Fast Human Body Motion Tracking Using 2-D Deformable Templates

George Hoffman ⟨*geh7@cornell.edu*⟩

May 16, 1997

Abstract

The paper presents a fast, two dimensional method of obtaining a model of human body motion. The method is based on a deformable template; the construction of this template is discussed and a minimization procedure is developed. Finally, the results of an implementation of the method are presented.

1 Justification

The data space which desktop computers manipulate continues to grow at an alarming rate. The average entry level desktop computer now has the ability not only to move large chunks of data quickly, but also process richer data, such 3D data sets and animation, in real time. However, the methods used to input and manpulate this data for the most part remain pathetically simple. The complexity of our interaction with this expanding data space is also increasing, but the pipes through which that interaction must take place are far too narrow. Users are left to manipulate and explore complex 3D environments using "point and click" interfaces and key commands, as mice, trackballs, and keyboards are yet the mainstay of computer input.

More complex input systems exists, such as trackers which use ultraviolet or electromagnetic fields to convey 3D position and orientation, and "datagloves" which convey hand position and gesture. However, aside from the fact that these systems are still expensive and not in widespread use, they have other disadvantages. For instance, such systems usually require complex set-up and are not easily scalable with the amount of data input, involving sensors placed on all points of interest, and perhaps multiple stationary receivers. Some have technical problems; for instance, EM systems can be fooled by the presence of extra metals in the region of question. More importantly, however, is that in a sense these systems are still highly limited as to the amount of input they allow the computer access to. For one thing, only the points of interest are tracked. No information about the surrounding environment is available. Such highly sparse input is of great use for specialized tracking applications, but of questionable worth for general purpose human-computer interaction. It is desirable to have a general purpose method of receiving and interpreting highly dense data. As humans, we have this in our visual system.

There are many uses for computer vision in human-computer interaction. As regards human body motion specifically, among other things, it would be useful to allow a computer to record the motions acted out by a body, in order to edit them and apply them to a model which could then be animated using these motions. Such activities are carried out now by highly specialized input systems designed for this specific purpose. A vision-based system could use the same general-purpose input for these specialized purposes which it uses for all other input.

2 Introduction

The analysis of human body motion has three stages. In the first stage, sometimes referred to as *segmentation*, the input image is decomposed into "meaningful" parts. It is not neccessary for the segmentation process to know which of these meaningful parts corresponds to which body parts, or even neccessarily for the segmentation process to be customized for human body motion. Usually, segmentation involves analysis of motion data using the difference picture or a similar method (citation). This initial stage isolates the "interesting" parts of the picture (in this case, the body itself) from the background.

The second stage is known as *labelling* or *structure analysis*, and involves deciding which portions of the segmented image correspond to particular body parts. The two names for this stage imply two slightly different approaches; a labelling process decides which edges belong to which body parts, while structure analysis implies that the actual orientation of these parts is decided. Thus, labelling tends to be both easier and less useful, although labelling might conceivably be used as a first pass in structure analysis. Structure analysis usually involves fitting a model to the data; in the case of human body motion, the model is often three dimensional, with many parameters, and the fitting procedure is very time consuming.

The third stage is the *recognition* stage, in which the time-varying posture of the fitted model is analyzed for qualitative data: is the body running, jumping, swinging, or climbing? Although not needed in all cases, this higher level data can be useful for later reference by the segmentation and structure analysis stages, which can use the qualitative posture recognized by this stage as a hint for subsequent frame analysis.

This paper focuses on the second stage, structure analysis. It presents an attempt to perform this stage in real-time using a 2-D deformable template of the body.

3 Methodology

Most human body structure analysis approaches are divisible into what I'll call *source based methods* and *destination based methods*. In source based methods, elements of the (segmented) input image are analyzed to see if they fit some constraints with relation to *a priori* knowledge and with respect to other analyzed elements. An approach of this type usually does most of its work by processing the source image to identify regions of interest and evolving the image towards something resembling its idea of the human body. Source based methods are commonly region or pixel-based methods, which decide respectively which regions or pixels belong to which body parts. Destination based methods, by contrast, work by altering expectations about the body pictured in the source image until the expectations match up with the original source image. These are usually model-based approaches, and work by fitting a model of the body to the input. The differences between what the model predicts the source image to be and what it is in reality are minimized by modifying the parameters of a parametrized model.

Much work has been done using both of these approaches. Destination based model fitting approaches are attractive, however, for several reasons. First, it is easy to express the knowledge we have about the model in terms of constraints placed on the model's parameters, as well as constraints implicit in the model's construction. Because human body motion analysis is heavily knowledge based, our a priori knowledge about the human form and it's range of motion are can be easily implicitly detailed by the model. Secondly, the construction of an analysis procedure is reduced to two well defined entities: the body model, and the minimization process.

The body model is a specialized case of a *deformable template*. A deformable template consists of a set of parameters which together define an exact repesentation of the feature in question, and a set of error terms which define an error function which compares the ideal feature defined by the template and the actual features as seen in the source image.

The parameters may define the feature in any way at all; some methods linearize features and use simple sets of points, but more ambitious methods use parametrized splines to get more exact fits, with the requisite additional computational work required to compare to the source image. Based on its parameters, the template defines certain qualities which it expects to find in the source image. For instance, the template may predict that between two points there should be an edge. The error term for this particular feature might be the Hausdorf distance between the pixels on a line between those two points and the precomputed edge pixels for the source image. The sum of these error terms define the error function, which is a function of all of the template's parameters, and this is what must be minimized.

The minimization process works by playing with the parameters of the template and trying to find the global minimum of the error function. The nice thing about the minimization procedure is that it can be stone cold dumb and, if the model is defined well, still work perfectly. The dumbest possible minimization procedure simply explores the total solution space. This is, in a word, non-optimal. However, it does allow a nice way to guarentee the correctness of the model, by allowing verification that the template does indeed experience its global minimum when it matches the feature in the source image. In practice, of course, performance dictates that the minimization procedure also have some knowledge of the model. For instance, one obvious optimization is to find the local minimum within some range of values. In vision, where points of interest move only slightly from frame to frame, the parameters acheiving the global minimum of frame $n + 1$ are bounded by some delta around the parameters acheiving the minimum for frame n. Other optimizations usually involve higher level processing, to predict a frame's minimum based on the motion patterns observed in previous frames.

Because the human body is complex, it has some very complex constraints. It is thus useful to have a simple way to express these constraints, and a deformable template provides that. In addition, some very impressive results have issued from the use of templates. Finally, the use of a deformable template is a very scalable method. Faster execution time can be obtained by simplifying the model, or limiting the minimization process, and if more time is available, that time can always be gainfully employed by a more thorough minimization.

Having chosen a method, it now remains to choose a body model and a minimization procedure.

3.1 A Human Body Template

The body is a difficult thing to model efficiently. For one thing, bodies are so heterogeneous; no one body is the same size and shape or has exactly the same proportions as any other. For another, bodies have an enormous range of motion and many degrees of freedom between body parts. For these reasons, attempts to construct models of the body which are useful in the general case have often ended up with complex three dimensional parametrizations, complete with physical models of joints and muscles and constraints involving mass, momentum and gravity.

Needless to say, these methods are not quite real-time. The vast number of parameters makes the solution space enormous, and it is not uncommon to spend many hours finding an adequate model for the motion in a thirty second video clip. Although the results obtained have been shown to be very good, such brute-force, non-real-time approaches show no sign of becoming applicable to interactive human-computer interface design any time soon. To satisfy the needs of these applications, a much faster method is needed.

To reduce the solution space (and thus the minimization time), the first thing we can do is to flatten the model. For many applications, such as animation capture or tracking a figure with the camera, a three dimensional model is not neccessarily needed. For other more demanding applications which require three dimensional information, it is still interesting to investigate obtaining a template fit quickly with two dimensional methods and then extrapolating 3D information from that. A two dimensional model has fewer parameters and a less expensive error function, and as a result is much faster than its 3D counterpart.

For ease of experimental design, we can impose some limits on the motion of the body. In this case, we will require that the body is upright, and facing the camera. While these may seem like strict requirements, keep in mind that many such templates could be at the analyzer's command, and higher-level procedures could decide which templates to match to the source image, based on motion history.

Next we make the assumption that the input image is reasonably uncluttered. This is by no means an unreasonable assumption, as separation of the interesting (i.e. moving) regions of the image from the background is dealt with by many approaches to the segmentation stage. In this way, edge pixels (and perhaps other primitives) which belong solely to the moving body can be analyzed as if existing by themselves in an uncluttered scene. Therefore, our assumption is either of an uncluttered source image or of a segmentation preprocessing stage which delivers to us an uncluttered input image.

Finally, we assume smoothness of motion over time. This is as relevant to defining the template as it is to the minimization procedure.

Given these assumptions, the template definition is reasonably straightforward. It consists of a number of models of individual body parts and a set of constraints which we impose on them and the relationships between them.

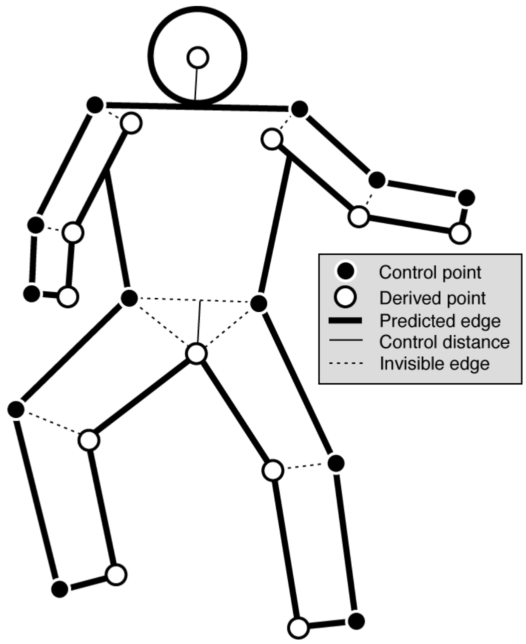

Figure 1: A human body model

3.1.1 Models

The body is represented by a collection of connected polygons. The torso is represented as a simple free form quadralateral, and is defined by four Cartesian points in the global image Cartesian coordinate system, whose origin is arbitrary. I will refer to the line segments the make up the torso as the shoulder line, the waist line, the left side and the right side. Each limb is modelled as a pair of two long quadralaterals which share an edge, and are defined by two sets of polar coordinates each. Each limb is connected to the corresponding corner point of the torso polygon.

The first polar coordinate has as its origin this corner point, and defines the line tracing the upper edge of the limb from the [hip, shoulder] to the [knee, elbow]. The second coordinate is relative to the [knee, elbow] and defines the line tracing the upper edge of the limb from there to the [hand, foot]. These lines form the entire "upper" edge of the limb. The "lower" edge is defined slightly differently depending on whether the limb is a leg or an arm. For arms, the "lower" edge consists of two line segments parallel to the two that make up the "upper" edge, each at a set distance from its opposite which is determined by the dimensions of the torso. The intersection of these line segments form the lower corner of the elbow. The end of the limb is capped with one last expected edge. For legs, the definition is similar, except that an extra point is used to locate the joining of the legs, what I affectionately refer to as the crotch point. The crotch point is defined as being some flexible distance below the waist line on the waist line's perpendicular bisector.

The head is modelled very simply. The head is a circle (the radius of which is determined from the dimensions of the torso) whose center lies on the perpendicular bisector of the shoulder line, and at a distance from that line upwardly bounded by its radius. This does not localize the head particularly well, but it does help a great deal in finding the torso, as will be seen shortly.

The picture expected by the model in a certain state consists of a set of edges and a set of regions which are expected to exibit certain properties, such as internal intensity consistency and intensity similarities between pairs of limbs. To compare this picture with the source picture, for each body part the expected edges are traced and edge pixels are placed, and the square of the Hausdorf distance in the source image at each of these pixels is summed. This sum is then divided by the number of edge pixels, to prevent limb length from having an effect on the picture comparison, and various weights are given to the sums from each particular body part. The sum of the edge comparisons from each body part is added to the total error. Inconsistencies within and between related regions count toward the total error as well.

3.1.2 Constraints

Because the body is so complex, many restraints can be found to limit the solution space of the model, and must be found to make that solution space converge tot he correct values. However, because the body has such a large range of motion, these constraints must also be somewhat loose. Establishing constraints for our model is thus an exercise in brinkmanship.

Some constraints are hard bounds on parameters (usually based upon other parameters) and others are soft bounds, which give an error "penalty" for unexpected states but allow them to exist, providing the rest of the model acheives a low enough error at that state. In addition, there are several "rewards" which subtract from error term, which attempt to "encourage" the model into a certain state, if that is possible while still maintaining a good fit. I shall talk about each of these types of constraints.

3.1.2.1 Soft Bounds

Soft bounds help define our idea of what a feature will "probably" look like.

1. **Torso shape**: The torso is represented as a general quadralateral, but obviously we have a much better idea of what it will look like than this. Therefore, we define constraints for:
   - **Corner angles** are expected to be something close to right angles. For each of the four corner angles θ, we add the weighted error term $sin^2(\theta)$ to penalize for non-right angles.
   - **Height/width ratio** of the torso is expected to be no more than about 1.2. If it is, we weight the square of the difference and add it to our error.
   - **Deformation** of the torso is expected to occur slowly over time. We therefore penalize the torso for any deformation that happens over the course of a frame by weighting and subtracting the sum of the squares of the differences between the length of the four sides of the torso from one frame to the next.
2. **Limbs**: The polar coordinates that represent each limb are expected to vary smoothly with time, as well. We weight and add the error terms

$$\frac{d_{upper} - d_{upper_{last}}}{d_{upper_{last}}}$$

and

$$2\left(\frac{\theta_{upper} - \theta_{upper_{last}}}{\pi}\right)$$

I tried many other soft bounds, but in the end I found that with the proper constants this small set acheived the same goals as a many larger sets I tried.

3.1.2.2 Rewards

The difference between soft bounds and rewards is conceptual; in practice, they are implemented in the same fashion. In general, those terms which ensure a fit close to a real body's shape are labelled as soft bounds, and those that try to encourage the minimization process to search in the right direction are labelled as rewards.

In some cases, we wish to encourage the model to tend towards a given state in the absence of any particular reason why *not* to. For instance, the way we define limbs gives the model no particular reason to ever have a limb longer than a single pixel, or even to have any length at all. Although a limb may remain foreshortened for some time, we would like to give the model incentive to try to "grow" its limbs outward when the error in doing so is low enough, to accurately track limbs which rotate into the plane of view. For this reason, we define a limb length reward which subtracts from the total error of the model for having the limbs extended. There are several such rewards which are subtracted from the total error, each with adjustable weights:

1. **Limb length reward**: $\frac{length_{limb_{total}}}{length_{limb_{max}}}$ for each limb, for the reasons described above.
2. **Crotch reward**: $\frac{distance_{crotch}}{distance_{crotch_{max}}}$, where $distance_{crotch}$ is the distance of the joining of the legs from the waist line; in other words, the height of the pelvis. This is an attempt to prevent the crotch from getting stuck on internal edges[1].
3. **Torso area reward**: This is similar to the limb length reward, in that there is no particular reason the torso should ever have any area, given our model. Thus, we try to encourage the torso model to cover the entirety of the torso by giving it a bonus for every pixel within the bounds of the torso polygon. This reward is kept in check by the internal consistency and exterior edge terms defined by the torso model proper.

3.1.2.3 Hard Bounds Hard bounds are employed to set hard limits on the solution space, and are set to prevent a reward or other factor from extending the model past the physical capabilities of the body. Many of the neccessary hard bounds are implied by the model (for instance, the limbs must originate at particular points on the torso), and so the remaining hard limits are ratio bounds. These ratio bounds use some constants which were experiementally determined; they are not neccessarily optimal, but loose limits are good enough for our purposes.

1. **Limb width**: Arms and legs are expected to be of a certain width, determined by the size of the torso. The values we have used are $width_{arm} = 0.27heighttorso$ and $width_{leg} = 0.43width_{waist}$.
2. **Limb length**: Although we permit foreshortening, nevertheless we can placed some wide bounds on limb lengths. Again, the bounds are calculated as a ratio of torso size, $0.27height_{torso} < length_{arm} < 1.8height_{torso}$ and $0.43width_{waist} < length_{leg} < 2.5height_{torso}$. In addition, each half-limb is allowed to be only half this length.
3. **Pelvis height**: This bounds the distance the crotch can be located from the waist.

   $0.2height_{torso} < distance_{crotch} < 0.35height_{torso}$
4. **Head radius and position**: The head is required to have a radius of $0.25width_{shoulder}$ and the distance of its center above those shoulders is upwardly bounded by that radius and bounded below by zero. This allows for the case in which the head is lowered or bent downward.
5. **Positioning within the image**: For simplicity, each control point is required to exist within the bounds of the image.

3.2 Preprocessing

The source image goes through a number of stages before the template ever sees it. First, a light Gaussian blur is applied. Next, edge pixels are determined and a distance map with the Hausdorf distance at each pixel in the source image is generated. The blurred image and the distance map are actually passed on to the template minimization function.

[1] I hate it when this happens. Ouch!

3.3 Minimization

I spent much time playing with minimization techniques attempting to find a method which explored as small a solution space as possible while still minimizing to the correct positions. Here is the best procedure I have found so far, which is applied for every frame.

1. Find the smallest error given by a particular translation of the whole template within some small radius. Place the template in this state.
2. Find the rotation, within some small angle bound, of the template around its center which minimizes error at this translation. Place the template in this new state.
3. Find the smallest error permitted by some boundedly small individual translations of the four individual points which form the torso quadralateral, and place the template in this new state.
4. Find the optimal $distance_{crotch}$ for this torso configuration, and leave it there.
5. Minimize the error of the legs, by allowing some boundedly small distance and angle changes in both polar coordinates for both legs.
6. Perform the same operation for both arms.

This procedure works nicely. First, we localize the whole body form with overall translation and rotation before localizing details, but those details help in localizing the overall body, since their error terms contribute to the error which is being minimized during these stages.

Secondly, deformation of the torso is handled by step three. Again, arms and legs are brought along for the ride when torso corner points are being found, and so greatly help in finding them by contributing their error terms.

Step four compensates for the torso deformation by refinding the crotch. Then the arms and legs are localized around the central torso.

A better fit could undoubtedly be found by more thorough minimization (for instance, minimizing each of the limbs for each translation or rotation), but my aim was to keep my methods applicable to real-time.

4 Implementation

For an experimental implementation, I have used as a testbed a BeBox with dual 200mhz 603e PowerPC processors runnning the BeOS. Although this is an SMP machine, I did not experiment overly with parallelism in my implementation. For video capture, I used a WinCast/TV card from Hauppauge. This card provides color video at 30 frame per second at up to 320x240. The input images were reduced to greyscale before primary vision processing. All implementation was done in C++.

As implementations of segmentation systems have been done but were not readily available, I did not implement a segmentation stage, but focused on structure analysis. Instead, I recorded sample image sequences of a figure moving against an uncluttered background, to simulate the input of such a segmentation process.

5 Results

My implementation works fairly well within the constraints I have set for it. Figure 2 shows the system tracking a figure against an uncluttered background. The template was calibrated to an initial position on the body and then left to handle itself in subsequent frames. My implementation is far from optimal; although effort was made to reduce order of execution, the error computations were not optimized and so the run-time constants involved were very high. As a result, the system takes several seconds per frame to minimize the template using the given minimization procedure. However, with a more optimal implementation and/or smarter minimization process, the method has definite real-time applicability.

Figure 2: The body template tracking a figure. Every fourth frame of the 64 frame sequence is shown.

Several problems became apparent when the system was implemented. One, internal edges make tracking limbs within the outline of the body nearly impossible. Although assumptions may be made about the homogeneous nature of the torso, these assumptions cannot be depended upon in the general case in which the individual in question might be wearing patterns or other such edge-creating garb. Lighting effects, too, can cause false edges within the body outline. Secondly, problems occur when the arms get close to the body or the legs move closer together; as the gap creates edges which look much like the limb itself. As a result, the gap is sometimes identified erroneously as the limb.

One way to avoid this might be to create a tighter model of the limbs. For instance, early on I thought to avoid this by permitting the template to more tightly fit the real limbs, by allowing variation in limb width along the limb. This actually caused worse results, because the limbs invariably tended to become as thin as possible and converge on one edge of the limb. For this reason, its desirable to disallow the edge of the limb model from matching the same edge pixels that the other side of that limb uses. A more structured preprocessing stage might allow this kind of constraint.

A third problem is that the model does not precisely match the construction of the body. For instance, when a body's arms are raised directly upward, the arm's attachment to the torso can only be roughly approximated by the model I have presented. A more complex model is neccessary to adequately model this, and care would be needed to not increase the size of the solution space unneccesarily.

6 Conclusions

The method discussed is a fast, viable approach to human body motion tracking. Results have been good, but thus far the system is limited as to its general applicability to real-world images. A hybrid method, combining these methods with a segmentation stage and perhaps some preprocessing to determine the location of model primitives (i.e. "ribbons" or similar elements) may have general applicability while providing the advantages of quick minimization and an implicit motion smoothness constraint which methods using such methods often lack.

References

[1] M. K. Leung and Y. H. Yang, "An Empirical Approach to Human Body Motion Analysis", Unpublished, Department of Computational Science, University of Saskatchewan, 1994

[2] M. K. Leung and Y. H. Yang, "Human body motion segmentation in a complex scene", *Pattern Recognition*, 20(1):55-64, 1987

[3] M. K. Leung and Y. H. Yang, "A region based approach for human body motion analysis", *Pattern Recognition*, 20(3):321-339, 1987

[4] W. Long and Y. H. Yang, "Background generation: An alternative to the difference of two images", *Pattern Recognition*, 23(5):1351-1359, 1990

[5] J. O'Rourke and N. I. Badler, "Model-based image analysis of human motion using constraint propogation", *IEEE Transactions On Pattern Recognition and Machine Intelligence*, 2(6):522-536, 1980

[6] R. A. Brooks, "Goal directed edge linking and ribbon finding", *Proceedings of the DARPA Image Understanding Workshop*, 72-76, 1979