Region-Based *vs.* Edge-Based Registration for 3D Motion Capture by Real Time Monoscopic Vision

David Antonio Gómez Jáuregui and Patrick Horain

INRIA / Projet MIRAGES
B.P. 105, 78153 Le Chesnay Cedex, France
Institut TELECOM ; TELECOM & Management SudParis
9 rue Charles Fourier, 91011 Evry Cedex, France
{David.Gomez,Patrick.Horain}@IT-SudParis.eu

Abstract. 3D human motion capture by real-time monocular vision without using markers can be achieved by registering a 3D articulated model on a video. Registration consists in iteratively optimizing the match between primitives extracted from the model and the images with respect to the model position and joint angles. We extend a previous color-based registration algorithm with a more precise edge-based registration step. We present an experimental analysis of the residual error *vs.* the computation time and we discuss the balance between both approaches.

Keywords: 3D motion capture, monocular vision, 3D / 2D registration, region matching, edges matching.

1 Introduction

Research in motion capture by computer vision has been motivated by many target applications: human-computer interfaces, animation, interaction with virtual environments, video surveillance, games, etc. We focus on 3D human motion capture in real-time without markers [8]. This is a difficult problem because of the ambiguities resulting of the lack of depth information, partial occlusion of human body parts, high number of degrees of freedom, variations in the proportions of the human body and different clothing of each person [14].

In this work, we extend a previous work for 3D human motion capture by registering a 3D articulated human body model on video sequences using a color-based step followed by an edge-based step [7]. In this work we shall experimentally characterize the contribution of color and edge information to model matching. We present a detailed analysis of the precision and processing time achieved by the color-based registration and the edge-based registration steps.

This paper is organized as follows. First, in section 2, we describe previous works related to 3D human motion capture by computer vision. In section 3, we introduce our 2 steps approach based on matching color regions and the edges. Then, our performance characterization experiments and the results obtained are presented in section 4. Finally, in section 5, we conclude and discuss how a balance can be found between both steps while facing limited computation resource.

A. Gagalowicz and W. Philips (Eds.): MIRAGE 2009, LNCS 5496, pp. 344–355, 2009.

2 Previous Work for 3D Human Motion Capture

Previous works rely on various appearance features such as color [10], [6], [7], edge [6], [7], [15], shape [1], [11], and motion [15], [9]. They can be divided into two main approaches: model-based and model-free approaches [14].

The model-based approaches use a 3D model of a human body and a matching cost function to find the 3D pose that best matches input images. Estimating the 3D pose from monocular images, is achieved by searching for the pose that minimizes some matching cost function [8]; some other works use human body part detectors to assemble the 3D pose using physical and proximity constraints [3]. Temporal coherence can be enforced with particle filters that allow multiple hypotheses matching [6], [15]. Some works use motion priors [16] to guide tracking within a motion model previously learned [16], or to learn a mapping between the pose space and a low-dimensional latent space in which the tracking occurs [13].

Model-free approaches do not use any 3D explicit human body model. Instead, they try to infer directly 3D poses from images. The learning-based approaches rely on training data to learn a function that maps the image observation to the 3D pose space [1]. Example-based approaches avoid this learning by saving in a database a collection of examples of 3D poses with their corresponding image descriptors and by searching this database and interpolating candidate poses for a 3D pose similar to the input image [11].

3 Our Approach for 3D Human Motion Capture

Our method consists in registering a 3D articulated model of upper human body on video sequences [8], [10]. Our 3D human model (figure 2a) has 3 global position parameters and 20 joint angles of the upper-body part (chest, arms, forearms, hands, neck and head). A 3D human pose is represented by a vector of parameters of the joint angles.

For each captured image frame, we extract color regions and edges. A colored silhouette and occluding edges are also computed for each candidate pose of the model. Our registration process consists in searching for the best matching correspondence between these primitives. We iteratively optimize registration using a color region-based criterion and then an edge-based registration (fig. 1) [7].

3.1 3D Human Model Calibration and Pose Initialization

To registering the 3D upper-body human model on a video sequence requires calibrating the body model to make it similar to the actor captured in the video. This is done by adjusting manually the 20 joint angles of the 3D model taking as reference the pose of the human in the first image and the dimensions (length, width and height) of each body part of the 3D model by watching the overlapping between each body part of the projected model and the human in the captured image (figure 2b).

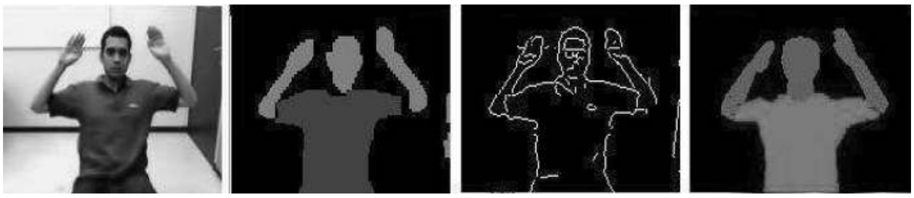

Fig. 1. Our prototype for 3D human motion capture. The images are respectively: the captured image, the segmented image, the edges in the foreground and finally the projection of the registered 3D human body model.

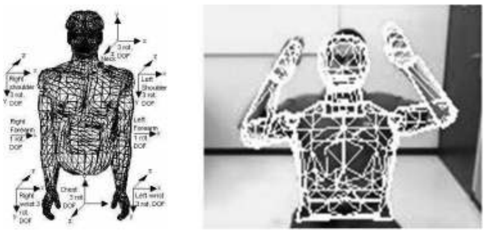

Fig. 2. 3D human model calibration step: a general design of our 3D human upper-body model with the 20 degrees (a), the 3D model projection superposed with the human in order to adjust the parameters of the model in the calibration and pose initialization steps (b)

3.2 Region-Based Registration

The human silhouette is detected by comparing the captured image with a reference image of the background. This silhouette (foreground) is segmented in two color classes (skin and clothes color). Color samples are extracted automatically from the first captured image. A skin color sample is taken in the face region found with Adaboost face detector [17]. A clothes sample is taken under the face. We model each sample using a simple gaussian model in a HSV color space. For each image, we project the 3D model [18] (using OpenGL API) according to the pose described in the vector of parameters. The 3D model is projected by rendering the skin and clothes colors. The matching between the 3D model projection and the segmented image is evaluated using a non-overlapping ratio:

$$F(q) = \prod_{c=1}^{m} \left(\frac{|A_c \cup B_c(q)| - |A_c \cap B_c(q)|}{|A_c \cup B_c(q)|} \right)^{\frac{1}{m}} . \tag{1}$$

where q is the vector of parameters describing a candidate 3D pose, m is the number of color classes, A_c is the set of pixels with the c color class in the segmented image, $B_c(q)$ is the set of pixels with the c color in the projection of the 3D model and $|X|$ represent the number of pixels in X.

This cost function is minimized using a downhill simplex algorithm [12] under biomechanical constraints. Further details can be found in [10].

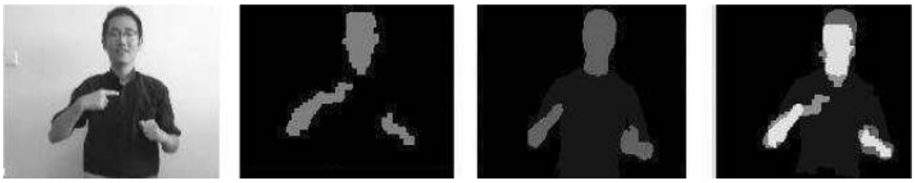

Fig. 3. Limited precision of the region-based registration. The images are respectively: the captured image, the segmented image, the projection of the registered 3D human body model and finally the projection of the 3D model superposed with the segmented image. The pose of the 3D model differs from the pose of the actor because the region-based registration is not precise.

It is important to note that for convergence towards a position approximately correct, this method initially requires only a partial overlapping between colored regions. However, it is not precise because the number of pixels in the border regions is few compared to the number of pixels inside the region (fig. 3).

3.3 Edge-Based Registration

We propose a further edge-based registration step to improve the precision. It works by matching edges of the captured image and occluding edges of the 3D model [9], [15]. The initial 3D pose of this step is the final state output by the region-based registration. Edges in the input video image are extracted with a Deriche filter [4] in the foreground region of the image. Then, a chamfer algorithm [2] allows computing a map of the distance between each pixel of the image and the nearest edge.

The occluding edges of the 3D model are the lines of the surface where the observation direction is tangent to the surface [5]. These occluding edges can easily and efficiently be extracted with the OpenGL API by rendering the surface mesh with culling based on the normal orientation. First, the back facing triangles and their edges are rendered with some foreground color while the front facing triangles are eliminated, then the inside of front facing triangles is rendered with background color while the back facing triangles are eliminated, so only the occluding edges remain highlighted in the image. Then the mean distance between the projected occluding edges of the model and the edges in the input video image is computed by masking the previous distance map with the projected binary image of the 3D model occluding edges:

$$D_c = \frac{1}{N_p} \sum_i I_{DT}(p_i) \ . \tag{2}$$

where D_c is the mean edge distance, I_{DT} is the distance transform image, p_i are the pixels in the projected occluding edges of the 3D model. We minimize this function with the downhill simplex algorithm [12] as previously in the region-based registration step.

Our registration process basically consists in using downhill simplex optimization algorithm [12] to minimize the non-overlapping ratio and the mean edge distance. The downhill simplex method has the advantage that it requires computing only values of the function to be optimized rather than its derivates. We use a measure of the size of the simplex (defined by the ratio between the highest and lowest value of the simplex) as a convergence criterion.

4 The Optimization Process

Iterative optimization in a high dimensional space usually requires a large variable number of iterations to converge. Because we are interested in real-time motion capture, we have to limit the computation time and thus, the number of iterations per image. Unfortunately, limiting the number of iterations decreases the precision of the registration process (fig. 4). For this reason, we experimentally analyzed the performance (precision and process time) of our registration process by varying the number of iterations of both registration step (region-based and edge-based), searching for an optimal balance between the precision and processing time in both registration steps (region-based and edge-based).

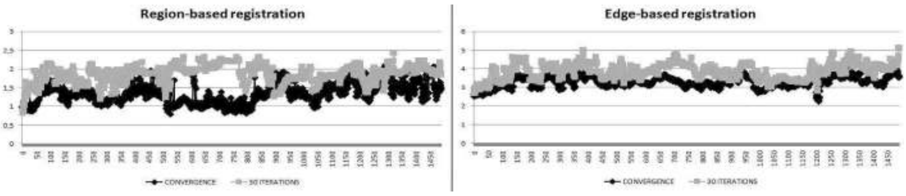

Fig. 4. Effect of a limited number of iterations on the residual error (ordinates). The abscissa is the image number in the video sequence. The left chart corresponds to the non-overlapping ratio minimized by region-based registration. The right chart corresponds to the mean edge distance minimized by edge-based registration. Black lines are the residual error at convergence while the gray line is limited to 30 iterations. A limited number of iterations decrease both the precision and robustness of registration.

We aim at real time tracking, so the available computation time for each captured image must be shared between the two steps of the registration process.

We used 6 video sequences showing various gestures with occlusions (e.g. arms crossed), fast movements, including in the depth direction (fig.5) and a person possibly not exactly facing the camera. These 160 x 120 pixels video sequences were captured using a Logitech QuickCam Pro 5000 webcam. The computation time varies with the number of iterations, the central processor (CPU) and the graphics card (GPU). Table 1 shows the computation time on two platforms[1] with varying number of iterations shared in our two-steps registration process.

[1] Experiments were run on a CPU Intel Pentium 4 3.6 GHz and a GPU NVIDIA Quadro FX 1400 (platform 1) and a CPU Intel Pentium M 1.4 GHz and a GPU NVIDIA GeForce 4200 Go (platform 2).

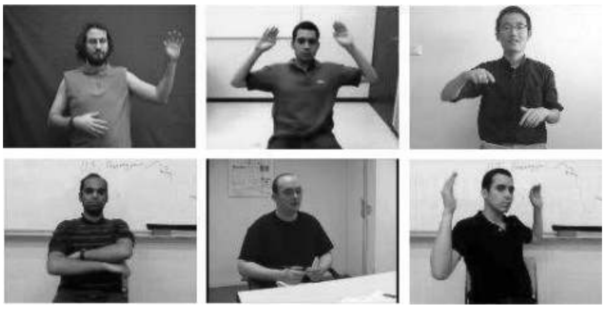

Fig. 5. The video sequences used in our experiments. The video sequence 1 (top left), the video sequence 2 (top center), the video sequence 3 (top right), the video sequence 4 (bottom left), the video sequence 5 (bottom center) and the video sequence 6 (bottom right) contains respectively 290 frames, 1497 frames, 1433 frames, 887 frames, 1032 frames and 551 frames. The first three sequences include various types of gesture. The sequence 4 includes principally gestures when arms are crossing each other. In the sequence 5, the person is not facing directly the camera. The sequence 6 includes movements in which the person is turning around himself.

Table 1. Computation time in milliseconds (average and standard deviation) with respect to the number of iterations shared in our two-step registration process on two platforms. In these experiments, 50% of the total number of iterations is given to each step.

Number of iterations	Avg. Time Platform 1 (ms)	Std. Dev. Time Platform 2 (ms)	Avg. Time Platform 2 (ms)	Std. Dev. Time Platform 2 (ms)
40	22.34	±6.58	88.18	±9.16
100	35.61	±6.68	137.24	±16.07
200	57.99	±6.84	223.30	±27.51
300	78.69	±6.90	300.13	±32.74
400	97.25	±7.12	370.06	±35.81
500	100.59	±9.80	436.21	±48.72

Table 2 shows the computation time of our 3D motion capture prototype[2] for images with higher definition. From these experiments, we found that the performance of the registration process with respect to the non overlapping ratio and the mean edge distance is similar for images with larger number of pixels (higher resolutions) since our approach does not require much accuracy in the segmentation and edge extraction.

4.1 Edge-Based Registration Precision Experiments

We recall that the region-based registration is used to initialize the registration process because is more robust, then, the edge-based registration is used to

[2] Computation times on a CPU Intel Pentium 4 3.0 GHz and a GPU NVIDIA GeForce 9600 GT.

Table 2. Average computation time in milliseconds for images with higher resolution. The image processing part includes background subtraction algorithm, color segmentation, edges extraction and distance transform computation. We show also the computation time of each matching cost function (non overlapping ratio and mean edge distance).

Image Resolution	Avg. Time (ms) Image processing	Avg. Time (ms) Non Overlapping Ratio	Avg. Time (ms) Mean Edge Distance
160 x 120	92.45	1.05	0.97
256 x 256	323.53	1.89	1.86
326 x 240	381.19	2.08	2.09
480 x 480	1187.92	3.09	3.05
512 x 512	1353.84	3.74	3.82
640 x 480	1581.86	5.29	5.58

improve the precision. In order to verify the increase of the precision given by the edge-based registration step with respect to the region-based registration, we experimentally compared the residual edge distance achieved by each step of the registration process (region-based and edge-based). Here, we iterated until convergence. Our results (fig. 6) show that the edge-based registration step allows correcting, for some images, incorrect region-based registrations that appear as peaks in the residual distance between edges. The figure 7 illustrates an example of such a correction.

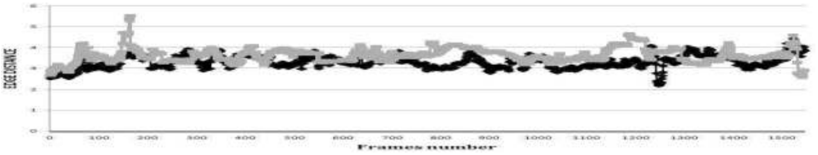

Fig. 6. Residual distance between edges achieved by the region-based registration (gray line) and the edge-based registration (black line)

4.2 Difficulties Encountered in Edge-Based Registration

After limiting the maximum number of iterations in the edge-based registration step, we encountered some difficulties related to the high instability and imprecision in the results. This is because the edges extracted from the image are not necessarily matched with the correct model edges by registration process, so the registration process is trapped in some local optimum. This issue can be limited by initializing the edge-based registration step using the usually smaller simplex after final iteration of the region-based registration step. Thus, the edge-based registration will start to search in a reduced search-space starting from the solution achieved by the previous step (region-based) avoiding being trapped in some local incorrect minimum. We can see the experimental results of the solution proposed in the figure 8.

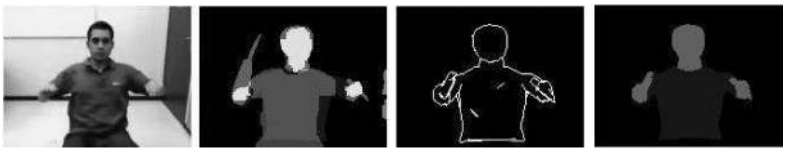

Fig. 7. Incorrect region-based registration corrected by the edge-based registration. The images are respectively: the captured image, the projection of the 3D model superposed with the segmented image that shows an incorrect region-based registration, the 3D model occluding edges that shows the correction by the edge-based registration and finally the 3D model projection showing the corrected 3D pose.

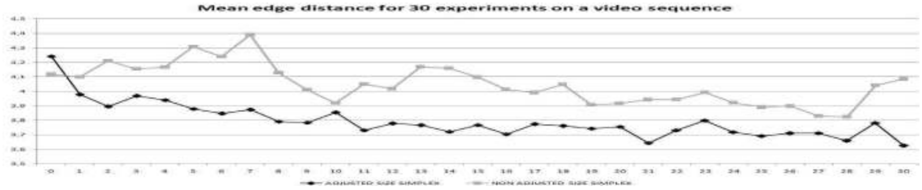

Fig. 8. Residual distance between edges on a video sequence. The abscissas is the number of iterations of the edge-based registration step (other iterations are devoted to the region-based registration step). The black line is the residual error with final simplex at step 1 (region) used as initial simplex at step 2 (edges). The gray line is the residual error using a large initial simplex. The residual distance between edges by reducing the size of the simplex (black line) is smaller because it can avoid more local wrong minimums.

4.3 Performance Experiments in Our Registration Process

In order to analyze the performance of our approach, we considered, for each experiment, the residual values of each evaluation function and also the number of failures (mistrackings) with varying numbers of iterations in the registration process.

We were interested in analyzing the performance from 1 to 500 iterations because the computation time is below 100 milliseconds (see table 1), thus allowing tracking at 10 Hz or more. In each experiment, we sampled the residual value of the non-overlapping ratio and the residual value of the mean edge distance. We considered only the mean residual for all the images in a video sequence. A way of measuring the robustness in each registration step is to count the number of failures for each experiment. We consider as failures or mistrackings the residual values above a defined threshold (a “peak”) for each evaluation function. In this way, if the residual value is relatively large, we consider that the solution output by the optimization algorithm is a “bad” registration. We show the experiments results for the video sequence 2 in the next figures (fig. 9, 10, 11 and 12).

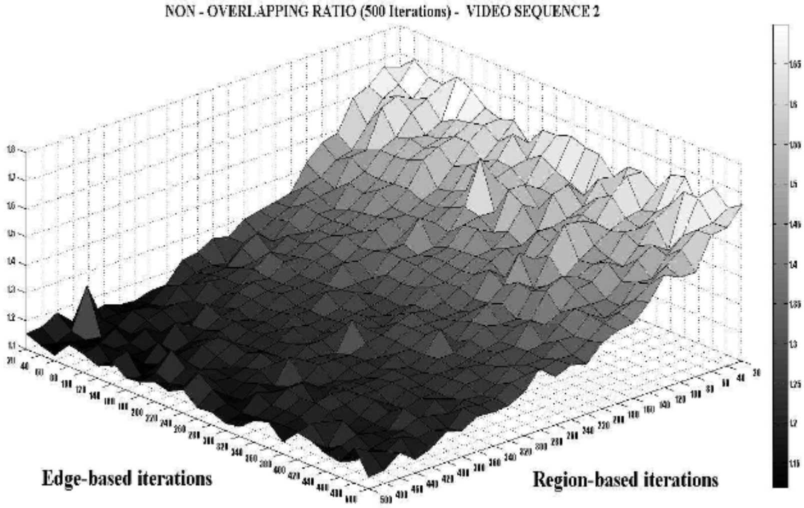

Fig. 9. Mean residual of the non-overlapping ratio (z-axis) with respect to the number of iterations of the region-based registration (x-axis) and the number of iterations of the edge-based registration (y-axis) achieved on video sequence 2. Experiments on video sequences 1, 3, 4, 5 and 6 showed similar results.

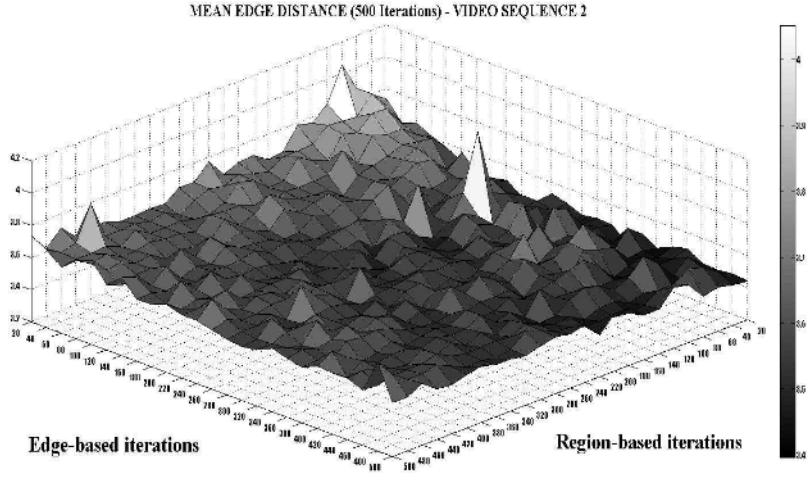

Fig. 10. Mean residual of the mean edge distance (z-axis) with respect to the number of iterations of the region-based registration (x-axis) and the number of iterations of the edge-based registration (y-axis) achieved on video sequence 2. Experiments on video sequences 1, 3, 4, 5 and 6 showed similar results.

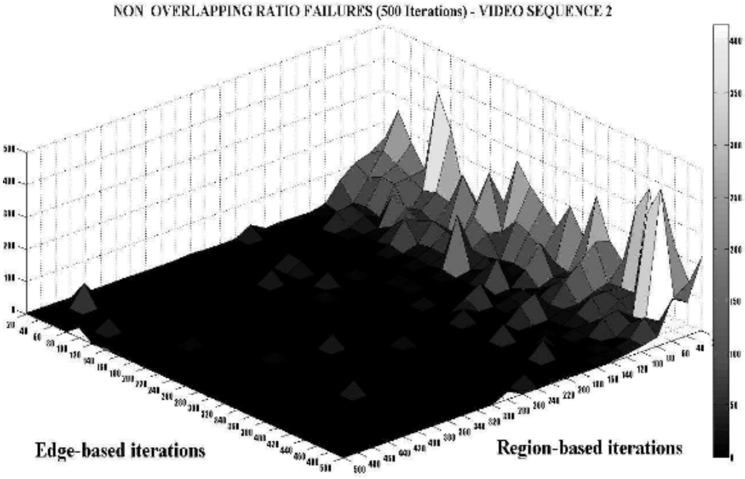

Fig. 11. Number of failures of the non-overlapping ratio (z-axis) with relation to the number of iterations of the region-based registration (x-axis) and the number of iterations of the edge-based registration (y-axis) obtained on the video sequence 2. Experiments on video sequences 1, 3, 4, 5 and 6 showed similar results.

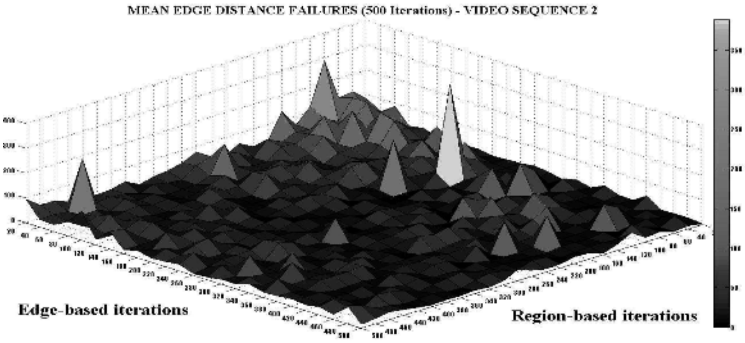

Fig. 12. Number of failures of the mean edge distance (z-axis) with relation to the number of iterations of the region-based registration (x-axis) and the number of iterations of the edge-based registration (y-axis) obtained on the video sequence 2. Experiments on video sequences 1, 3, 4, 5 and 6 showed similar results.

5 Conclusions

We have presented a 3D human motion capture algorithm by monocular vision in real-time, based on registering a 3D articulated model on color regions and then an edge distance criterion. Through experimental results (the surfaces 3D displayed above), we can understand the performance of the region-based and edge-based registration steps. From figure 9 and 10 we can see how the region-based registration step reaches to convergence faster that the edge-based registration step. The figures 11 and 12 show the instability of the edge-based registration step compared to the stability of the region-based registration step. So we need to combine the robustness and stability of the region-based registration and the precision of the edge-based registration.

In order to have the best performance in real-time for our approach, we decided to give priority to the stability of the registration when the number of iterations is below 200 (found experimentally from figure 11), thus, in this case, all the total iterations will be executed by the region based step. However, when the total number of iterations is above 200, the number of failures in region-based step registration is relatively small (figure 11), thus, we can take advantage of the precision achieved by the edge based step (figure 6 and 7) by sharing in the same proportion the total number of iterations between each step (50% of iterations for region-based step and 50% of iterations for edge-based step). Although the performance variation (fig. 9, 10, 11 and 12) was similar for all tested videos, the video sequence 6 (fig. 5) presented the highest number of failures (mistrackings) due to the ambiguity caused by the limited depth information in monocular images. Our future work aims at estimate the gradient of the edge distance in order to use a gradient descent optimization algorithm to improve the precision of our approach.

References

1. Agarwal, A., Triggs, B.: Recovering 3D human pose from monocular image. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 44–58 (2006)
2. Borgefors, G.: Distance transformations in digital images. Computer Vision, Graphics and Image processing 34, 344–371 (1986)
3. Cheung, G., Baker, S., Kanade, T.: Shape-from-silhouette for articulated objects and its use for human body kinematics estimation and motion capture. In: Computer vision and pattern recognition, Madison, Wisconsin, USA, pp. 16–22 (2003)
4. Deriche, R.: Fast algorithms for low-level vision. IEEE Transactions on Pattern Analysis and Machine Intelligence 12, 78–87 (1990)
5. Franco, J.-S., Boyer, E.: Une approche hybride pour calculer l'enveloppe visuelle d'objets complexes. In: ORASIS 2003, pp. 67–74. Gérardmer (2003)
6. Fontmarty, M., Lerasle, F., Danes, P.: Data Fusion within a modified Annealed Particle Filter dedicated to Human Motion Capture. In: IEEE / RSJ International Conference on Intelligent Robots and Systems IROS 2007, San Diego, CA, USA, October 29-November 2, pp. 3391–3396 (2007)
7. Gómez Jáuregui, D.A., Horain, P., Baroud, F.: Acquisition 3D des gestes par vision monoscopique en temps réel. In: Conférence MajecSTIC 2008, Marseille, France (2008)
8. Horain, P., Bomb, M.: 3D Model Based Gesture Acquisition Using a Single Camera. In: Proceedings of IEEE Workshop on Applications of Computer Vision WACV 2002, Orlando, Florida, December 3-4, pp. 158–162 (2002)
9. Lu, S., Huang, G., Samaras, D., Metaxas, D.: Model-based integration of visual cues for hand tracking. In: Proceedings of IEEE workshop on Motion and Video Computing, Orlando, Florida, pp. 119–124 (2002)

10. Marques Soares, J., Horain, P., Bideau, A., Nguyen, M.H.: Acquisition 3D du geste par vision monoscopique en temps réel et téléprésence. In: Actes de l'atelier Acquisition du geste humain par vision artificielle et applications, pp. 23–27. Toulouse (2004)
11. Mori, G., Malik, J.: Recovering 3D human body configurations using shape contexts. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 28, 1052–1062 (2006)
12. Nelder, J.A., Mead, R.: A simplex method for function minimization. Computer Journal 7, 208–313 (1965)
13. Pang, J., Qing, L., Huang, Q., Jiang, S.: Monocular Tracking 3D People by Gaussian Process Spatio-Temporal Variable Model. In: International Conference on Image Processing, ICIP 2007, San Antonio, Texas, USA, vol. 5, pp. 41–44 (2007)
14. Poppe, R.W.: Vision-based human motion analysis: An Overview. Computer Vision and Image Understanding 108(1-2), 4–18 (2007)
15. Sminchisescu, C., Triggs, B.: Estimating Articualted Human Motion with Covariance Scaled Sampling. International Journal of Robotics Research 22, 371–393 (2003)
16. Urtasun, R., Fleet, D.J., Fua, P.: 3D people tracking with gaussian process dynamical models. In: Proceedings of the Conference on Computer Vision and Pattern Recognition CVPR 2006, New York, vol. 1, pp. 238–245 (2006)
17. Viola, P., Jones, M.: Rapid Object Detection Using a Boosted Cascade of Simple Features. IEEE Computer Vision and Pattern Recognition 1, 511 (2001)
18. Wright Jr., R.S., Lipchak, B., Haemel, N.: OpenGL SuperBible: Comprehensive Tutorial and Reference, 4th edn., pp. 127–172. Addison-Wesley Professional, Ann Arbor (2007)