Proceedings of the 2001 IEEE
International Conference on Robotics & Automation
Seoul, Korea • May 21-26, 2001

Design and Implementation of Visual Servoing System for Realistic Air Target Tracking

Wei Guan Yau [1], Li-Chen Fu [1,2], and David Liu [1]

Department of Electrical Engineering [1]
Department of Computer Science and Information Engineering [2]
National Taiwan University, Taipei, Taiwan, R.O.C.
r7921018@ms.cc.ntu.edu.tw

Abstract

In this paper, a real-time visual tracking system based on our proposed motion estimation algorithms is developed. The proposed motion estimation algorithm is used to predict the location of target and then generate a control input so as to keep the target stationary in the center of image. The work differs from previous ones in that it is capable to decouple the estimation of motion from the estimation of structure. The major contribution of this work is that simple, none computation intensive, correspondence-free, and numerically stable 3D motion estimation algorithms are developed. The robust target detection method in simple environment and a time reduction of SSD method in complex environment are minor contributions. The visual tracking system can achieve at a rate of 30 Hz. The robustness of the visual tracking system is validated by a number of experiments.

1. INTRODUCTION

Recently, research on motion estimation has been widely conducted [16, 26, 20, 21, 18, 1, 2, 13, 12]. However, one cannot estimate the target motion without knowledge of its structure or of the camera motion. In light of this observation, an adaptive method [23, 25, 11, 14, 15] or fuzzy controller is used to overcome this problem. On the other hand, the Interacting Multiple Model (IMM) algorithm is another famous target tracking framework [9, 28, 27, 4, 8].

The methods described above however are either too complicated in form or have their own drawbacks, which in turn increase the computational burden or deteriorates the tracking performance. Therefore, this paper proposes another close form solution for estimating the object motion during the tracking phase. This yields better performance than the previous methods.

Target detection algorithms such as Sum of Squared Difference (SSD) [23, 24, 26], and Normalized Cross-Correlation (NCC) [3, 5] are primitive methods. A small area of the image is used as template, which is later searched throughout the interested region in the next frame for finding the target location. However, these methods are computationally expensive and the SSD method is sensitive to the changes of the illumination. Sawasaki [5] used a specific hardware to speed up the computation. Hager and Belhumeur [19] also used the SSD method, but they provided an illumination-insensitive tracking algorithm. However, they need the images under various lighting conditions as the illumination basis. Another template matching method called contour matching [17, 6, 22] uses a non-rigid contour as model and is more robust to cluttered background.

Another well-known algorithm is motion-based recognition [10]. Motion-based tracking systems have the advantage of tracking a moving object regardless of its shape [29] [30]. This method is suitable for real-time implementation. However, they are easily corrupted in outdoor environment when the background contains other moving objects such as waving tree leaves.

Several existing target detection algorithms have already demonstrated appealing performance [17, 6, 22, 7, 29]. However, capability of real-time implementation is our main consideration. Therefore, we will design a simple, efficient, real-time monocular visual tracking system in this paper.

The remainder of this paper is organized as follows. Section 2 describes the target detection methods which are operated in simple and complex environment. The trajectory tracking algorithm is illustrated in Section 3. The implementation of these methods and the experimental results are presented in Section 4. Finally, conclusions are drawn in Section 5.

2. TARGET DETECTION METHODS

2.1 Target Detection in Simple Environment

Detecting the target by finding the centroid of edges is prone to fail as shown in Fig. 1. To solve this problem, we calculate the centroid of edges in a small attention window instead of the whole image as shown in Fig. 2. But at first we have to determine the location and the size of the attention window.

To determine the location of the attention window is critical for containing the target. Therefore, we use the direction of optical flow to provide the location of the attention window, which is the so-called optical flow based attention window as shown in Fig. 3. In particular, for fast sampling rate systems, the current centroid of the target can be used as the center location of the attention window in the next frame. Then we determine the size of the attention window. Assume the size of the attention window is $m \times n$, and the target has a maximum velocity $\mathbf{v}_m$. Then the size of attention window can be determined as follow:

$$m \geq |u_m| \quad \text{and} \quad n \geq |v_m| \tag{1}$$

where $[u_m \ \ v_m]^T$ is the optical flow induced by $\mathbf{v}_m$. The concept of appropriately determining the size of the attention window not only reduces the computation time but also makes this method more robust.

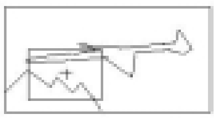


Fig. 1. Finding centroid of edges in whole image.

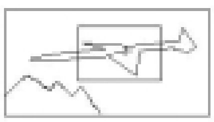


Fig. 2. Finding centroid of edges in attention window.

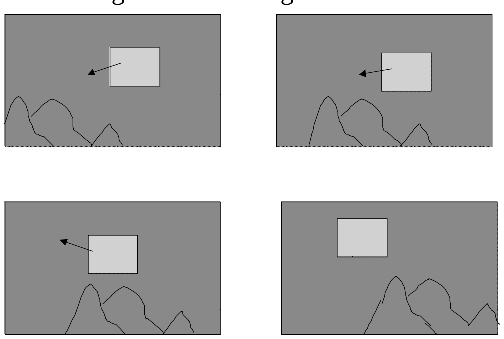


Fig. 3. Optical flow based attention window.

2.2 Target Detection in Complex Environment

The SSD measure having the moving edges as candidate is used to detect the target in complex environment. The method of finding the moving edges is based on motion energy detection [29] except for the background compensation algorithm. The background compensation algorithm used in [29] is based on Kanatani's relationship which is valid only when the pan/tilt angles between successive frames are not larger than 3°.

We propose another background compensation algorithm that can compensate for arbitrary pan/tilt angles if the object is still visible in the current frame after pan/tilt motion. Denote Δx and Δy be the displacement in x and y direction respectively. From [25], Δx and Δy are

$$\Delta x = f \tan(\phi) \tag{2}$$

$$\Delta y = [-\Delta x \sin(\phi) - f \cos(\phi)] \tan(\theta) \tag{3}$$

where ϕ and θ are the pan and tilt angles. By using the actual coordinate system $\{F_a\}$ in Fig. 4, the relationship between pixel position in the previous and the current frame is :

$$x_{t-1} = x_t - \Delta x \tag{4}$$

$$y_{t-1} = y_t - \Delta y \tag{5}$$

where $[x_{t-1} \ \ y_{t-1}]^T$ and $[x_t \ \ y_t]^T$ are the pixel position in the previous and the current frame. Figure 5 and Fig. 6 show image subtraction with and without the proposed background compensation algorithms. Executing a logical AND operation between subtracted image and edge image can detect the moving edges as shown in Fig. 7.

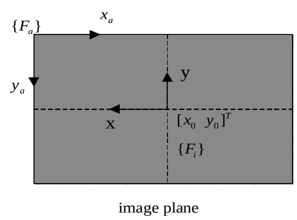


Fig. 4. Image coordinate system.

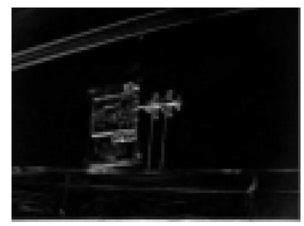


Fig. 5. Image subtraction with compensation.

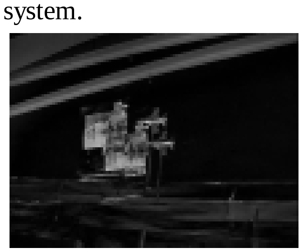


Fig. 6. Image subtraction without compensation.

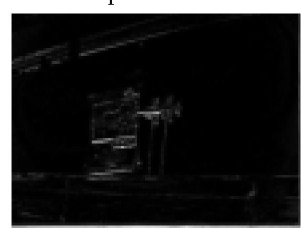


Fig. 7. Moving edges in 3.5° pan and tilt angles.

In the SSD algorithm, assume the attention window size is $m \times n$ and the template size is $T_a \times T_b$, then the template is shifted $(m - T_a) \times (n - T_b)$ times. Obviously the computation is redundant if we have knowledge about the location of the target in the attention window. In light of this observation, moving edges are chosen as the candidates for the SSD measure. Let set $\Im$ of the candidates be the set of points:

$$\Im = \{\mathbf{x} = [x \ \ y]^T \mid I_m(x, y) > 0\} \tag{6}$$

where $I_m(x, y)$ is the brightness value of the result of motion energy detection. Thus, any $\mathbf{x} \in \Im$ is a candidate for the SSD measure. Obviously, the number of elements which belong to the set $\Im$ is less than $(m - T_a) \times (n - T_b)$. Accordingly, not only the computation time is reduced significantly, but also the noise is reduced and thus the confidence of SSD measure is increased.

If the object is static, then this concept can be extended by using the edges as the candidates for the SSD measure. Similarly, define the set

$$\Im_e = \{\mathbf{x} = [x \;\; y]^T \mid \;\; \mathrm{G}(x, y) > 0\} \qquad (7)$$

where $G(x, y)$ is the gradient magnitude at the location $\mathbf{x}$ in the attention window. Thus, any $\mathbf{x} \in \Im_e$ is a candidate for the SSD measure.

Table 1 shows the comparison of the processing time among these methods.

Table 1. The processing time among the different target detection methods.

Different methods / Processing time	SSD	SSD having moving edges as candidate	SSD having edges as candidate
In simple environment	540 ms	30 ms	58 ms
In complex environment	550 ms	30 ms	228 ms

3. DESIGN OF TRAJECTORY TRACKING ALGORITHM

We will make clear our main concept and notion for designing the trajectory tracking algorithm in this section. Our goal here is to design a simple, none computation intensive, and numerically stable algorithm. Our main idea is that if the target motion is known, then the camera motion which will eliminate the relative motion of the target may be generated such that the tracking is achieved. In particular, if the motion of target is known well, then the "perfect tracking" may be achieved, i.e., image position of the target can always be kept at the center of image.

It is apparent that the estimation of the target motion is the first step in our proposed algorithm. In general, the motion of target is not known a *priori*. However, it can be estimated through the change of the image position which is called "optical flow" or image displacement. Using the observation of image displacement or optical flow which induced by target motion, one can estimate the motion of target if the camera is stationary. However it is not the case for the tracking phase since both the camera and target are moving during the tracking phase and therefore the optical flow is subject to both camera and target motion. In light of this observation, we must modify the optical flow equation such that the optical flow is induced only by the target motion and irrelevant to the camera motion.

We can then estimate the target motion by using the modified optical flow. Motion estimation task becomes peculiarly difficult in trajectory tracking since neither the structure (depth) nor the target motion is known. Accordingly, we proposed a motion estimation algorithm which is independent of the estimation of structure. In other words, our algorithm is capable to decouple the estimation of motion from the estimation of structure. Thus, the structure (depth) need not be known or estimated before solving the motion estimation task. To this end, the weak perspective projection which is used to alleviate this problem, gives a good approximation when the size and the depth variation of the object are small compared with the distance between the object and the camera.

The second step in our proposed algorithm is prediction. The prediction task is to predict where the target image location in the current frame will 'move to' in the next frame. The last step is tracking. Using the predicted target position, we can then calculate a feasible camera motion in order to track the moving target. The whole process of these algorithms makes up of three steps and the flow chart is shown in Fig. 10. At any time instant k, the target image position at this time instant k and at the previous time instant $k-1$ are as the input of the motion estimation module. Then, it output the motion parameters to the prediction and filtering module. After that, the prediction and filtering module will calculate the predicted target image position in the next frame and eventually the tracking module generate the desired camera motion which will achieve tracking.

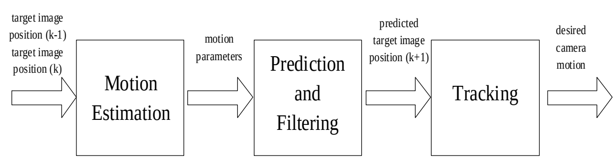


Fig. 8. Flow chart of our proposed algorithm.

For the lack of the space, the equation of this algorithm is summarized in Fig. 11 and the details can be found in [31].

Trajectory Tracking Algorithm

1. The target induced optical flow is calculated as follow:

$$u_o(k-1) = x(k) - x(k-1) + \left(x_{\text{predict}}(k-1) - x_o\right)$$

$$v_o(k-1) = y(k) - y(k-1) + \left(y_{\text{predict}}(k-1) - y_o\right)$$

where $[x_o \;\; y_o]^T$ are the image center and $[x_{\text{predict}}(k-1) \;\; y_{\text{predict}}(k-1)]^T$ are the predicted target image position.

2. Select three different edge points $[x_1 \;\; y_1]^T$, $[x_2 \;\; y_2]^T$, $[x_3 \;\; y_3]^T$ which lie around the centroid of target and use the following equation to estimate the target motion parameters.

$$\mathbf{v} = \mathbf{G}^{-1}\mathbf{c}$$

where

$$\mathbf{G} = \begin{bmatrix} f & 0 & -x_1(k-1) & -\frac{x_1(k-1)y_1(k-1)}{f} & \frac{x_1^2(k-1)+f^2}{f} & -y_1(k-1) \\ 0 & f & -y_1(k-1) & -\frac{y_1^2(k-1)+f^2}{f} & \frac{x_1(k-1)y_1(k-1)}{f} & x_1(k-1) \\ f & 0 & -x_2(k-1) & -\frac{x_2(k-1)y_2(k-1)}{f} & \frac{x_2^2(k-1)+f^2}{f} & -y_2(k-1) \\ 0 & f & -y_2(k-1) & -\frac{y_2^2(k-1)+f^2}{f} & \frac{x_2(k-1)y_2(k-1)}{f} & x_2(k-1) \\ f & 0 & -x_3(k-1) & -\frac{x_3(k-1)y_3(k-1)}{f} & \frac{x_3^2(k-1)+f^2}{f} & -y_3(k-1) \\ 0 & f & -y_3(k-1) & -\frac{y_3^2(k-1)+f^2}{f} & \frac{x_3(k-1)y_3(k-1)}{f} & x_3(k-1) \end{bmatrix}$$

$$\mathbf{v} = \begin{bmatrix} \frac{T_x(k-1)}{Z_c(k-1)} \\ \frac{T_y(k-1)}{Z_c(k-1)} \\ \frac{T_z(k-1)}{Z_c(k-1)} \\ \omega_x(k-1) \\ \omega_y(k-1) \\ \omega_z(k-1) \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} u_0^1(k-1) \\ v_0^1(k-1) \\ u_0^2(k-1) \\ v_0^2(k-1) \\ u_0^3(k-1) \\ v_0^3(k-1) \end{bmatrix}.$$

3. Evaluate the predicted target centroid coordinate by assuming $\mathbf{T}(k) \approx \mathbf{T}(k-1)$, $\boldsymbol{\omega}(k) \approx \boldsymbol{\omega}(k-1)$, and $Z(k) \approx Z(k-1)$ for short sampling time as follow:

$$\begin{bmatrix} u_o(k) \\ v_o(k) \end{bmatrix} = \begin{bmatrix} f & 0 & -x_c(k) \\ 0 & f & -y_c(k) \end{bmatrix} \begin{bmatrix} \frac{T_x(k)}{Z(k)} \\ \frac{T_y(k)}{Z(k)} \\ \frac{T_z(k)}{Z(k)} \end{bmatrix} + \begin{bmatrix} -\frac{x_c(k)y_c(k)}{f} & f+\frac{x_c^2(k)}{f} & -y_c(k) \\ -\left(f+\frac{y_c^2(k)}{f}\right) & \frac{x_c(k)y_c(k)}{f} & x_c(k) \end{bmatrix} \begin{bmatrix} \omega_x(k) \\ \omega_y(k) \\ \omega_z(k) \end{bmatrix}$$

$$x_c(k+1) = x_c(k) + u_o(k)$$

$$y_c(k+1) = y_c(k) + v_o(k)$$

where $[x_c \;\; y_c]^T$ is the target centroid coordinate in image plane.

4. Use the following equations to compute the accommodated tracking command:

$$\phi = \tan^{-1}\left(\frac{x_o - x_c(k+1)}{f}\right),$$

$$\theta = \tan^{-1}\left(\frac{y_o - y_c(k+1)}{-\left(x_{org} - x_c(k+1)\right)\sin\phi - f\cos\phi}\right),$$

where ϕ is the pan angle and θ is the tilt angle.

Fig. 9. Trajectory Tracking Algorithm

4. IMPLEMENTATION AND EXPERIMENTS

This section illustrates the performance of the trajectory tracking algorithms under a variety of circumstances. All experiments were performed on live image sequences which were grabbed by a JAI MCL-1500 DSP color camera and then processed by the MATROX CORONA image processing card in the Pentium III-450 PC. The control command is then sent to the ADVANTECH stepping motor control card so as to generate an appropriate pan/tilt motion that achieves the goal of tracking.

4.1 Validation of the Trajectory Tracking Algorithm

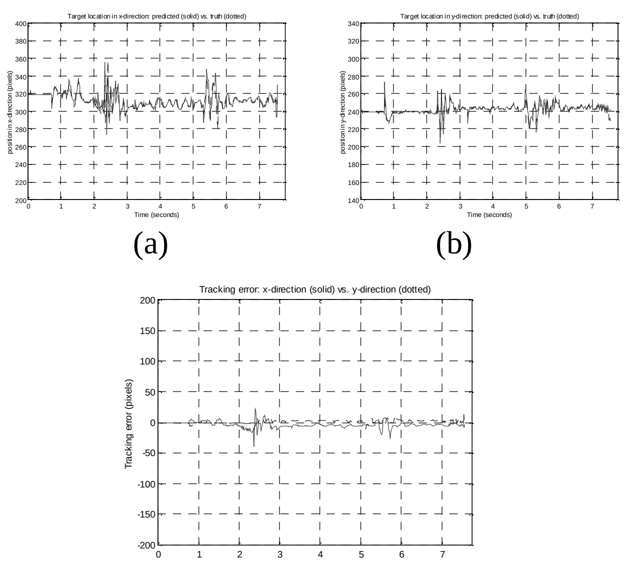
To validate the performance of the proposed trajectory tracking algorithm, we will use the target detection method in simple environment as described in Section 2. Timing of the control cycle indicates that the proposed visual tracking system can perform frame rate (30 Hz) tracking of image regions with size 640×480 pixels.

- **Indoor Experiment**

An aircraft model is used as our target. At any time instant, the location of the target in the image plane is fed to the trajectory tracking algorithm to predict the future position as shown in Fig. 10(a) and 10(b). The tracking error is shown in Fig. 10(c). Note that the center of the image plane is $[320 \;\; 240]^T$. It is clear that the predicted target position is close to the true target position. This implies that our proposed algorithm provides accurate prediction of the maneuvering target trajectory.

- **Outdoor Experiment**

We use a remote controlled helicopter as our target. The helicopter is first near the camera. Then, it flies away, turns around and flies back toward the camera as shown in Fig. 11. The data of these image sequences are shown in Fig. 12. It demonstrates that the algorithm is robust against fast varying target size. Finally, it is interesting to notice Fig. 12(a) and 12(b), which show that the predicted target position well matches the motion of target.

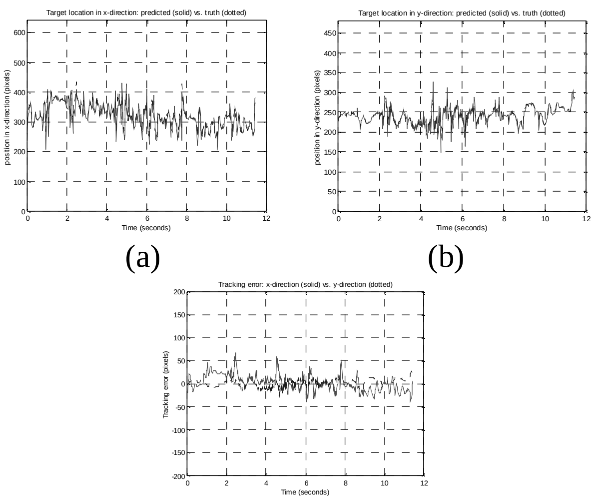


(a) (b)

(c)

Fig. 10. Experimental results of indoor experiment.

Fig. 11. Image sequence.



(a) (b)

(c)

Fig. 12. Experimental results of outdoor experiment.

4.2 Target Tracking in Complicated Scene

Image sequences in Fig. 13 demonstrate that we can successfully track an aircraft model which passes through highly cluttered scenes.

Fig. 13. Complicated environment scene image sequence.

5. CONCLUSION

In this paper, we have proposed a real-time visual tracking system. The whole system consists of a target detection method and a trajectory tracking algorithm.

We have proposed two different target detection methods which operate in simple and complex environment, respectively. In simple environment, the centroid of edges in the attention window is recognized as the location of target. The proposed method has been proved robust in general conditions if the target is not occluded by other objects. On the other hand, the SSD measure which has the moving edges as candidate is used to detect the target in the complex environment. The capability of tracking in highly cluttered scenes has been validated by the experiment. The computational burden of both algorithms is modest such that our system can be implemented real-time.

The centroid of target which is detected by the proposed target detection method is used as the tracking information which is needed by the trajectory tracking algorithm. The trajectory tracking algorithm based on proposed motion estimation algorithm provides accuracy of the prediction of the trajectory of the maneuvering target. The robustness of proposed visual tracking system is validated by a number of experiments.

REFERENCES

[1] G. Adiv, “Determining Three-Dimensional Motion and Structure from Optical Flow Generated by Several Moving Objects,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. PAMI-7, no. 4, pp. 384-401, Jul 1985.

[2] S. Soatto, R. Frezza, and P. Perona, “Motion Estimation via Dynamic Vision,” *IEEE Trans. Automatic Control*, vol. 41, no. 3, pp. 393-413, Mar 1996.

[3] Robert J. Schilling, *Fundamentals of Robotics Analysis and Control*, Englewood Cliffs, N.J. : Prentice-Hall, c1990.

[4] Y. Bar-Shalom, K. C. Chang and H. A. P. Blom, “Tracking a Maneuvering Target Using Input Estimation Versus the Interacting Multiple Model Algorithm”, *IEEE Trans. on Aero. and Electro. Sys.*, vol. AES-25, no. 2, pp. , Mar 1989.

[5] N. Sawasaki, T. Morita, and T. Uchiyama, “Design and Implementation of High-Speed Visual Tracking Systems for Real-Time Motion Analysis,” *IEEE Int. Conf. Pattern Recognition*, vol. 3, 478-483, 1996.

[6] A. Blake, R. Curwen, and A. Zisserman, “A Framework for Spatiotemporal Control in the Tracking of Visual Contours,” *Int. J. Computer Vision*, vol. 11, no. 2, pp. 127-145, 1993.

[7] Andrew Blake and Michael Isard, *Active Contours*,

London : Springer-Verlag, c1998.
[8] H. A. P. Blom and Y. Bar-Shalom, “The Interacting Multiple Model Algorithm for Systems with Markovian Switching Coefficients”, *IEEE Trans. Automatic Control*, vol. 33, no. 8, pp. 780-783, Aug 1988.
[9] K. J. Bradshaw, I. D. Reid, and D. W. Murray, “The Active Recovery of 3D Motion Trajectories and Their Use in Prediction,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 3, pp. 219-233, Mar 1997.
[10] C. Cedra and M. Shah, “Motion-based recognition: a survey,” *Image and Vision Computing*, vol. 13, no. 2, pp. 129-155, Mar 1995.
[11] L. Chen and S. Chang, “A Video tracking system with adaptive predictors,” *Pattern Recognition*, vol. 25, no. 10, pp. 1171-1180, Oct 1992.
[12] B. Sabata and J. K. Aggarwal, “Estimation of Motion from a Pair of Range Images: A Review,” *CVGIP: Image Understanding*, vol. 54, no. 3, pp. 309-324, Nov 1991.
[13] Z. Duric, J. A. Fayman, and E. Rivlin, “Function From Motion,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 6, pp. 579-591, June 1996.
[14] J. T. Feddema, C. S. G. Lee, “Adaptive image feature prediction and control for visual tracking,” *IEEE Trans. Systems, Man and Cybernetics*, vol. 20, no. 5, pp. 1172-1183, Sept-Oct 1990.
[15] J. T. Feddema, C. S. G. Lee, “Adaptive image feature prediction and control for visual tracking with a moving camera,” *Proc. IEEE Int. Conf. Systems, Man and Cybernetics*, pp. 20-24, Nov 1990.
[16] D. B. Gennery, “Visual Tracking of Known Three-Dimensional Objects,” *Int. J. Computer Vision*, vol. 7, no. 3, pp. 243-270, 1992.
[17] G. D. Hager and P. N. Belhumeur, “Efficient Region Tracking With Parametric Models of Geometry and Illumination,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 10, pp. 1025-1039, Oct 1998.
[18] D. J. Heeger and A. D. Jepson, “Subspace Methods for Recovering Rigid Motion I: Algorithm and Implementation,” *Int. J. Computer Vision*, vol. 7, no. 2, pp. 95-117, 1992.
[19] Horace H. S. Ip and D. G. Shen, “An affine-invariant active contour model (AI-snake) for model-based segmentation,” *Image and Vision Computing*, vol. 16, pp. 135-146, 1998.
[20] T. S. Jebara and A. Pentland, “Parametrized Structure from Motion for 3D Adaptive Feedback Tracking of Faces,” *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pp. 144-150, 1997.
[21] T. Joshi, “Structure and Motion Estimation from Dynamic Silhouettes under Perspective Projection,” *Int. J. Computer Vision*, vol. 31, no. 1, pp. 31-50, 1999.
[22] J. S. Park and J. H. Han, “Contour matching: a curvature-based approach,” *Image and Vision Computing*, vol. 16, pp. 181-189, 1998.
[23] N. P. Papanikolopoulos and P. K. Khosla, “Adaptive Robotic Visual Tracking: Theory and Experiments,” *IEEE Trans. Automatic Control*, vol. 38, no. 3, pp. 429-445, Mar 1993.
[24] S. Nassif and D. Capson, “Real-Time Template Matching Using Cooperative Windows,” *IEEE Canadian Conf. Electrical and Computer Engineering*, vol. 2, pp. 391-394, 1997.
[25] D. Maravall and L. Baumela, “Adaptive control of a video camera for the automatic detection and Tracking of Mobiles,” *Proc. IEEE Int. Conf. Systems, Man and Cybernetics*, pp. 53-58, Oct 1993.
[26] L. Matthies, T. Kanade, and R. Szeliski, “Kalman Filter-based Algorithms for Estimating Depth from Image Sequences,” *Int. J. Computer Vision*, vol. 3, pp. 209-236, 1989.
[27] E. Mazor, A. Averbuch, Y. Bar-Shalom and J. Dayan, “Interacting Multiple Models methods in Target Tracking: A Survey”, *IEEE Trans. Aero. and Electro. Sys.*, vol. 34, no. 1, Jan 1998.
[28] A. Munir & D. P. Atherton, “Maneuvring target tracking using different turn rate models in the interacting multiple model algorithm,” *Proc. 34th IEEE Conf. Decision & Control New Orleans*, pp. 2747-2751, Dec 1995.
[29] D. Murray and A. Basu, “Motion Tracking with an Active Camera,” *IEEE Trans. Pattern Analysis and Intelligence*, vol. 16, no. 5, pp. 449-459, May 1994.
[30] D. Murray and A. Basu, “Active Tracking,” *Proc. 1993 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems Yokohama*, Japan, pp. 1021-1028, Jul 1993.
[31] W. G. Yau, “Design and Implementation of Visual Servoing System for Realistic Air Target Tracking,” Master’s thesis, Dept. Elec. Eng., National Taiwan University, 2000.