Real-time Upper Body 3D Pose Estimation from a Single Uncalibrated Camera

Antonio S. Micilotta Eng Jon Ong Richard Bowden

CVSSP, University of Surrey, Guildford, UK

Abstract

This paper outlines a method of estimating the 3D pose of the upper human body from a single uncalibrated camera. The objective application lies in 3D Human Computer Interaction where hand depth information offers extended functionality when interacting with a 3D virtual environment, but it is equally suitable to animation and motion capture. A database of 3D body configurations is built from a variety of human movements using motion capture data. A hierarchical structure consisting of three subsidiary databases, namely the frontal-view Hand Position (top-level), Silhouette and Edge Map Databases, are pre-extracted from the 3D body configuration database. Using this hierarchy, subsets of the subsidiary databases are then matched to the subject in real-time. The examples of the subsidiary databases that yield the highest matching score are used to extract the corresponding 3D configuration from the motion capture data, thereby estimating the upper body 3D pose.

Categories and Subject Descriptors (according to ACM CCS): I.4.8 [Image Processing and Computer Vision]: Scene Analysis

1. Introduction

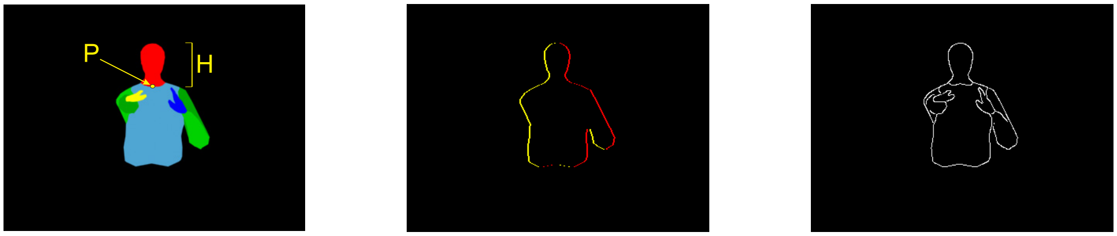
Human animation can be done laboriously via key framing or via motion capture which can be expensive. The ability to animate directly from video would be a beneficial tool with applications in many areas such as 3D broadcasting, games, HCI and animation.

Statistical methods of reconstructing the 3D pose from a monocular sequence track multiple body points and compute prior probabilities of 3D motions with the aid of training data [BM00, HLF00]. Sidenbladh [Sid01] employed strong motion priors in a particle filter framework to overcome visual ambiguity and presented a tracked walking human in a monocular image sequence. The matching of shape and edge templates has also received attention in hand pose estimation [STTC04] where shape matching follows a cascaded approach to reduce the number of edge template comparisons. We apply a similar method to reconstruct the upper human body, and use hand positions to initially extract corresponding silhouettes.

2. Data acquisition

Using a 3D graphics package, a skeleton is skinned with a generic human mesh (Figure 4 (b)) to resemble a person wearing loose fitting clothing. The mesh material is assigned an ‘Ink ’n Paint’ material with one level of colour so that the rendered model has a clean ‘cell shaded’ effect. A rendered model with one colour level resembles a simple silhouette as the outline of the arms is not visible when moving in front of the torso. We therefore colour the respective body parts independently to preserve these edges. The head body part extends from the top of the head to the bottom of the neck, and is comparable to the visible upper body skin tone of the user (from the hairline to the collar of the shirt). The left and right hands are coloured blue and yellow respectively, thereby providing independent labelling. The material from the waist down is transparent and the rendered model therefore consists of a multi-coloured upper body against a black background (see Figure 1 (a)).

A single target camera (a camera whereby the camera-to-target distance remains fixed) is then attached to the chest bone of the skeleton, and is allowed to roll in accordance with it. The skeleton is then animated using a variety of mo-



(a) (b) (c)

Figure 1: *(a) Frontal 2D representation of 3D model (b) Boundary image (c) Edge map*

tion capture data to produce a database of 3D body configurations. This sequence, consisting of 5000 frames, is rendered from this camera view, and yields a database of 2D frontal view images (Frontal View Database) of an upright upper body that has a fixed scale, and is centred at position *P* (Figure 1 (a)).

2.1. Subsidiary datasets

The images of the Frontal View Database are then used to produce a hierarchy of three subsidiary databases. These are computed off-line, and are loaded when the application is executed. From parent down:

1. **Hand Position Database.** This consists of the 2D positions of the left and right hands that are obtained by determining the centroid of the blue and yellow (hand) regions of each frame.
2. **Silhouette Database.** This is easy to create as the background of each example is black. However, due to the size of the dataset, storing a silhouette image for each frame is unrealistic as the entire dataset occupies several Gigabytes in raw format. It is more efficient to represent each silhouette image in terms of its boundary, as shown in Figure 1 (b) and is stored as entry and exit pairs for each row of the silhouette. This representation not only minimises RAM requirements, but offers a fast and efficient method of comparison to the input silhouette, which is represented as an integral image (see Section 3.5).
3. **Edge Map Database.** Conducting an edge detection on the cell shaded and multi-coloured model provides clean edge images (Figure 1 (c)). Again, to conserve RAM, only the edge locations are stored.

All examples in these databases are indexed according to the Frontal View Database, and hence the 3D body configuration database that generated it.

3. Model matching

The sections below discuss the processes that occur at runtime, after the subsidiary databases have been loaded.

3.1. Background suppression

In this paper, the *input image* refers to the image captured from the camera at run time, and consists of a subject (or *user*) facing the camera with a cluttered background. Segmenting the user from the input image plays an important role in tracking the various body parts, and in matching a 3D model. A simple solution would be to use a blue screen background where chroma keying can be performed. However, such a controlled environment is limiting, and we therefore make use of a background suppression algorithm that can isolate a user from a *cluttered* background. Our algorithm was originally developed for exterior visual surveillance and relies upon modelling the colour distribution with a Gaussian mixture model on a per pixel basis. This model is learned in an online fashion using an iterative approximation to expectation maximisation – once the background has been learned, sudden changes in pixel intensity are associated with foreground movement. Background is represented by ‘0’, and foreground by ‘1’.

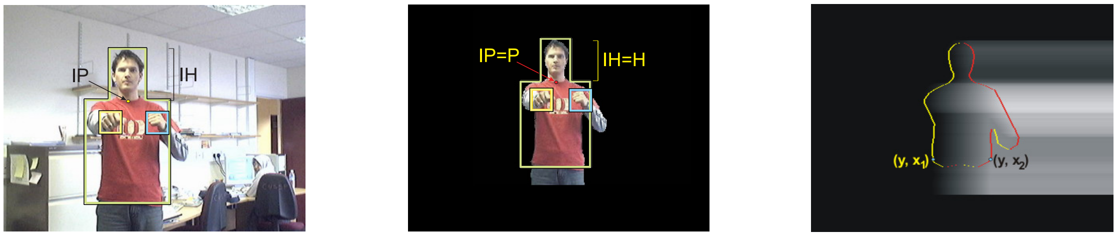
3.2. Tracking the user

In order for the entire system to run in real-time, we require a robust method to track the user’s torso, face and hands. Using the segmented image, we make use of a robust tracking algorithm that uses a coarse estimate to body shape to track the torso, and learns a user-specific skin model to track the face and the hands (see Figure 2 (a)). The reader is directed to [MB04] for full implementation details.

3.3. Input image adjustment

Referring to an example of the Frontal View Database (Figure 1 (a)), the length from the top of the head to the neckline *H*, is constant across all examples, and is used as the reference point with which to scale the input image. Position *P* and length *H* are pre-computed.

Comparing the Frontal View Database and its subsidiaries to the input image requires that the input image foreground exists in same spatial domain (see Figure 2 (b)). To do this, the input image neck centre *IP* and head length *IH* must be determined. The tracking system of Section 3.2 provides



(a) (b) (c)

Figure 2: *(a) Input image (b) Adjusted input image (c) Integral image / boundary overlap*

the positions and dimensions of the torso and hands. IP is approximated to be the same as the shoulder height, and IH is therefore the length from the top of the head to IP.

The scale factor is determined by $S = IH/H$, and the offset from P to IP is determined by $offset = P - IP/S$. The input image is scaled and translated in a single pass, creating the *adjusted input image* $(AdjIm)$ of Figure 2 (b):

$$\forall x,y \; AdjIm(x,y) = inputImage(x,y)/S + offset; \qquad (1)$$

We then extract an adjusted input silhouette IS and edge map from this adjusted input image.

3.4. Extracting subsidiary database examples

Before conducting silhouette matching, we initially extract a subset of the Silhouette Database by considering the user's hand positions. Using the left and right hand bounding boxes provided by the tracking algorithm as reference, we search through the Hand Position Database for hand positions that are simultaneously contained by these bounding boxes, and extract the corresponding examples from the Silhouette Database. It is likely that several possible examples will be identified; a matching score is therefore calculated for each example as per Section 3.5.

3.5. Silhouette matching using integral images

We determine a set of matching scores for the Silhouette Database subset by computing the percentage pixel overlap between the IS and each example. A crude method would be to reconstruct a silhouette image from the boundary database, and to perform a comparison on a per pixel basis. This is prohibitive as each example silhouette contains approximately 15 000 pixels – computing this multiple times would clearly limit real-time performance. The matching procedure is made more efficient by using an intermediate representation of the input silhouette IS, called an integral image II.

The II encodes the shape of the object by computing the summation of pixels on a row by row basis. The value of the $II(x,y)$ equals the sum of all the non-zero pixels to the left of, and including $IS(x,y)$:

$$II(x,y) = \int_{i=0}^{x} IS(i,y)\mathrm{d}i \qquad (2)$$

The entire II can be computed in this manner for all (x,y), however for efficiency we compute this incrementally:

$$\forall x,y \; II(x,y) = IS(x,y) + II(x-1,y) \qquad (3)$$

Figure 2 (c) offers a visualisation of the II of the IS (extracted from Figure 2 (b)), with a silhouette boundary example of the Silhouette Database superimposed. Referring to Figure 2 (c), the number of pixels between boundary pair (y,x_1) to (y,x_2) is computed as $N_B(y) = x_2 - x_1 + 1$. The number of pixels of the input silhouette for the corresponding range is therefore computed as $N_{IS}(y) = II(y,x_2) - II(y,x_1) + 1$. $\sum N_B$ and $\sum N_{IS}$ are computed for all boundary pairs, and the matching score is therefore computed as $S = \sum N_{IS} / \sum N_B$. This score is computed in a few hundred operations; considerably less than tens of thousands of pixel-pixel comparisons.

Once matching scores are computed for the examples of the Silhouette Database subset, the top 10% are used to extract a subset of the Edge Map Database.

3.6. Chamfer matching and final selection

Poses with the arms directly in front of the body produce similar silhouettes, and we therefore also consider the edge information to resolve ambiguities. Having extracted a subset of the Edge Map Database, we then compare each of these edge maps to that of the input image to compute a second matching score.

As humans vary in physique, it is unlikely that the edges of the input and the examples will overlap exactly. We therefore apply a distance transform [FH04] to the input edge image (Figure 3 (a)) to 'blur' the edges (Figure 3 (b)). The distance transform specifies the distance of each pixel to the

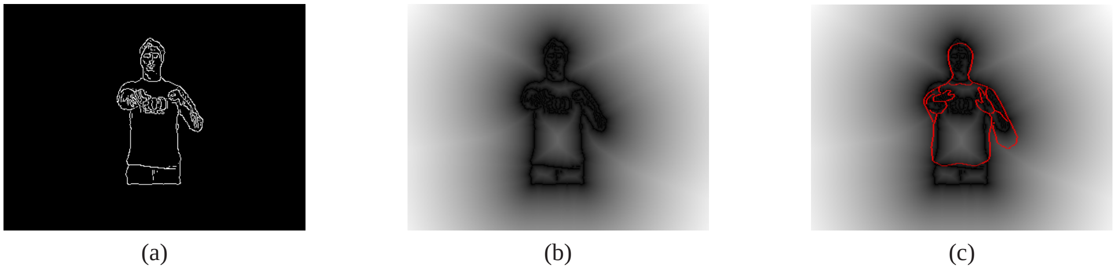


Figure 3: *(a) Edge image (b) Distance image (c) Chamfer matching*

nearest non-zero edge – the darker the pixel, the closer it is to an edge.

We then superimpose the example edge map on the distance image, and determine the *edge distance* – the mean of the distance image pixel values that co-occur with example edge maps. The example that yields the shortest distance represents the best match, and is used to access the 3D body configuration from the original database. This method of matching edge images is referred to as Chamfer matching [BTBW77].

4. Results

Figure 4 (a) shows a tracked subject in various scenes. A representative CG model, corresponding to the best silhouette and edge match, is shown in Figure 4 (b). The model illustrated here is that used for the example database and can be easily replaced with another model. The system runs at 16 frames/sec and is invariant to the user's scale and position.

5. Conclusion

We have been successful in matching a corresponding 3D model to a subject. The 3D hand positions can be extracted for HCI, or the CG model itself could be used for animation purposes. Matching by example does however require a large example dataset, and we have therefore stored our datasets in their simplest forms. Not only can these simple representations be accessed quickly, but they also contribute to the fast matching methods employed. Furthermore, the hierarchical structure restricts analysis to subsets of the subsidiary databases, thereby contributing to the real-time aspect of the approach.

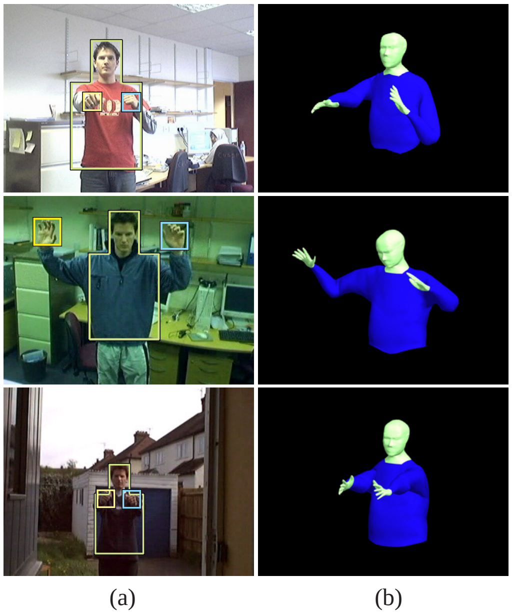


Figure 4: *Frontal pose with corresponding 3D model*

References

[BM00] Bowden R., Mitchell T.: Non-linear statistical models for the 3d reconstr. of human pose. In *Image and Vision Computing* (2000), vol. 18, pp. 729–737.

[BTBW77] Barrow H., Tenenbaum J., Bolles R., Wolf H.: Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proc. of Joint Conf. AI* (1977), pp. 659–663.

[FH04] Felzenszwalb P., Hurrenlocher D.: *Distance Transforms of Sampled Functions*. Tech. Rep. TR2004-1963, Cornell Computing, 2004.

[HLF00] Howe N., Leventon M., Freeman W.: Bayesian reconstruction of 3d human motion from single camera video. In *NIPS* (2000), vol. 12, pp. 820–826.

[MB04] Micilotta A., Bowden R.: View-based location and tracking of body parts for visual interaction. In *Proc. of BMVC* (2004), vol. 2, pp. 849–858.

[Sid01] Sidenbladh H.: *Probabilistic Tracking and Reconstruction of 3D Human Motion*. PhD thesis, Royal Institute of Technology, CVAPL, Nov 2001.

[STTC04] Stenger B., Thayananthan A., Torr P., Cipolla R.: Hand pose estimation using hierarchical detection. In *Workshop on HCI* (2004), pp. 105–116.