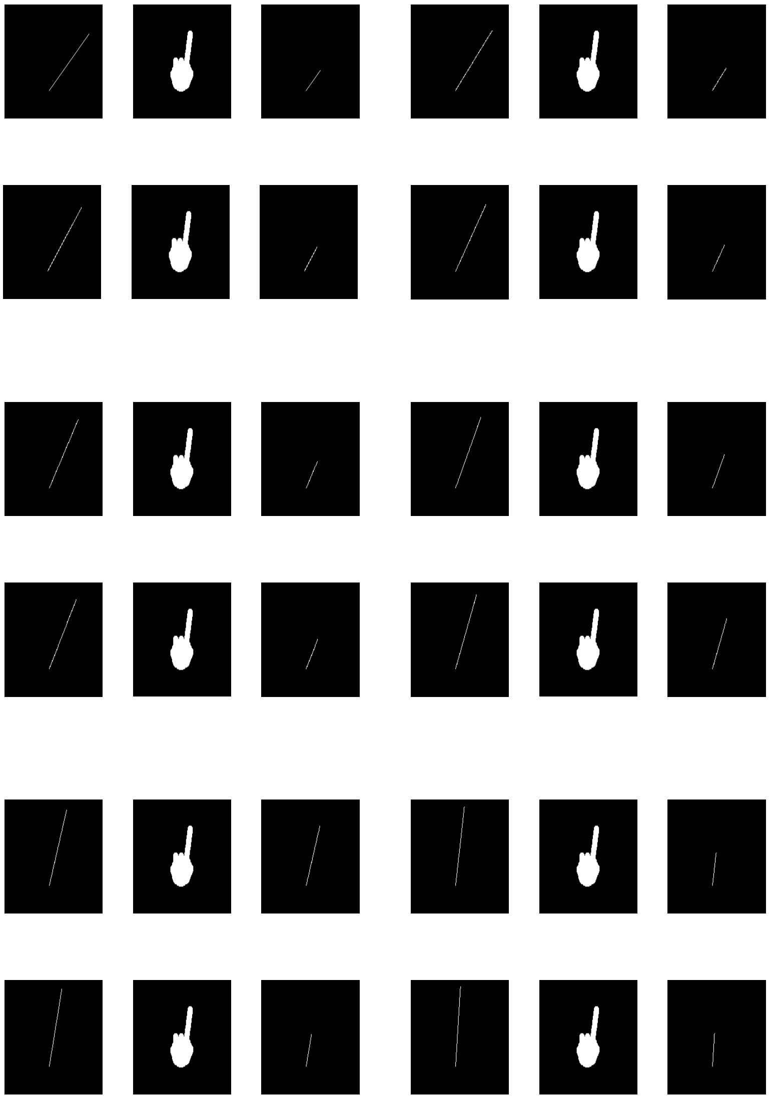


Figure 5.4: Line image, binarized image and result image for the AND operation.

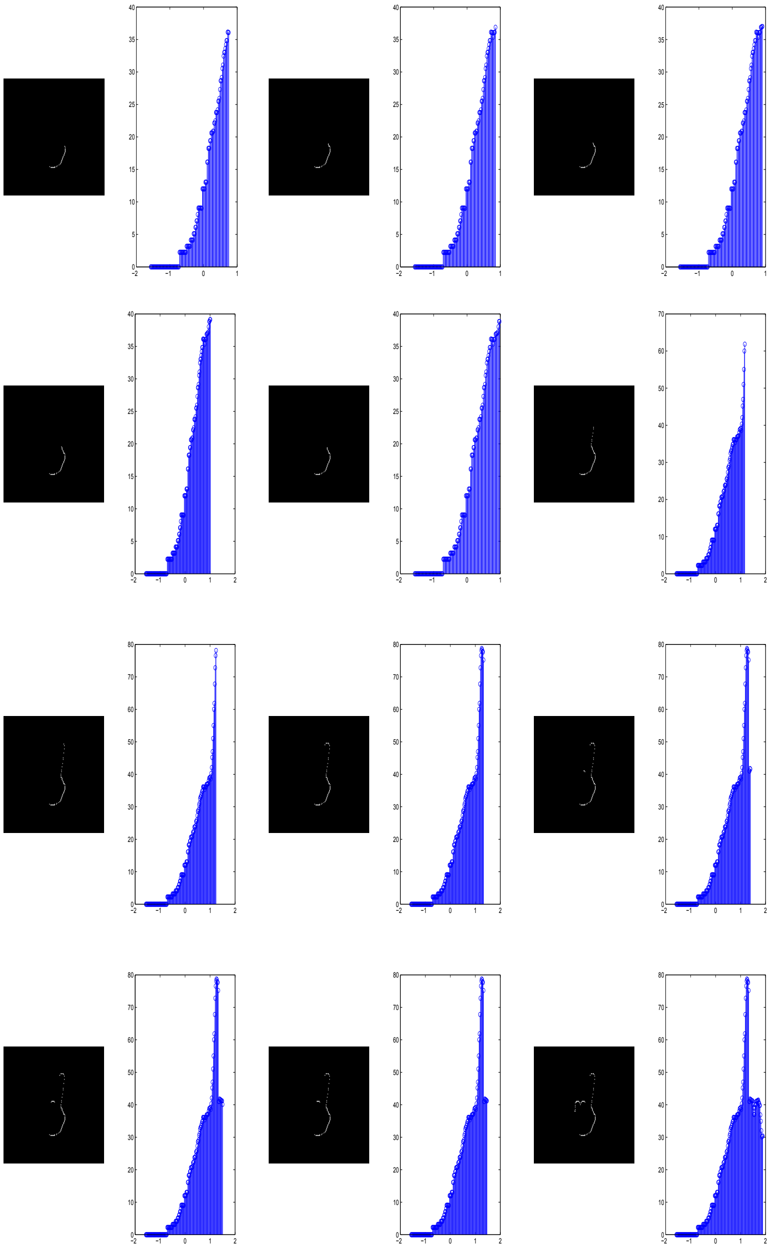


Figure 5.5: Distance from wrist to contour points.

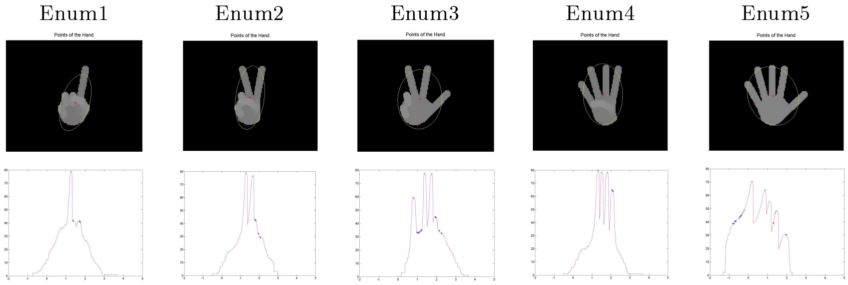
of the function larger than their neighbours. The input parameters available to set up the function work are:

1. *Basic configuration:* Find all peaks presented in the input function.
2. *Min-peak-height configuration:* A real scalar threshold which needs to be exceeded by local maximums in order to be considered peaks of the function. This threshold determines the minimum acceptable height for a peak.
3. *Min-peak-distance configuration:* A positive integer which defines the minimum distance between indices of the maximum under consideration and the peaks already detected. This specification avoids detecting non-valid peaks in the case of having a maximum with an undesired transition or a "glitch".
4. *Threshold configuration:* A non negative real scalar value that needs to be exceeded by the difference of height of a maximum with the heights of the neighbour points..
5. *N-Peaks configuration:*The maximum number of peaks to be found in the input data.

The basic configuration of the function was firstly used to find peaks without any restriction. Then, each of the other configurations was applied to result in some peaks rejected from the first selection. The restrictions applied in the configurations number 2 and 4 obtained better results than the specification applied in the number 3. In both situations the selection of the the threshold was tested as a possible way to reduce false positives in the detection of peaks using different estimations.

The basic local calculation (*configuration number 1*) finds some peaks that are false positives, because it considers every little increment in the function as a peak. This is why the first restriction (*configuration number 2.*)was applied, remaining those peaks which values are higher than the defined threshold. Several values were defined for the threshold:

- The first value used was **the average value of initial peaks**. In some cases, when the detected peaks do not match with real ones (false



Enum1 Enum2 Enum3 Enum4 Enum5

Figure 5.6: Configuration number 1 to detect peaks with threshold defined with **the average of initial peaks.**

positives detection), this restriction works, as in the case of the three first gestures in Figure 5.6. In this Figure the green points represent the final peaks of the function resulting from the configuration number 2. meanwhile the peaks in blue are the original ones generated by the basic configuration number 1. On the other hand, when there are false peaks closed to real ones, the applied restriction could erroneously discard valid maximums (false negatives detection), like in the case of the image with four fingers. In the right image, the presence of multiple false positives closed to a peak with a high value of the function from the first basic configuration, makes the threshold too high to be exceeded by the last peak, loosing them in the final result (resulting again in the detection of false negatives).

- The next value used to define the threshold was **the 3% of the median value of initial distance function**. This percentage was chosen testing several values to select the minimum number of false positives peaks. The reason to use all initial values from the distance function instead values from preselected peaks to calculate this new threshold is reducing the limit of the height specified by this restriction and then, avoid rejecting real peaks (reduce the number of false negatives). Therefore, the smallest peaks will not be rejected due to the consideration of multiple points in the estimation of the average. Nevertheless, the results obtained here did not introduced any kind of improvement of the results given by the previous threshold uses. So, images yielded from this restriction are not showed here .

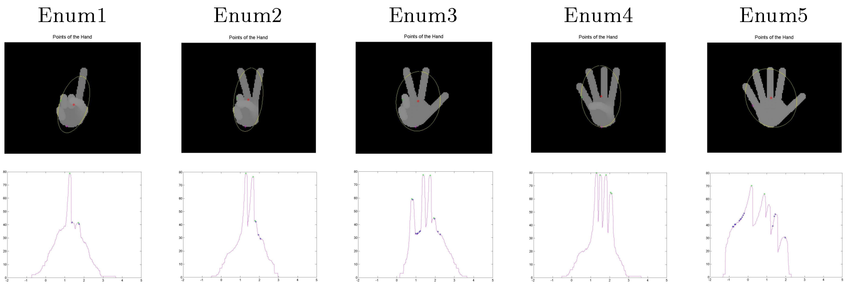


Figure 5.7: Min-peak-distance configuration to detect peaks with threshold defined with **the mean of differences between neighbours**

The next simulation made was the implementation of the *configuration number 4.* i, using differences between neighbours instead of absolute heights. The threshold is defined here was **the mean of differences between neighbours** from distance function. Finally, initial chosen peaks are rejected as final peaks when the difference of height with their neighbours is lower than this threshold, as is shown in Figure 5.7. The points in green colour represents the final remained peaks (from configuration number 4.) meanwhile the blue points belong to the original peaks extracted from configuration number 1. Due to the lower values obtained from this threshold, several peaks were detected erroneously (false positives detections), as it can be observed in the Figure.

This specification was also tried with a little modification:

- The initial vector of peaks selected with configuration number 1. was analysed in windows of 5 elements to calculate the difference of each peak with the four nearest neighbours instead of their difference with all their previous neighbours. Therefore, the presence of a high maximum next to a peak,would not interfere in the selection of that peak using the average of the high differences between neighbours as the threshold. However, the obtained results were not as they were expected to be.

- The combination of configurations number 2 and 4 was tried to improve maxima detection. This means that a set of peaks were selected using the average of distance function as the threshold. Another set of peaks were selected from distance function using configuration number 4. based on differences between neighbours. Finally, common peaks

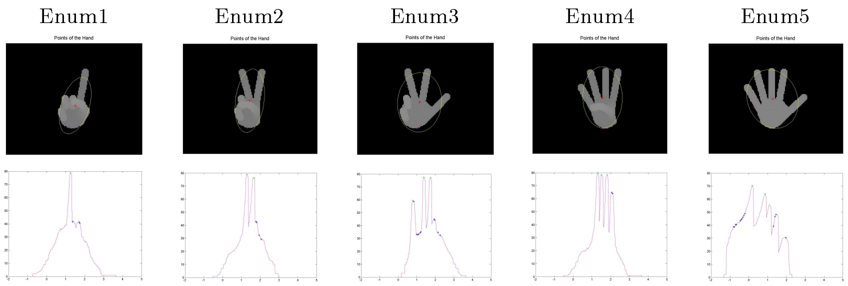


Figure 5.8: Combination of second and third restrictions configuration.

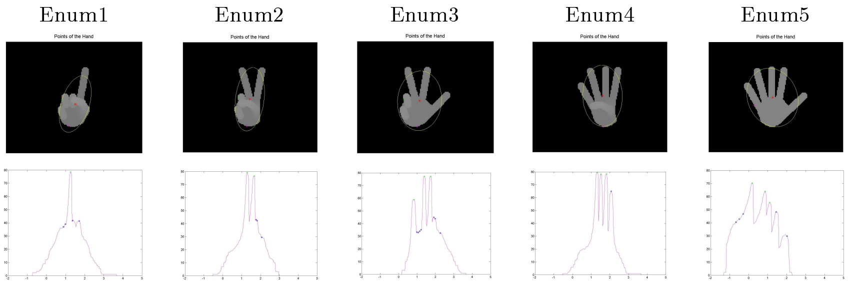


Figure 5.9: Detection of peaks increasing steps between detections

presented in both sets were defined as final peaks of the function. An example of the results obtained from this configuration is shown in Figure 5.8.

The last try made toe improve the detection of peaks consist on increasing of the step in α, reducing the parameter $NumSamples$ to 250. The objective is to reduce the number of false detection of local maxima by using a high sampling frequency. The results obtained from this configuration were better than in the previous implementations. We can conclude that using values from function in more abrupt intervals reduce the problem of detect several maximums from the same local peak (see Figure 5.9).

Finally, the conclusion obtained from the implementation of this existing function is that there is no reference non fixed value which can be used as threshold in the detection of peaks in distance function. This is because the correct selection of the peaks depends, in the first estimation, on all possible peaks of the function. When there is not false positives peaks from initial maximums (outputs of configuration number 1), a threshold based on the mean or average of these peaks would reject valid maximums in the final result, because a mean is always higher than the minimum value of preselected peaks. In the other hand, when there are false positives in the first calculation of possible peaks, the most of undesired maxima have low values of the function, and the threshold used to reject them must be higher than these maximums. Therefore, is impossible to accomplish both objectives at the same time.

Calculation based on a Gaussian fitting Due to the lack of effectiveness in previous tests using Basic Local Calculation 5.2.4.2, finally, the estimation of maxima was made using an analytical Gaussian function result of a fitting process. The chosen fitting function consisted of the sum 5 gaussians and provides better results than a polynomial or exponential function.

The main advantage or performing a function fitting to the input data is that analytical operations can be performed over the resulting fitted function. This way, the candidates to be final peaks can be detected checking basic restrictions in the first and second analytic derivative of the fitted function.

After maxima calculation, the candidates to final peaks are analysed to discard false positives. Several restrictions were analysed for finally applying a restriction based on calculation of areas under the curves of the function. Some tested restrictions are:

- *Restriction in the second derivative slope*: A restriction was applied to the second derivative of the function in the preliminary selection of the peaks. Besides the search of points of the function $f(x)$ where its first derivative function $f(x)^{'}$changes the sign, called critical points, the slope of this function $f(x)^{''}$ was also considered during the selection of maxima. Due to the second derivative of the function represents the speed of the changes in the function at the critical points, when they are an abrupt minimum or maximum $f(x)^{''}$ reaches high levels (independent of its sign, using the absolute value of $f(x)^{''}$). So this restriction will reject peaks when their length or duration is too short (rapid changes) . Therefore, maxima of the function with short duration and quick variation would be rejected with this restriction. In contrary, when a maximum belongs to a finger of the hand, its duration will be long enough to make the values from $f(x)^{''}$ too small to

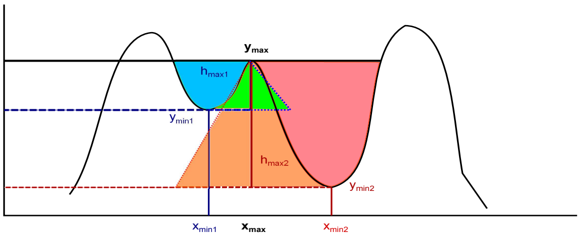


Figure 5.10: Reference heights of the peaks of the function in Tanibata

be rejected. In addition, same steps were followed in order to define minima of function but using a different threshold in the restriction. Furthermore, the first threshold used to define the limit of the function $f(x)''$ depended on the maximum value of $f(x)''$. In particular the 80% of the maximum value of $f(x)''$ were defined as the limit for the maxima detection, and the 20% of the same maximum value for the minima detection. However, in the final tried, a global threshold was used instead these local thresholds due to its few meaningful when the number of maxima were small. In order to determine the value of these thresholds, several graphics of $f(x)''$ from multiple images were analysed.

Therefore, the best results obtained before using the final restriction to calculate protrusions of the hand were the next:

$$u_{maximos} = 3 \cdot 10^4$$

$$u_{minimos} = 1,8 \cdot 10^4$$

- *Restriction in areas under the peaks:*The last restriction used to estimate maxima in the function is focused in computing bulk areas as the area of a triangle. In this approach, the height of each maximum relative to the height of the minima next to it was used. Each maximum usually has two possible relative heights associated to each of its neighbours, the minimum with the smallest height will be used to defined the height of the maximum. Moreover,the point belonging to this last minimum will be used to define the base of the triangle enclosed under the maximum.

- In Figure 5.10 the neighbours of the analysed maximum (the second maximum of the function) are the minima in blue, x_{min1}, and red x_{min2} colour. The area under the maximum resulting from the use of the each minimum is the triangle in green colour, if we use the height of x_{min1}, and the triangle in orange colour, if we use the height of x_{min2}. In the Figure are also defined the heights from each minimum, y_{min1} and y_{min2} as well as the relative heights of the maximum h_{max1} and h_{max2} associated to each minimum. So the area finally obtained with this restriction would be the green triangle generated by the use of the parameters with the blue minimum.

  The area enclosed in the green triangle of the figure 5.10 is estimated with the next equations:

$$a_{max} = \frac{b_{max}.h_{max}}{2}$$

$$b_{max} = 2.(x_{max} - x_{min1})$$

$h_{max} = h_{max1} = y_{max} - y_{min1}$Once the parameters of the area, h_{max_i} and b_{max_i}have been defined for each maximum of the function, the final peaks of the function would be chosen attending to the size of their areas, rejecting the smallest ones.

- *Final Configuration. Restriction in derivatives and triangle areas:* In this paragraph is described the final process followed to obtain the number of protrusions based on the fitting with a Gaussian function and applying restrictions over the derivatives and the areas of the function.

  As it was already mentioned, the input data is fitted with a five gaussians function in order to apply analytic mathematics (see equation REF 5.2.4.2). The number of Gaussian addends in the fitting function was chosen because the maximum number of valid local maxima (fingers) is five. Parameters of the function were obtained with a Matlab function which minimizes the Least Square Mean Error for a fitting to the input data. This fitting results in a function smoother than input data, eliminating candidates to being wrongly detected as peaks. Taking advantage of the implicit restrictions of the input data, some parameters

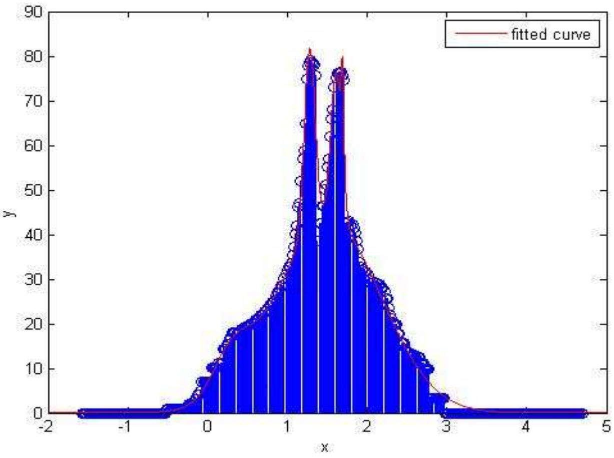


Figure 5.11: Fitted function to contour to wrist point distances.

were limited, such as upper and lower values of the amplitude, variance and mean of the five Gaussian.

$$f(x) = \sum a_i.e^{-\frac{(x-b)^2}{2c_i^2}}; i \in [1, 5], \text{ where the limit values are:}$$

$$a_{min} = 0,\ a_{max} = \infty, ,\ b_{min} = 0,\ b_{max} = 2\pi, c_{min} = 0 \text{and} c_{max} = 2\pi$$

An example of the fitted function to an input data can be found in Figure 5.11. The fitted function is the red continue line over the blue samples of distance to the wrist. The result function is a sum of Gaussian functions, nevertheless this method does not adjust properly in critical points of the function, as we can see in Figure 5.11 where the fit function has two maxima around the second maximum of the original function.

Once the fitted function was obtained, the values of their first and second derivatives were calculated in order to find candidates to local maxima and minima. The method used in this configuration is based on the own definition of maxim and minimum, paying attention to the derivatives of the function during the searching of these critical points. In addition, a variation of restrictions previously defined5.2.4.2 and 5.2.4.2 is used to detect the final peaks which represent the fingers of the hand pose in the image. Definitions used to the establishment of the local maxima and minima are:

- The first specification is focused in finding every local extrema in the distance function as stationary points in accordance with *Fermat's theorem:* If $f(x)$ is a real and continuous function, the point x_0 belongs to the same domain than the function, and $f(x)$ is differentiable at that point $f'(x_0) = 0$, then x_0 is a local extremum of $f(x)$ and if the second derivative $f''(x_0)$ exists, it can classify the point as a maximum, minimum, or inflection point.
- The *first derivative test* checks the value of the first derivative to classify whether these stationary points belong to a local maximum, a local minimum, or neither of them (a turning point).
- The *second derivative test* is a criterion for determining whether a function in a certain point already labelled as particular presents a a local maximum , a local minimum or a possible inflection point. The test determines that the function $f(x)$ has a local maximum at x_0 if $f(x)$ is twice differentiable at that stationary point x_0 besides having $f^{'}(x_0) = 0$, and $f^{''}(x_0) < 0$. In the same way, the function $f(x)$ has a local minimum at x_0 if $f(x)$ is twice differentiable at that stationary point x_0 besides having $f^{'}(x_0) = 0$, and $f^{''}(x_0) > 0$.

Due to the fact that the fitting distance function of our implementation is a real and differentiable function in its interval of definition, the previous definitions could be used to find the maxima and minima of the function. Therefore, values of the fitting function were covered to find out the values of x in which the first derivative of the function changed its sign instead of finding the values where the function took exactly null value. This is because we were working with discrete samples and consequently, a point with $f^{'}(x) = 0$ is difficult to be found within a finite sampling of the function. Furthermore, in order to avoid detecting the same critical point twice, those points where $f^{'}(x) = 0$ are not considered in the selection.

In Figure 5.12 an example of the fitting function , its $1^{st}$and 2^{nd} derivatives is shown.

In figures of distance functions, green points represent the values of the fitting function, points in pink are the values of the first derivative meanwhile the values of the second derivative are drown in black colour.

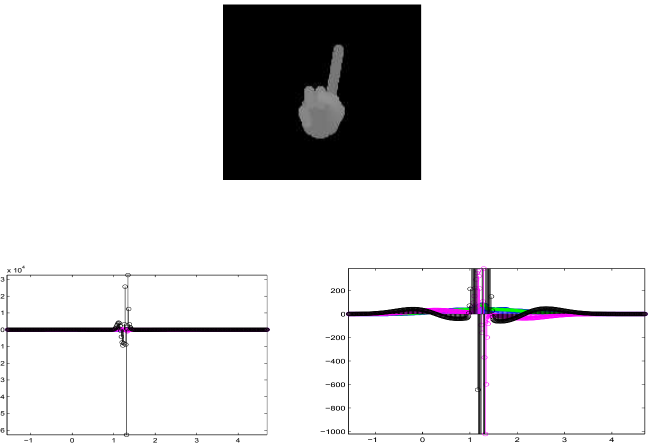


Figure 5.12: Depth image; fitted function, $1^{st}$and 2^{nd} derivatives; detail of the fitted function, $1^{st}$and 2^{nd} derivatives.

The second image belongs to the whole distance function, where the 2^{nd} derivative take high values, and the third image shows with more resolution the values of the fitted function and its first derivative.

To continue with the processed followed, after the search of first candidates to be local maxima and minimums of the function applying the approach to the first derivative test, the second derivative test was combined with some restrictions to choose the first set of peaks.Therefore, when a critical point x_i is found in the first derivative function, the values of the first derivative are analysed between the two consecutive samples x_i and x_{i+1} increasing the accuracy of the range, this is subsampling the function in the interval $X_{sampl} = [x_i : \frac{x_{i+1}-x_i}{1000} : x_{i+1}]$. Then, if the local extremum is a possible minimum $(f'(x_i) < f'(x_{i+1}))$, the second derivative subsampled function is analysed in the point of X_{sampl} where the positives values of $f''(x)$ is the minimum. On the basis of the the second derivative test, if $f''(x)$ is positive in that found point, then it is defined as a local Minimum of the function. Following the same argument, if the local extremum is a possible maximum $(f'(x_i) > f'(x_{i+1}))$, the second derivative subsampled function is analysed in the point of X_{sampl} where the negative values of $f''(x)$ is the closest to 0. So, if $f''(x)$ is negative in that found point, then it is defined as a local Maximum of the function.

Once the initial set of local Maxima and Minima have been estimated, some restrictions are applied to discard false positives Maxima. To begin the discarding, the slope of the Maxima (the second derived of the function) is analysed to reject those which amplitude is very high for a short range of x values. An example of this case is similar to a glitch in the function, an undesired maximum because reach large values very quickly. Therefore, initial local Maxima which second derivative function is higher than a specific threshold will be rejected. The chosen reference level of the second derivative is $6 \cdot 10^4$.

Every time a local extremum is removed its associated local extremum (if it was a maximum, the associated minimum) must change. So, in this case, after removing a local Maximum, the local Minima next to it must change. This means if the rejected maximum had a minimum in both sides, these two minima would be replaced by a single one defined as the average of both old points. Nevertheless, if the rejected maximum had only a single minimum closer, this one would remain meanwhile the maximum would be removed from candidates list.

The next discarding method applied to the survival candidates consists on calculating areas using the area of a triangle enclosed under the curves of the minimums, at first, and the curves under the maximums, at last . This is the restriction defined above about the triangle areas under the maxima 5.2.4.2

Orthogonal-triangular decomposition is the method used to obtain these areas using 3 points of the curve for each region. Required points for the computation of areas under the local Minima are:

- ***p1:***coordinates of the maximum with lowest value of x.
- ***p2:*** coordinates of the minimum.
- ***p3:*** coordinates of the maximum with higher value of x.

In the same way, required points for the computation of areas under the local Maxima are:

- ***p1:*** coordinates of the minimum with lowest value of x.
- ***p2:*** coordinates of the maximum.
- ***p3:*** coordinates of the minimum with higher value of x.

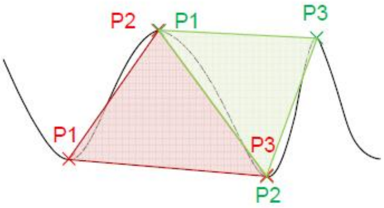


Figure 5.13: Points to calculate triangle area in distance function of the hand

In Figure 5.13 areas enclosed under both, maximum and minimums curves, as well their three points required to define the triangle´s area are showed. Moreover, the area of the first maximum is filled in red and the area of the second minimum is filled in green.

Nevertheless, in this example both curves are closed to other maximum and minimum. When these required points belonging to the extremes of the function (**P1** **and P3)** are not available, the information of the other two points has to be used to estimate the third one. For example, if in the previous Figure, if the point **P1** of the first maximum did not exist, it would have to be defined as:

$$x_{P1} = x_{P2} - (x_{P3} - x_{P2}) = 2x_{P2} - x_{P3}$$

$$y_{P1} = y_{P3}$$

Following the same method, if the point **P3** of the second minimum did not exist, it would have been defined as:

$$x_{P3} = x_{P2} + (x_{P2} - x_{P1}) = 2x_{P2} + x_{P1}$$

$$y_{P1} = y_{P1}$$

In linear algebra, a QR decomposition[3] (also called a QR factorization) of a matrix is a decomposition of a matrix A into a product $A = QR$ of an orthogonal matrix Q and an upper triangular matrix R. This

[3] http://en.wikipedia.org/wiki/QR_decomposition

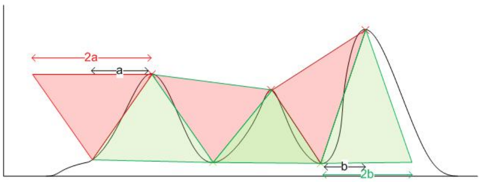


Figure 5.14: Area under the curves to calculate peaks in distance function of the hand

method is often used to solve the linear least squares problem, and is the basis for a particular eigenvalue algorithm, the QR algorithm.

If A has linear independent columns (say n columns), then the first n columns of Q form an orthonormal basis for the column space of A. More specifically, the first k columns of Q form an orthonormal basis for the span of the first k columns of A for any $1\leq k\leq n$. The fact that any column k of A only depends on the first k columns of Q is responsible for the triangular form of R.

The implementation of the method is developed by Dirk-Jan Kroon who used an existing Matlab function„ triangle_area[4], which calculates the area and angles of any triangle described with 3 points of an space with any dimension. Furthermore, the function allow to calculate areas using Heron´s formula besides Orthogonal-triangular decomposition. In geometry, Heron's formula, named after Heron of Alexandria, states that the area A of a triangle whose sides have lengths a, b, and c is

$$A = \sqrt{s(s-a)(s-b)(s-c)}$$

where s is the semiperimeter of the triangle:

$$s = \tfrac{a+b+c}{2}$$

In the Figure 5.14, an example of the calculated areas from Maxima (green triangles) and Minima (red triangles)a is shown:

Once the process used to calculate areas of maxima and minima is explained, the area restrictions applied to remaining local Maxima and Minima are described.

[4] http://www.mathworks.com/matlabcentral/fileexchange/16448-triangle-area-and-angles-v1-3

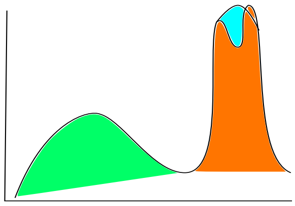


Figure 5.15: Example of a false minimum tooth mark case

As in the previously applied restriction to areas under the peak,the main goal here is rejecting local Maxima with small areas to avoid false positives in the estimation of the number of peaks , hence it has to be identified the possible cases where a maximum is considered a false positive or a negative:

– *Tooth mark case. False positive maximum:* The first case can be define as the situation when we want to remove a false maximum which has a small minimum which splits the maximum in two sections, like a tooth mark. In this case, if the area of each minimum is calculated, the smallest ones can be removed, obtained a bigger single maximum instead. The process followed to reject these kinds of minima are the next:

  First, the areas of each local minima are calculated. Then, these areas are compared with a threshold ($Area$= 2), and they are discarded if their areas are smaller than this value. Once again, if there are two maximum close to a removed minimum, they have to be replaced by the combination of them into a single one, giving to both x and y values the average of the older maxima. In contrary, if they have only one or none maximum close, only rejected minima would be removed. In Figure 5.15 an example of this case can be shown. The graphic represents both initial maxima, the green and orange regions. The final maximum resulting from reject the minimum within the orange region is represented in blue colour.

– *Small minimum next to the initial maximum case. False negative maximum:* The reason to discard minima with small areas before removing the maxima of the function is because if there is a small minimum close to the first or the last maxima, it would

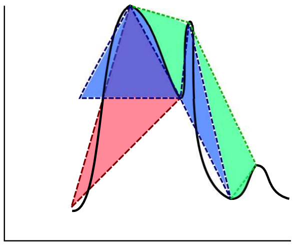


Figure 5.16: Reduced area of the first maximum of the function due to a small minimum

make the calculated area of this maximum smaller than it is in real and hence this maximum would be rejected for being a false negative. In the example of Figure 5.16the first maximum has a wide area (red triangle), nevertheless, the minimum next to it would provide it with an smaller area (blue triangle) than the real one. Therefore, both maxima with the blue area would be rejected. The solution is applying the same restriction explained before this last configuration,t his is the *Restriction in the second derivative slope.* Therefore, the abrupt local Maxima are rejected from the function.

- *Small maximum next to a large maximum. False positive minimum:* This case happens when a minimum which had to be discarded, have the same area as their next and valid minimum. In addition in this case there is a small maximum between both minima like it is represented in Figure 5.17, where the non-valid minimum is in green and the valid one is in pink. Furthermore, the problem introduced by this small maximum located between both minima, is the same as the one described in the previous case . Due to the effect of the second an third maxima, the last minimum (pink) has a similar area to the previous one (green) which is wanted to be removed. it could not been rejected by area restrictions. The solution found to this problem is to reject abrupt local maximum before computing the areas of the remained maxima and minima of the function, and then apply the restriction of areas to remained local Minima.

Once difficult cases found during the implementation of the last configuration have been described, their associated solutions already ex-

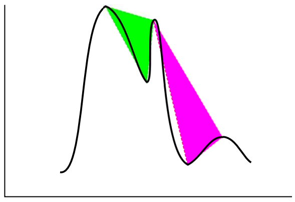


Figure 5.17: Area of a valid minimum versus the area of valid minimum

plained are summarized:

1. Searching of the local extremes by means of the changes in $f'(x)$.
2. Classification of first critical points found by sub-sampling $f''(x)$.
3. Restriction of the maximum values in $f''(x_{max})$ in points associated to initial candidates of local maxima.
4. Restriction of the minimum area $A(x_{min})$ in points associated to initial candidates of local minima

Nevertheless, the are more restrictions and processes in this last configuration to be described.

The next step after removing small minima was the recalculation of the areas from surviving maxima in order to discard maxima with too small areas, like it was made with the minima. At this point of the process we found a critical situation: when all possible minima have been removed but one maxima still remains as candidate. Therefore, when the area of this maximum has to be estimated, two of the three required points do not exist. The solution for this situation is to define this missing points as the lowest and highest points of x where the function is defined ($f(x) \neq 0$). Like in the previous similar restrictions, the area of the remainder maxima is computed and rejected in the case of being too small. Finally, the minima next to the removed maximum have to be unified under its average value.

The last restriction made consists on rejecting the maxima which have a height lower than an specific threshold. This value was determined by the distance from the wrist to the limit of the palm of the hand. This condition allows to remove maxima from distance function of fist images where no protrusion should be identified. Therefore, the maximum distance reached by an outline point in fist images was defined as the minimum distance which a finger of the hand has to exceed. The defined threshold was found out from different observations.

Finally, the number of protrusions was obtained from the number of surviving candidates to local maxima of the distance function. Thus, the above mentioned list of applied restrictions can be completed now:

1. Restriction of the minimum area $A(x_{max})$ in points associated to remainder candidates of local maxima.
2. Restriction of the amplitude of the local maxima $f(x_{max}) > d_{fist}$
3. Number of protrusions: $N_p = \#(x_{max})$

As it was explained in the beginning of this section, only the flatness of the ellipse $r5.2.2$ and the number of protrusions of the image N_p are defined as parameters of the description. Results obtained from this implementation will be describe in the next Chapter 6 besides yielded conclusions of this method.

5.3 Roussos Implementation Concerns

5.3.1 Introduction

In this Section a detailed explanation of the implemented modules, described in [2], will be given.

In the implementation made of Roussos descriptor, 3 different stages can be enumerated:

- Preprocesing: 2D alignment of hand images.
- Base Generation: a set of synthetic images is used to generate the hand base.
- Extraction of the descriptions: the descriptions for the input images are obtained by their projection to the generated base.

Each stage above was implemented to extract descriptions from two sets of images: firstly from TRAINING set and later from TEST images described in Chapter 3. Previously to these stages, images from TEST set were segmented on the basis of the depth information (see Section 4.1), meanwhile in the paper [2] the segmentation of the hand is done using an skin colour detector and templates. Besides the segmentation of images, a normalization process was required to prepare this subset of images for the base generation. This consisted in subtracting the mean image and dividing by the standard deviation image.

The process followed consists on generating the vectors of the base from a subset of the TRAINING set images after their alignment. Then, images from both sets are extracted aligning them firstly with an average image A_0 and projecting them to the base. Finally, the verification of the descriptor performance is made with the reconstruction of images on the basis of their projections.

Now , each of the implemented stages is explained in detail.

5.3.2 Preprocessing Alignment

Before explaining the followed process for the recursively alignment, which is quite similar to the method implemented in the paper [2], it is important to point out why it is mandatory: Principal Component Analysis (PCA) is very sensitive to translations, scale and rotations in the images. This makes very important the alignments previous to the base calculation. For the generation of the base are recursive alignment process (based on the affine alignment describes in Section 4.2.1). It can be summarized in the following points:

1. *Selection of the first image from the training set as A_0 image*: Assuming that the input images for the base generation have to be aligned, A_0 was initialized as the average of these base images. The size of this image, and of the rest of the images involved in the base generation is incremented adding black rows and columns of pixels. This way we avoid loosing information of the images being aligned when the applied transformations put pixels out of the original dimensions of the image.

- *Alignment of images with current A_0 using the affine transform.* Images have to be aligned to avoid rotations, scales and translations. The details about the whole process involved in the affine transform can be found in Section 4.2.1. The three principal points (see Section 4.2.1.1) of each image are required to be aligned with the principal points of A_0, which define the reference of the transformation.

- *Estimation of the new reference image$A_0^{'}$*: After the first iteration of alignment of images, the resulting image A_0 needs to be reestimated in the new $A_0^{'}$ by the average of aligned images. Obviously, this $A_0^{'}$ has the same size as the biggest aligned image.

- *Comparison of the new $A_0^{'}$ with the previously calculated A_0:* The mean A_0 is updated iteratively until the mean squared of the difference with the previous mean image is under a certain threshold: the unit. Therefore, steps from the second to this last one, are repeated changing A_0 by $A_0^{'}$ until there is no significant difference between both reference images.

5.3.3 Base Generation

The implementation of the hands base generation is based on a code originally proposed by Alex P. Pentland and Matthew A. Turk from MIT in 1991 for the description of faces using a Eigenfaces based approach[5]. Authors of this method obtain the mean image of a predefined set of face images. Then, the recognition of face images is applied weighting the difference between face images projections to a set of eigenvectors. If the difference does not exceeds a specific threshold, projected image is assumed as an known face, which will be defined by the eigenface with the lowest difference among the previously calculated. Due to the similarity with the goal of this project, the mentioned method was used for the implementation of Roussos descriptor extractor. In next lines the process carried out is described.

In general lines, the process consists on computing the covariance matrix of previously aligned training images resulting from the preprocessing of images (see Section 5.3.2); extracting eigenvalues and eigenvectors from this matrix and applying PCA to select the principal components of the new computed hand base.

[5] http://www.pages.drexel.edu/~sis26/Face%20recognition.htm

Before continuing with the explanation of the implemented techniques, the PCA foundations will be pointed out. The next explanation is based in [59] and documentation from Wikipedia[6].

Principal component analysis (PCA) is a mathematical procedure which generates a new reduced set of uncorrelated variables, the principal components, from a wide range of interrelated variables of a data set using an orthogonal transformation. Besides the reduction of the data dimension, another goal of this procedure is to retain as much as possible the variability in the data ordering the principal components by their variance, higher in the first component than in subsequence ones. This can be understood as a reduction in the number of components in a way in which we keep the most relevant information of the input data. Afterwards, the resulting component can be sort by the quantity of information they are able to express.

PCA analysis is usually used as a tool in exploratory data analysis as well as in generation of predictive models. One of the possible methods to apply PCA which we have used in this study is the estimation of a covariance matrix and its decomposition in eigenvalues. Results from this method are classified in terms of component scores and loadings. The first concept corresponds to the values from data transformations, this is a particular observation of the experiment. In contrary, loadings are the weights required to obtain a specific value in the transformation domain by their multiplication with the standard original variable. In terms of the hand descriptor, scores of each variable or image would be the description of the variable or input image.

Moreover, if this procedure is used in a multivariate dataset, each variable can be represented as a new axis in a high-dimensional space where data is represented by the coordinates of their projections into this new space.

Now, it is going to be described the implementation of the principal components extraction using the covariance method, which provides input base images with an orthogonal linear transformation to project input images into the new coordinate system.

The first step consists on arranging the data set chosen for the generation of the base in order to reduce the number of elements of the base after being

[6] http://en.wikipedia.org/wiki/Principal_component_analysis

aligned with the image A_0. Because each image has to be considered as a a different repetition or observation of the same experiment, pixels of each image are placed in the same row along different columns (one column for each image). Therefore, if we had M images with N pixels, the initial data set matrix obtained is a NxM matrix.

$$X_{NM} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1M} \\ x_{21} & x_{22} & \cdots & x_{2M} \\ \vdots & \vdots & \vdots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{NM} \end{bmatrix} \tag{5.1}$$

PCA proposes an orthogonal decomposition of symmetric positive semidefinite (PSD) matrix, in particular of the Covariance matrix, to provide images with an orthogonal basis of eigenvectors.

If we attend to the definition of the Covariance matrix of M random variables with a Gaussian distribution, we can explain the next steps of the implementation by the equations of that Covariance matrix:

$$C_{x_1,x_2 \ldots x_M} = \begin{bmatrix} C_{11} & C_{12} & ... & C_{1M} \\ C_{21} & C_{22} & \ldots & C_{2M} \\ \vdots & \vdots & \vdots & \vdots \\ C_{M1} & C_{M2} & \ldots & C_{MM} \end{bmatrix} \tag{5.2}$$

$$C_{i,j} = E\left[(x_i - \bar{x_i})(x_j - \bar{x_j})\right] = \sum_{k=1}^{N}\sum_{l=1}^{N}(x_{ik} - \bar{x_i})(x_{jl} - \bar{x_j}); i \neq j \tag{5.3}$$

$$C_{i,i} = \sigma_i^2 = E[(x_i - \bar{x_i})^2] = \sum_{k=1}^{N}(x_{ik} - \bar{x_i})^2; i = j \tag{5.4}$$

Because each element of the covariance matrix is the covariance of each image with the rest of images, except the elements from the diagonal which are their own variance, the next step of the process is the calculation of the average from each observation into an empirical mean vector with M elements.

$$u_i = \frac{1}{N}\sum_{i=1}^{N} x_i; i \in [1, M] \tag{5.5}$$

$$M_{MxN} = \begin{bmatrix} u_1 & u_1 & \cdots & u_1 \\ u_2 & u_2 & \cdots & u_2 \\ \vdots & \vdots & \vdots & \vdots \\ u_M & u_M & \cdots & u_M \end{bmatrix} \tag{5.6}$$

After calculating the empirical mean of each image, they have to be subtracted from each of its pixel, making that resulting principal components minimize the mean square error of approximating data. This is because the global mean and variance from the resulting data distribution function were the same with independence of hand distances during the capture, etc. This also allows the covariance matrix resulting from these normalized set to be centred on 0 with unit variance. Therefore, if the input data matrix is transposed and centred on the empirical mean matrix later, the resulting matrix will be a MxN matrix.

$$H_{MxN} = X' - M$$

Finally, using the proprieties of the outer product and the conjugate transpose the covariance matrix is generated as the sum of outer products of the previous matrix H.

$$C_{MxM} = \begin{bmatrix} C_{11} & C_{12} & ... & C_{1M} \\ C_{21} & C_{22} & \ldots & C_{2M} \\ \vdots & \vdots & \vdots & \vdots \\ C_{M1} & C_{M2} & \ldots & C_{MM} \end{bmatrix} = E\left[H \otimes H\right] = E\left[H \cdot H^{\star}\right] = \frac{1}{N}\sum_{i=1}^{N} H \cdot H^{\star} \tag{5.7}$$

The next step consists on extracting eigenvalues and eigenvectors of this covariance matrix which satisfy the next equation.

$$C.v = \lambda v$$

As it was mentioned before, extracted eigenvector and eigenhands by this process will be the base elements of a new orthogonal basis of images. The method used allows to extract directly eigenvalues from the diagonal matrix . Moreover, this diagonal matrix results from multiplying our covariance matrix with the matrix which contains eigenvectors and which can diagonalize this covariance matrix. However, there are some preconditions to be fulfilled by matrix involved in these calculations. These conditions are those which ensure that the covariance matrix is a real symmetric and square matrix, allowing extract simply the values of its eigenvectors and eigenvalues:

1. If C is the square complex Covariance matrix, C is normal if and only if:

   (a) $C * .C = C.C*$

   Among complex matrices, all unitary, Hermitian, and skew-Hermitian matrices are normal. Likewise, among real matrices, all orthogonal, symmetric, and skew-symmetric matrices are normal.

2. If the Covariance matrix is real besides normal, then:

   (a) $C* = C^T$

   (b) $C^T.C = C.C^T$

3. U is an unitary matrix if and only if:

   (a) $U * U = UU* = I; \Rightarrow$

   (b) $U^{-1} = U*$

The ***spectral theorem*** says that every normal matrix is unitary similar to a diagonal matrix, this is if A is a normal matrix ($AA* = A * A$) then there exists a unitary matrix U such that $U * AU$ is diagonal. It can be demonstrated by the ***Schur decomposition***, where the normal matrix can be written as $A = UTU*$, defining U as an unitary matrix and T as an upper-triangular matrix. Since A is normal, $TT* = T * T$. Therefore T must be diagonal. Finally, A can be said is a normal matrix if and only if there exists a unitary matrix U such that $A = UTU*$, where the entries of the diagonal matrix T are the eigenvalues of A and where the column vectors of U are the orthonormal eigenvectors.

Therefore, if previous 1,2 and 3 conditions are fulfilled by the Covariance matrix, this means that it is a real square and normal matrix, there exists a unitary matrix V which generates the diagonal matrix D , an MxM matrix containing the eigenvalues of the matrix C. In addition, we can extract the characteristic equation from the ***spectral theorem, the Schur decomposition*** and above-mentioned properties:

1. Spectral theorem and Schur decomposition:
   (a) $V * .C.V = D$
   (b) $C = V.D.V^*$
   (c) $C = V.DV^{-1}$
2. Diagonal matrix containing eigenvectors:
   (a) $D = \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0_1 & 0 & \lambda_M \end{bmatrix}$
3. Unitary matrix containing orthonormal eigenvectors
   (a) $V^{-1} = V*$
   (b) $V * .V = V.V* = I;$
4. Multiplying matrix V in both right sides of 1.c
   (a) $C.V = V.D.V^{-1}.V$
   (b) $C.V = V.D.I$
   (c) $C.V = V.D$
5. Finally, the characteristic equation from the square Covariance matrix is achieved.The solutions to this equation are the desired eigenvalues λ_i of the matrix D which diagonalize the covariance matrix C

   (a) $C.v_k = \lambda.v_k$

   (b) $(C - \lambda_k I)v_k = 0$

6. If we want to obtain non-zeros vectors (if λ is wanted to be a eigenvalue of C) the matrix $C - \lambda_k I$ has to be a singular or non-invertible matrix. This goal can be achieved making $det(C - \lambda_k I) = 0$. Consequently, resulting roots of λ_i will be eigenvalues of Covariance matrix, and their corresponding eigenvectors will be used to create the V diagonalizable matrix. Finally, we can obtain the M different eigenvalues from each eigenvector of the M ones (each eigenvector has M elements):

   (a) $\lambda_k = [\lambda_1, \lambda_2, ..., \lambda_M]$

   (b) $$\begin{bmatrix} C_{11} & C_{12} & ... & C_{1M} \\ C_{21} & C_{22} & \dots & C_{2M} \\ \vdots & \vdots & \vdots & \vdots \\ C_{M1} & C_{M2} & \dots & C_{MM} \end{bmatrix} \begin{bmatrix} v_{11} \\ v_{12} \\ \vdots \\ v_{1M} \end{bmatrix} = \lambda_1 \begin{bmatrix} v_{11} \\ v_{12} \\ \vdots \\ v_{1M} \end{bmatrix} \Rrightarrow \overrightarrow{v_1} = [v_{11}, v_{12}, ..., v_{1M}]$$

   (c) $$\begin{bmatrix} C_{11} & C_{12} & ... & C_{1M} \\ C_{21} & C_{22} & \dots & C_{2M} \\ \vdots & \vdots & \vdots & \vdots \\ C_{M1} & C_{M2} & \dots & C_{MM} \end{bmatrix} \begin{bmatrix} v_{M1} \\ v_{M2} \\ \vdots \\ v_{MM} \end{bmatrix} = \lambda_M \begin{bmatrix} v_{11} \\ v_{12} \\ \vdots \\ v_{1M} \end{bmatrix} \Rrightarrow \overrightarrow{v_M} = [v_{M1}, v_{M2}, ..., v_{MM}]$$

Nevertheless, for the implementation of this algorithm, an existing Matlab function, eig [7] ,was used to extract the set of eigenvalues and eigenvectors of the matrix resulting from multiplying our centred set of images H , without the need of computing these tedious calculations. The Eigenvectors are the principal components of the covariance matrix, and their eigenvalues, are extracted from the covariance matrix.. Then, non zero eigenvalues which are in a diagonal matrix are stored besides theirs associated eigenvectors. Moreover, each eigenvector have M elements related to each eigenvalue.

The next step consist on rearrange both eigenvectors and eigenvalues to sort decreasing the eigenvalues, having the largest ones at the top of the matrix. This means that largest eigenvalues correspond to upper positions of the matrix, as well as the matrix with eigenvalues has to be rearranged in the

[7] http://www.mathworks.es/help/techdoc/ref/eig.html

same manner. Furthermore, the principal-components that are associated with most of the covariability will be yielded from the first columns of the matrix.

Once eigenvectors and their corresponding eigenvalues have been rearranged, they have to be also normalized in order to avoid problems when input images are measures with different sources.

$$\overline{v_k} = \frac{\overrightarrow{v_k}}{\sqrt{\sum_i (v_{ki})^2}}$$

At this time, ***eigenhands*** defined as the orthonormal image basis can be calculated. So the eigenvectors set has to be normalized with respect to its variance, this is the root of its eigenvalues. Finally, the loading or the final values of the eigenhands are resulting from the product of the aligned centred images with unit eigenvectors of the basis.

$$\overrightarrow{e_k} = \frac{(X-\overline{X}).\overrightarrow{v_k}}{\sqrt{{d_k}^2}} = \frac{H.\overrightarrow{v_k}}{\sqrt{{d_k}^2}}$$

Finally, the M ***unit eigenhands*** are consequently extracted from the normalization of the previous vectors with the same size than input images:

$$\bar{e}_k = \frac{\overrightarrow{e_k}}{\sqrt{\sum_i (e_{ki})^2}}$$

In order to apply ***PCA***, the projections of the images from the base set have to be obtained to calculate the cumulative energy among the eigenhands. Therefore, the weights or projections associated to the k^{th} image can be calculated as the dot product of the k^{th} image with M eigenvectors of the base. Each image will have one weight associated to each eigenhand:

$$w_k = [< e_1, h_k >; < e_2, h_k >; ... < e_M, h_k >]$$

The final selected eigenhands are chosen from the cumulative energy of these projections. Therefore, depending on the desired resolution of the regenerated hand images, we can chose the percentage of energy to be accumulated by the projections of base images from the chosen set of eigenhands. In fact, different resolutions can be appreciated in the implementation of the descriptor, where three different amounts of energy haven been stored: 95%, 99% and 100%.

$$g = \sum_{i=1}^{N_c} w_i^2$$

$$\frac{g}{\sum_{i=1}^{M} w_i^2} \leq 95\%; 99\%; 100\%$$

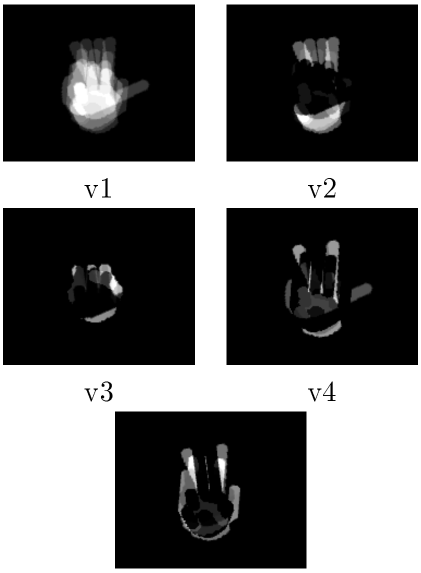
The more energy is stored with eigenhands, the higher precision the reconstructed images will present, although the reduction applied to the principal components will be smaller. This issue is the responsible of defining the number of the elements which compounds the hand base. In [2] this number is named Nc. There are two contrasting trends considered in the paper. The first one attends to the fact that the smaller is the order of the descriptor, the easier the test images descriptions are classified in the correct cluster: this implies high recognition rates. In addition, the discrimination among different points in the eigenhands feature space is low. However, the greater the order of the descriptor, the more accuracy in the discrimination of the different gestures.

Finally, when definitive eigenhands are calculated with the cumulative energy, they are stored to be used as vector images of the base.

The relationship between PCA and the implemented descriptor is that each input image used in the generation of our base can be considered as a particular observation of the experiment, where there are interrelated variables. Therefore, the extraction of principal components of this set of images obtains a new set of uncorrelated variables (or images) which can be similar or smaller than the original set, but t enough to regenerate an image which bears a strong resemblance to the original one.

The right reconstruction of images from their description depends on the set of images used in the generation of the base. This set has to be representative enough to provide a hand base with the minimum number of elements required to describe the input image.

Figure 5.18 represents eigenhands of a particular set of synthetic images. However the vectors of the base are similar to images from the base hand



v1 v2

v3 v4

v5

Figure 5.18: Eigenhands of the [6] dictionary generated from a set of synthetic images with range of $\theta_1^x, \theta_1^y, \theta_1^z = \pi/8$, 1 POV and 95% of the energy

set, but they are not the same and even less when the number of its elements is lower than in the original set of base images (before applying PCA). But in the implementation of the descriptor, their mean and average had been modified in order to allow eigenhands to be shown as images. In this Figure we can notice the resemblances of eigenhand with the images used for the generation of the base (see Figure 3.7c).

5.3.4 Description Extraction

After the generation of the eigenhands of the base, input images from both set of images, TRAINING and TEST, can be described with their projections into this new base.

The description proposed in [2]is the same than we used in its implementation: the weights/values of the projection from each image into the hand base. These values result from the dot product of input aligned images with the eigenhands of the base. However, like images used for the generation of the base, images to be described have to be prepared before extracting their projections. This preparation consists on:

- A standard normalization of images, consisting on subtracting them its average and dividing them by its standard deviation.
- An affine alignment of images with A_0, performed as explained in Section 4.2.1 .

As it was commented at the beginning of this paragraph, the projection was obtained from the dot product of these aligned and centred images with the eigenhands of the base. These projection of images are the weights of the image with each image of the base, how much can contribute each element of the base for the generation of the input image:

$$\lambda_k =< e_k, (x_k - \overline{x}_k) > \tag{5.8}$$

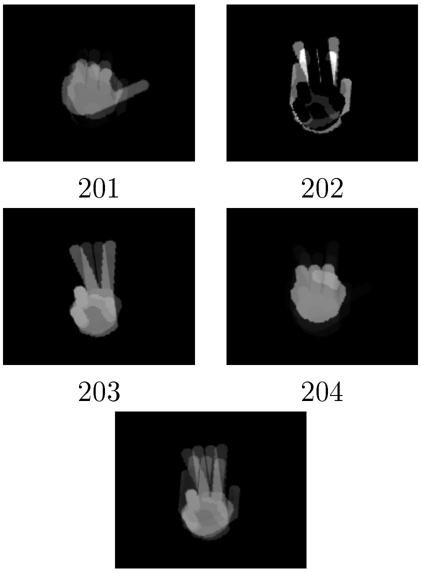
$$\lambda = [\lambda_1, \lambda_2, ..., \lambda_{Nc}] \tag{5.9}$$

The reconstruction of the images from the description and the eigenhands was implemented too. The method used for the reconstruction is the same as proposed in [2]: by the linear combination of the hand base images and the mean reference image A_0 as well as the projection of the images:

$$f(x) = A_0 + \sum_{i=1}^{N_c} \lambda_i \cdot E_i \tag{5.10}$$

In Figure 5.19 synthetic reconstructed images are extracted.

To conclude it is important to emphasized the quality of the reconstructed images depends on the variability of the images for the base generation. This means that the more different the images for the base are, the better the reconstruction of the new images to be described will be. After applying PCA to the elements of the base, with a particular amount of energy stored



201 202

203 204

205

Figure 5.19: Reconstructed synthetic images

by eigenvectors, only the images with the highest weights in their projections will remain.

For example, the palm of the hands is a common feature of all images, therefore it will be present in the first component which includes the highest quantity of information of the data set. In contrast, fingers involve a smaller contribution of the images depending on the kind of gesture. But, the forefinger is the most common of all the fingers in numeric dictionaries. This implies that it will be probablyincluded in the final set of principal components, remained after the selection of the 95% of cumulative energy. Nevertheless, the thumb will be probably not included within this set, due to his lower contribution in base images , unless more images of hands with outstretched thumb were used in the base set.

The results obtained in the implementation of this descriptor will be described in the next chapter 6, besides the demonstrations and conclusions extracted from the descriptor discussed above.

Chapter 6

Results and Conclusions

6.1 Evaluation Scheme

6.1.1 Framework:Weka

Weka[1] open source software is included in GNU General Public License and it was selected in this study to the evaluation of implemented descriptors. It provides with a collection of machine learning algorithms , as well as with useful tools for data pre-processing, classification, regression, clustering, association rules, and visualization. Therefore, there are two specific test fields yielded from Weka analysis which are the centre of attention for the evaluation stage of this project: to estimate the separability of the categories defined..

This way we will be able to estimate the performance of each model to describe the hand poses captured in the test images. Moreover, the more separated the components of the description are, the easier it will be to classify data in different categories.

Given a set of training examples, with their correspondent descriptions and annotations (i.e. its belonging to one of the possible categories, in our case the gestures of a dictionary), a unsupervised learning technique is applied to build a model that later assigns a predicted label to new images belonging to the test set. This model tries to map that examples into separate categories as wide as possible.

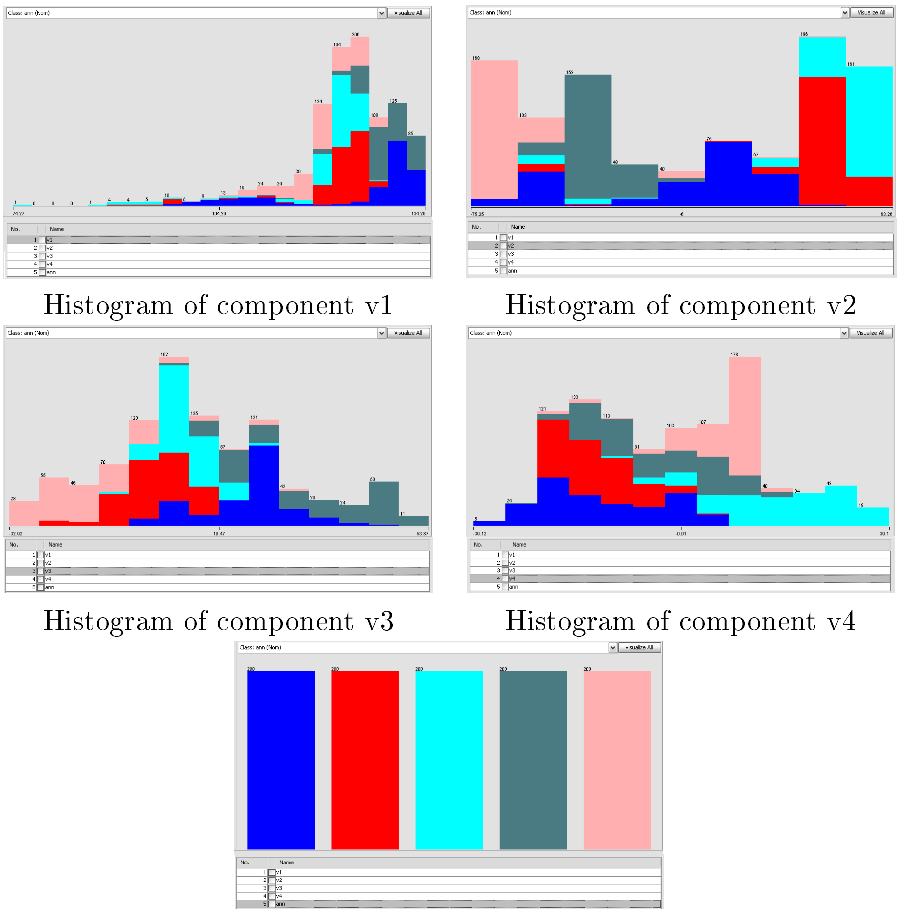
[1] http://www.cs.waikato.ac.nz/ml/weka/

As an illustrative way of visualizing the set of descriptions is to represent each coordinate of each local description in an histogram, assigning a different colour depending on to which class the description belongs to. In Figure 6.1 we find an example of these histograms for the synthetic training set Roussos descriptions. Only the first 4 coordinates of the descriptions were taken under consideration, this is, the 95% of the energy of the original images is achieved. Thus, the description of each image, this is each instance, has four components or attributes. Analysing descriptions and annotations attached to each instance, each component of the description can be separated in different categories attending to its values in each hand pose of the training images. Therefore, Weka generates the histogram of each component of the description from instances of the training set, and identifies for each income the different class of hand poses which belongs to it.

In Figure 6.1 we can find the histograms for each coordinate of the pattern for a classification problem with four features plus the annotation label. For example, the first component of the description belongs to v1 and reaches high values in all the hand poses, although the highest ones are taken by the poses that belong to blue and grey classes.

A possible approach for the classification of these descriptions in hand poses classes would be the definition of rules. For example, basing on the histograms of we could deduce that the hand posture associated to pale blue class will probable be the descriptions which have the highest values of components v4 and v2. In contrast, gestures represented by grey samples can be easily detected if the component v3 of the descriptor are higher than 50 because only this class of hand postures reaches these values with this attribute. Anyway, this approach gets too complex with the increment of the number of coordinates or simply when the instances are not easily separable. This is why the followed approach for this evaluation stage consists on the use of unsupervised learning techniques, more concretely, the use of a Multi-Layer Perceptron (MLP).

Another important concept necessary for understanding outcome from Weka analysis is the confusion matrix. It is generated from the predictions made over the test set of images. Based on the annotations of that test set, the



Histogram of component v1

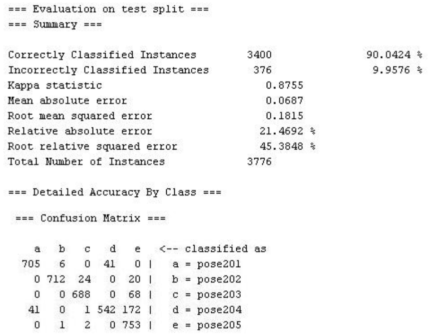
Histogram of component v2

Histogram of component v3

Histogram of component v4

Histogram from annotations of each hand pose

Figure 6.1: Pattern classification from Roussos descriptor using a base of the 95% from the input set energy .



```
=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances        3400               90.0424 %
Incorrectly Classified Instances       376                9.9576 %
Kappa statistic                          0.8755
Mean absolute error                      0.0687
Root mean squared error                  0.1815
Relative absolute error                 21.4692 %
Root relative squared error             45.3848 %
Total Number of Instances             3776

=== Detailed Accuracy By Class ===

 === Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 705   6   0  41   0 |   a = pose201
   0 712  24   0  20 |   b = pose202
   0   0 688   0  68 |   c = pose203
  41   0   1 542 172 |   d = pose204
   0   1   2   0 753 |   e = pose205
```

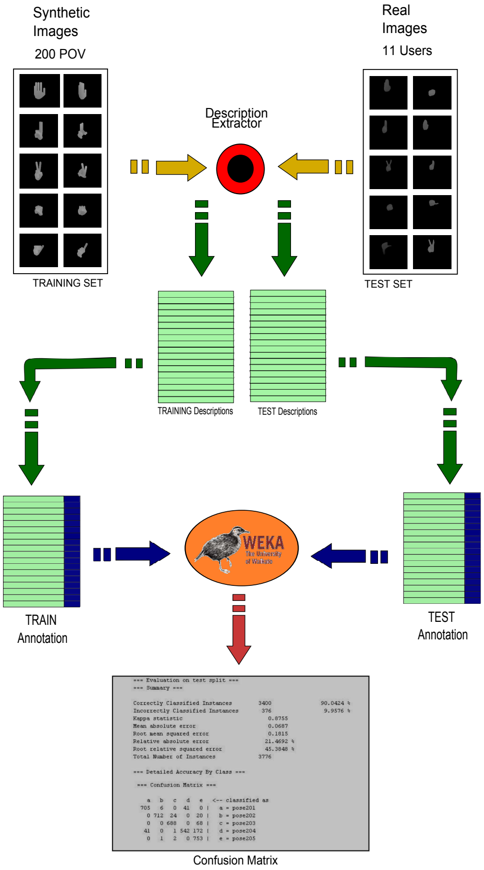
Figure 6.2: Confusion Matrix from Roussos descriptions using a base of the 95% of image energy

predictions given by the neuronal network can be evaluated to asses the quality of the descriptor. Thus the matrix shows which hand poses have been classified correctly and which of them have been confused with other ones. In addition, Weka shows the percentage of these hand poses correctly described. An example of confusion matrix is shown in Figure 7.2,

In the matrix shown in this Figure 7.2 it can be appreciated that the pose 204 is confused with the pose 201 in some cases. It can also be noticed that pose 205 is the best classified hand gesture, a low number of images were assigned to other hand poses. In contrast, the images belonging to pose 204 class show the worst detection results of all images because in several cases it was confused with pose 205.

6.1.2 Modular Scheme

Following the process described in the Figure 6.3, the evaluation of both implemented descriptors Tanibata and Roussos is described in this paragraph. Moreover, due to the independence of the evaluation machine with the implemented methods to extract description data, the followed scheme was the same for the two studied models of the project.



```
=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances        3400               90.0424 %
Incorrectly Classified Instances       376                9.9576 %
Kappa statistic                          0.8755
Mean absolute error                      0.0687
Root mean squared error                  0.1815
Relative absolute error                 21.4692 %
Root relative squared error             45.3848 %
Total Number of Instances             3776

=== Detailed Accuracy By Class ===

 === Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 705   6   0  41   0 |   a = pose201
   0 712  24   0  20 |   b = pose202
   0   0 688   0  68 |   c = pose203
  41   0   1 542 172 |   d = pose204
   0   1   2   0 753 |   e = pose205
```

Figure 6.3: Followed processes for the evaluation of the descriptors.

As it can be shown in the Figure 6.3, the evaluation scheme firstly requires the descriptions and annotations from synthetic and real images. The example shown in the Figure includes hand gesture images from Soutschek dictionary. Regarding to the collection of images we used to evaluate our descriptors, two kinds of images were selected attending to the training and test stages. The training images set is compound of 200 different synthetic images for each hand pose of the dictionary. This collection was used to train the MLP algorithm. The test set of images are captures of hand poses extracted from 11 different users.

Once the description data is extracted, the annotation related to each hand pose is added to instances of test and train sets. Then, the machine learning software establishes the relationships between descriptions and associated hand poses in order to generate the predictions from the descriptions of the test data. Weka framework provides several classifying functions to extract and assess predictions from data descriptions, however the multilayer perceptron is the algorithm we used to this issue. Thus, back-propagation technique applied in the train stage of the project tries to find the statements which allow to classify better each hand pose attending to their descriptions. Finally, after generating predictions of the test set of images, annotations included are used to assess those predictions. Therefore, knowing the hand poses corresponding to each instance of the test, the confusion matrix is yielded from the outcomes of the evaluation program. Moreover, besides the confusion matrix, Weka provides the percentage of gestures correctly classified during this modular scheme of the test stage.

However, results obtained from this evaluation process, were generated from normalized training instances and the multilayer perceptron classifier algorithm. Due to the the results obtained from the normalized data are better than using data without being pre-processed, the next configurations were applied to instances of the training set:

- Normalization of instance Unsupervised: This normalization only affects to numeric attributes, ignoring class index. Configuration parameters of this function:
    - Norm: The norm of the instance after normalization.
        - Norm =1.0:

  - Lnorm: The Lnorm to use
    - ∗ Lnorm=L2.0

However, the results obtained from this evaluation process are generated on the basis of normalized training instances. The applied normalization method is widely used and it normalizes its instance independently from the othersAs it was above mentioned, among the provided by Weka framework classifiers, the multilayer perceptron, a fed-forward artificial neuronal network was chosen to make the predictions during the evaluation process. These kind of networks are made up with multiple fully connected layers of nodes in a direct graph. Each processing element of the network is a node which has a non-linear activation function. Based on the supervised learning back-propagation technique to the automatic pattern recognition, some parameters of the algorithm were configured as it is defined below to performance the descriptions trials.

- Multilayer Perceptron Classifier: Back-propagation technique is used here to classify instances. The nodes created in this network are all sigmoid (except for when the class is numeric in which case the the output nodes become unthresholded linear units). Set-up parameters were the default ones, with the "normalize Attributes" flag disactivated:
  - Learning rate: The amount the weights are updated, L=0.3.
  - Momentum: Momentum applied to the weights during updating, M=0.2.
  - Training time: The number of epochs to train through. If the validation set is non-zero then it can terminate the network early, N=500.
  - Seed: This parameter is used to initialise the random number generator. Random numbers are used for setting the initial weights of the connections between nodes, and also for shuffling the training data, V=0.
  - Validations set size: The percentage size of the validation set. (The training will continue until it is observed that the error on the validation set has been consistently getting worse, or if the training time is reached). If This is set to zero no validation set

Syn. angles ranges ($\theta_1^x, \theta_1^y, \theta_1^z$)	$([0, \pi/8], [-\pi/8, \pi/8], [-\pi/8, \pi/8])$	$([0, \pi/4], [-\pi/4, \pi/4], [-\pi/4, \pi/4])$
Dict [6]	61.2843 %	60.873 %

Table 6.1: Tanibata descriptor performance for different dictionaries and with different set-ups.

will be used and instead the network will train for the specified number of epochs.S=0.

– Validation threshold: Used to terminate validation testing. The value here dictates how many times in a row the validation set error can get worse before training is terminated, E=20.

To conclude, the results obtained for Roussos and Tanibata descriptors are presented in next Section.

6.2 Results

The experiments carried out take into account different set-ups. Common to Roussos and Tanibata descriptors we vary the variation ranges of the three global rotation angles of the synthetically generated images. For Roussos there are two more parameters to vary: relative PCA energy of the selected eigenvalues relative to the all of them, and the number of Points Of View (POV) of each pose in the collection for the generation of the base.

Here we present some results for dictionary [6].

PCA energy (%) | Syn. angles ranges$(\theta_1^x, \theta_1^y, \theta_1^z)$	95%| $([0, \pi/8], [-\pi/8, \pi/8], [-\pi/8, \pi/8])$	95%| $([0, \pi/4], [-\pi/4, \pi/4], [-\pi/4, \pi/4])$	99%| $([0, \pi/8], [-\pi/8, \pi/8], [-\pi/8, \pi/8])$	99%| $([0, \pi/4], [-\pi/4, \pi/4], [-\pi/4, \pi/4])$	100%| $([0, \pi/8], [-\pi/8, \pi/8], [-\pi/8, \pi/8])$	100%| $([0, \pi/4], [-\pi/4, \pi/4], [-\pi/4, \pi/4])$
Dict [6]	71.31%	36.746 %	48.355 %	58.7662 %	59.7691 %	58.7662 %

(a) 1 POV for each pose in the collection for base generation.

PCA energy (%) | Syn. angles ranges$(\theta_1^x, \theta_1^y, \theta_1^z)$	95%| $([0, \pi/8], [-\pi/8, \pi/8], [-\pi/8, \pi/8])$	95%| $([0, \pi/4], [-\pi/4, \pi/4], [-\pi/4, \pi/4])$	99%| $([0, \pi/8], [-\pi/8, \pi/8], [-\pi/8, \pi/8])$	99%| $([0, \pi/4], [-\pi/4, \pi/4], [-\pi/4, \pi/4])$	100%| $([0, \pi/8], [-\pi/8, \pi/8], [-\pi/8, \pi/8])$	100%| $([0, \pi/4], [-\pi/4, \pi/4], [-\pi/4, \pi/4])$
Dict [6]	69.3795 %	72.9076 %	65.57 %	70.5051 %	47.4242 %	71.1977 %

(b) 9 POV for each pose in the collection for base generation.

Table 6.2: Roussos descriptor performance for different dictionaries and with different set-ups.

-

Chapter 7

Conclusions and Future Work

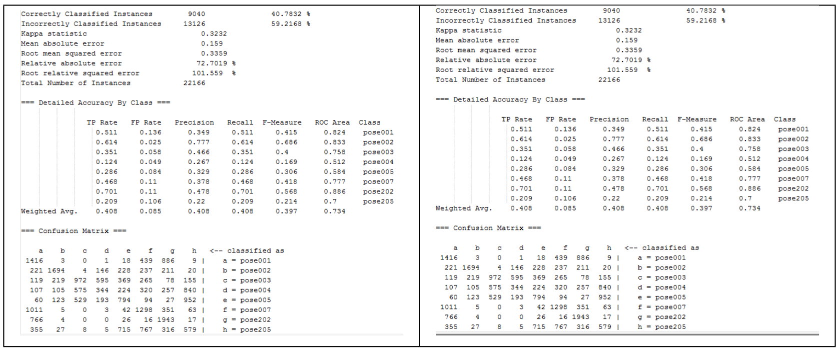
7.1 Conclusions

7.1.1 Tanibata Descriptor

The main differences of the results obtained from the simulations of Tanibata descriptor are not directly related to the kind of configuration involved in the trials, but it is related with the kind of gestures included in each dictionary. In fact, due to the fact that only two parameters, the ratio of the ellipse and the number of protrusions of the hand, are used to describe images, the quality of each element of the description depends on the kind of the hand gesture to be described. Attending to each parameter of the description, some conclusion can be extracted from the experiments:

- *Ratio of ellipse, r.* This feature of the hand is especially useful to differentiate hand poses with extended fingers from fist images, where only the knuckles of the hand match with the contour of the ellipse. This ratio also allows to distinguish hand poses without a thumb from gestures containing the thumb extended, because the difference of the aspect ratio between both images.

  The correct estimation of the wrist point in images is one of the tasks of the descriptor which affects to the computation of the ratio of the ellipse. As it has already commented in Chapter 5, the ellipse extracted from hand images are not always orientated in the same direction as



```
Correctly Classified Instances        9040               40.7832 %
Incorrectly Classified Instances     13126               59.2168 %
Kappa statistic                          0.3232
Mean absolute error                      0.159
Root mean squared error                  0.3359
Relative absolute error                 72.7019 %
Root relative squared error            101.559  %
Total Number of Instances            22166

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.511     0.136      0.349     0.511     0.415      0.824    pose001
                 0.614     0.025      0.777     0.614     0.686      0.833    pose002
                 0.351     0.058      0.466     0.351     0.4        0.758    pose003
                 0.124     0.049      0.267     0.124     0.169      0.512    pose004
                 0.286     0.084      0.329     0.286     0.306      0.584    pose005
                 0.468     0.11       0.378     0.468     0.418      0.777    pose007
                 0.701     0.11       0.478     0.701     0.568      0.886    pose202
                 0.209     0.106      0.22      0.209     0.214      0.7      pose205
Weighted Avg.    0.408     0.085      0.408     0.408     0.397      0.734

=== Confusion Matrix ===

    a    b    c    d    e    f    g    h   <-- classified as
 1416    3    0    1   18  439  886    9 |    a = pose001
  221 1694    4  146  228  237  211   20 |    b = pose002
  119  219  972  595  369  265   78  155 |    c = pose003
  107  105  575  344  224  320  257  840 |    d = pose004
   60  123  529  193  794   94   27  952 |    e = pose005
 1011    5    0    3   42 1298  351   63 |    f = pose007
  766    4    0    0   26   16 1943   17 |    g = pose202
  355   27    8    5  715  767  316  579 |    h = pose205
```

```
Correctly Classified Instances        9040               40.7832 %
Incorrectly Classified Instances     13126               59.2168 %
Kappa statistic                          0.3232
Mean absolute error                      0.159
Root mean squared error                  0.3359
Relative absolute error                 72.7019 %
Root relative squared error            101.559  %
Total Number of Instances            22166

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.511     0.136      0.349     0.511     0.415      0.824    pose001
                 0.614     0.025      0.777     0.614     0.686      0.833    pose002
                 0.351     0.058      0.466     0.351     0.4        0.758    pose003
                 0.124     0.049      0.267     0.124     0.169      0.512    pose004
                 0.286     0.084      0.329     0.286     0.306      0.584    pose005
                 0.468     0.11       0.378     0.468     0.418      0.777    pose007
                 0.701     0.11       0.478     0.701     0.568      0.886    pose202
                 0.209     0.106      0.22      0.209     0.214      0.7      pose205
Weighted Avg.    0.408     0.085      0.408     0.408     0.397      0.734

=== Confusion Matrix ===

    a    b    c    d    e    f    g    h   <-- classified as
 1416    3    0    1   18  439  886    9 |    a = pose001
  221 1694    4  146  228  237  211   20 |    b = pose002
  119  219  972  595  369  265   78  155 |    c = pose003
  107  105  575  344  224  320  257  840 |    d = pose004
   60  123  529  193  794   94   27  952 |    e = pose005
 1011    5    0    3   42 1298  351   63 |    f = pose007
  766    4    0    0   26   16 1943   17 |    g = pose202
  355   27    8    5  715  767  316  579 |    h = pose205
```

Dict1	Dict3

Figure 7.1: Confusion Matrix of Tanibata Dictionaries

the palm. So in some case, like in fist images, the Minor axis of the ellipse is the axis generated over the fingers region instead of the Major one.

Now, analysing the generated confusion matrix from the two different dictionaries(see Figure 7.1), Dict1 and Dict3 (see Section 6.2), the previous comments can be checked.

Also, the affine alignment have a big influence in the results obtained from the classification of hand poses and it is necessary to the correct extraction of the number of protrusions for particular images, like it is explained in the next paragraph.

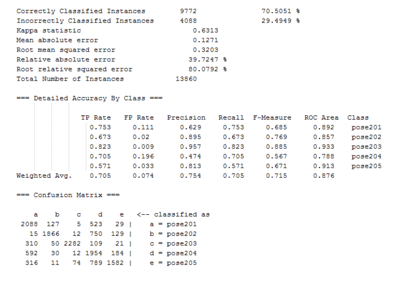
- *Number of protrusions N_p.* This parameter is the most intuitive to classify different hand poses. In fact, if it would be always correctly extracted, gestures from the [1:5] Dictionary will only need this parameter to be described. Nevertheless, the number of peaks or protrusions is not enough when not all the elements of a dictionary are numeric gestures. For example, in the case of [6] Dictionary, poses "Ok_left" and "Ok_rigth" have both one peak in their distance function. In the same way, in [6], the pose "e" also has one number of protrusion. Moreover, because in the computing of the distance function almost all outpoints of the hand are covered, the fist image also presents a peak in their function. Although this peak is not so abrupt like in the case of finger detection, it is also a maximum of the function. Because of this, the relative height of the peaks were analysed in order to reject maximums lower than the peak of a fist image. Therefore, in this kind of image the number of protrusions should be zero.

As well the estimation of the point of the wrist is very important for the later calculation of $\underline{N_p}$. As in the case of the ratio of the ellipse, it is necessary the correctly location of this point of the hand. Since almost all the contour of the hand is covered during the estimation of the distance function, this problem especially affects when the analysed pose is a fist. Hence, if the wrist is erroneously located in the thumb, for example, the relative height of the distance function will be higher than if it were in the right place. So, the number of protrusions here would be one instead of zero.

7.1.2 Roussos Conclusions

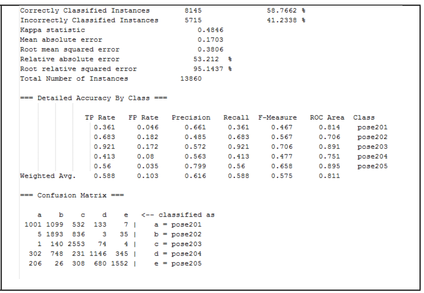
In this descriptor the configurations of the base and the synthetic TRAINING set used in the simulation are very important. Therefore, some parameters used in this descriptor are affected by the number of images defined for the generation of the base (by means of the number of POVs per pose and the relative energy of the eigenhands), as well as the range chosen for synthetic images of the Training set._The number of eigenvectors depends on the outcome of the PCA analysis, which allows to reduce the number of principal components used to describe images. Hence, the higher number of element of the base, N_c,the lower is the difference between the reconstructed image and the original one, meanwhile the run time and complexity of the alignment increase.

- The results obtained for the experiments made with Dict1, generated with base images of 9 and 1 points of view, both of them with the 99% of the energy are very illustrative. Therefore, the 70.501 % obtained for 9 POV in contrast to the 58.766 % for 1 POV, which can be shown in Table 6.2b and 6.2a respectively, demonstrates the argue discussed above: Within a medium number of elements of the base of the same dictionary, the more images used in the base generation, the better results will be obtained in the classification. Therefore, using a higher number of points of view of each pose, 9 POV, generated results are better, almost 71%. These results can be shown in Figure 7.2.



```
Correctly Classified Instances        9772               70.5051 %
Incorrectly Classified Instances      4088               29.4949 %
Kappa statistic                          0.6313
Mean absolute error                      0.1271
Root mean squared error                  0.3203
Relative absolute error                 39.7247 %
Root relative squared error             80.0792 %
Total Number of Instances            13860

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.753     0.111      0.629     0.753     0.685      0.892    pose201
                 0.673     0.02       0.895     0.673     0.769      0.857    pose202
                 0.823     0.009      0.957     0.823     0.885      0.933    pose203
                 0.705     0.196      0.474     0.705     0.567      0.788    pose204
                 0.571     0.033      0.813     0.571     0.671      0.913    pose205
Weighted Avg.    0.705     0.074      0.754     0.705     0.715      0.876

=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
 2088  127    5  523   29 |    a = pose201
   15 1866   12  750  129 |    b = pose202
  310   50 2282  109   21 |    c = pose203
  592   30   12 1954  184 |    d = pose204
  316   11   74  789 1582 |    e = pose205
```

Confusion Matrix with 9 POV



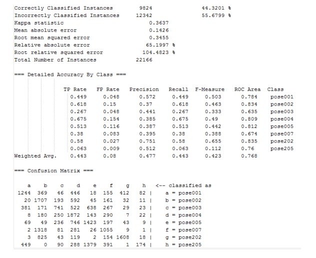
```
Correctly Classified Instances        8145               58.7662 %
Incorrectly Classified Instances      5715               41.2338 %
Kappa statistic                          0.4846
Mean absolute error                      0.1703
Root mean squared error                  0.3806
Relative absolute error                 53.212  %
Root relative squared error             95.1437 %
Total Number of Instances            13860

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.361     0.046      0.661     0.361     0.467      0.814    pose201
                 0.683     0.182      0.485     0.683     0.567      0.706    pose202
                 0.921     0.172      0.572     0.921     0.706      0.891    pose203
                 0.413     0.08       0.563     0.413     0.477      0.751    pose204
                 0.56      0.035      0.799     0.56      0.658      0.895    pose205
Weighted Avg.    0.588     0.103      0.616     0.588     0.575      0.811

=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
 1001 1099  532  133    7 |    a = pose201
    5 1893  836    3   35 |    b = pose202
    1  140 2553   74    4 |    c = pose203
  302  748  231 1146  345 |    d = pose204
  206   26  308  680 1552 |    e = pose205
```

Confusion Matrix with 1 POV

Figure 7.2: Confusion Matrix from Dict1 using a base with the 99% of the energy

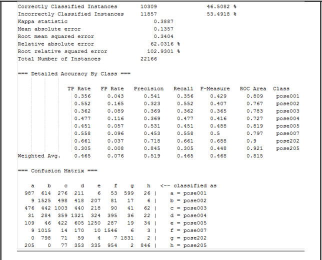
- *The elements of the base.* As it is explained in Chapter 5, the extraction of principal components of the initial set of base images generates a new set of uncorrelated images smaller than the original set, but enough to regenerate with fidelity an image. Furthermore, the average image A_0contains the higher variability of all images from the base set meanwhile the rest of eigenimages have a progressive lower contribution. This characteristic allows images to be described by the main component A_0 besides a reduced number of secondary components. Therefore, the correctly reconstruction of images from their descriptions depends on the set of images used in the generation of the base. This set has to be representative enough to provide a hand base with the minimum number of elements required to describe the input image. So, the more different the images used are , the wider the base will be and consequently, the expressive capacity. As well, the success of the alignment is also very important because it allows describing the same kind of gesture with independence of the scale, orientation and shift without the need of adding more elements into the base.

  For example, the influence of the little finger is lower than the influence of the index finger in [56] Dictionary, where the most of hand poses included have this last finger extended. So it is easy to deduce, that the eigenimage containing this region of the finger will have a weight higher in the most of images projected into the base than the eigenhand related to the little finger region.



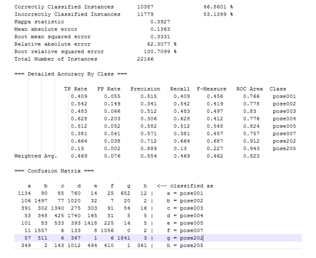
```
Correctly Classified Instances        9824               44.3201 %
Incorrectly Classified Instances     12342               55.6799 %
Kappa statistic                          0.3637
Mean absolute error                      0.1426
Root mean squared error                  0.3455
Relative absolute error                 65.1997 %
Root relative squared error            104.4823 %
Total Number of Instances            22166

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.449     0.048      0.572     0.449     0.503      0.784    pose001
                 0.618     0.15       0.37      0.618     0.463      0.834    pose002
                 0.267     0.048      0.441     0.267     0.333      0.635    pose003
                 0.675     0.154      0.385     0.675     0.49       0.809    pose004
                 0.513     0.116      0.387     0.513     0.442      0.812    pose005
                 0.38      0.083      0.395     0.38      0.388      0.674    pose007
                 0.58      0.027      0.751     0.58      0.655      0.835    pose202
                 0.063     0.009      0.512     0.063     0.112      0.76     pose205
Weighted Avg.    0.443     0.08       0.477     0.443     0.423      0.768

=== Confusion Matrix ===

    a    b    c    d    e    f    g    h   <-- classified as
 1244  369   46  446   18  155  412   82 |    a = pose001
   20 1707  193  592   45  161   32   11 |    b = pose002
  381  171  741  522  638  267   29   23 |    c = pose003
    8  180  250 1872  143  290    7   22 |    d = pose004
   69   49  236  746 1423  197   43    9 |    e = pose005
    2 1318   81  281   26 1055    9    1 |    f = pose007
    3  825   43  119    2  154 1608   18 |    g = pose202
  449    0   90  288 1379  391    1  174 |    h = pose205
```

95% of the energy



```
Correctly Classified Instances       10309               46.5082 %
Incorrectly Classified Instances     11857               53.4918 %
Kappa statistic                          0.3887
Mean absolute error                      0.1357
Root mean squared error                  0.3404
Relative absolute error                 62.0316 %
Root relative squared error            102.9301 %
Total Number of Instances            22166

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.356     0.043      0.541     0.356     0.429      0.809    pose001
                 0.552     0.165      0.323     0.552     0.407      0.767    pose002
                 0.362     0.089      0.369     0.362     0.365      0.783    pose003
                 0.477     0.116      0.369     0.477     0.416      0.727    pose004
                 0.451     0.057      0.531     0.451     0.488      0.819    pose005
                 0.558     0.096      0.453     0.558     0.5        0.797    pose007
                 0.661     0.037      0.718     0.661     0.688      0.9      pose202
                 0.305     0.008      0.845     0.305     0.448      0.921    pose205
Weighted Avg.    0.465     0.076      0.519     0.465     0.468      0.815

=== Confusion Matrix ===

    a    b    c    d    e    f    g    h   <-- classified as
  987  614  276  211    6   53  599   26 |    a = pose001
    9 1525  498  418  207   81   17    6 |    b = pose002
  476  442 1003  440  218   90   41   62 |    c = pose003
   31  284  359 1321  324  395   36   22 |    d = pose004
  109   46  422  605 1250  287   19   34 |    e = pose005
    9 1015   14  170   10 1546    6    3 |    f = pose007
    0  798   71   59    4    7 1831    2 |    g = pose202
  205    0   77  353  335  954    2  846 |    h = pose205
```

99% of the energy



```
Correctly Classified Instances       10387               46.8601 %
Incorrectly Classified Instances     11779               53.1399 %
Kappa statistic                          0.3927
Mean absolute error                      0.1363
Root mean squared error                  0.3331
Relative absolute error                 62.3077 %
Root relative squared error            100.7099 %
Total Number of Instances            22166

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.409     0.055      0.515     0.409     0.456      0.766    pose001
                 0.542     0.149      0.341     0.542     0.419      0.778    pose002
                 0.483     0.066      0.512     0.483     0.497      0.83     pose003
                 0.628     0.203      0.306     0.628     0.412      0.776    pose004
                 0.512     0.052      0.582     0.512     0.545      0.824    pose005
                 0.381     0.041      0.571     0.381     0.457      0.757    pose007
                 0.664     0.038      0.712     0.664     0.687      0.912    pose202
                 0.13      0.002      0.889     0.13      0.227      0.943    pose205
Weighted Avg.    0.469     0.076      0.554     0.469     0.462      0.823

=== Confusion Matrix ===

    a    b    c    d    e    f    g    h   <-- classified as
 1134   90   85  760   14   25  652   12 |    a = pose001
  106 1497   77 1020   32    7   20    2 |    b = pose002
  391  302 1340  275  303   91   54   16 |    c = pose003
   53  348  425 1740  165   31    5    5 |    d = pose004
  101   83  533  393 1418  225   14    5 |    e = pose005
   11 1557    6  133    8 1056    0    2 |    f = pose007
   57  511    6  347    1    6 1841    3 |    g = pose202
  349    2  143 1012  494  410    1  361 |    h = pose205
```

100% of the energy

Figure 7.3: Confusion Matrix from Dict3 using a base with the 1 POV

Moreover, this kind of problem is shown with the confusion matrix of Figure 7.2. There, it can be noticed that in the case of the base with 9 POV the pose 205 is usually confused with the 204 (see Figure 3.7c), due to the problem mentioned in the next bullet about the elements which form the base images. In the same way, results from base with 1 POV shown that the pose 201 is usually confused with the 202 (see Figure 3.7c) due to the resemblance between both images.

- *The amount of energy stored by the eigenhands defined by PCA.* This energy parameter is related to the definition of the number of elements which form the base. Hence, the more energy we want to store with eigenhands, the higher precision in the reconstructed images. This implies that with more energy the expected recognition rates are higher..

  Now, analysing the confusion matrix of the three different sets of stored energies, 95%, 99% and 100%, of the same Dictionary, Dict 3, generated with 1 POV the previous observation can be noticed in Figure7.3. The results obtained are better, 47% for a base with a higher number of elements, this is the ones generated with the 100% of the energy, meanwhile the results obtained from the base with the minimum energy, 95% generated the lowest results, 42.2% of the energy.

Moreover, it can be observed in these matrixes that the most misclassified hand poses are those that do not belong to numeric poses, like the pose 205 which has the worst results from all the classified gestures in all the configurations (see Figure 7.3).

The initial alignment of images (see Section 4.2) contributes in the reduction of the number eigenhands because they are extracted from the covariance matrix, which depends on the variability and resemblance of the images used to the base generation. Furthermore, PCA selects the regions of the hand with the high representation of images used to the generation of the base. Therefore, the same hand gestures with different scale, direction or center would produce multiple and different elements during the base generation, meanwhile aligned images of the same gesture would generate fewer elements to describe hand images.

7.2 Future Work

Some future work lines can be enumerated:

- The testing of more descriptors with different evaluation schemes.
- The inference of the position of each joint of the hand for a given input image.
- The processing of a temporal image sequence, making use of temporal coherence over pose detection or over the original depth information.

Bibliografía

[1] N. Tanibata, N. Shimada, and Y. Shirai, "Extraction of hand features for recognition of sign language words," in *In International Conference on Vision Interface*, pp. 391–398, 2002.

[2] V. P. A. Roussos, S. Theodorakis and P. Maragos, "Affine-invariant modeling of shape-appearance images applied on sign language handshape classification," (Hong Kong), Proc. IEEE Intel Conf. on Image Processing (ICIP-10), 2010.

[3] L. B. Noureddine Aboutabit, Denis Beautemps, "Hand and lip desynchronization analysis in french cued speech: Automatic temporal segmentation of hand flow," *IEEE ICASSP, Toulouse*, pp. 1–4, France, 2006.

[4] X. Yi, S. Qin, and J. Kang, "Generating 3d architectural models based on hand motion and gesture," *Comput. Ind.*, vol. 60, no. 9, pp. 677–685, 2009.

[5] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly, "Vision-based hand pose estimation: A review," *Computer Vision and Image Understanding*, vol. 108, no. 1-2, pp. 52 – 73, 2007. Special Issue on Vision for Human-Computer Interaction.

[6] S. Soutschek, J. Penne, J. Hornegger, and J. Kornhuber, "3-d gesture-based scene navigation in medical imaging applications using time-of-flight cameras," pp. 1–6, 2008.

[7] K. Nickel and R. Stiefelhagen, "Visual recognition of pointing gestures for human-robot interaction," *Image and Vision Computing*, vol. 25, no. 12, pp. 1875–1884, 2007.

[8] "Energy management and building automation system," 1996.

[9] 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*, vol. 20, pp. 1371–1375, 1998.

[10] O. Aran, I. Ari, L. Akarun, B. Sankur, A. Benoit, A. Caplier, P. Campr, A. H. Carrillo, and F.-X. Fanard, "Signtutor: An interactive system for sign language tutoring," *IEEE Multimedia*, vol. 16, pp. 81–93, 2009.

[11] J. K. T. Konstantinos G. Derpanis Richard P. Wildes, "Definition an drecovery of kinematic features for recognition of american sign language movements, imagevision comput.," *York University, Department of Computer Science and Engineering,*, Received 23 August 2006; revised 16 February 2008; Accepted 3 April 2008. Available online 22 April 2008.

[12] M.-H. Yang, N. Ahuja, and M. Tabb, "Extraction of 2d motion trajectories and its application to hand gesture recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 1061–1074, 2002.

[13] J. Alon, V. Athitsos, Q. Yuan, and S. Sclaroff, "A unified framework for gesture recognition and spatiotemporal gesture segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 9, pp. 1685–1699, 2009.

[14] Y. Cui and J. Weng, "Appearance-based hand sign recognition from intensity image sequences," *Computer Vision and Image Understanding*, vol. 78, no. 2, pp. 157 – 176, 2000.

[15] S. C. Ong, S. Ranganath, and Y. Venkatesh, "Understanding gestures with systematic variations in movement dynamics," *Pattern Recognition*, vol. 39, no. 9, pp. 1633 – 1648, 2006.

[16] C. Vogler and D. Metaxas, "A framework for recognizing the simultaneous aspects of american sign language," *Computer Vision and Image Understanding*, vol. 81, no. 3, pp. 358 – 384, 2001.

[17] R. Yang and S. Sarkar, "Coupled grouping and matching for sign and gesture recognition," *Computer Vision and Image Understanding*, vol. 113, no. 6, pp. 663 – 681, 2009.

[18] O. Al-Jarrah and A. Halawani, "Recognition of gestures in arabic sign language using neuro-fuzzy systems," *Artificial Intelligence*, vol. 133, no. 1-2, pp. 117 – 138, 2001.

[19] E.-J. Holden, G. Lee, and R. Owens, "Australian sign language recognition," *Machine Vision and Applications*, vol. 16, pp. 312–320, 2005. 10.1007/s00138-005-0003-1.

[20] W. Kong and S. Ranganath, "Signing exact english (see): Modeling and recognition," *Pattern Recognition*, vol. 41, no. 5, pp. 1638 – 1652, 2008.

[21] W. Gao, G. Fang, D. Zhao, and Y. Chen, "A chinese sign language recognition system based on sofm/srn/hmm," *Pattern Recognition*, vol. 37, no. 12, pp. 2389 – 2402, 2004.

[22] J. F. Lichtenauer, E. A. Hendriks, and M. J. Reinders, "Sign language recognition by combining statistical dtw and independent classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 2040–2046, 2008.

[23] O. Aran, T. Burger, A. Caplier, and L. Akarun, "A belief-based sequential fusion approach for fusing manual signs and non-manual signals," *Pattern Recognition*, vol. 42, no. 5, pp. 812 – 822, 2009.

[24] P. Dreuw, D. Keysers, T. Deselaers, and H. Ney, "Gesture recognition using image comparison methods," in *Gesture in Human-Computer Interaction and Simulation* (S. Gibet, N. Courty, and J.-F. Kamp, eds.), vol. 3881 of *Lecture Notes in Computer Science*, pp. 124–128, Springer Berlin / Heidelberg, 2006.

[25] R. G. S. I. d. C. e. R. a. A. P. P. Infantino, I.; Rizzo, "A framework for sign language sentence recognition by commonsense context," *IEEE Systems, Man, and Cybernetics Society*, vol. 37, pp. 1034 – 1039, 2007.

[26] J. Han, G. Awad, and A. Sutherland, "Modelling and segmenting subunits for sign language recognition based on hand motion analysis," *Pattern Recognition Letters*, vol. 30, no. 6, pp. 623 – 633, 2009.

[27] M. Flasinski and S. Myslinski, "On the use of graph parsing for recognition of isolated hand postures of polish sign language," *Pattern Recognition*, vol. 43, no. 6, pp. 2249–2264, 2010.

[28] Y.-J. Oh, K.-H. Park, and Z. Bien, "Korean manual alphabet (kma) recognition system using usb camera," in *ISITC '07: Proceedings of the 2007 International Symposium on Information Technology Convergence*, (Washington, DC, USA), pp. 148–151, IEEE Computer Society, 2007.

[29] M. I. Carver Mead, *Analog VLSI implementation of neural systems.* Kluwer Academic Publishers, 1989.

[30] H. Popescu, A.C.; Farid, "Exposing digital forgeries in color filter array interpolated images," *IEEE Signal Processing Society*, vol. 53, pp. 3948 – 3959, Oct. 2005.

[31] G. Molenaar, "Sonic gesture," Master's thesis, Universiteit van Amsterdam, October 6, 2010.

[32] P. Breuer, C. Eckes, and S. Muller, "Hand gesture recognition with a novel ir time-of-flight range camera: A pilot study," pp. 247–260, 2007.

[33] O. Hall-Holt and S. Rusinkiewicz, "Stripe boundary codes for real-time structured-light range scanning of moving objects," *Computer Vision, IEEE International Conference on*, vol. 2, p. 359, 2001.

[34] A. Cassinelli, S. Perrin, and M. Ishikawa, "Smart laser-scanner for 3d human-machine interface," in *CHI '05 extended abstracts on Human factors in computing systems*, CHI EA '05, (New York, NY, USA), pp. 1138–1139, ACM, 2005.

[35] N. T. K. Yasuda, T.; Suehiro, "Strategies for collision prevention of a compact powered wheelchair using sokuiki sensor and applying fuzzy theory," in *Robotics and Biomimetics (ROBIO), 2009 IEEE International Conference on*, Dec. 2009.

[36] S.-H. Nam and S.-Y. Oh, "Real-time dynamic visual tracking using psd sensors and extended trapezoidal motion planning," *Applied Intelligence*, vol. 10, pp. 53–70, 1999. 10.1023/A:1008385515068.

[37] M. Van den Bergh and L. Van Gool, “Combining rgb and tof cameras for real-time 3d hand gesture interaction,” pp. 66–72, 2011.

[38] G. A. Poghosyan and H. G. Sarukhanyan, “Decreasing volume of face images database and efficient face detection algorithm,” *International Journal Information Theories and Applications*, vol. 17, Number 1, pp. 62–69, 2010.

[39] b. M. Yining Deng, “ünsupervised segmentation of color-texture regions in images and video,",” *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, vol. vol. 23, no. 8,, pp. pp. 800–810, Aug. 2001.

[40] X. Zhu, J. Yang, and A. Waibel, “Segmenting hands of arbitrary color,” in *FG ’00: Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition 2000*, (Washington, DC, USA), p. 446, IEEE Computer Society, 2000.

[41] J. K. R. Kjeldsen, “"finding skin in color images,",” in *Second IEEE International Conference on Automatic Face and Gesture Recognition (FG ’96),fg,*, pp. pp.312,, 1996".

[42] T. Wenjun, W. Chengdong, Z. Shuying, and J. Li, “Dynamic hand gesture recognition using motion trajectories and key frames,” in *Advanced Computer Control (ICACC), 2010 2nd International Conference on*, vol. 3, pp. 163 –167, 2010.

[43] J. Usabiaga, A. Erol, G. Bebis, R. Boyle, and X. Twombly, “Global hand pose estimation by multiple camera ellipse tracking,” *Machine Vision and Applications*, vol. 21, pp. 1–15, 2009.

[44] Q. Fei, X. Li, T. Wang, X. Zhang, and G. Liu, “Real-time hand gesture recognition system based on q6455 dsp board,” *Intelligent Systems, WRI Global Congress on*, vol. 2, pp. 139–144, 2009.

[45] M. Zobl, M. Geiger, B. Schuller, M. Lang, and G. Rigoll, “A real-time system for hand gesture controlled operation of in-car devices,” in *Multimedia and Expo, 2003. ICME ’03. Proceedings. 2003 International Conference on*, vol. 3, pp. III – 541–4 vol.3, july 2003.

[46] S. Reifinger, F. Wallhoff, M. Ablassmeier, T. Poitschke, and G. Rigoll, "Static and dynamic hand-gesture recognition for augmented reality applications," in *Human-Computer Interaction. HCI Intelligent Multimodal Interaction Environments* (J. Jacko, ed.), vol. 4552 of *Lecture Notes in Computer Science*, pp. 728–737, Springer Berlin / Heidelberg, 2007.

[47] S. Tamura and S. Kawasaki, "Recognition of sign language motion images," *Pattern Recognition*, vol. 21, no. 4, pp. 343 – 353, 1988.

[48] D. Gavrila and L. Davis, "Towards 3-d model-based tracking and recognition of human movement: a multi-view approach.," in *Workshop on Face and Gesture Recognition*, 1995.

[49] K. A.-R. M. D. o. C. S. A. U. o. S. Shanableh, T.; Assaleh, "Spatio-temporal feature-extraction techniques for isolated gesture recognition in arabic sign language," *IEEE Systems, Man, and Cybernetics Society*, vol. 37, pp. 641 – 650, June 2007.

[50] R. Bowden, D. Windridge, T. Kadir, A. Zisserman, and M. Brady, "A linguistic feature vector for the visual interpretation of sign language," in *Computer Vision - ECCV 2004* (T. Pajdla and J. Matas, eds.), vol. 3021 of *Lecture Notes in Computer Science*, pp. 390–401, Springer Berlin / Heidelberg, 2004.

[51] F. Kuhl and C. Giardina, "Elliptic fourier features of a closed contour," *Computer Graphics and Image Processing*, vol. 18, pp. 236–258, 1982.

[52] L. Bretzner, I. Laptev, and T. Lindeberg, "Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering," in *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pp. 405–410, 2002.

[53] Y. T. Chen and K. T. Tseng, "Developing a multiple-angle hand gesture recognition system for human machine interactions," in *Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE*, pp. 489–492, 2007.

[54] V. Athitsos and S. Sclaroff, "Estimating 3d hand pose from a cluttered image," *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 2, p. 432, 2003.

[55] D. Castilla, I. Miralles, M. Jorquera, C. Botella, R. Baños, J. Montesa, and C. Ferran, “Analysis and testing of metaphors for the definition of a gestual language based on real users interaction: vision project,” in *13th International Conference on Human-Computer Interaction*, (San Diego, CA, USA), 2009.

[56] J. Molina, M. Escudero-Vi?olo, A. Signoriello, M. Pard?s, C. Ferrán, J. Bescós, F. Marqués, and J. Martínez, “Real-time user independent hand gesture recognition from time-of-flight camera video using static and dynamic models,” *Machine Vision and Applications*, pp. 1–18, 2011. 10.1007/s00138-011-0364-6.

[57] 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*, vol. 5, no. 3/4, pp. 334–343, 2008.

[58] D.-Y. Huang, W.-C. Hu, and S.-H. Chang, “Gabor filter-based hand-pose angle estimation for hand gesture recognition under varying illumination,” *Expert Syst. Appl.*, vol. 38, pp. 6031–6042, May 2011.

[59] I. T. Jolliffe, *Principal Component Analysis.* Springer, second ed., oct 2002.

Parte I

Presupuesto

1) Ejecución Material: 1650 €

Desglose por Conceptos:

- Compra de ordenador personal (Software incluido)....... 1.500 €
- Material de oficina....... 150 €

2) Gastos generales

- 16 % sobre Ejecución Material 264 €

3) Beneficio Industrial

- 6 % sobre Ejecución Material 99 €

4) Honorarios Proyecto

- 1500 horas a 18 € / hora 27000 €

5) Material fungible: 280 €

Desglose por Conceptos:

- Gastos de impresión 80 €
- Encuadernación 200 €

6) Presupuesto antes de Impuestos

- Subtotal Presupuesto 29293 €

7) I.V.A. aplicable

- 18 % Subtotal Presupuesto 5273 €

8) Total presupuesto • Total Presupuesto 34566 €

Madrid, Diciembre de 2011

El Ingeniero Jefe de Proyecto

Fdo.: Laura de las Heras Muñoz

Parte II

Pliego de condiciones

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de un sistema de reconocimiento de gestos manuales. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego. Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

Condiciones generales

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.

2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.

3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que suponvo en los casos de rescisión.

8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.

9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.

10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no,

se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.

11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.

12. Las cantidades calculadas para obras accesorias, aunque figuren por partida alzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.

13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.

14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.

15. La garantía definitiva será del 4 % del presupuesto y la provisional del 2 %.

16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.

17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.

18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.

19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.

20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.

21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.

22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.

23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad “Presupuesto de Ejecución de Contrata” y anteriormente llamado ”Presupuesto de Ejecución Material” que hoy designa otro concepto.

Condiciones particulares

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.

2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.

3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.

4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.

5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.

6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.

7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.

8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.

9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.

10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.

11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha

aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.

12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.

e este precio en relación con un importe límite si este se hubiera fijado.

4. La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.

5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.

6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.

7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.

8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.

9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo

a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.

10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.

11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.

12. Las cantidades calculadas para obras accesorias, aunque figuren por partida alzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.

13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.

14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.

15. La garantía definitiva será del 4 % del presupuesto y la provisional del 2 %.

16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.

17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la

provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.

18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.

19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.

20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.

21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.

22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.

23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad “Presupuesto de Ejecución de Contrata” y anteriormente llamado ”Presupuesto de Ejecución Material” que hoy designa

otro concepto.

Condiciones particulares

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.

2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.

3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.

4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.

5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.

6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.

7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.

8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.

9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.

10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.

11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.

12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.