Real-time Upper-body Human Pose Estimation using a Depth Camera

Himanshu Prakash Jain, Anbumani Subramanian

HP Laboratories
HPL-2010-190

Keyword(s):

Haar cascade based detection, template matching, weighted distance transform and pose estimation

Abstract:

Automatic detection and pose estimation of humans is an important task in Human- Computer Interaction (HCI), user interaction and event analysis. This paper presents a model based approach for detecting and estimating human pose by fusing depth and RGB color data from monocular view. The proposed system uses Haar cascade based detection and template matching to perform tracking of the most reliably detectable parts namely, head and torso. A stick figure model is used to represent the detected body parts. Then, the fitting is performed independently for each limb, using the weighted distance transform map. The fact that each limb is fitted independently speeds-up the fitting process and makes it robust, avoiding the combinatorial complexity problems that are common with these types of methods. The output is a stick figure model consistent with the pose of the person in the given input image. The algorithm works in real-time and is fully automatic and can detect multiple non-intersecting people.

External Posting Date: November 21, 2010 [Fulltext] Approved for External Publication
Internal Posting Date: November 21, 2010 [Fulltext]

Real-time Upper-body Human Pose Estimation using a Depth Camera

Himanshu Prakash Jain
Indian Institute of Technology, Madras
himanshu@cse.iitm.ac.in

Anbumani Subramanian
HP Labs, Bangalore
anbumani@hp.com

Abstract

Automatic detection and pose estimation of humans is an important task in Human- Computer Interaction (HCI), user interaction and event analysis. This paper presents a model based approach for detecting and estimating human pose by fusing depth and RGB color data from monocular view. The proposed system uses Haar cascade based detection and template matching to perform tracking of the most reliably detectable parts namely, head and torso. A stick figure model is used to represent the detected body parts. Then, the fitting is performed independently for each limb, using the weighted distance transform map. The fact that each limb is fitted independently speeds-up the fitting process and makes it robust, avoiding the combinatorial complexity problems that are common with these types of methods. The output is a stick figure model consistent with the pose of the person in the given input image. The algorithm works in real-time and is fully automatic and can detect multiple non-intersecting people.

Keywords: Haar cascade based detection, template matching, weighted distance transform and pose estimation.

1. Introduction

Motion capture for humans is an active research topic in the areas of computer vision and multimedia. It has many applications ranging from computer animation and virtual reality to human motion analysis and human-computer interaction (HCI) [1] [2]. The skeleton fitting process may be performed automatically or manually, as well as intrusively or non-intrusively. Intrusive manners include, for example, imposing optical markers on the subject [3] while non-automatic method could involve interacting manually to set the joints on the image, such as in [4]. These methods are usually expensive, obtrusive, and not suitable for surveillance or HCI purposes. Recently, due to the advances on imaging hardware and computer vision algorithms, markerless motion capture using a camera system has attracted the attention of many researchers. One of the commercial solutions for markerless motion capture currently under development includes Microsoft's Kinect system for console systems.

Since the application domain is less restrictive with only a monocular view, human pose estimation from monocular image captures has become an emerging issue to be properly addressed. Haritaoglu et al. [8] tries to find the pose of a human subject in an automatic and non-intrusive manner. It uses geometrical features to divide the blob and determine the different extremities (head, hands and feet). Similarly, Fujiyoshi and Lipton [9] 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. Guo et al. [7] attempts to find the exact positions of all body joints (like the neck, shoulder, elbow, etc.) by minimizing the distance based criterion function on the skeletonized foreground object to fit the stick model. Neural networks [5] and genetic algorithms [6] have also been used to obtain the complete position of all of the joints of the person.

The simplest representation of a human body is the stick figure, which consists of line segments linked by joints. The motion of joints provides the key to motion estimation and recognition of the whole figure. This concept was initially considered by Johansson [12], who marked joints as *moving light displays* (MLD). Along this vein, Rashid [20] attempted to recover a connected human structure with projected MLD by assuming that points belonging to the same object have higher correlations in projected positions and velocities.

The organization of the paper is as follows: Section 2 discusses the proposed approach with subsections giving

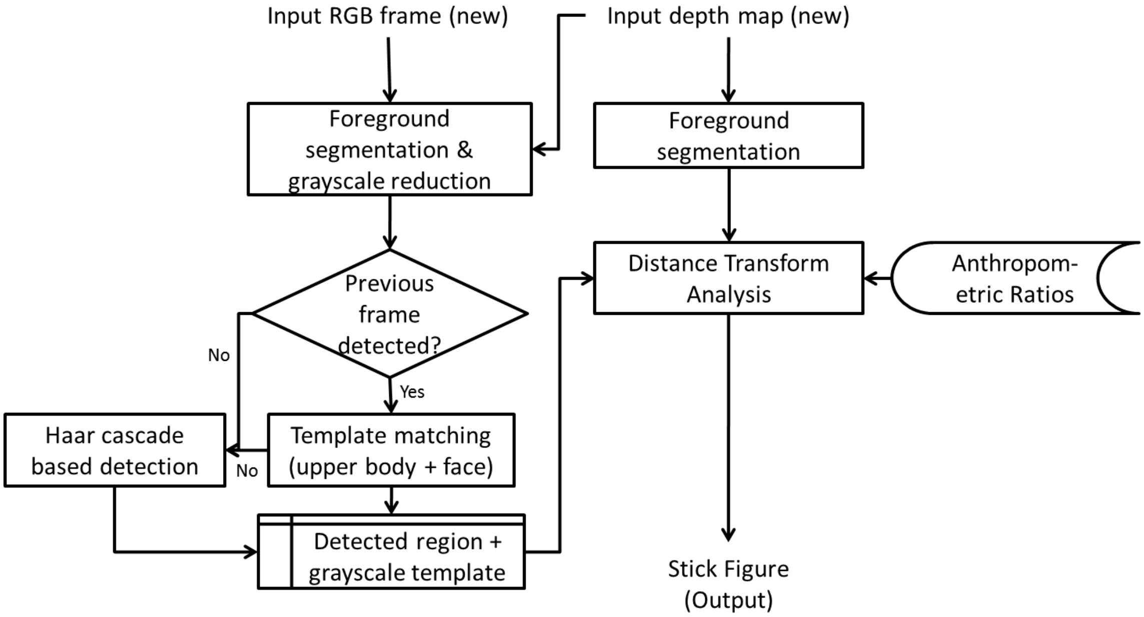

Input RGB frame (new)
Input depth map (new)
Foreground segmentation & grayscale reduction
Foreground segmentation
Previous frame detected?
No
Yes
Distance Transform Analysis
Anthropom-etric Ratios
Haar cascade based detection
No
Template matching (upper body + face)
Detected region + grayscale template
Stick Figure (Output)

Figure 1: Flowchart of the proposed system.

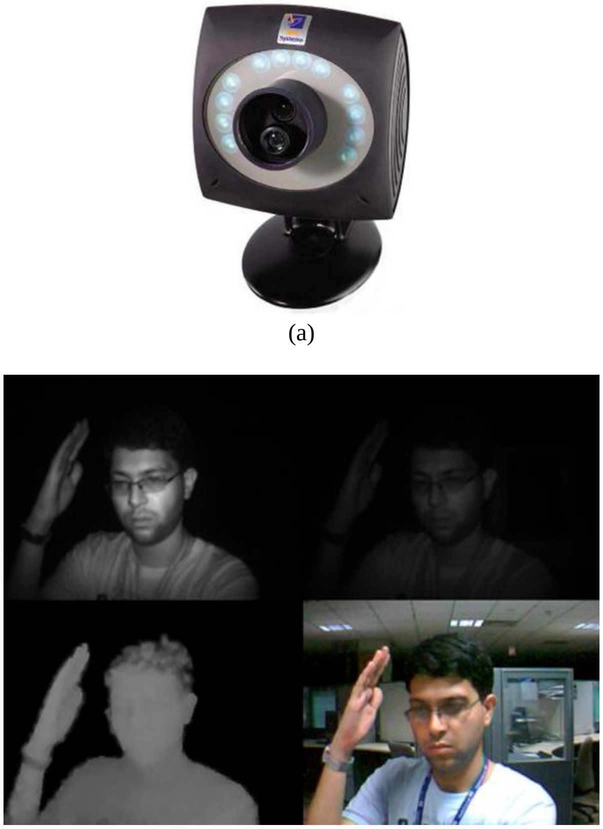
details about each module used in the system. Section 3 extends the discussion towards the implementation details about the proposed prototype. Finally, Section 4 concludes the paper and delineates possible directions for future research.

2. Overview of the entire system

In this work, we assume that a depth-camera is static and is at human height. It is also assumed that users' interaction spaces are non-intersecting and upper-body and face are visible without any occlusion. A block diagram of the human detection and pose estimation approach used in our work is shown in Fig. 1. The following subsections provide details of each module incorporated in the system.

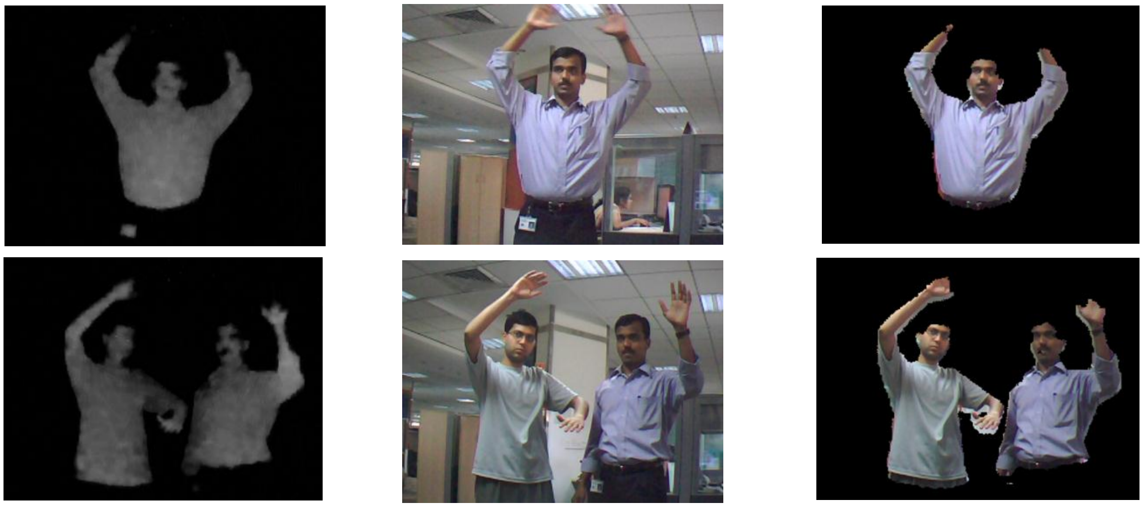
2.1. Depth camera

We use ZCam from 3DV Systems (shown in Fig. 2(a)) in our work. This camera uses active illumination for depth sensing – it emits modulated infra-red (IR) light and based on the time-of-flight principle, the reflected light is used to calculate depth (distance from camera) in a scene. This camera provides both RGB (VGA size) image and a grayscale depthmap (half-VGA size) image at 30 frames per second (fps).


(a)

(b)

Figure 2: (a) ZCam from 3DV Systems (b) Data output from ZCam – primary and secondary infrared images, a depthmap and a RGB color image.


(a)

(b)

(c)

Figure 3: (a) Input depthmaps (b) input RGB images (c) Foreground segmented RGB images obtained using (a) and (b).

Figure 2(b) shows a sample of four images obtained from the camera. The top row shows active brightness (left) and passive brightness (right) IR images and the bottom row shows the depthmap (left) and the RGB (right) image respectively. It can be observed in the depthmap, the depth values of objects near the camera appear bright while those of objects that are farther appear darker.

2.2. Foreground Segmentation

We use RGB image and the depthmap as input images to the system. A sample input frames are shown in Fig. 3. The raw depth map is threshold to remove noise with low intensity values. Then, blob analysis is done to remove very small blobs. The depthmap is then used for obtaining the foreground object in the RGB image.

2.3. Haar cascade based detection

The object detector described in [10] and [11] is based on Haar classifiers. Each classifier uses rectangular areas (Haar features) to make the decision if the region of the image looks like the object of interest or not.

$$f_i = \begin{cases} +1 & v_i \geq t_i \\ -1 & v_i < t_i \end{cases}$$

Figure 4 shows different types of Haar features used. The Haar detector uses a form of AdaBoost but organizes it as a rejection cascade of nodes, where each node is a multitree AdaBoosted classifier designed to have high (say, 99.9%) detection rate (low false negatives) at the cost of a low (near 50%) rejection rate (high false positives). For each node, a “not in class” result at any stage of the cascade terminates the computation, and the algorithm then declares that no object exists at that location.

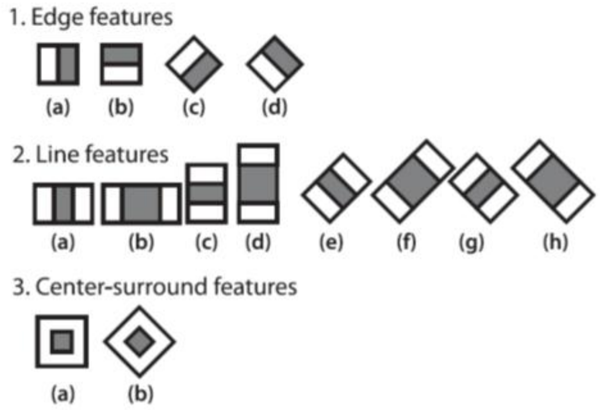


Figure 4: Types of Haar-like features used by object detection classifier in [11].

$$F = sign(w_1 f_1 + w_2 f_2 + \cdots + w_n f_n)$$

Here, the sign function returns -1 if the number is less than 0, 0 if the number equals 0, and +1 if the number is positive. On the first pass through the dataset, the threshold t_i of f_i is learnt. Boosting then uses the resulting errors to calculate the weighted vote w_i. Thus, true class detection is declared only if the computation makes it through the entire cascade (see Fig. 5).

For upper body detection, the classifier trained for upper-body (head + torso) [13] [14] is used. The detected

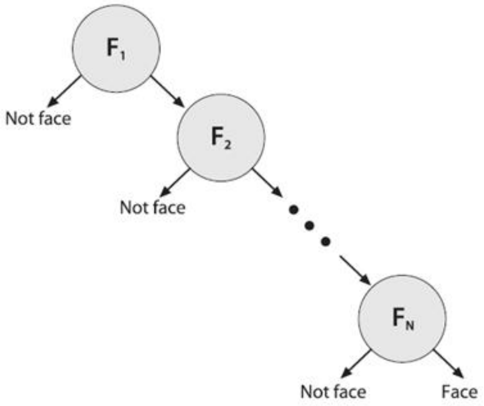


Figure 5: Rejection cascade of the classifier where each node represents a multitree boosted classifier [11].

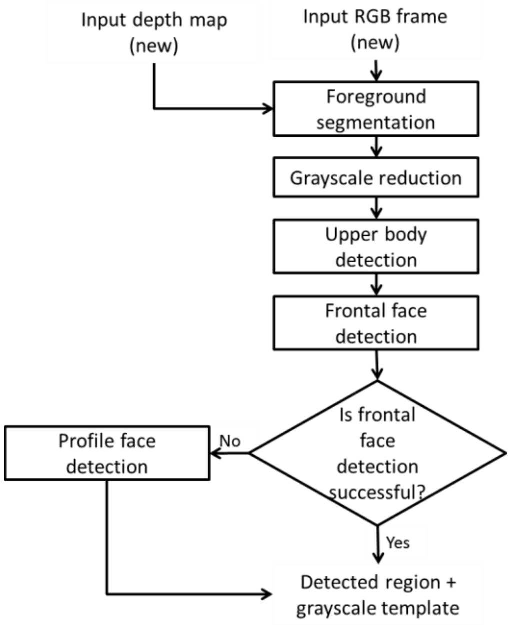
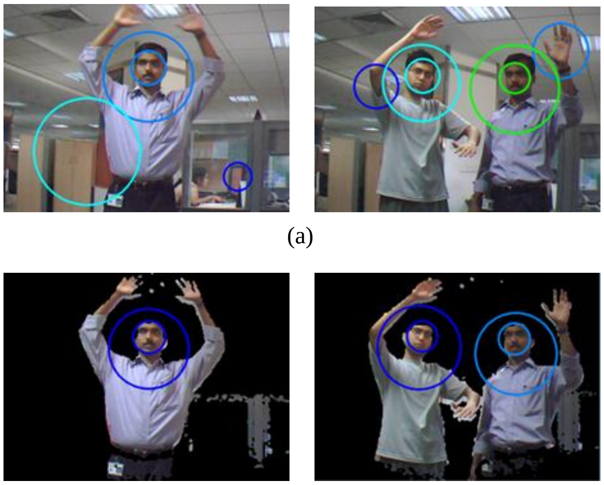


Figure 6: Haar cascade based detection logic.



(a)

(b)

Figure 7: Haar cascade based detection for upper-body and face. Circles circumscribing another circle denote successful face (inner circle) and upper body detection (outer circle) whereas single circle denotes a successful upper-body (either false positive or true positive) detection and unsuccessful face detection (either false negative or true negative). (a) Haar cascade based detection on original grayscaled RGB image. (b) Haar cascade detection on foreground segmented grayscaled RGB image.

regions are then passed on to frontal face detector classifier (see Fig. 6). In case frontal face detection fails to detect any faces, profile face detector [15] is used to detect faces. If either upper body detector or the profile face detector fails to produce any positive results then the frame is completely rejected and the next frame is analyzed for any possible upper-body detections. If no face is detected in the detected upper body region, then it is assumed to be false positive and the detection is rejected for further analysis. This successive detection logic helps in reliably determining the positive detections and pruning out the false positive detections. In order to reduce the computation time as well as the false positives, Haar detection is done on foreground segmented image (see Fig. 7).

2.4. Template matching based tracking

The template-based approach determines the best location by matching an actual image patch against an input image, by “sliding” the patch over the input search image using normalized cross-correlation, defined as:

$$R(x,y) = \frac{\sum_{x',y'}\left(T_{RGB}^{G}{}'(x',y').I_{RGB}^{G}{}'(x+x',y+y')\right)}{\sqrt{\sum_{x',y'}T_{RGB}^{G}{}'(x',y')^2.\sum_{x',y'}I_{RGB}^{G}{}'(x+x',y+y')^2}}$$

where:

$$T_{RGB}^{G}{}'(x,y) = T_{RGB}^{G}(x,y) - \overline{T_{RGB}^{G}}$$

$$I_{RGB}^{G}{}'(x,y) = I_{RGB}^{G}(x,y) - \overline{I_{RGB}^{G}}$$

T_{RGB}^{G} = grayscaled RGB template image

I_{RGB}^{G} = input grayscaled RGB image

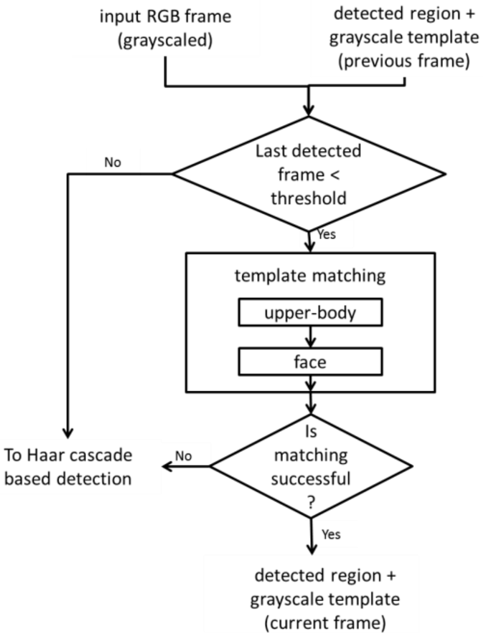

input RGB frame (grayscaled)
detected region + grayscale template (previous frame)
Last detected frame < threshold
No
Yes
template matching
upper-body
face
Is matching successful ?
No
To Haar cascade based detection
Yes
detected region + grayscale template (current frame)

Figure 8: Template Matching based tracking logic.

Since template-based matching requires sampling of a large number of points, we can reduce the number of sampling points by reducing the resolution of the search and template images by the same factor (in our case, down sampled by a factor of 2) and performing the operation on the resultant downsized images. Advantages of using template matching, over Haar cascades, is reduced computation time and higher true positives, since Haar cascades misses variations in object orientation and pose.

The template images/ patches are obtained from the successful detection in the last frame; either by Haar cascade based detection or by template based matching (see Fig. 8). Haar cascade based detection is used only when there are no templates to do matching or when the template matching fails to track the template in the input image. Haar cascade based detection is forced after certain empirically chosen time-lapse/frames (threshold) to handle drifting errors and appearance of new people into the scene. Figure 9 shows examples of template matching on input images.

2.5. Stick human body model

The skeleton model is represented by a vector of 7 body parts (*bp_1* to *bp_7*) as shown in Figure 10. The proportions between the different parts are fixed and were determined based on NASA Anthropometric Source Book [16] and [17] (see fig. 11). Each body part has its own range of

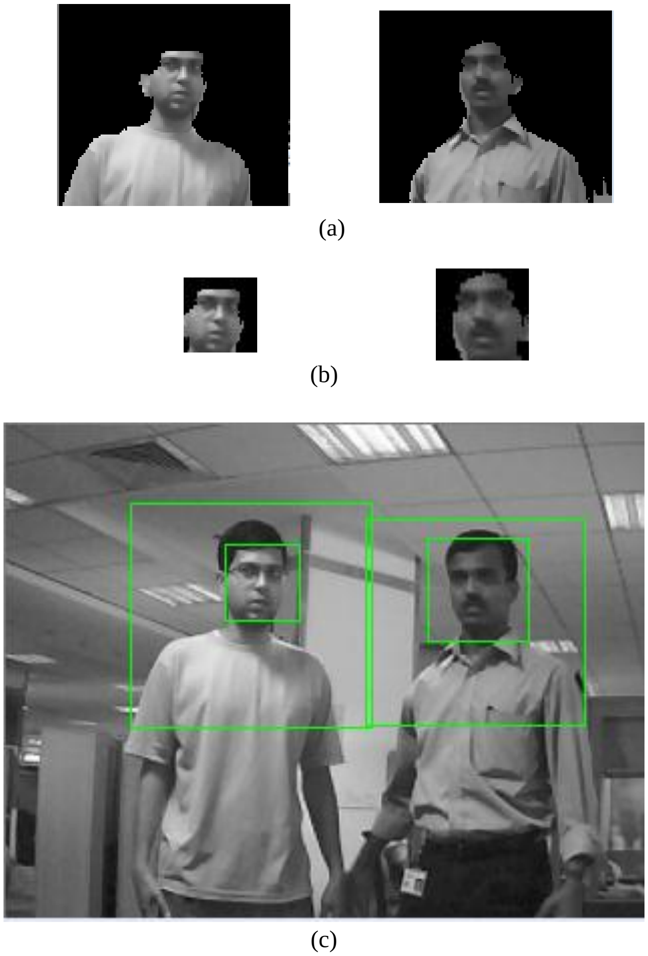


Figure 9: Results for template matching based tracking. Templates are grayscaled and down-sampled to half VGA to reduce computation time. Similarly input RGB image is also grayscaled and down-sampled to half VGA (a) upper-body template identified in previous frame (b) face templates identified in previous frames (c) input image grayscaled and down-sampled with marked rectangular regions denoting successful template based tracking.

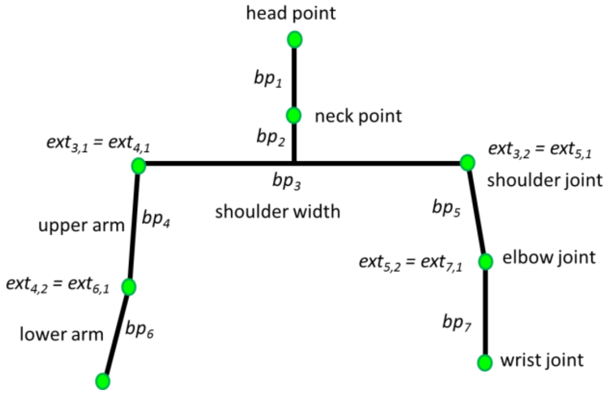


Figure 10: The stick model used for human upper-body skeleton fitting.

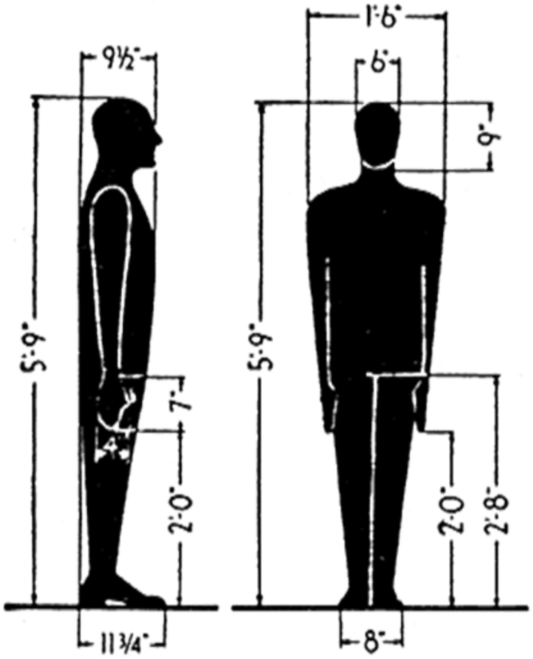


Figure 11: Anthropometric ratios of typical human body [17].

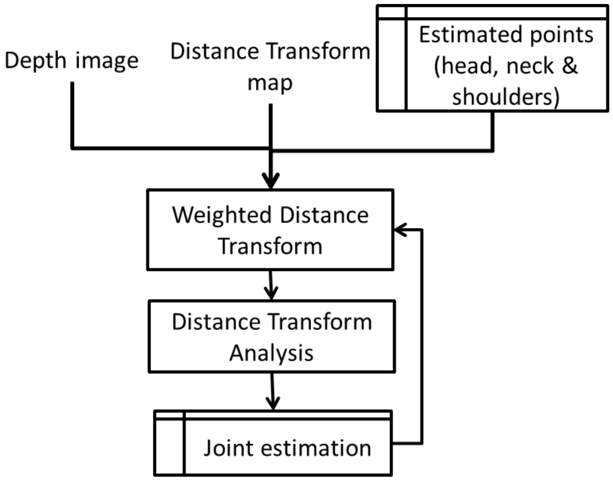

Depth image
Distance Transform map
Estimated points (head, neck & shoulders)
Weighted Distance Transform
Distance Transform Analysis
Joint estimation

Figure 12: Limbs fitting based on linear regression of sampled weighted distance transform map.

possible motion. Each body part is composed of two extremities, representing the coordinates of the body part in the image plane:

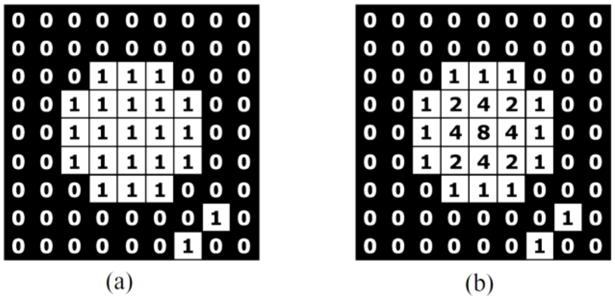
$$bp_i = \{ex_{i,1}, ex_{i,2}\}$$

where,

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

$x_{i,j}$ 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.

The head, neck, shoulder (both left and right) joints are estimated based on detected upper-body and head region.



0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0
0	0	1	1	1	1	1	0	0
0	0	1	1	1	1	1	0	0
0	0	1	1	1	1	1	0	0
0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	1	0	0

(a)

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0
0	0	1	2	4	2	1	0	0
0	0	1	4	8	4	1	0	0
0	0	1	2	4	2	1	0	0
0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	1	0	0

(b)

Figure 13: Numerical example of distance transform. (a) shows an input binary image. In (b) the Euclidean distance of each pixel to the nearest black pixel is shown. The distance values are squared so that only integer values are stored. [21]

The centroid of the detected head template is taken as head point. The shoulder joints are taken as the lower extremities of the detected upper body region in input image. Based on the anthropometric ratios, the neck point is estimated to be at 2/3 of the vertical distance from head to shoulder points. Similarly length of upper arms is taken as 2/3 of shoulder width and 5/9 of shoulder width in case of lower arms.

2.6. Limbs fitting

In order to estimate the remaining joints (elbow and wrist, both left and right) and limb inclinations (upper and lower arm, both left and right), linear regression on sampled weighted-distance transform map (distance transform analysis) is performed (see Fig. 12). Once the elbow joints are estimated, weighted-distance transform w.r.t. these joints are computed for estimating wrist joints and 2D inclinations for lower arms.

The Distance Transform (DT) maps each image pixel into its smallest distance to regions of interest [18].

$$D(p) := \min\{d(p,q) |\, q \in \ O^c\} = \min\{d(p,q)|I_d(q) = 0\}$$

where:

D is distance map of depth image (I_d)

O^c is complement of foreground object

Euclidean metric is used for calculating the DT:

$$d(p,q) = \sqrt{(p_x - q_x)^2 + \left(p_y - q_y\right)^2}$$

Figure 13 shows a numerical example of Euclidean Distance Transform (EDT) [19]. For each pixel in fig. 13(a), the corresponding pixel in the DT of fig. 13(b) holds the smallest Euclidean distance between this pixel

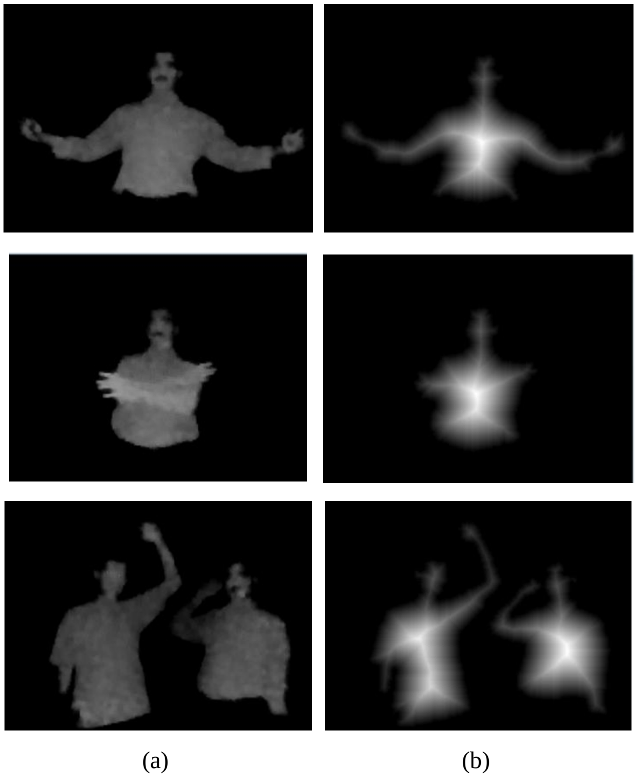


Figure 14: EDT on foreground segmented depthmap normalized to 0 -256 range for visualization (a) foreground segmented depthmap (b) distance transform map

and all other black pixels. The squared Euclidean distance is used for saving storage: since the pixel coordinates are integers, the square of the Euclidean distance $d^2(p,q)$ is also an integer. Figure 14 shows some examples of EDT on input images.

Since limb movements for human body can be out of image plane which EDT fails to capture in the depthmap. In order to take into account the projected lengths of the limbs weighted-distance transform is calculated. The distance map of the image is multiplied with variance factor representing the variance ratio of the point w.r.t to the reference point (parent joint) in the direction orthogonal to the image plane. This variance can easily be calculated from the input depthmap.

The weighted –distance transform $D^w(p,c)$ for point p w.r.t. c in depth image (I_d) is defined as:

$$D^w(p,c) = D(p).\left(1 + \frac{|I_d(p) - I_d(c)|}{I_d(c)}\right) \forall\, I_d(c) \neq 1$$

where:

$D(p)$ is DT value at point p in image I_d.

I_d is input depth map

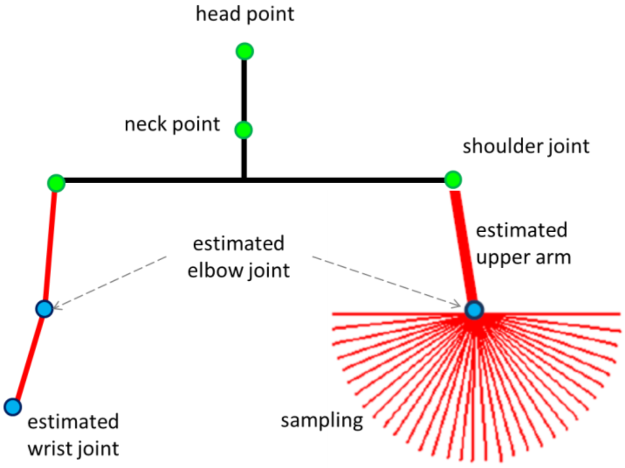


Figure 15: Sampling of weighted distance transform map for left lower arm pose estimation. The green color points have already been estimated based on upper body and head region detection. The blue colored joints are estimated by sampling followed by linear regression.

c is the reference point (parent joint) for estimating the angles for upper and lower arms. e.g. for estimating the inclination of upper left arm, reference point (c) is left shoulder joint and similarly for estimating the lower right arm, reference point (c) is right elbow joint. Sampling of the Weighted-Distance Transform map is done upto length *l* from the reference point (parent joint) c in an angular region varying from 0 to 2π, and with a predefined sampling angle. Temporal information can be incorporated to improve computational efficiency by imposing range constraints on the angular region for sampling the map (see fig. 15). The length *l* of arms is estimated based on anthropometric ratios as discussed in section 2.5. The step size for sampling angle influences the robustness and speed of the technique. If it's too large, a good solution could be overlooked. However, the whole process might take too long if the step size is chosen small. It then becomes possible to sample points along and for each candidate solution. In estimation of both upper arms and lower arms, a second global maximum is taken as the estimated pose of the limb. In case of upper arms, the global maxima always denotes the angle from left or right shoulder joint towards torso's center region; since weighted-distance transform map value is always maxima along this path (see Fig. 14). Similarly for lower arms, a global maximum denotes the angle connecting the elbow joints to shoulder joints, since due to physical structure of human body upper arms are broader in width compared to lower arms. Due to these reasons second maxima is universally chosen to represent the estimated limb’s inclination.

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 fitting process becomes more time-consuming. 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, while the other limbs will be successfully fitted if the upper body detection is successful. In the case of a global method, a small error can lead to the failure of the whole fitting module. However, because of the local fitting method, even if one part is missed, the overall fitting is often acceptable.

3. Experimental results

We have developed a working prototype of our human detection and pose estimation logic. The prototype was implemented using C/C++ and OpenCV library, on a windows platform. The prototype works in real-time using live feeds from 3DV camera mounted on top of a personal computer. We have tested the above prototype for single as well as multiple (upto 3) non-intersecting people with appearance and disappearance of people at random and for various different upper body poses. The input RGB stream is of 640x480 resolution at 30fps (VGA) and the depth stream is of 320x240 resolution at 30fps (half-VGA). For foreground segmentation, blob with size less than 400 pixels (empirically chosen) are considered as non-humans. Haar cascade based detection is done on full-VGA size grayscaled RGB image to increase true positive detections. Template matching based tracking is done on half-VGA size grayscaled RGB image to reduce computation time. Threshold used for enforcing Haar cascade based detection is empirically chosen as 15 frames. Since foreground segmentation is the most critical step in pose estimation, poor foreground segmentation can sometimes lead to incorrect pose estimation. Figure 16 shows a few examples of our analysis done on input frames of humans interacting in various poses. Table 1 gives the time taken (on a machine with Intel Core 2 Extreme processor, 3 GHz & 3GB RAM) for various processes in the prototype. The average running time of the entire process is less than the total time used for detection (~68 ms/frame) since Haar cascade based detection is enforced only once in every 15 frames while for the rest of the frames, template matching based tracking (~ 8ms/frame) is used.

Table 1: Computational time for various modules in our system.

Modules	Time/frame (in ms)
Haar cascade based upper-body & face detection	~ 57 ms/frame
Skeleton fitting	~ 11 ms/frame
Total time using detection	**~ 68ms/frame**
Template matching based tracking	~ 3 ms/frame
Skeleton fitting	~ 5 ms/frame
Total time using tracking	**~ 8 ms/frame**
Average Running Time*	**~14 ms/frame**

* Threshold = 15frames/sec

4. Conclusion

In this paper, we have presented a viable vision-based human pose estimation technique using RGB and depth streams from a monocular view. An articulated graphical human model is created for pose estimation of upper-body parts for HCI applications. Our technique uses a balance of Haar cascade based detection and template matching based tracking. Haar based detection handles appearance of humans and drifting errors in tracking, while template matching based tracking is able to handle variations in object pose and makes the approach computationally light. Our approach uses anthropometric statistics to estimate the pose and also guides the estimation process of the model. The limbs of the model are fitted individually by generating all possible positions and selecting the best position. This technique is performed progressively, one limb at a time, instead of globally. This way, the process is faster and robust. We have demonstrated the technique for various real world input data. Some improvements are possible in this framework. Incorporating skin detection and edge detection would reduce false positive configurations for lower arms. Occlusion handling and comparative studies with published work form nice scope of work in the future.

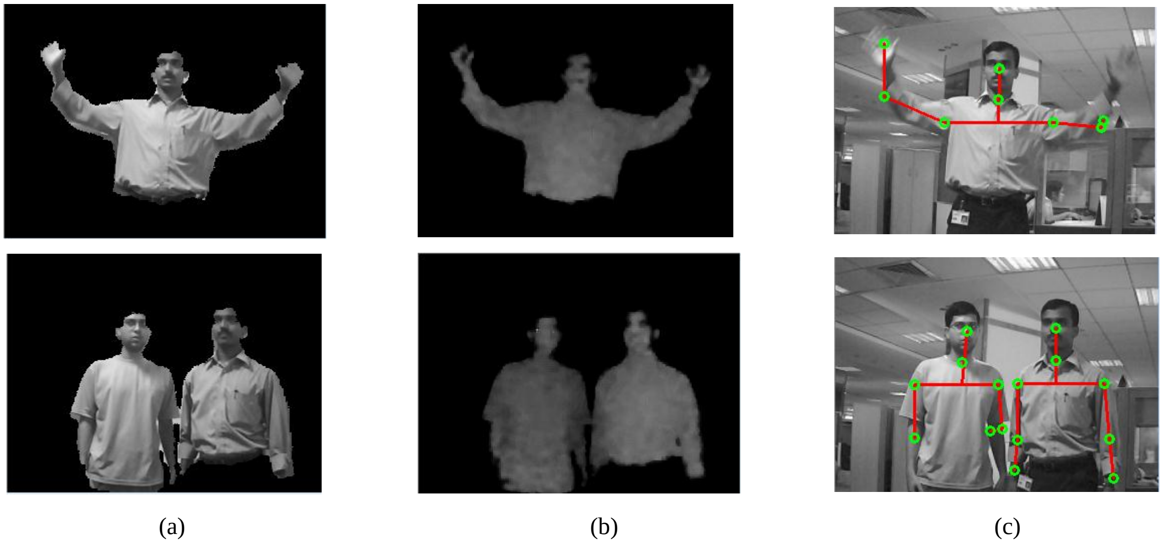


Figure 16: (a) Foreground segmented grayscaled RGB image (b) input depthmap (c) Estimated upper body human stick figure overlaid upon the grayscaled RGB image.

References

[1] J. K. Aggarwal and Q. Cai. “Human motion analysis: A review”, In Proceedings of IEEE Computer Society Workshop on Motion of Non-Rigid and Articulated Objects, Puerto Rico, pp. 90–102, 1997.

[2] Thomas B. Moeslund, Adrian Hilton, and Volker Krüger. “A survey of advances in vision-based human motion capture and analysis”, Computer Vision and Image Understanding (CVIU), vol. 104, no. 2, pp. 90–126, 2006.

[3] 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, pp. 77–83, 2000.

[4] Carlos Barr´on and Ioannis A. Kakadiaris. “Estimating anthropometry and pose from a single uncalibrated image”. Computer Vision and Image Understanding (CVIU), vol. 81, no. 3, pp. 269–284, 2001.

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

[6] 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, vol. 2, pp. 477–486, 2000.

[7] Y. Guo, G. Xu and S. Tsuji. “Tracking Human Body Motion Based on a Stick Figure Model”, Journal of Visual Communication and Image Representation, vol. 5, no. 1, pp. 1-9, March 1994.

[8] 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, pp. 222–227, 1998.

[9] H. Fujiyoshi and A. Lipton. “Real-time human motion analysis by image skeletonization”, In Proceedings of the 4th IEEE Workshop on Applications of Computer Vision, pp. 15–21, 1998.

[10] P. Viola and M. J. Jones. “Rapid Object Detection Using a Boosted Cascade of Simple Features”, IEEE CVPR (2001).

[11] R. Lienhart and J. Maydt. “An Extended Set of Haar-like Features for Rapid Object Detection”, IEEE ICIP, vol. 1, pp. 900–903, 2002.

[12] G. Johansson. “Visual motion perception”, Scientific American, vol. 232, no. 6, pp. 76-88, 1975.

[13] Hannes Kruppa, Modesto Castrillon Santana and Bernt Schiele, “Fast and Robust Face Finding via Local Context”, Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (2003).

[14] Castrillon Santana, M. and Deniz Suarez, O. and Hernandez Tejera, M. and Guerra Artal, C., “ENCARA2: Real-time Detection of Multiple Faces at Different Resolutions in Video Streams”, Journal of Visual Communication and Image Representation, 2007, vol. 18, no. 2, pp. 130-140.

[15] D. Bradley (2003), “Profile face detection” http://opencv.willowgarage.com.

[16] NASA. Anthropometric Source Book, vol. 2, Springfield VA, Johnson Space Center, Houston, TX, 1978.

[17] 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.

[18] A. Rosenfeld and J. Pfaltz, “Distance function on digital pictures”, Pattern Recognition vol. 1, no. 1, pp. 33-61 (1968).

[19] Gunilla Borgelors, “Distance Transformations in Digital Images”, Computer Vision, Graphics and Image Processing 34, pp. 344-371 (1986).

[20] R. F. Rashid, “Towards a system for the interpretation of moving light display”, *IEEE Trans. PAMI*, vol. 2, no. 6, pp. 574–581, November 1980.

[21] R. Fabbri, L. F. Costa, J. C. Torelli and O. M. Bruno, “2D Euclidean distance transform algorithms: A comparative survey”, ACM Computing Survey, vol. 40, no. 1, pp. 1-44, February 2008.