Progressive Human Skeleton Fitting

Jérôme Vignola, *Jean-François Lalonde* and *Robert Bergevin*

Laboratoire de vision et systèmes numériques (LVSN),
Département de génie électrique et de génie informatique,
Université Laval, Ste-Foy (Qc), Canada, G1K 7P4.
E-mail: {vignolaj, lalond02, bergevin}@gel.ulaval.ca

Abstract

This paper proposes a method to fit a skeleton or stick-model to a blob to determine the pose of a person in an image. The input is a binary image representing the silhouette of a person and the ouput is a stick-model coherent with the pose of the person in this image. A torso model is first defined, and is then scaled and fitted to the blob using the distance transform of the original image. Then, the fitting is performed independently for each of the four limbs (two arms, two legs), using again the distance transform. The fact that each limb is fitted independently speeds-up the fitting process, avoiding the combinatorial complexity problems that are frequent with this type of method.

Keywords: Skeleton fitting, Stick-model, Distance transform, Pose estimation.

1 Introduction

A method fitting a skeleton to the image region occupied by a person is needed as part of a monitoring system which attempts to recognize the same person from two different points of view or at different times. Matching persons is to be done according to the appearance of its different limbs in the image. This is why a part-based description of the person is needed, i.e. which groups of pixels represent the arm, the leg, etc.

The skeleton is a stick-model that represents the pose of the person in the image and makes it possible to segment the person into different parts. However, to reduce the high combinatorial complexity typical of the problem at hand, the fit should be obtained in a progressive manner, i.e. one limb at a time, after each limb has been previously scaled with respect to the blob's size.

Typical situations the system should handle include people walking parallel to or facing the camera (see figure 2). However, the system must be robust and tolerate more complex situations. The main assumption to be made is that people shall be in an upright position.

2 Related work

The ideal solution to the skeleton fitting problem would be to process the whole image instead of only a binary blob. This way it would be possible to obtain more details about the actions of a person (for example, is the person facing the camera or moving away from it?). Such details are unavailable when using a binary blob. Unfortunately, at this time, there is no technique that can segment a person's silhouette and extract his different parts from a real complex scene accurately and fast enough for a system such as the one described herein [9, 10].

The fitting process may be performed automatically or non-automatically, as well as intrusively or non-intrusively. Intrusive manners include, for example, imposing optical markers on the subject [8] while non-automatic method could involve interacting manually to set the joints on the image, such as in [2]. These methods are inappropriate for a monitoring system such as the one described herein which strives to be non-intrusive and automatic. People have to be monitored without interactions and this operation must be processed without human interaction.

Many methods have been tested to find the pose of a human subject in an automatic and non-intrusive manner. Some of these methods only provide the position of extremities (in most cases these extremities are the head, hands and feet), while other systems give the complete position of all of the joints of the person (these joints usually include the neck, shoulder, elbow, etc.). The system described by Haritaoglu *et al.* [7] belongs to the former category and uses geometrical features to divide the blob and determine the different extremities. Fujiyoshi and Lipton [5] have no model but rather determine the extremities of the blob with respect to the centroid and assume that these points represent the head, hands and feet. The exact position of the body parts is not required for their application. In the second class, involving systems which provide the exact position of all body joints, one can find [6], which uses a stick-model and tries to fit it

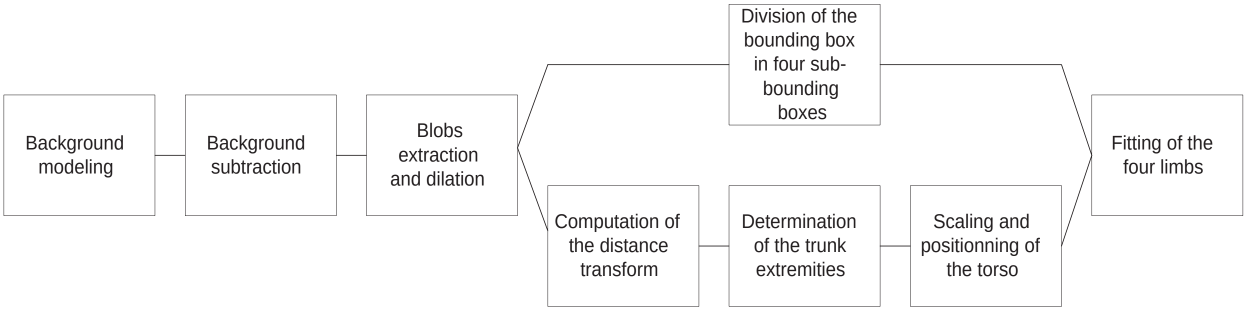

Background modeling
Background subtraction
Blobs extraction and dilation
Division of the bounding box in four sub-bounding boxes
Computation of the distance transform
Determination of the trunk extremities
Scaling and positionning of the torso
Fitting of the four limbs

Figure 1: Overview of the system.

to the blob. Neural networks [13] and genetic algorithms [12] are also used. Finally, [4] presents a method dealing specifically with the detection of armed robbery. This method analyses the skeletonization of the blob to decide whether or not a robbery is taking place in the scene.

The interest of the method described herein is that it combines, in the same system, the speed of some techniques and the robustness of others, while giving a complete description of the body. These three elements are almost never present simultaneoustly in a system.

3 Proposed approach

A general overview of the system is presented in 1.

3.1 Blobs Extraction

A custom background subtraction method is used to extract the silhouette of the person. That is, the mean and the standard deviation of each pixel is computed in a series of images without any person. Then, a pixel is regarded as belonging to a moving object if the difference between the mean and the current value of that pixel is higher than a certain threshold related to the standard deviation. Figure 2 b) shows background subtraction.

Once this step is completed, a test is made on each of the blobs obtained to ensure that the area is sufficient, and not composed only of noise. Blobs which are too small are eliminated. A filling algorithm is then used to ensure that the blobs are exempt of any holes. Finally, two steps of dilation are carried out to obtain a smooth silhouette.

Then, a distance transform is computed on this image, using an implementation of the two pass algorithm [3]. The result, shown in figure 3, is an image which gives, for each pixel, the distance from the nearest contour. This result is used in further processing.

In our method, the fitting is carried out progressively, one limb at a time. It is necessary to ensure that each limb covers his part of the blob. This is why the bounding box is divided into four parts. To do this, the center of mass is computed for the whole blob. Then the blob is divided in four sections by tracing vertical and horizontal lines through the center of mass. The new bounding boxes are computed for each of the four sub-images. These four bounding boxes will serve in the limbs fitting module.

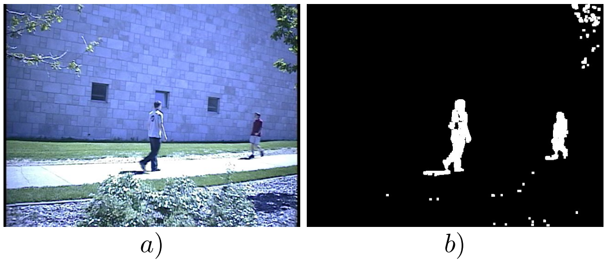


Figure 2: a) Typical situation the system has to handle. b) Blobs extraction before any treatment.

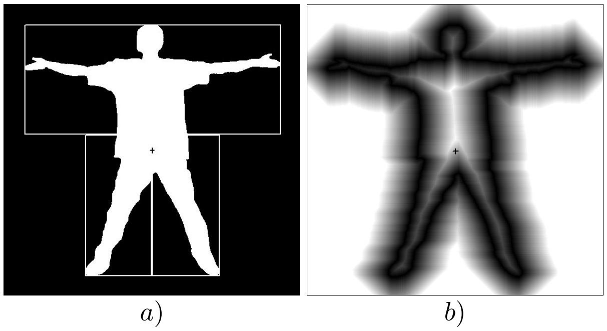


Figure 3: a) A binary blob and the four corresponding bounding boxes. b) The distance transform obtained for image a) and normalized between 0 and 255.

3.2 Skeleton model

The skeleton model used herein is represented by a vector of 14 body parts. It is shown in figure 4.

$$B = \{bp_1, bp_2, \ldots, bp_{14}\} \quad (1)$$

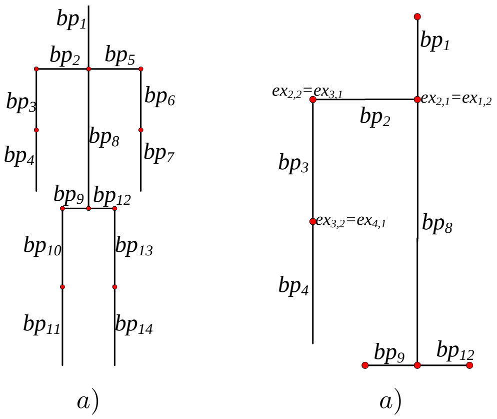


Figure 4: a) The stick-model used to do the fitting. b) A closer look at the right arm.

The proportions between the different parts are fixed and were determined by consulting the NASA Anthropometric Source Book [11] as well as work reported by other research teams specializing in human modeling [1, 2]. Each body part has its own range of possible motion. Angle constraints ensure that the stick-model will not take on any undesirable positions, i.e. positions a human cannot take.

Each body part is composed of two extremities, these two extremities representing the coordinates of the body part in the image plan:

$$bp_i = \{ex_{i,1}, ex_{i,2}\} \tag{2}$$

where

$$ex_{i,j} = (x_{i,j}, y_{i,j}) \tag{3}$$

$x_{i,j}$ is the x coordinate of extremity *j* of body part *i* and $y_{i,j}$ is the y coordinate of extremity *j* of body part *i*.

3.3 Torso fitting

The torso (*T*) is defined as being a subset of *B*, i.e.

$$T = \{bp_1, bp_2, bp_5, bp_8, bp_9, bp_{12}\} \tag{4}$$

The trunk (bp_8) schematically represents the spinal cord of the person. The success of positioning the whole skeleton relies on the trunk, which is the process' starting point and is based on human morphological proportions. The neck ($ex_{8,1}$) has been located at 2/15 of the total human height, starting from the head, while the hips ($ex_{8,2}$) are located at 8/15.

However, the total height of the person might not be the same as the total height of the bounding box. For example, a person could have an arm raised above his head. To overcome this problem, the height is measured by sampling points in the y axis in the upper part of the blob and finding the maximum of the distance transform for each of the sampled points. Using linear regression, a line is fitted on the points sampled, and the height is found by raising a segment up following the line direction until the blob's frontiers are reached.

Let *DT* be the distance transform image. *DT(x,y)* would be the value (between 0 and 255) of the pixel at coordinate *(x,y)* in the distance transform image. The first extremity of the trunk is $ex_{8,1} = (x_{8,1}, y_{8,1})$. $y_{8,1}$ is set constant to 2/15 of the person's height and $x_{8,1}$ is computed as being the pixel that maximize the value of the distance transform. Let x_l be the left x coordinate of the bounding box, and x_r be the right coordinate. We can compute

$$x_{8,1} = x|\max_{x_l<i<x_r}(DT(i, y_{8,1})) \tag{5}$$

The same process is repeated for $x_{8,2}$ where $y_{8,2}$ is set to 8/15 of the total human height. In this way, a rough approximation of the two trunk points is obtained. In order to have a better approximation, the position of these points is refined by using the distance transform image once again. The points are moved toward the ascending gradient until they reach a local maximum. This process is repeated a fixed number of iterations. Once these two points are calculated, the torso can be scaled with respect to the distance between those two points. The fact that the trunk extremities are positioned relative to the person's height justifies the need for the real height of the person, instead of only the blob's height. Figure 5 illustrates torso fitting.

Once the trunk has been placed, the system has to choose between a frontal and a side model. To do this, a perpendicular line to the trunk is traced, crossing $ex_{8,1}$, and the width of the blob is computed. If this value is above a fixed threshold, the frontal model is selected. Otherwise, the side model is chosen. One should note that the only difference in the two models is the clavicles (bp_2 and bp_5) length.

3.4 Limbs fitting

Limbs include the right arm, left arm, right leg and left leg. As for the torso, each limb is a subset of *B*. It is composed of two body parts. For example,

$$Arm_{right} = \{bp_3, bp_4\} \tag{6}$$

Because of the skeleton model, all limbs have the following form

$$L_i = \{bp_i, bp_{i+1}\} \tag{7}$$

and are linked to bp_{i-1}.

Once the torso is fitted, all four limbs are scaled based on the torso size. Then, to fit bp_i and bp_{i+1} , $ex_{i,1}$ is first

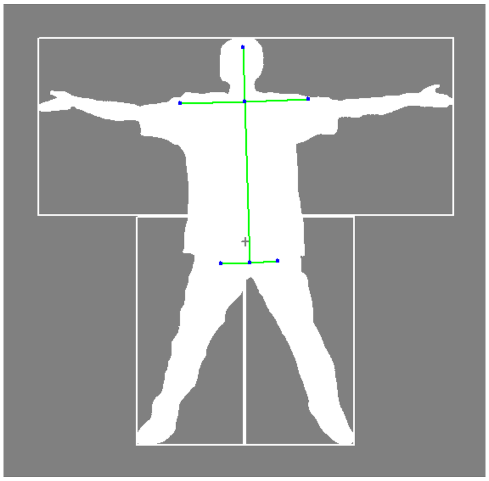


Figure 5: Torso fitting. The lenght of the torso is defined based on the blob's size. Then the torso model is scaled and his position is determined using an algorithm based on the distance transform.

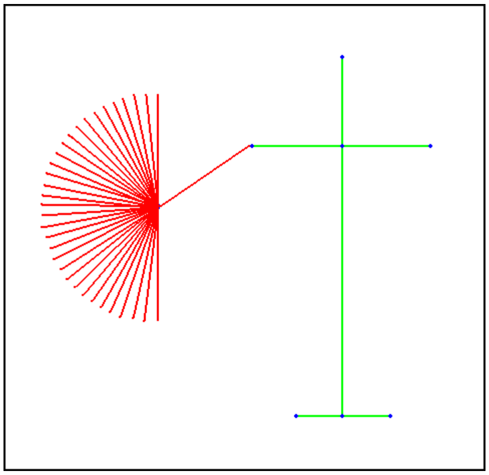


Figure 6: One particular position for bp_3 and all the candidate positions generated by the system for bp_4. This process is repeated for all possible positions of bp_3 admitted by the angle constraints. The sampling angle in this case is π/32.

fixed as having the same coordinates as $ex_{i-1,2}$ (these coordinates are known because $ex_{i-1,2}$ belongs to the torso and the torso has already been fitted). Then, the set *S* of all candidate solutions for this limb is generated. In other words, all the possible positions for bp_i and bp_{i+1} are generated according to the angle constraints that were imposed, and with a predefined sampling angle (see figure 6). This angle influences the robustness and speed of the technique. If the sampling angle is too large, a good solution could be overlooked. However, the whole process might be too long if the sampling angle is chosen small. It then becomes possible to sample points along bp_i and bp_{i+1} for each candidate solution. For a particular solution, if the angle between bp_{i-1} and bp_i is α and angle between bp_i and bp_{i+1} is β, these sampled points are

$$P_i^\alpha = \{p_{1,i}^\alpha, \ldots, p_{n,i}^\alpha\} \tag{8}$$

$$P_{i+1}^\beta = \{p_{1,i+1}^\beta, \ldots, p_{m,i+1}^\beta\} \tag{9}$$

where $p_{1,i}^\alpha$ is the coordinate of the first sampled point of the body part bp_i in the candidate solution with angle α. Here *n* and *m* depend on the sampling rate. The sampling rate is an adjustable parameter that also influences the robustness and speed of the method. Indeed, the more points there are along a line to validate a solution, the more robust the system is if a part of a limb has been poorly extracted. However, the more time-consuming the fitting process becomes.

Two criteria have been developed which determine if a solution is good or not. The first one is the *interior rating (IR)*, which is computed with the distance transform image:

$$IR_i^{\alpha\beta} = \sum_{k=1}^{n} DT(p_{k,i}^\alpha) + \sum_{k=1}^{m} DT(p_{k,i+1}^\beta) \tag{10}$$

where $DT(p_{k,i}^\alpha)$ is the value of the pixel $p_{k,i}^\alpha$ in the distance transform image.

The greater the distance between the limb and any contour, the highest the *IR* value is.

The second criterion is the *coverage rating (CR)*, which is related to the bounding box. It is a boolean variable. Since the bounding box has been divided into four parts, there is one *CR* for every limb. When L_i is fitted, a test is processed to ensure that the blob is covered, i.e. ex_{2i+1} is close enough from the limb's bounding box. If so, *CR* takes the value true. Otherwise, it assumes the value false. At this point, the minimal distance to fix the *CR* criterion as true has been set to 10% of the person's height. If *CR* is true, the solution is accepted and the corresponding *IR* value is compared to those of the other potential solutions. The solution meeting the *CR* criterion and having the highest *IR* is considered as being the best solution.

4 Strength

This method is fast compared to a global fitting method. The whole process takes about one second on a Pentium III 550MHz. The fact that all limbs are fitted independently of each other speeds-up the process and avoids the combinatorial complexity problem which would occur with a global method.

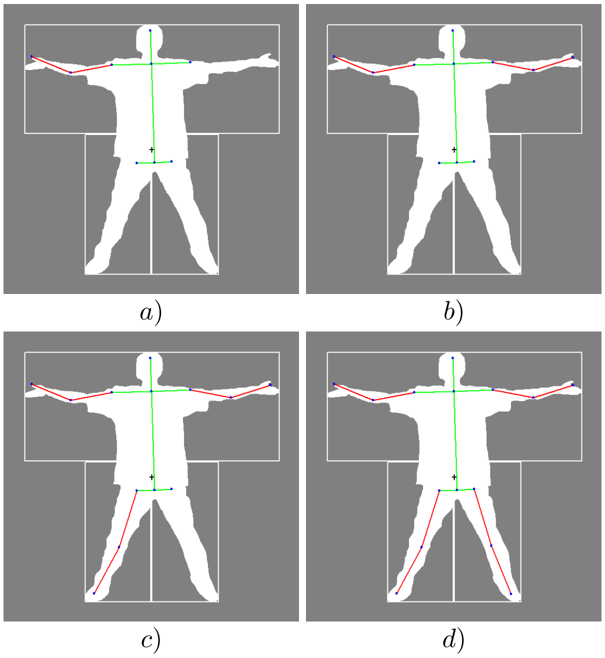


Figure 7: Limbs fitting. Fitting is done independently for each limb. This local method speeds up the process and improves the robustness if some regions of the blob have been poorly extracted.

This method also facilitates the segmentation of the body into different parts. Indeed, the system does not only extract the position of an extremity (a hand or a foot, for example), but rather a segment that represents the whole limb. This element will be very useful when the pixels representing this limb must be extracted. A system which only provides the extremity position does not give any indication as to where the limb connected to this extremity is located.

A local method such as the one presented here also increases the robustness of the whole system in the following way. If some region of the blob has been poorly extracted, it is likely that only this part will be poorly fitted and that the other limbs will be succesfully fitted, at least if the torso has been successfully fitted. In the case of a global method, a small error can lead to the failure of the whole fitting module.

5 Experimental results

The different modules of the system have been tested on a series of 500 images. These blobs represent silhouettes with varying scales, point of vues, standing poses (including poses where the head does not represent the highest point of the blob or where the person is bending) and levels of precision in the blob extraction.

Tested images have 640x480 pixels and typical silhouettes are between 137 and 320 pixels high. For a blob with defaults (shadow, incomplete parts, etc.), the local fitting method permits in most cases to obtain satisfactory results, at least for the body parts that have been correctly extracted. Figure 9 shows that the method is able to process blobs that are not well shaped, for instance more difficult cases with different kinds of shadows.

An evaluation technique to analyse the results has been developed to classify a given solution for a particular blob as being either acceptable or not by comparing how the stick-model has been fitted to a blob by the system and by humans. First, the skeleton is fitted by the computer and the joints of the skeleton are saved. The same blob is then fitted by a human. This is considered as the optimal solution. To compare these two solutions, the distance between each of the main joints is computed. This distance is then normalized to compensate for the scale effect. The results presented herein are for a trunk of 100 pixels. Mean and standard deviations shown in table 1 have been determined for 500 silhouettes. Table 2 presents results for the 10 best fitted skeletons and table 3 gives results for average fitted skeleton. Finally, table 4 shows results for poorly fitted skeletons.

Experiments show that an error of less than 5 pixels for a joint is excellent and that an error of about 10 pixels is acceptable. The reader will notice that poor fitting is mainly due to scaling errors, i.e. the height of the blob has not been correctly determined, and to unusual positions, for example if a limb is not visible in the image. However, because of the local fitting method, even if one part is missed, the overall fitting is often acceptable.

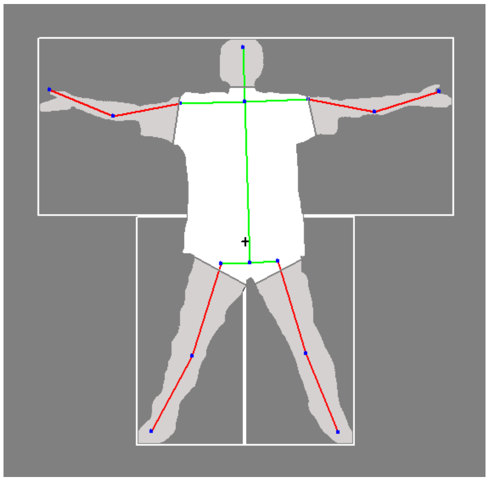


Figure 8: The segmentation of the body in his different parts is the next step toward building an operational and efficient system.

6 Conclusion

In this paper, a new method of stick-model or skeleton fitting has been presented. This technique is original in that it is performed progressively, one limb at a time, instead of globally. This way, the process is faster. A skeleton model was defined and scaled with respect to the person's height. However, the blob's height does not always represents the person's height and this could lead to an error in the scaling factor. To overcome this problem, an algorithm was developed to compute the height of the person even in situations where the head is not the highest point of the blob.

The four limbs of the model are scaled with respect to the torso size. Then, they are fitted individually by generating all possible positions and selecting the best position. This best solution is computed using two criteria. First, the *IR* criterion gives a measure of the *depth* of a limb in a blob, i.e. how far the limb is located from any contour, by using the distance transform of the blob's binary image. The *CR* criterion then involves the validation of the position of the limb by checking if the limb covers the total bounding box area. As the fitting is conducted separately for each limb, a different bounding box is computed for each part of the blob.

Future work includes segmenting the person into different parts (see figure 8) as well as possibly improving the system by adding a module to analyse the posture of the subject based on the skeleton position.

Acknowledgements: This research was supported by an NSERC grant to R. Bergevin.

References

[1] Norman I. Badler, Cary B. Phillips, and Bonnie Lynn Webber. *Simulating Humans: Computer Graphics Animation and Control*. Oxford University Press, New York, 1993. ISBN 0-19-507359-2.

[2] Carlos Barrón and Ioannis A. Kakadiaris. Estimating anthropometry and pose from a single uncalibrated image. *Computer Vision and Image Understanding: CVIU,* 81(3):269–284, march 2001.

[3] Gunilla Borgefors. Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing*, 34(3):344–371, june 1986.

[4] Jaime Dever, Niels da VitoriaLobo, and Mubarak Shah. Automatic visual recognition of armed robbery. In *Proceedings of the 16th IEEE International Conference on Pattern Recognition*, volume 1, pages 451–455, 2002.

[5] H. Fujiyoshi and A. Lipton. Real-time human motion analysis by image skeletonization. In *Proceedings of the 4th IEEE Workshop on Applications of Computer Vison*, pages 15–21, 1998.

[6] Yan Guo, Gang Xu, and Saburo Tsuji. Understanding human motion patterns. In *Proceedings of the 12th International Conference on Pattern Recognition*, pages 325–330, 1994.

[7] I. Haritaoglu, D. Harwood, and L. Davis. Who, when, where, what: A real time system for detecting and tracking people. In *Proceedings of the 3th Face and Gesture Recognition Conference*, pages 222–227, 1998.

[8] L. Herda, P. Fua, R. Plankers, R. Boulic, and D. Thalmann. Skeleton-based motion capture for robust reconstruction of human motion. In *Proceedings of the Computer Animation*, pages 77–83, 2000.

[9] S. Ioffe and D. Forsyth. Finding people by sampling. In *Proceedings of the 7th International Conference on Computer Vision*, volume 2, pages 1092–1097, 1999.

[10] Anuj Mohan, Constantine Papageorgiou, and Tomaso Poggio. Example-based object detection in images by components. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(4):349–361, 2001.

[11] NASA. *Anthropometric Source Book*, volume 2. Springfield VA, Johnson Space Center, Houston, TX, 1978.

[12] Jun Ohya and Fumio Kishino. Human posture estimation from multiple images using genetic algorithm. In *Proceedings of the 12th International Conference on Pattern Recognition*, pages 750–753, 1994.

[13] Kazuhiko Takahashi, Tetsuya Uemura, and Jun Ohya. Neural-network-based real-time human body posture estimation. In *Proceedings of the 2000 IEEE Signal Processing Society Workshop*, volume 2, pages 477–486, 2000.

Stats	Head	Upper Trunk	Right Elbow	Right Hand	Left Elbow	Left Hand	Lower Trunk	Right Knee	Right Foot	Left Knee	Left Foot	Global
μ	9.32	8.40	13.22	16.67	9.41	10.33	10.40	16.17	15.51	15.43	15.51	12.76
σ	9.58	10.20	15.70	30.63	7.30	8.05	8.01	8.58	8.79	8.50	8.79	7.66

Table 1: Mean difference (μ) and Standard deviation (σ) for the 500 images. The distance is computed in pixels and normalized for a trunk of 100 pixels. The **Global** column represents the mean of all distances.

ID	Head	Upper Trunk	Right Elbow	Right Hand	Left Elbow	Left Hand	Lower Trunk	Right Knee	Right Foot	Left Knee	Left Foot	Global
406	3.34	2.11	1.49	2.11	4.72	5.38	6.67	6.33	1.49	10.55	1.49	4.15
295	2.31	5.17	3.27	4.90	1.63	4.62	0.00	13.96	1.63	6.74	1.63	4.17
164	6.56	1.82	3.64	5.75	4.07	6.56	4.07	7.71	4.07	4.07	4.07	4.76
297	1.61	4.82	3.21	6.81	6.81	2.27	3.59	18.79	1.61	7.18	1.61	5.30
475	5.02	1.22	0.00	6.56	4.39	6.09	1.72	8.18	8.18	9.83	8.18	5.40
236	7.93	3.55	3.55	10.64	5.72	8.09	1.59	7.93	4.49	1.59	4.49	5.41
381	3.15	2.81	1.41	4.45	5.07	3.98	8.56	8.90	7.96	5.80	7.96	5.46
403	4.94	1.56	3.13	4.94	6.25	9.38	7.97	6.99	3.49	9.50	3.49	5.60
301	3.65	4.61	3.26	6.73	0.00	4.61	7.30	12.42	3.65	11.88	3.65	5.61
410	5.65	1.94	4.94	3.06	3.06	3.06	10.96	8.33	4.94	12.63	4.94	5.77

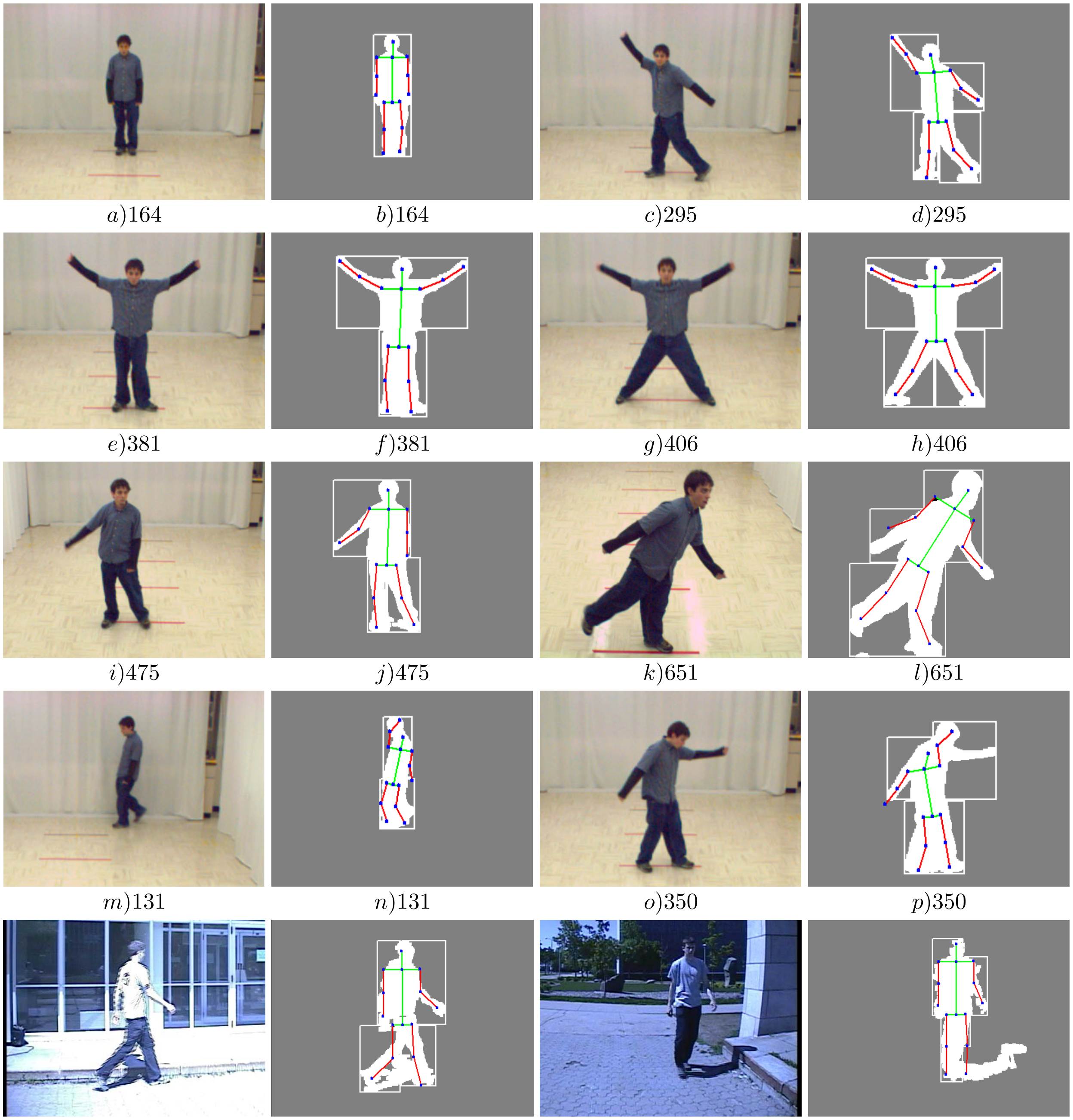
Table 2: The difference for joint location for the ten skeletons with the best fitting. The distance is computed in pixels and normalized for a trunk of 100 pixels. The **ID** column represents the frame number and some results presented herein can be referenced in figure 9.

ID	Head	Upper Trunk	Right Elbow	Right Hand	Left Elbow	Left Hand	Lower Trunk	Right Knee	Right Foot	Left Knee	Left Foot	Global
207	5.17	5.45	12.06	6.89	11.56	11.56	5.45	16.97	12.06	11.56	12.06	10.07
307	3.72	6.00	30.15	29.50	6.00	6.86	5.26	10.13	4.99	3.33	4.99	10.09
316	6.64	5.81	10.81	14.85	6.83	8.06	1.61	11.73	14.41	16.11	14.41	10.12
152	6.67	5.85	6.67	6.67	0.00	9.25	3.70	14.45	21.10	15.81	21.10	10.12
651	9.82	6.09	16.69	32.90	5.81	5.29	6.61	8.37	13.41	11.56	13.41	11.81

Table 3: The difference for joint location for skeletons with average fitting. The distance is computed in pixels and normalized for a trunk of 100 pixels.

ID	Head	Upper Trunk	Right Elbow	Right Hand	Left Elbow	Left Hand	Lower Trunk	Right Knee	Right Foot	Left Knee	Left Foot	Global
461	52.28	36.28	74.24	157.59	26.74	2.05	18.57	16.41	4.35	13.37	4.35	36.93
350	50.52	52.34	44.89	29.72	68.85	91.97	20.39	19.37	10.75	10.75	10.75	37.30
501	31.50	42.15	50.83	34.69	28.10	21.51	50.37	55.04	53.22	53.72	53.22	43.12
131	27.41	29.59	64.10	152.37	28.16	27.79	14.40	19.58	41.04	33.61	41.04	43.55
132	33.98	33.98	61.53	150.00	36.52	30.81	30.81	35.16	36.52	36.52	36.52	47.49

Table 4: The difference for joint location for skeletons with poor fitting. The distance is computed in pixels and normalized for a trunk of 100 pixels.



$a)164$ $b)164$ $c)295$ $d)295$

$e)381$ $f)381$ $g)406$ $h)406$

$i)475$ $j)475$ $k)651$ $l)651$

$m)131$ $n)131$ $o)350$ $p)350$

$q)$ $r)$ $s)$ $t)$

Figure 9: Some results obtained with the described method. First and third columns represent the original images, second and fourth columns represent the fitted skeleton. Figures a) to j) show very good results obtained in different situations. Figures k) and l) are an average fitted skeleton. Shoulders are a bit too large, but the scale factor is the good one and the overall fitting is acceptable. Figures m) to p) present poor fitting. In these two cases, the height is not the good one, which leads to scaling error. This is due to the head position. However, the overall fitting is still not so bad. Finally, figures q) to t) demonstrates, with images took in different conditions, that using the bounding box constraints, the skeleton is well fitted even if there is shadow between the two legs (r) or side shadow (t).