POSE TRACKING FOR VIRTUAL WALK-THOUGH ENVIRONMENT CONSTRUCTION

K.H. WONG AND S.H. OR
Dept. of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong
E-mail: khwong@cse.cuhk.edu.h, shor@cse.cuhk.edu.hk

M.M.Y. CHANG
Dept. of Information Engineering, The Chinese University of Hong Kong
E-mail: mchang@ie.cuhk.edu.hk

During the construction of a virtual walk-through environment, the most tedious work is to take pictures of the interior of an environment for forming the textures of the walls. The texture images must be taken with known intrinsic and extrinsic parameters of the camera used. Many existing methods can obtain the intrinsic parameters correctly. However, the inverse problem of tracking extrinsic parameters (or tracking the pose) of a camera from a long image sequence is a more difficult task because of the problem of point correspondence. Our work attempts to solve this problem by a probabilistic method that rejects poor correspondences to increase the accuracy for the tracking. Our approach is based on a recursive Bayesian filtering method called Condensation working in conjunction with a pose algorithm. Simulation results show that it is robust for tracking object poses in a long image sequence.

1 Introduction

Pose estimation is important in many applications such as robotics, model constructions and virtual reality system development. For most pose estimation problems we generally assume that we have the model of the object and an image sequence, and our target is to find the pose (rotation and translation) of the object with respect to the camera [1]. The problem is usually solved by an iterative algorithm based on the correspondences of the 3D model features and their 2D image points. And the pose of the object obtained is defined by 3 rotational angles and 3 translations in the 3D Cartesian space. The pose estimation problem has two variations: (a) pose estimation of an object based on one image. For example, if we want to recognize objects found in the Internet, we have to find the pose of the object before actual object recognition begins. (b) The second problem is the pose-tracking problem for an image sequence of an object in motion. Of course, one can solve it by finding the pose for each image separately, and then combine the result at the end. However, the dynamics of the object motion may give extra information for the tracking. The Kalman filter method is an example for this kind of approaches [5].

We are interested in applying pose estimation in virtual environment system development. For example, in constructing a virtual environment for the interiors of a real building, developers have to capture large number of images of the environment and paste them onto a 3D graphics rendering system. This image taking process is a tedious one because the camera extrinsic and intrinsic parameters should be recorded for each picture taken. Our aim is to automate this picture taking process by a robot and a robust pose estimation algorithm. That is the reason why we studied the problem of pose tracking.

In fact, pose estimation algorithms for one single image already exist, the lowest number of point correspondence pairs needed is 3 and there are algorithms use 4, 5 or 8 or more [3,4]. By combining the pose estimation results of individual images within an image sequence we can have a pose tracking system. However, all these algorithms assume the point correspondences are available and accurate. Usually least square algorithms are employed for handling noise problems in these pose estimation algorithms. However, if a large percentage of mismatches occur these algorithms may fail. This mismatch problem becomes more serious as the video length becomes longer. Many researchers have experienced this lost track problem in dealing with long image sequences. That is, at the beginning of an image sequence the pose tracking is quite accurate, however, the error accumulates fast and the tracking will soon fail. The main problem is usually not the pose estimation itself, it is mainly caused by the fact that some 2D to 3D point correspondences are incorrect, and the least square pose estimation cannot recover this mistake. There are many causes to this mismatch problem in the correspondence process. Kalman filtering can provide a solution to reduce the effect of mismatch but it cannot eliminate the effect totally.

We use a Bayesian filtering method called Condensation [2] to solve this problem. First we know that the minimum number of point correspondences is 3, so if we have a large number of point correspondences available,we do not need to use all of them. The reason is, within the pool of correspondences some are erroneous. Our approach is to select randomly a subset of the n point correspondences (hoping to select only the good correspondences) say m (where n is greater than m), to form a pose. By repeating this selection process many times, we will have a collection of poses found. Then we can use a statistical method to determine the accurate result. However, if all possible choices (C^n_m) of selections are considered, it is too large. To solve this, a Monte Carlo type search method may be suitable. We use this method for tracking the pose of the object with mismatch noise and found that the algorithm can

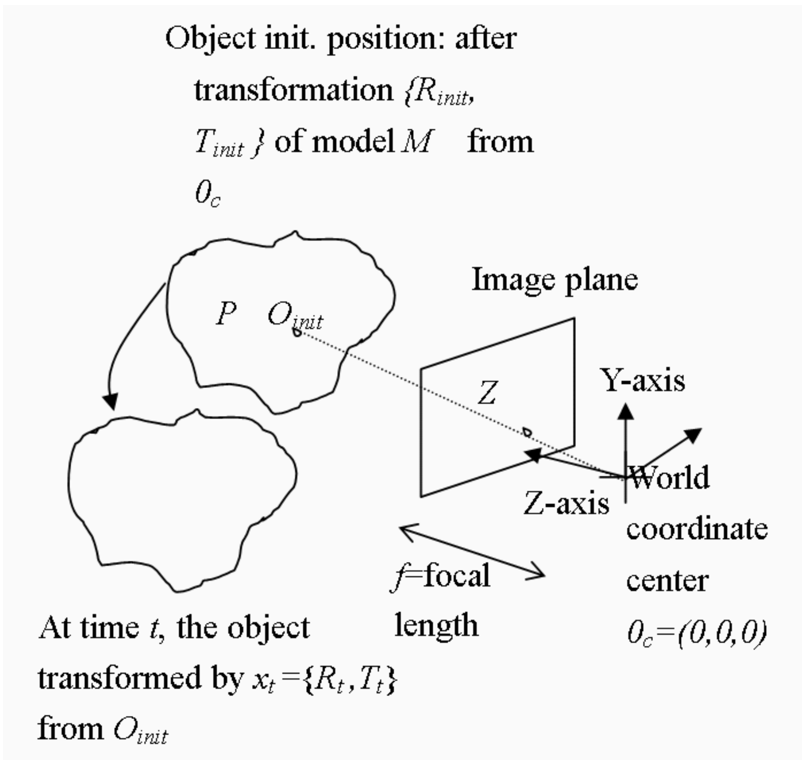
Object init. position: after transformation $\{R_{init}, T_{init}\}$ of model M from 0_c
P
O_{init}
Image plane
Z
Y-axis
World coordinate center $0_c=(0,0,0)$
Z-axis
f=focal length
At time t, the object transformed by $x_t=\{R_t,T_t\}$ from O_{init}

Figure 1. The object and the Camera

improve the accuracy of the tracking.

In section 2, we will describe the theory used in our approach. In section 3, we will describe the implementation. And in section 4, we will show the experimental results. Conclusion can be found in section 5.

2 Theory

2.1 Problem formulation

We have a camera viewing an object as in Figure 1, the camera is at the center of the 3D world coordinate center O_c. The movement of any point from $p = [p_x, p_y, p_z]^T$ to p' in the 3D space can be described by a transformation of $p' = Rp + T$ ($[]^T$is the matrix transpose operator). In here, $R(\theta_x, \theta_y, \theta_z)$

is a 3x3 rotation matrix and $T = [T_x, T_y, T_z]^T$ is a 3x1 translation vector, where $\theta_x, \theta_y, \theta_z$ are rotation angles about the X, Y, Z-axis, and T_x, T_y, T_z are translations along the X, Y, Z-axis, respectively. Also, the poses (or states) of the object up to time t is described by $\chi_t = \{x_1, x_2, ..., x_t\}$, where $x_t = (R_t, T_t)$.

The object centered at O_c has a model M containing a set of 3D feature points $p = \{p_1, p_2, ..., p_n\}$ on the surface. Before tracking, assume the object center is moved from O_c to O_{init} by a transformation governed by a rotation matrix R_{init} and a translation vector T_{init}. The 3D feature points are projected onto the image plane by the camera of focal length f to form a set of 2D features $Z_t = \{z_1, z_2, ..., z_n\}_t$, where $z = (u, v)$ and the perspective projection formulas $u = f\frac{p_x}{p_z}$ and $u = f\frac{p_y}{p_z}$. From the initial position the object moves to a new position at time t, therefore we will have a set of 2D-measurement (correspondences) history$\bar{Z}_t = \{Z_1, Z_2, ..., Z_t\}$. Our aim is to find χ_t from $\bar{Z}_t$ and M.

2.2 Noise problems in tracking correspondences and pose

Mismatch of correspondences is the major cause of tracking failure. We identify the major sources of mismatch as follows: (1) 2D noise caused by lighting condition changes. (2) Pose change may introduce mismatch error. (3) 2D Clutter noise. Our algorithm can reduce the effect of these noises.

2.3 The probability framework

A probabilistic approach is necessary to reduce the effect of the mismatch error. The following formulas describe the probability framework of the Condensation approach as in [2].

$$p(x_t|\bar{Z}_t) = k_t p(Z_t|x_t) p(x_t|\bar{Z}_{t-1}) \tag{1}$$

$$p(x_t|\bar{Z}_{t-1}) = \int_{x_t-1} p(x_t|x_{t-1}) p(x_{t-1}|\bar{Z}_{t-1}) \tag{2}$$

Equation (1) calculates the probability of having the current pose as x_t if the measurement history $\bar{Z}_t$ is known. The Bayesian rule decomposes the conditional probability $p(x_t|\bar{Z}_t)$ into two components. (a) The density $p(x_t|\bar{Z}_{t-1})$calculates the probability that the object is predicted to have the pose x_t if the measurement history up to the previous time frame is $\bar{Z}_{t-1}$; (b) $p(Z_t|x_t)$ is the likelihood function of finding the probability of producing the current measure $\bar{Z}_t$ if the current state is predicted to be x_t. We treat x_t as a temporal Markov chain, therefore Equation (2) decomposes into two terms: (a) $p(x_{t-1}|\bar{Z}_{t-1})$ is the probability of reaching state x_{t-1} based on the observations till the previous frame; (b) $p(x_t|x_{t-1})$ is the probability of all possible state transitions from the previous time to the current state x_t. Details of the formulation can be found in [2].

Equation (1) and (2) serve as an iterative loop for finding x_t by maximizing $p(x_t|\bar{Z}_t)$ for the given measurement history if the initialized parameters are known. However, in actual calculation, the possible space for x_t over time is very large, and the complexity of finding $p(x_t|\bar{Z}_t)$ of all possible x_t is huge. So we use the Condensation algorithm to make the complexity limited to a manageable level.

2.4 The Condensation algorithm for pose tracking

For the pose estimation of an object with n 3D features, we discussed that it is not practical to assume we can identify all 2D to 3D point correspondences without mistakes, because the matching may be corrupted by 2D and 3D noise. One approach is that we can select a subset m out of all n (where $m < n$) 3D-feature points on the object and use them to find their correspondences in the image. Then, we will use the correspondence set to find a pose by some known iterative methods such as the algorithm in [1]. We can repeat the above process many times, then we will have a set of poses. Then a voting algorithm can be used to give an overall result. It is essentially a statistical approach, however, this approach has two problems. (a) The number of combination of selecting m out of n features can be very large, say the combination of selection $m = 10$ out of $n = 20$ features is $C^{n=20}_{m=10} = 184756$, which is a huge number to be handled since each set has be processed by an iterative optimization method to find the pose. (b) Wrong matches using this method will still be bad, it will still corrupt the final result, and for the tracking of an object in an image sequence the error will accumulate and end up with intolerable errors.

In this work, we will solve it by the Condensation technique. It is a way to reduce the complexity of the algorithm and at the same time minimizes the error created by the mismatch correspondences. Assume that we have a set of 2D-measurement (correspondences) history $\bar{Z}_t = \{Z_1, Z_2, ..., Z_t\}$ and a Model M. At time t, we perform N_{cond} selections. For each selection, we select m out of the n features from Z_t. That is, for the j^{th}selection, it creates a set $I_t^{(j)} = \{i_1, i_2, .., i_m\}$ containing the indexes of the selected elements from the corresponding 2D measurements Z_t. Hence the selected set is $Z_t^{(j)} = \{z_{i_1}, z_{i_2}, .., z_{i_m}\}$, where $z = (u, v)$ is a 2D feature point and $j \in \{1, 2, ...N_{Cond}\}$. And from $z_t^{(j)}$ we can form a pose $x_t^{(j)} = (R_j, T_j)$ by the model M. So after each time frame, $Z_t = \{z_t^{(1)}, z_t^{(2)}, ...z_t^{(j)}, ..., z_t^{(N_{Cond})}\}$ and $X_t = \{x_t^{(1)}, x_t^{(2)}, ...x_t^{(j)}, ..., x_t^{(N_{Cond})}\}$ are formed.

By using the pose set X_t, M and its time derivative dX_t, we can find the prediction $X_{t+1}$and the possible measurement set Z_{t+1} for the next time frame. The probability $P(x_{t+1}^{(j)}|x_t^{(j)})$ tells how to predict new pose for thej^{th} selection. By matching the predicted measurement and real measurement, we can calculate $P(z_{t+1}^{(j)}|x_{t+1}^{(j)})$. If it is high, the j^{th} selection is good enough and it can make prediction for the next selection of $z_{t+1}^{(j)}$, otherwise a new index set $I_{t+1}^{(k)}$ of randomly selected 2D image points $z_{t+1}^{(k)}$ will be selected for the replacement. The actual implementation can be found in the following section.

The advantage of this method is that we only handle N_{cond} processing steps for each time frame, therefore the complexity of the algorithm can be controlled to a tolerable level.

3 Implementation

3.1 Initialization and sampling

During initialization, assume the model M with n 3D features points and (R_{init}, T_{init}) are known. Then we use a feature extraction method to locate the initial feature set Z_1 of n 2D feature points, and establish the 2D to 3D correspondences between Z_1 and M. From Z we perform N_{cond} selections and for each selection (e.g. the j^{th} selection) we select m features out of the available n features to form the index set s_j and $Z_{t=1}^{(j)} = \{z_1, z_2, ..z_m\}$, and then by using M and an iterative algorithm [1] we can calculate the corresponding pose $x_{t=1}^{(j)}$. We also initialize all likelihood probabilities $P(z_{t=1}^{(j)}|x'_{t=1}{}^{(j)})$ and accumulated probabilities $C_{t=1}^{(j)}$ (for all j) as ones.

After initialization we have $S_{t=1} = \{s_1, s_2, ..s_j, .s_{N_{cond}}\}$, $X_{t=1} = \{x_1, x_2, ..x_j, x_{N_{cond}})$ and $P(X_2|X_1)$.

3.2 The main iteration loop modified from [2]:

From $(t = 2)$ to $(t = T^{(MAX)})$ frame, iterate the following steps:

1. Re-sampling for the jth selection

   (a) Generate a uniformly distributed random number $r \in [0, 1]$.

   (b) Find, by binary subdivision, the smallest q for which $c_t^{(q)} \geq r$.
   (c) Assign $s_t^{(j)}$ =$s_{t-1}^{(q)}$

2. Prediction and measurement loop for constructing the normalized likelihood probability set $\{\pi_{t+1}^{(j)}\}$ for j ={1,2,..,N_{Cond}}

   (a) Prediction for the jth selection { From $s_t^{(j)}$ and its related pose $x_t^{(j)}$ and $dx_t^{(j)}$ make predicted pose: ${x'_{t+1}}^{(j)} = x_t^{(j)} + dx_t^{(j)} + prediction\_noise$}
   (b) Transform the model points M according to ${x'_{t+1}}^{(j)}$ and project them to the image plane to form the prediction measurement ${z'_{t+1}}^{(j)}$.
   (c) Measurement for the j^{th} selection

$$P(z_{t+1}^{(j)}|{x'_{t+1}}^{(j)}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-1}{2\sigma^2}E\right)$$

$$\text{where } E = \sqrt{\sum_{w=all\ m} \left({z'_{t+1}}^{(w)} - z_{t+1}^{(w)}\right)^2} \text{ and } \sigma = 0.398$$

   (d) Transforming the $P(z_{t+1}^{(j)}|x_{t+1}^{(j)})$ into a normalized probability function π_{t+1}, so that $\sum_m \pi_{t+1}^{(j)} = 1$.

3. Form a cumulative probability density c_{t+1}, where,

$$c_{t+1}^{(0)} = 0,$$
$$c_{t+1}^{(j)} = c_{t+1}^{(0)} + \pi_{t+1}^{(j)} \ (j = 1, ..., N_{Cond})$$

3.3 Result generation

We can use the expected value of x_t as the output result.

4 Experiments

4.1 Synthetic results

We generated an object of 20 features randomly at a range of 30cm^3 and 1 meter away from the camera. Then moved them in the 3D space, and projected them to form an image sequence. Our pose-condensation algorithm was used to track the object and we can see that the tracking was correct even mismatch noise was introduced. However when no Condensation was used, tracking failed. Parameters used for this test are as follows: total number of features is n=20, selected number of features for each Condensation sample is m=10, features corrupted by mismatch noise is 25 % of all features, the mismatch noise used is a unit random generator * 0.04 * image feature positions, N_{Cond}=200 .

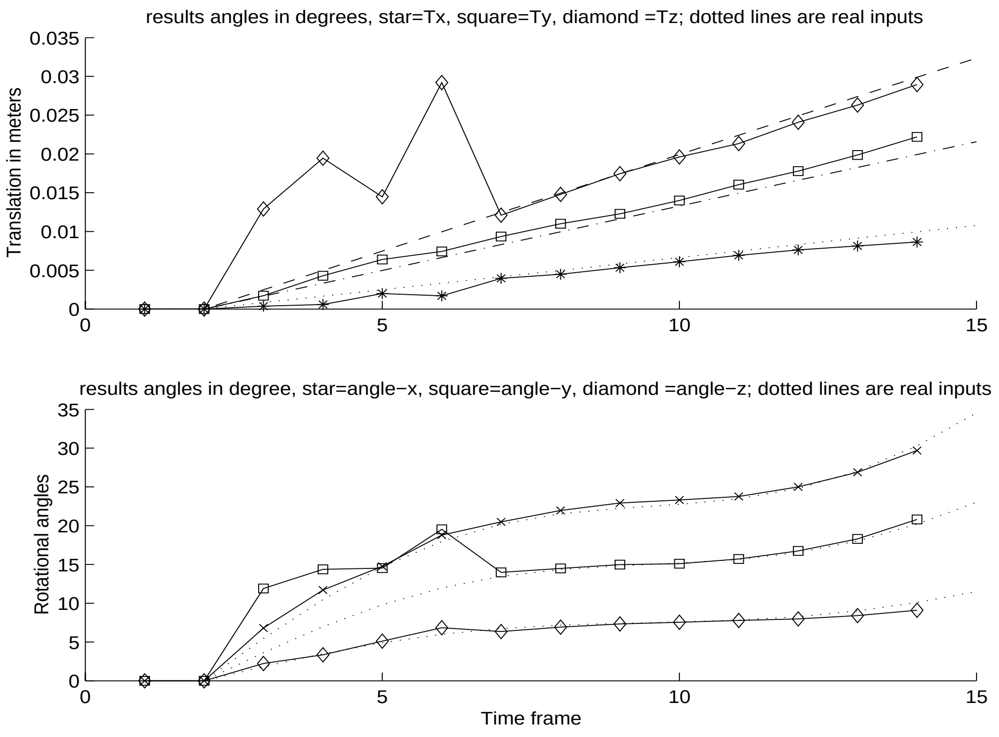

Figure 2. Tracking result of simulated data

In the figure, the upper part are the translations and the lower part are the rotational angles. The dotted lines are real values and the solid lines are tracked result. The focal length used is 4 mm. The object is a cluster of points 1 meter away from the camera. All parameters were tracked except some initial error especially at the translation T_z, it can be explained that the iterative solution is not very sensitive to depth change. And it can be improved if good initial guess is given.

5 Conclusion

In this work we have successfully used the Condensation algorithm for pose tracking. The algorithm has been described and simulation results have been produced. The next step is to apply it to track poses of objects in real images.

6 Acknowledgements

The work described in this work was fully supported by a grant from the Research Grant Council of Hong Kong Special Administrative Region. (Project Number. CUHK4389/99E)

References

1. Helder Araujo and Rodrigo L. Carceroni and Christopher M. Brown, A Fully Projective Formulation to Improve the Accuracy of Lowe's Pose-Estimation Algorithm", Computer Vision and Image Understanding: CVIU, Vol.70, number 2, pages 227–238,1998.
2. Michael Isard and Andrew Blake , CONDENSATION – conditional density propagation for visual tracking", Int. J. Computer Vision, 29, 1, 5–28, (1998).
3. M.L. Liu and K.H. Wong, "Pose Estimation Using Four Corresponding Points", Pattern Recognition Letters, Volume 20, Number 1 January 1999, pp. 69-74.
4. S.H. Or, W.S. Luk, K.H. Wong, I.King, "An efficient iterative pose estimation algorithm", Image and Vision Computing Journal, Volume 16, Issue 5, pp.355-364, May 1998.
5. E. Koller-Meier and F. Ade, "Tracking Multiple Objects Using the Condensation Algorithm", Journal of Robotics and Autonomous Systems, vol. 34, 2-3, 2001, pp. 93-105
6. Camera Calibration Toolbox for Matlab: http://www.vision.caltech.edu/bouguetj/calib_doc/