Hand Detection and Tracking Using the Skeleton of the Blob for Medical Rehabilitation Applications

Pedro Gil-Jiménez[1], Beatriz Losilla-López[1], Rafael Torres-Cueco[2], Aurélio Campilho[3], and Roberto López-Sastre[1]

[1] Universidad de Alcalá, Alcalá de Henares, Madrid, Spain
pedro.gil@uah.es
http://agamenon.tsc.uah.es/Investigacion/gram/
[2] Universidad de Valencia, Valencia, Spain
[3] INEB - Instituto de Engenharia Biomédica, Portugal
Universidade do Porto, Faculdade de Engenharia, Portugal

Abstract. This article presents an image processing application for hand detection and tracking using the *4-connected* skeleton of the segmentation mask. The system has been designed to be used with techniques of virtual reality to develop an interactive application for phantom limb pain reduction in therapeutic treatments.

One of the major contributions is the design of a fast and accurate skeleton extractor, that has proven to be faster than those available in the literature. The skeleton allows the system to precisely detect the position of all the interest points of the hand (namely the fingers and the hand center).

The system, composed of both the hand detector and tracker, and the virtual reality application, can work in real-time, allowing the patient to watch the virtual image of his own hand on a screen.

Keywords: hand detection and tracking, blob skeleton, virtual reality, phantom limb pain reduction.

1 Introduction

In the management of patients with *Complex Regional Pain Syndrome* (CRPS) or *Phantom Limb Pain* (PLP), there is compelling evidence that *Mirror Visual Feedback* (MVF) is effective in reducing pain and disability [13] [14]. The use of virtual reality in medical applications has been given a great interest in recent years. One of these approaches is the *Virtual Reality Mirror Visual Feedback* (VRMVF), with the same principles of the MVF, but, instead of using a mirror, it uses a different equipment, sometimes quite complex, like sensors inserted in a glove [3] [15]. This equipment can not be used in many cases, since it implies that the patient has to wear a glove (or a similar type of sensor) which some patients can not tolerate. Furthermore, in this kind of rehabilitation techniques,

A. Campilho and M. Kamel (Eds.): ICIAR 2012, Part II, LNCS 7325, pp. 130–137, 2012.

for an application to be useful, it needs to offer some kind of entertainment or amusement for the patient to get his compliance.

In our approach, we allow the patient to be bare hand. This is shown in Fig. 1(a). The setup consists of a camera inside a structure. This structure is intended to prevent the user to see his own hand, as proposed by the experts. A mirror placed in the top of the structure allows us to increase the optical field of view without the use of complex lenses for the camera. At the front of the structure, a hole allows the patient to introduce his hand. Finally, a screen placed at the other side of the structure displays the virtual images, that can be watched by the patient, but also by the doctor if needed.

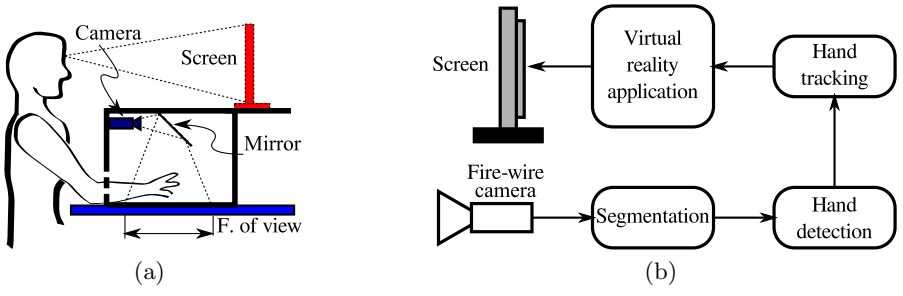


Fig. 1. (a): Graphical schema of the setup (b): Flow diagram

In Fig. 1(b) we can see the flow diagram of the system. Once the video has been captured, a segmentation block extracts the mask corresponding to the patient's hand. As we will see later, the system setup described above simplifies the segmentation algorithm to a simple 2^{18} bytes Look-Up Table (LUT). A careful build of the segmentation LUT, along with a proper design of the setup can render a clear mask of the image, plus isolated noisy pixels that can be removed with a morphological close operation. Afterwards, the *4-connected* skeleton of the current blob is extracted. The last step of the algorithm is devoted to localize and track the position of the fingers and the center of the hand.

As it can be deduced, the goal of the algorithm described is the localization of the interest points of the hand. This information comprises the data needed by the virtual reality block, in this case, the position of the fingers and the hand to interact with the virtual objects used in the application. In section 6 we will see see the applications designed to help the patient exercise his non affected limb.

2 State of the Art

There are many works that have been proposed in the literature to detect and track the hand, which have been developed either with the use of glove sensors or artificial vision techniques. These techniques include color segmentation [7] [16] [8], contour detection [9] [20] and infrared segmentation [1], among others.

The hand gesture is normally extracted by recognition algorithms, for example, searching the fingertips [7], estimating the optical flow [16] or detecting the valleys between the fingers and the fingertips [20].

In the studies mentioned above, the input images have been obtained by a camera (web, fire-wire, etc.) and the recognition will be on real-time [12] [7] [19] or not [9][17]. Some authors have improved the system with the use of the Microsoft Kinect sensor, which has a depth sensor and a RGB camera integrated in one device [17] [19][18]. Although Kinect has its own library (OpenNI) to recognize the individual's skeleton position, it does not focus on the skeleton of the hand.

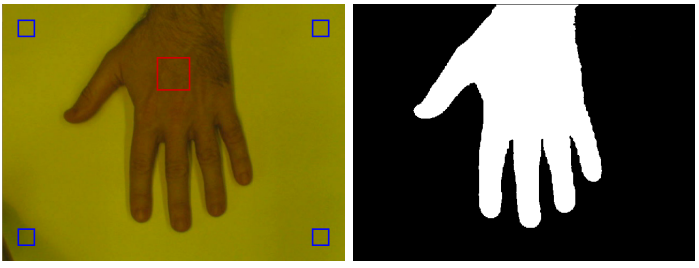
3 Setup and Segmentation

As we saw in Fig. 1(a), the camera, and its field of view, are inside a closed box. Furthermore, the bottom of the box can be chosen to have a color or texture as different as possible as the color of the hand. Typically, some kind of colored sheets that are frequently used by physiotherapists can be used. Nevertheless, the color or texture of the sheet can be interchanged to increase the contrast between the hand and the background. For instance, a patient wearing a glove would require a different sheet than one bare hand.

A careful design of the setup will simplify enormously the segmentation process. In many image processing systems, segmentation is the first and more critical step. A poor segmentation will make the whole system fail. On the other hand, a proper segmentation mask would make the design of the rest of the system easier. Many algorithms have been designed for hand, or more generally, for skin segmentation [4]. However, those algorithms have one or two drawbacks. On the one hand, many algorithms look for some predefined palette of colors, making it quite difficult for the user to fix it when the skin color of the patient, or just the proper parameters of the camera, make the segmentation fail. On the other hand, some works implement a complex algorithm which makes it impossible for the system to cope with the real-time requirements.

In this work, we perform the segmentation in a pixel wise fashion, with the use of a *Look-up Table*, [5], built with a set of *Support Vector Machines* (SVMs) [2]. The SVMs are trained with a set of pixels values obtained from a given image of the patient's hand. The system works as follows.

The patient is asked to put his hand inside the setup described in Fig. 1(a), in such a way that the image of his hand occupies a central square, and does not occlude a set of four squares at the corners of the image. Fig. 2 shows the process. During this step, the patient and the doctor can see the image of the hand, including the five squares superimposed in the image. The red one must be occupied completely by the hand, while the other four blue ones must only be on the background. In this situation, the doctor only needs to press a key and the process to build the LUT is started.



(a) original image (b) segmentation mask

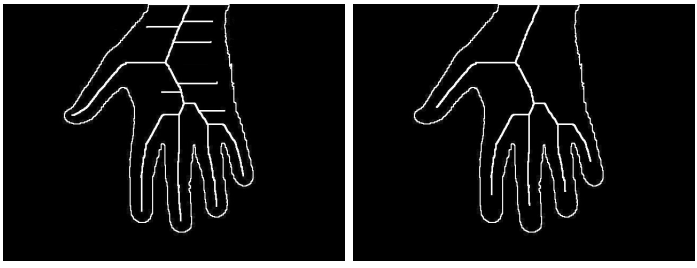
Fig. 2. Automatic procedure to set the segmentation block. The patient is asked to put his hand (a), so that his hand occupies the central square, and does not occupy the exterior ones. The doctor can see the segmentation mask (b) in real time.

To build the LUT, the system collects all the RGB values of the pixels inside the five squares in Fig. 2. The blue squares are 30×30 pixels size each, while the red one is 60×60 pixels. Then, a two-class SVM is trained with two set of vectors. The first one, belonging to the class *hand*, is composed of all the RGB values of the $60 \times 60 = 3600$ pixels inside the central square, while the second, named *background*, is obtained from the $4 \times (30 \times 30) = 3600$ pixels of the other four squares. After the SMV has been trained, the LUT can be built. To reduce the size of the LUT, only the 6 MSB (*More Significant Bits*) are used, which yields a LUT of $2^{6\times3}$ elements, that is, some 262 kbytes are needed.

4 Skeleton Computation

Skeletons are important shape descriptors in object representation and recognition. They capture essential topology and shape information of the object in a simple form [6]. Although the skeleton is a widely known tool to describe binary images, there is not a publicly available implementation which could be useful for the purpose of the project [11]. Many of the implementation tested fail in one or two drawbacks. Many algorithms render, apart from the correct skeleton, some branches due to noise, that would make it impossible for the next block to find the correct position of the hand center and the fingers. This effect is illustrated in Fig. 3(a) which depicts the skeleton computed with the algorithm described in [6]. As we can see, there are some branches, normally running from the contour of the object in horizontal or vertical direction, that do not correspond to any real segment of the object. These false branches are generated by protrusions in the contour of the blob, that are caused by segmentation noise or the shape of the segmented object.

The other problem inherent to some of the algorithms described in the literature is that they do not ensure a connected set of pixels for the whole skeleton, or that the skeleton is not one pixel width throughout the whole skeleton.



(a) (b)

Fig. 3. Skeleton computed with (a) the algorithm described in [6] and (b) proposed in this paper. In both cases, the skeleton has been morphologically dilated, and the contour of the blob superimposed for illustration purposes.

In this work, we have implemented an algorithm that always overcomes the drawbacks described above. Furthermore, in contrast to the algorithms proposed for instance in [6], we define the skeleton as a set of 4-connected pixels. Fig. 4 shows the difference between the 4- and 8-connected skeleton of the same object. One of the advantages of a *4-connected skeleton*, in contrast to an *8-connected* version, is that the algorithm is faster, since only half of the comparisons need to be done. That is, for the 8-connected skeleton, for a pixel belonging to the object in a given iteration, its 8 neighbors must be checked. In the 4-connected version, only the horizontal and vertical neighbors are needed. Our experiments showed an improvement of 30% reduction in the execution time of our algorithm compared to the implementation of the one described in [6]. The other advantage is that a 4-connected skeleton can be analyzed easier. As we will see in the next section, the skeleton must be scanned looking for the interest points of the hand, and the 4-connected skeleton has been proven to be more suitable for this task.

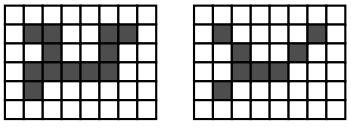


Fig. 4. Comparison between 8- and 4-connected skeletons

5 Point Finding and Tracking

The last step consists in finding the interest points of the hand from the skeleton previously computed. Two kind of points must be found. On the one hand, those points where three (or four) branches meet are called *cross points*. In a 4-connected skeleton, only five patterns exist, as shown in Fig. 5, so that the process reduces to a search of these configurations.

The other kind of points, called *end points*, corresponds to the end of the branches. These points should correspond to the fingertips. The algorithm must scan the skeleton from each cross point to the end of each branch, which can end either in another cross point, or in an end point. The length, orientation and position of the branches are used to discard false branches from the real ones. Fig. 6 shows some results of the algorithm[1].

Finally, using the information collected from previous images, the algorithm tracks the position of each interest point, specially the fingertips. This tracking is intended to improve the behavior of the system in the presence of segmentation noise, for instance, vibration of the interest points from frame to frame or segmentation occlusions. The algorithm could be used to handle real occlusions, for instance when one finger moves beneath another, although these problems do not need to be handle in this project.

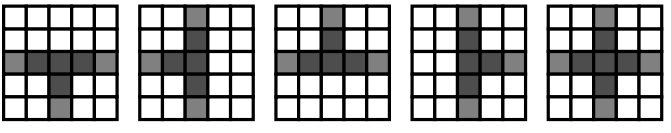


Fig. 5. Patterns indicating a cross point in the skeleton. The central pixel is the *cross point* of the skeleton.

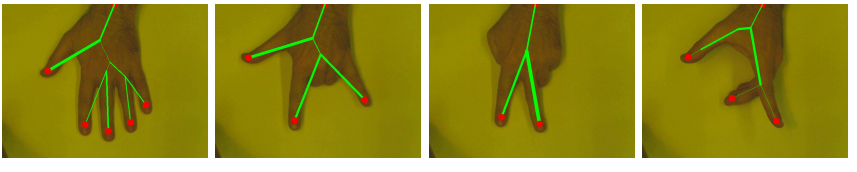


Fig. 6. Some illustrations of the performance of the hand detection algorithm

6 Virtual Reality Application

The system described has been used to develop a virtual reality application that can be used in the rehabilitation process of patients with a phantom limb. In this case, the system includes several games that can be played by the patient. Some videos showing the performance of the system can be downloaded from the web page.

The system has been implemented using the OpenCV library, version 2.3. For the segmentation block, we used the SVMs implementation included in OpenCV. The skeleton and the interest point finding and tracking were implemented in C++. The input of the system is a fire-wire camera, with 640×480 pixels color images. The system works at 15 frames per second over a Pentium 4 2.4 GHz on a Linux kernel 2.6.32-25.

[1] The complete video, and some more examples can be found at `http://agamenon.tsc.uah.es/Investigacion/gram/papers/ICIAR12`

7 Conclusion and Future Work

In this paper, we have presented an algorithm for hand and finger detection and tracking, that allows the development of a virtual reality application to be used in rehabilitation processes, specifically, to reduce the pain in patients with a phantom limb. The main contributions are the development of a skeleton extractor algorithm, that overcomes the achievement of the ones described in the literature, and an algorithm which locates the interest points of the hand from the skeleton. This algorithm has been used to develop a virtual reality application, which includes a set of games, that can be played by the patient within its rehabilitation process.

Future work includes several directions. The closest one is the development of several more applications or games. This must be done working together with doctors and patients, since each patient could require a different set of activities for his personal rehabilitation process.

The system is also open to depth images fusion. In our case, we intend to use a Microsoft Kinect sensor mainly to get the information about the height of each finger with respect to the table. This will allow the system to detect when the patient is pressing the background. This would widen the set of applications that can be developed.

Although the main motivation of our work was the development of a virtual reality application, the applications of the system can be extended to other kind of projects, such as Sign Language Recognition [10,18].

Acknowledgements. This work was supported by the Spanish Ministry for Science and Innovation project number TIN2010-20845-C03-03, project UAH2011/EXP-030 from the University of Alcalá and the INNPACTO Program N. exp. IPT-2011-1366-390000.

References

1. Breuer, P., Eckes, C., Müller, S.: Hand Gesture Recognition with a Novel IR Time-of-Flight Range Camera–A Pilot Study. In: Gagalowicz, A., Philips, W. (eds.) MIRAGE 2007. LNCS, vol. 4418, pp. 247–260. Springer, Heidelberg (2007)
2. Chang, C., Lin, C.: LIBSVM: a library for support vector machines (2001), software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
3. Cole, J., Crowle, S., Austwick, G., Slater, D.H.: Exploratory findings with virtual reality for phantom limb pain; from stump motion to agency and analgesia. Disability and Rehabilitation 31(10), 846–854 (2009)
4. Elgammal, A., Muang, C., Hu, D.: Skin detection - a short tutorial. In: Encyclopedia of Biometrics (2009)
5. Gómez-Moreno, H., Maldonado-Bascón, S., Gil-Jiménez, P., Lafuente-Arroyo, S.: Goal evaluation of segmentation algorithms for traffic sign recognition, vol. 11(4), pp. 917–930 (July 2010)
6. González, R., Woods, R.: Digital Image Processing. Addison-Wesley (1993)
7. von Hardenberg, C., Bérard, F.: Bare-hand human-computer interaction. In: Proceedings of the 2001 Workshop on Perceptive user Interfaces, PUI 2001, pp. 1–8 (2001)

8. Hsieh, C.C., Liou, D.H., Lee, D.: A real time hand gesture recognition system using motion history image. In: 2010 2nd International Conference on Signal Processing Systems (ICSPS), vol. 2, pp. V2-394–V2-398 (July 2010)
9. Imai, A., Shimada, N., Shirai, Y.: 3-d hand posture recognition by training contour variation. In: Proceedings of Sixth IEEE International Conference on Automatic Face and Gesture Recognition, pp. 895–900 (May 2004)
10. Isaacs, J., Foo, J.: Hand pose estimation for american sign language recognition. In: Southern Symposium on System Theory, pp. 132–136 (2004)
11. Lakshmi, J.K., Punithavalli, M.: A survey on skeletons in digital image processing. Digital Image Processing, 260 – 269 (2009)
12. Letessier, J., Bérard, F.: Visual tracking of bare fingers for interactive surfaces. In: Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology, UIST 2004, pp. 119–122. ACM (2004)
13. Lewis, J.S., Kersten, P., McCabe, C.S., McPherson, K.M., Blake, D.R.: Body perception disturbance: A contribution to pain in complex regional pain syndrome. In: CRPS (2007)
14. McCabe, C.S., Haigh, R.C., Ring, E.F., Halligan, P.W., Wall, P.D., Blake, D.R.: A controlled pilot study of the utility of mirror visual feedback in the treatment of complex regional pain syndrome (type 1). Rheumatology 42(1), 97–101 (2003)
15. Murray, C.D., Patchick, E., Pettifer, S., Howard, T., Caillette, F., Kulkarni, J., Bamford, C.: Investigating the efficacy of a virtual mirror box in treating phantom limb pain in a sample of chronic sufferers. Disabil Human Dev. 5(3), 227–234 (2006)
16. Nope, R., Sandra, E., Humberto Loaiza, C., Eduardo Caicedo, B.: Estudio omparativo de técnicas para el reconocimiento de gestos por visión artificial. Avances en Sistemas e Informtica 5(3) (2009)
17. Oikonomidis, I., Nikolaos, K., Argyros, A.A.: Efficient model-based 3d tracking of hand articulations using kinect. In: Tracking Hand Articulations using Kinect (2011)
18. Pugeault, N., Bowden, R.: Spelling it out: Real-time ASL fingerspelling recognition. In: IEEE Workshop on Consumer Depth Cameras for Computer Vision (2011)
19. Ren, Z., Meng, J., Yuan, J., Zhang, Z.: Robust hand gesture recognition with kinect sensor. In: Proceedings of the 19th ACM International Conference on Multimedia, MM 2011, pp. 759–760. ACM (2011)
20. Yoruk, E., Konukoglu, E., Sankur, B., Darbon, J.: Shape-based hand recognition. IEEE Transactions on Image Processing 15(7), 1803–1815 (2006)