MVA2000 IAPR Workshop on Machine Vision Applications, Nov. 28–30, 2000, The University of Tokyo, Japan

13—13

Model-based 3D Motion Analysis via Robust Estimation

Jairo Rocha and Arnau Mir
University of the Balearic Islands *

Abstract

Within a human motion analysis system, body parts are modeled by simple virtual 3D rigid objects. Its position and orientation parameters at frame $t+1$ are estimated based on the parameters at frame t and the image intensity variation from frame t to $t+1$, under kinematic constraints. An optimization procedure calculates the 3D parameters that make a goal function that measures the intensity change minimum. The goal function is *robust*, so that outliers located especially near the virtual object projection borders have less effect on the estimation. Since the object's parameters are relative to the reference system, they are the same from different cameras, so more cameras are easily added, increasing the constraints over the same number of variables. A successful experiment is presented for an arm motion of three parts seen from two cameras.
Keywords: Human motion, robust estimation, twist.

1 Introduction

Despite the great amount of work done in the area, e.g., [9, 6, 4, 7], [2, 5, 1], human motion analysis is still a challenging topic. Bregler and Malik [4] built a system able to track human motion with great precision, even for Muybridge's photograph sequences. They defined a 3D virtual model of the subject and a goal function over body part position parameters that measures the changes in image intensities. Using the twist representation for rigid transformations and the flow constraint equation, they managed to make the goal function lineal in the parameter variables. They could therefore apply an iteration of linear optimization techniques and a warping routine to obtain a very reliable procedure for 3D position estimation. Our system is a modification of this one. The differences are the following: First, our goal function is robust, so that no EM procedure is needed afterwards; instead our optimization directly performs a robust parameter estimation. Second, we do not use the flow constraint equation but direct difference of pixel intensity, thus fewer assumptions, such as small motion and constant intensity, are assumed and no warping procedure is needed; due to these two previous differences, we cannot apply a linear optimization technique as we will explain below. Third, our parameters are reference-based instead of camera-based, so additional cameras do not increase the number of variables to be estimated. Our system is not a finished product and therefore its performance cannot be compared to Bregler and Malik's, but we will try to convey why we think this project is promising.

The *Cardboard People* system [7] performs robust estimation of motion parameters for 2D regions. Assuming the flow constraint equation and a model for the motion of each patch, motion parameters are robustly estimated using a non-linear optimization procedure. We can say that our system is roughly a 3D version of Cardboard People.

All these systems and ours as well assume that a virtual humanoid that matches the real subject in size and initial position can be defined by other means in practice, by user interaction.

2 Problem Formulation

Given the film $I(x, y, t)$, let us take $I_0(x, y) = I(x, y, t_0)$ and $I(x, y) = I(x, y, t_0 + 1)$, two consecutive frames.

The virtual model of a body part is an ellipsoid of appropriate dimensions. Virtual cameras of the real cameras are defined by a camera calibration routine. Assume that the 3D pose (position and orientation) of the ellipsoid at time t_0 is known. The problem is to find the change in the 3D pose of the ellipsoid so that the motion coincides with the real image'motion. Let ϕ be the pose transformation of the ellipsoid from t to $t+1$; ϕ is defined by 6 real parameters that will be discussed in detail later.

Let (x, y) be a pixel, and $(u_x(x, y, \phi), u_y(x, y, \phi))$, its displacement vector when a 3D point that is projected onto the camera pixel moves according to ϕ. The goal functional E is the brightness change sum over the point projections before and after the pose transformation.

We define the functional $E(\phi)$ by

*Address: E-07071 Palma de Mallorca, Spain, e-mail: arnau.mir@uib.es, jairo@ipc4.uib.es

$$\sum_{(x,y)\in\mathcal{R}} \rho(I_0(x,y) - I(x+u_x(\phi), y+u_y(\phi))) = \sum_{(x,y)\in\mathcal{R}} \rho(\Delta I) \text{ where } \rho(t) = \frac{t^2}{\sigma^2+t^2} \quad (1)$$

where $\phi \in R^6$ are the 3-D motion parameters, $I_0(x,y)$ is the image brightness (intensity) function for the initial frame, $I(x,y)$ is the image brightness function for the final frame, $u_x(\phi), u_y(\phi))$ are the horizontal and vertical components of the flow image at the point (x,y), which is the projection from R^3 of the motion associated with the parameters ϕ; $\mathcal{R}$ is the patch to consider, in this case, the projection of the virtual ellipsoid that models that tracked part and, finally, ρ is the function that reduces the influence of some outlying measurements of the brightness difference and allows an estimation of the dominant parameters; there are other ρ-functions that can be considered as well to obtain a different robust estimation. In the rest of this paper, we will use the one above.

To minimize the functional (1), we use a *continuation method*, the same used by [3]. It is the following iterative scheme:

$$\phi_i^{(n+1)} = \phi_i^{(n)} - \omega \frac{\frac{\partial E}{\partial \phi_i}(\phi^{(n)})}{T_{\phi_i}},$$

where T_{ϕ_i} is an upper bound on the second partial derivative of E : $T_x \geq \left|\frac{\partial^2 E}{\partial x^2}\right|, \quad \forall \phi.$

We do not know how to precisely calculate the upper bound T_{ϕ_i}. We decided to sample $\frac{\partial^2 E}{\partial^2 \phi_i}(\phi)$ for approximately 4000 values of ϕ through several images. Experimentally, the result of this sampling allows us to get correct upper bounds on the value of the second derivatives, for all the tests we have done.

The objective is to minimize E relative to ϕ, for which a precise definition of (u_x, u_y) and its gradient is needed.

3 Motion Projection

The object pose relative to the reference frame can be represented as a rigid body transformation in R^3 using homogeneous coordinates:

$$q_r = G \cdot q_0 = \begin{pmatrix} r_{11} & r_{12} & r_{13} & d_x \\ r_{21} & r_{22} & r_{23} & d_y \\ r_{31} & r_{32} & r_{33} & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot q_0,$$

where $q_0 = (x_0, y_0, z_0, 1)^\top$ is a point in the object frame and $q_r = (x_r, y_r, z_r, 1)^\top$ is the corresponding point in the reference frame. $q_c = (x_c, y_c, z_c, 1)^\top$ is the corresponding point in the camera frame: $q_r = M_C \cdot q_c$, where M_C is the transformation matrix associated with the camera frame.

Using orthographic projection with scale s, the point q_r in the reference frame gets projected onto the image point $(x_{im}, y_{im})^\top = s \cdot (x_c, y_c)^\top$. s is equal to the focal distance divided by the distance of the ellipsoid center to the camera, which happens to be a good approximation for all the points on the ellipsoid.

It can be shown ([8]) that for any arbitrary $G \in SE(3)$, there exists a vector $\xi = (v_1, v_2, v_3, w_x, w_y, w_z)^\top$, called the twist representation, with associated matrix

$$\tilde{\xi} = \begin{pmatrix} 0 & -w_z & w_y & v_1 \\ w_z & 0 & -w_x & v_2 \\ -w_y & w_x & 0 & v_3 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

such that $G = e^{\tilde{\xi}} = \mathrm{Id} + \tilde{\xi} + \frac{\tilde{\xi}^2}{2} + \frac{\tilde{\xi}^3}{6} + \cdots$

We define the pose of an object as $\xi = (v_1, v_2, v_3, w_x, w_y, w_z)^\top$. A point q_0 in the object frame is projected onto the image location (x_{im}, y_{im}) with:

$$\begin{pmatrix} x_{im} \\ y_{im} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \cdot s \cdot M_C^{-1} \cdot e^{\tilde{\xi}} \cdot q_0. \quad (2)$$

The image motion of point (x_{im}, y_{im}) from time t to time $t+1$ is:

$$\begin{pmatrix} u_x \\ u_y \end{pmatrix} = \begin{pmatrix} x_{im}(t+1) - x_{im}(t) \\ y_{im}(t+1) - y_{im}(t) \end{pmatrix}.$$

We assume that the scale change due to the motion is negligible from frame to frame, since the objects are far from the camera. Therefore $s(t+1) = s(t) = s$. By using (2) we can write the previous expression as:

$$\begin{pmatrix} u_x \\ u_y \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \cdot M_C^{-1} \left(e^{\tilde{\xi}'} - \mathrm{Id}\right) \cdot s \cdot q_r,$$

with $\xi' = (v_1', v_2', v_3', w_x', w_y', w_z')^\top = \xi(t+1) - \xi(t)$ and $s' = \frac{s(t+1)}{s(t)} - 1$.

Assuming that the motion is small, i.e., the ellipsoid center moves a few centimeters and the axis orientation changes a few degrees, we have $\|\xi'\| \ll 1$. We approximate the matrix $e^{\tilde{\xi}'}$ by $\mathrm{Id} + \tilde{\xi}'$. Experimentally, we confirm that the approximation is very good.

We rewrite the previous expression as:

$$\begin{pmatrix} u_x \\ u_y \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \cdot M_C^{-1} \cdot \tilde{\xi}' \cdot s(t) \cdot M_C \cdot q_c. \quad (3)$$

The value of ϕ is $(v_1', v_2', v_3', w_x', w_y', w_z')^\top$, which are the optimization variables of the problem.

Based on (3), for a pixel (x,y), we need only calculate q_c to describe the image motion in terms of

the motion parameters ϕ. The 3D point q_c is calculated by intersecting with the ellipsoid a ray orthogonal to the camera image pixel. To do so, we associate with each pixel at t_0 the corresponding z_c of the closest point on the ellipsoid surface that is projected onto that pixel.

The parameters ϕ are independent of the camera. Hence, when we add more cameras, no new variables are needed but more constraints are added.

4 Kinematic Chain

The parameterization of a single body part has been discussed in the previous section. Assume that a second body part is attached to the first one in a point and that $E_1(\phi_1)$ and $E_2(\phi_2)$ are the functionals to be minimized if the parts were to be tracked independently.

Let p_1 and p_2 be the coordinates of the shared point in the two object frames. Let $\tilde{\xi_1}, \tilde{\xi_1}', \tilde{\xi_2}$ and $\tilde{\xi_2}'$ be the twist and its change for each part as in the previous section. In order to keep the parts attached, the following equality must be true: $e^{\tilde{\xi_1}+\tilde{\xi_1}'} \cdot p_1 = e^{\tilde{\xi_2}+\tilde{\xi_2}'} \cdot p_2$. If the point was shared at frame t, it is true that $e^{\tilde{\xi_1}} \cdot p_1 = e^{\tilde{\xi_2}} \cdot p_2 = p_r$ where p_r are the coordinates of the joint in the reference system. Hence, the constraint simplifies to $e^{\tilde{\xi_1}'} \cdot p_r = e^{\tilde{\xi_2}'} \cdot p_r$ and using again the first order approximation of the exponential function, it leads to $\tilde{\xi_1}' \cdot p_r = \tilde{\xi_2}'$, which corresponds to three linear equations that allow us to eliminate three variables. In short, each part pose is described by 3 variables except the first one that needs six variables.

The technique generalizes to n parts although the number of variables is $3n$.

5 Implementation

Using the camera calibration, the user sets the ellipsoid poses at time 0 so that its projections coincide with the three real body parts to be tracked in each view. This is done by calculating the best 3D coordinates for the image positions such as the wrist and the elbow. The three axis are found and used to define the ellipsoids. The initial object transformation matrix is calculated with the *Rodrigues formula*[8]. The shape parameters of each part are also set manually for the whole film.

The system then tracks each virtual part in 3D: at frame t, for each pixel in the image range, it calculates the 3D position of a point on the ellipsoid that is projected onto the pixel. Frame $t+1$ is then loaded and its first and second spatial derivatives are calculated. An iterative procedure begins that initializes $\phi_0 = 0$. At iteration n, the (u, v) displacement is estimated using the pose change ϕ_{n-1} for each pixel (x, y). Using the first and second derivates of the image intensity at position $(x + u_x, y + u_y)$ of frame $t + 1$, the difference of image intensities with frame t, accumulated for all pixels, and the kinematic constraints a new pose change ϕ_n is calculated using a non-linear optimization procedure. The value for ω is 0.5. For the experiments, 2500 iterations are used. The value of ϕ that minimizes the functional within all the iterations is used to calculate the $t + 1$ pose. The procedure starts again for the next frame.

The value for σ is set as follows. Grey levels within the arm vary a maximum of 20% of the maximum variation of intensity Max_I, i.e., between the values for black and white. This means that variations above this value should be considered outliers. Since above $\sigma/\sqrt{3}$ the influence of outliers first begins to decrease [3], we take $\sigma = \sqrt{3} \times 0.2 Max_I$. Larger values produce worse solutions, which implies that non-robust estimators could perform poorly and shows that the use of a ρ-function creates great estimation stability. No *cooling schedule* for σ as in [7] was implemented, but there is some evidence that our system might benefit from one.

6 Experiments

A human subject was videotaped from two synchronized black and white cameras. An arm was selected for tracking because of its large motion and the visibility of three articulated parts. Arm articulation points were manually marked at the first frame from each view, so that the ellipsoid's center 3D coordinates and orientation of its main axis are defined by a rotation axis and an angle with respect to the reference system defined during the camera calibration. The frames are 640×480 pixels, and the cameras are situated between three and five meters from the subject.

The system tracks the arm for 15 frames. Sections of the first 8 frames are shown in Figure 1. The system works reasonably well, taking approximately 1.5 min. per frame on a Pentium II processor at 333 MHz. The non-linear local optimization procedure may cause some problems for a general solution, since w has been tuned for this experiment, according to the upper bounds experimentally found, for the σ used.

Frame 0 is the first one in Figure 1. The ellipsoids are situated "manually" in this frame, and their projections are displayed by the ellipse borders. The program starts calculating the best ellipsoid motions for the next frame. From frame to frame, some displacement error is made. These errors are visually accumulated, since the process finds the next motion based only on the previous position, e.g., motion calculation for frame 2 forgets about frame 0. Therefore, after 15 frames, the hand's position is not

at the real hand's, but the motion corresponds to a functional minimum.

7 Conclusions and Future Work

We have presented the theoretical framework for robust tracking of human parts based on a 3D model. Our experiments show that the robust estimation of motion parameters works well, but more testing is needed.

Acknowledgements. José María Buades calibrated and synchronized the cameras. Isabel Miró helped with the implementation. This research is supported by the Spanish Education and Culture Ministery, under the project TIC98-0302-C0201.

References

[1] A. Azarbayejani and A. Pentland. Real-time self-calibrating stereo person tracking using 3d shape estimation from blob features. In *Proceedings of ICPR*, pages 90–103, Vienna, Austria, 1996.

[2] A. Bharatkumar, K. Daigle, M. Pandy, Q. Cai, and J. K. Aggarwal. Lower limb kinematics of human walking with the medial axis transformation. In *IEEE Computer Society Workshop on Motion of Non-Rigid and Articulated Objects*, pages 70–76, Austin, Texas, 1994.

[3] M. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, January 1996.

[4] C. Bregler and J. Malik. Video motion capture. http://www.cs.berkely.edu/~bregler/digmuy.html, 1997. UCB-CSD-97-973.

[5] I. Haritauglu, D. Harwood, and L. Davis. W^4S: A real-time system for detecting and tracking people in $2\frac{1}{2}$D. *Computer Vision- ECCV'98*, 1406:877–892, 1998.

[6] E. Hunter, P. Kelly, and R. Jain. Estimation of articulated motion using kinematically constrained mixture densities. In *Proceedings of IEEE Non-Rigid and Articulated Motion Workshop*, pages 10–17, Puerto Rico, USA, 1997.

[7] S. Ju, M. Black, and Y. Yacoob. Carboard People: A Parameterized model of articulated image motion. In *2nd International Conference on Face and Gesture Analysis*, pages 38–44, Vermont, USA, 1996.

[8] R. Murray, Z. Li, and S. Sastry. *A Mathematical Introduction to Robotic Manipulation.* CRC Press, 1994.

[9] S. Wachter and H. Nagel. Tracking of persons in monocular image sequences. In *Proceedings of IEEE Non-Rigid and Articulated Motion Workshop*, pages 2–9, Puerto Rico, USA, 1997.

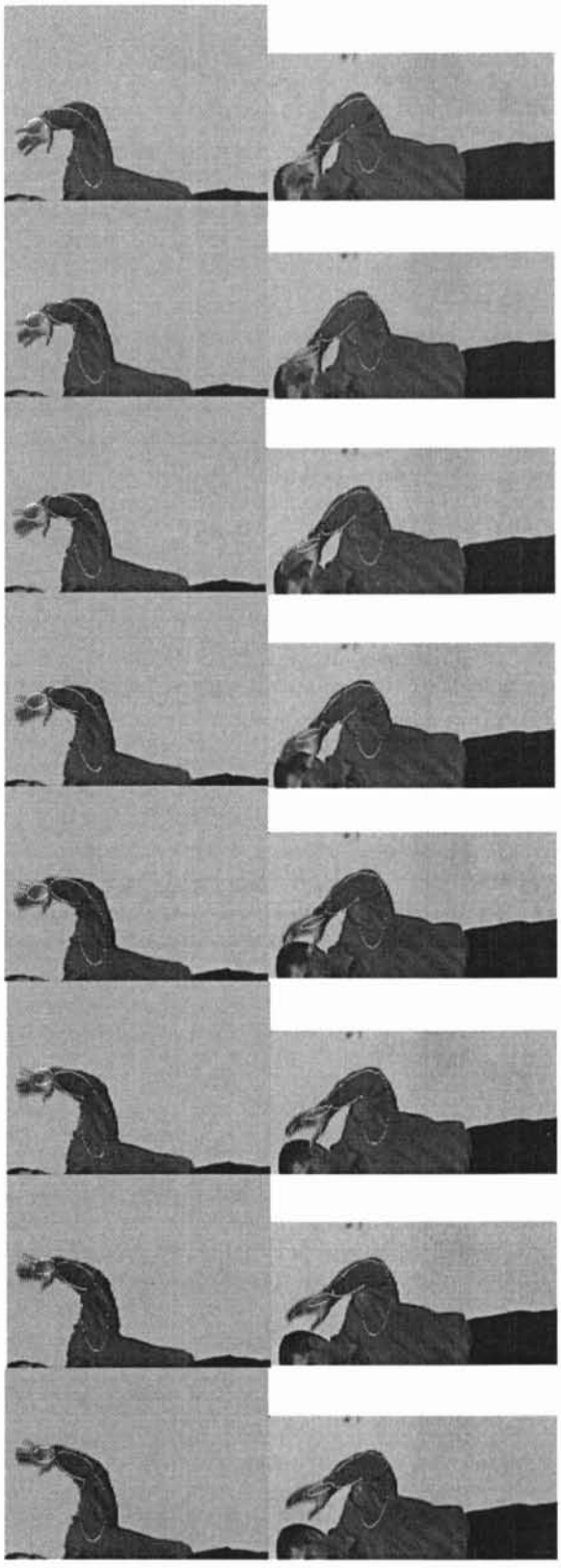

Figure 1: Arm tracking.