Robust Human Motion Estimation Using Detailed Shape and Texture Models

Ales Ude*1+2+3

*1Kawato Dynamic Brain Project, ERATO, JST *2Dept. of Automatics, Biocybernetics and Robotics, Jožef Stefan Institute *3Cyberhuman project, ATR

1. Introduction

There is a vast application field for a low cost vision system that can track and estimate human motion. While specialized optical and magnetic tracking devices are in regular use nowadays, human motion capture from video recordings is still considered to be a very difficult problem.

Although textured polygonal models were used in the past for the estimation of human finger movements [5] and for the head tracking [2] [6], they have not been applied yet to the problem of capturing full-body movements. The main reason is that texture of fingers and faces remains similar over a wide range of persons. On the other hand, textured body models must take into account the person’s clothing and therefore always require an initial calibration phase to ensure the accuracy of the model. Recent progress in the acquisition of textured models [3] makes such models more practicable also for full-body tracking.

2. Models

Our motion estimation framework makes use of a geometric, kinematic and texture model of a human body. The model generation process starts from a generic geometric and kinematic model. These models contain some parameters such as dimensions of body parts and joint positions, which must be modified to fit the observed person’s body. We have developed an automatic calibration procedure to find a kinematic model of a person [7].

The next step is the generation of a textured model. For this purpose we record images of a person from a number of characteristic viewpoints (e.g. front, back, left, right). The person’s posture in each image frame is determined and the geometric model is projected from the calculated posture onto the image plane. Texture from every image, which is first smoothed by a low-pass filter, is mapped onto the visible polygons. In addition, a weight describing the reliability of the assigned texture is estimated. All polygons that are not visible are assigned the weight of 0 and are thus ignored by the tracking process. Multiple texture models computed from images of a person taken from different viewpoints are combined to acquire a more complete texture model as more polygon textures become known. The combined texture is obtained by averaging weighted textures observed from different viewpoints.

3. Human Motion Estimation Using Robust Optimization

The placement of a human body in Cartesian space is determined by the position and orientation of a global body coordinate system rigidly attached to one of the body parts and by the values of the joint angles. We use twist coordinates to model the observed person’s kinematics, but any other kinematic parameterization would result in the same sort of an optimization problem. Let the coordinates of a body point in a local coordinate system of a rigid body part to which it belongs be given by $y_j$. Its 3-D position $\hat{y}_j$ at body configuration $(R, d, \theta_1, \ldots, \theta_n)$ can then be calculated as follows

$$\hat{y}_j = g(R, d) \cdot \exp(\theta_1, \xi_{i_1}) \cdot \ldots \cdot \exp(\theta_{n,x}, \xi_{i_n,x}) \cdot G_x \cdot y_j$$

$$= h_j(R, d, \theta_1, \ldots, \theta_n).$$  (1)

Here exp is a function transforming twists $\xi_i$ into rigid body transformations. $G_x$ is a homogeneous matrix combining the position and orientation of a coordinate system attached to the body part $x$ with respect to the
Ales Ude

558

global body coordinate system at zero configuration, $R$ is a rotation matrix specifying the orientation and $d$ is a vector specifying the position of a global body coordinate system with respect to a camera (world) coordinate system, and $g(R, d)$ denotes the $4 \times 4$ homogeneous matrix corresponding to $R$ and $d$. Note that the set of twists affecting the motion of a body point varies with the identity of the body part which the point belongs to. The camera image of the body point $y_j$ after motion is given by

$$u_j(t) = f(h_j(R(t), d(t), \theta_1(t), \ldots, \theta_n(t))),$$  \hspace{1cm} (2)

where $f$ is a projective, nonlinear camera mapping that can be obtained using a standard camera calibration procedure.

Our task is to estimate kinematic parameters describing the motion of a human body from a video stream. Instead of trying to determine feature correspondences, we rather exploit differential properties of the observed body surfaces to track the person of interest. The first approach that we implemented is based on a brightness constancy assumption which states that color intensity values of a point in a 3-D space remain constant over a short time interval, i.e.

$$I(u_j(t), t) \approx I(u_j(t + \Delta t), t + \Delta t),$$  \hspace{1cm} (3)

where $I(u, t)$ denotes the color intensity of an image point at time $t$ and $u_j(t)$, $u_j(t + \Delta t)$ are the projections of the same body point $y_j$ at two different measurement times as given by (2). Assuming that $R(t + \Delta t) = \Delta R \ast R(t)$, $d(t + \Delta t) = d(t) + \Delta d(t)$, $\theta_i(t + \Delta t) = \theta_i(t) + \Delta \theta_i$, we can estimate the updating step by minimizing

$$\min_{\Delta R, \Delta d, \Delta \theta} \sum_{j=1}^{N} ||I(u_j(t), t) - I(u_j(t + \Delta t), t + \Delta t)||^2.$$  \hspace{1cm} (4)

The point set $\{u_j, y_j\}_{j=1}^{N}$ is given by the integer image pixels for which there exists a body point $y_j$ so that $u_j(t) = f(h_j(R(t), d(t), \theta_1(t), \ldots, \theta_n(t)))$ at the initial configuration $(R(t), d(t), \theta_1(t))$. Such a criterion function was used also in [1] [8].

Before describing our approach to the resolution of the optimization problem (4), we present an alternative formulation of the motion estimation problem. In (4) we only make use of the geometric and the kinematic model of the observed person. Considering also the texture model, we can state the motion estimation problem as

$$\min_{R, d, \theta} \sum_{j=1}^{N} ||T(y_j) - I(u_j(t))||^2,$$  \hspace{1cm} (5)

where $u_j$ is given by (1) and (2) and $T(y_j)$ is a function returning values from the texture model. The point set $\{u_j, y_j\}$ is determined as before. The motivation for replacing (4) by (5) is that errors can accumulate when successively optimizing (4). If the current estimate for the body configuration is not precise, then this is reflected in the point set $\{u_j, y_j\}$. It is difficult to correct such errors in the next step because some textures from the background are assigned to model points. This does not happen when minimizing (5) because in this case the texture is fixed. On the other hand, criterion (5) makes a stronger assumption than criterion (4) in which we assume that color intensity values between two measurement times do not change much; here we assume that color values from the texture model do not change much over the whole measurement period. Although this looks like a pretty strong assumption, it is still approximately fulfilled in many practical cases because lighting conditions normally don’t change abruptly. Also, we map the RGB values into the HSV space and use only hue and saturation values that are less sensitive to variations in lighting conditions.

Nonlinear optimization problems such as (4) and (5) are usually solved by an iterative process (Gauss-Newton, Levenberg-Marquardt iteration) that involves successive calculation of the Jacobian of the vectorized criterion function and the resolution of a system of linear equations. Unfortunately, such an approach does not work in this situation. This happens because the projection of body parts from the initial posture only partially overlaps with these body parts in the actual image. Parts of the surfaces that do not overlap with the real image of the body must be ignored by the optimization procedure because useful information for differential methods can only be found in regions where the projected model image overlaps with the real body image.

A possible way to alleviate this problem is to solve linear systems arising in the Gauss-Newton or Levenberg-Marquardt iteration using a robust optimization technique. Robust optimization methods can ignore points that violate model assumptions. This makes the underlying iteration more reliable while, provided the robust
estimator works well, its convergence properties are not altered. Among the most useful robust estimators are the M-estimators, which were thoroughly studied in the case of linear regression [4]. In our experiments we used the M-estimator defined by the Geman-McClure $\rho$ function

$$\rho(x, \sigma) = \frac{x^2}{\sigma + x^2}. \quad (6)$$

Parameter $\sigma$ is used to ensure convergence towards a global minimum. We start with a larger $\sigma$ and calculate the minimum of the criterion function (4) or (5) using the proposed robust iteration. The new posture estimate is then used to determine a new point set $\{u_j, y_j\}$ and the iteration is restarted using smaller $\sigma$. Such an approach is both robust and has good convergence properties because the region of convexity of the $\rho$ function increases with larger $\sigma$, while the robustness of the estimator increases with smaller $\sigma$.

The presented approach can be extended to handle information from more than one camera view by incorporating measurements from different cameras into criterion functions (4) and (5).

Since we work with complex shape and texture models, we needed to implement rendering in an efficient way. We made use of the OpenGL library to accomplish this goal. This required us to implement some specialized procedures as OpenGL does not provide all information needed by our system directly. For example, we often need to know the identity of a model polygon that is projected onto a pixel in the image. To extract this information, we assigned a unique integer number ID to each polygon in the model. This ID is broken up into three parts arithmetically to generate a unique color value. This color is assigned to the polygon and used in a flat color rendering of the model from the current posture. The color of each pixel is then reversely mapped back and recombined into an integer number which gives the identity of the polygon that the pixel belongs to. The two-way mapping between color values and polygon IDs is implemented via a lookup table and the actual values depend on the capability of the display device. Thus we must render the graphical model of a person twice at each measurement time; once to generate the textured image of a body projected from the current posture and once to calculate the polygon IDs via flat color rendering. Similar problems had to be addressed in [6] in the case of head tracking.

### 4. Experiments

We carried out several experiments to test our system.
Image sequences of moving persons were captured either from a video tape or by a high speed camera. We used both simpler ellipsoidal body models (see Fig. 2), which were "polygonized" to acquire texture models, and more complex polygonal mesh body models (see Fig. 1). Our experience is that more accurate shape models increase the reliability of tracking, especially when the subject is taken from an oblique view (as in Fig. 1). The textured models fulfilled our expectation in the sense that it was possible to recover from partially false postures during tracking. However, the convergence of the optimization procedure was faster when minimizing the criterion function (4), which does not involve the texture model. The reason for this is that the texture from the previous image is a better approximation for the texture in the next image than the fixed texture model obtained by the calibration procedure. It is well known that the Gauss-Newton iteration converges better when the underlying criterion function tends to zero.

An important advantage of the proposed approach is that it is not affected by the complexity of the background. We did not observe any significant differences when tracking people in complex environments such as in Fig. 2, or in a specialized studio shown in Fig. 1. We were able to process both outdoor and indoor sequences. Finally, the proposed approach works also when using low-quality video images. The recreation of the reconstructed motion with a computer graphics agent and with a humanoid robot can be seen in Fig. 3 and 4.

5. Conclusion

We proposed a new approach to human motion estimation from image sequences based on differential properties of the observed surfaces and robust optimization. Our technique can accommodate both the estimation based on the brightness constancy assumption between two consecutive image frames and the matching of the model textures to the image textures. The only conditions for our method to work are that the model textures are piecewise differentiable and that they differ from the background. To our best knowledge, this is the first approach that implements full-body motion estimation in a robust optimization framework and that can make use of textured models. Our method is accurate enough to generate naturally looking computer animations and humanoid robot motions.

In the future we plan to extend our approach to allow tracking in the presence of varying lighting conditions. We also intend to improve our system by making use of 3-D information that can be generated by a stereo camera. This will enable a more reliable estimation of truly three dimensional motions.

References


Aleš Ude

The author studied applied mathematics at the University of Ljubljana, Slovenia, and computer science at the University of Karlsruhe, Germany, where he received a doctoral degree in 1996. From 1998 to 2000 he was an STA fellow in the Kawato Dynamic Brain Project, ERATO, JST. Currently he holds a research position at the Jožef Stefan Institute, Ljubljana, Slovenia and is also associated with the Cyberhuman project, ATR-I, Kyoto, Japan. His research interests include various topics in computer vision and robotics.