A Kalman Filter Based Visual Tracking Algorithm for an Object Moving in 3D

Joon Woong Lee*, Mun Sang Kim**, and In So Kweon*,Δ
*Department of Automation and Design Engineering,
Korea Advanced Institute of Science and Technology.
207-43, Cheongryangridong, Dongdaemoongu, Seoul, KOREA.
Internet kweon @design.kaist.ac.kr
**Korea Institute of Science and Technology
Δ Member of ERC_ACI, Seoul National University

Abstract

Robust and effective real-time visual tracking is realized by combining the first order differential invariants with the stochastic filtering. The Kalman filter as an optimal stochastic filter is used to estimate the motion parameters, namely the plant state vector of the moving object with the unknown dynamics in successive image frames. Using the fact that the relative motion between the moving object and the moving observer causes the image deformation, we compute the first order differential invariants of the image velocity field. The surface orientation and the depth estimate between the observer and the object are computed based on these first order differential invariants.

We demonstrate the robustness and feasibility of the proposed tracking algorithm through real experiments in which an X-Y Cartesian robot tracks a toy vehicle moving along 3D rails.

1. Introduction

Processing of a stream of many image frames is indispensable for many computer vision algorithms to accomplish given tasks in dynamic environments. Image sequence analysis provides intermediate results for a conceptual description of events in a scene. We introduce a novel algorithm for tracking a rigid object moving under three dimensional motion with unknown dynamics using only monocular image. Specifically, the first order differential invariants of the image velocity field combined with the stochastic filtering is used to estimate the surface normal and the depth from the viewer to the object moving in 3D. Visual tracking has two inherent problems[3]: 1)the system or plant has the unknown exogenous disturbances, 2)the sensor for observing the system state has also noises. These are two main reasons why the Kalman filtering is introduced to track the object moving in 3D. In addition, our algorithm utilizes the property of image deformation occurred due to the relative motion between the observer and the object[2]. This deformation can be described by the first order differential invariants of the image velocity field - curl, divergence and pure shear components. The surface orientation and the depth estimate between the observer and the object are computed based on these first order differential invariants. In the original work of Cipolla[2] deliberate motion is endowed to the observer under the static scene and observed the image deformation. However, we don't know the motion information in advance. Because the target moves with unknown dynamics and observer also moves to track this target passively. Even though the relative motion is existed it's not easy to extract the first order differential invariants due to concurrent motion of observer and target. Kalman filtering plays the important role of solving these problems.

In this article we take advantages of the geometric properties of a triangle instead of direct usage of the detected image feature points. First, we construct a network of triangles using detected feature points of an image frame. Second, we compute the center of gravity and the higher order shape moments of a moving object. The shape moments are then used in calculating the first order differential invariants and the principal axis. The center of gravity, the first order differential invariants and the principal axis play an important role for estimating the system state vector of the Kalman filtering.

2. Models for the motion and measurement of an object moving in 3D

2.1. Motion model

0-8186-7108-4/95 $4.00

Let's assume that a target and a tracker moves concurrently. Therefore, the depth from the viewer to the target and the viewed surface orientation varies as the target moves. Furthermore the velocity of the target changes linearly and rotationally and is not constant in practice. However, we assume the target moves smoothly. $\mathbf{P}(t) = (x(t), y(t), z(t))^T$ represents the target position with respect to camera coordinate system with its origin **Oc**, and $\phi(t)$ is the orientation angle between the x-axis of the camera coordinate system and the principal axis of the target in Fig. 1. We define the linear velocity of the target, $\mathbf{v} = (\dot{x}, \dot{y}, \dot{z})^T$ along the each axis of camera coordinate system, the rotational velocity of the target $\omega = (\omega_1, \omega_2, \omega_3)^T$ about the each axis of camera coordinate system, surface normal vector of the target, $\mathbf{n} = (n_1, n_2, n_3)$, slant σ which is an angle between surface normal and visual direction, and tilt τ of the surface tangent plane.

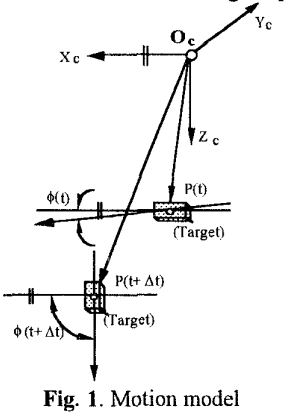


Fig. 1. Motion model

The system state vector $\mathbf{x}_k$ at time t_k is given as :

$$\mathbf{x}_k = (x_k, y_k, z_k, \dot{x}_k, \dot{y}_k, \dot{z}_k, \omega 1_k, \omega 2_k, \omega 2_k, n1_k, n2_k, n3_k, \sigma_k, \tau_k)^T \quad (1)$$

and the system model which describes the state transition from t_k to t_{k+1} as :

1. Position vector

$$\mathbf{P}(k+1) = \mathbf{P}(k) + \dot{\mathbf{P}}(k)\Delta t \quad (2)$$

2. Linear velocity

$$\dot{\mathbf{P}}(k+1) = \mathrm{Rot}(z,\theta)\mathrm{Rot}(y,\phi)\mathrm{Rot}(x,\varphi)\dot{\mathbf{P}}(k)$$

$$= \begin{bmatrix} c\theta c\phi & c\theta s\phi s\varphi - s\theta c\varphi & c\theta s\phi c\varphi + s\theta s\varphi & 0 \\ s\theta c\phi & s\theta s\phi s\varphi + c\theta c\varphi & s\theta s\phi c\varphi - c\theta s\varphi & 0 \\ -s\phi & c\phi s\varphi & c\phi c\varphi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}(k) \\ \dot{y}(k) \\ \dot{z}(k) \\ 1 \end{bmatrix} \quad (3)$$

where c stands for cos and s sin.

3. Rotational velocity

$$\omega(k+1) = \omega(k) \quad (4)$$

4. Surface normal vector

$$\mathbf{n}(k+1) = \begin{bmatrix} 0 & n_3(k) & -n_2(k) & n_1(k) \\ -n_3(k) & 0 & n_1(k) & n_2(k) \\ n_2(k) & -n_1(k) & 0 & n_3(k) \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \varphi \\ \phi \\ \theta \\ 1 \end{bmatrix} \quad (5)$$

5. Surface slant and tilt

$$\begin{bmatrix} \sigma(k+1) \\ \tau(k+1) \end{bmatrix} = \begin{bmatrix} -\sin\tau(k) & \cos\tau(k) & 0 & \sigma(k) \\ -\dfrac{\cos\tau(k)}{\tan\sigma(k)} & -\dfrac{\sin\tau(k)}{\tan\sigma(k)} & 1 & \tau(k) \end{bmatrix} \begin{bmatrix} \varphi \\ \phi \\ \theta \\ 1 \end{bmatrix} \quad (6)$$

where $\theta = \omega_3(k)\Delta t$, $\phi = \omega_2(k)\Delta t$, and $\varphi = \omega_1(k)\Delta t$, $\Delta t = t_{k+1} - t_k$.

By denoting the state transition function by $\mathbf{f}(\cdot)$ and adding the system random noise $\mathbf{w}_k \sim N(0, \mathbf{Q}_k)$, we can express the system model as : $\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{w}_{k-1}$. (7)

Because this model is non-linear, the first order Taylor series expansion is necessary to obtain a state transition matrix, Φ_{k-1} at $\mathbf{x} = \hat{\mathbf{x}}_{k-1}^+$[3]. Then the predicted state estimate, $\hat{\mathbf{x}}_k^-$ and the error covariance, $\mathbf{P}_k^-$ are obtained as follows :

$$\hat{\mathbf{x}}_k^- = \mathbf{f}(\hat{\mathbf{x}}_{k-1}^+) \quad (8)$$

$$\mathbf{P}_k^- = \Phi_{k-1}\mathbf{P}_{k-1}^+\Phi_{k-1}^T + \mathbf{Q}_{k-1} \quad (9)$$

where,

$$\Phi_{k-1} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{k-1}^+} \quad (10)$$

2.2. Measurement model

Since we use CCD camera as a sensor for observing the system state, the designing of a measurement model is the same as the modeling of this camera. The ideal pin-hole camera model is considered as the measurement model and a target is projected on the image plane by perspective transformation. Computer image coordinate X_f and Y_f are defined as follows:

$$X_f = \frac{1}{S_x}\frac{f \cdot x}{z} + C_x, \quad Y_f = \frac{1}{S_y}\frac{f \cdot y}{z} + C_y. \quad (11)$$

where, (S_x, S_y) are camera parameters, and (C_x, C_y) is the center of computer frame memory. And we also measure the first order differential invariants, divergence(an isotropic expansion) specifying the change in scale or size, divv, and a pure shear describing the distortion of the image shape, defv, and the orientation of the axis of expansion μ. From these first order differential invariants we can obtain the *time to*

contact tc and the surface normal vector **n** as follows[2]:

$$t_c = \frac{2}{\mathrm{div}\mathbf{v} - \mathrm{def}\mathbf{v}\cos(2\mu - 2\angle\mathbf{A})} \quad (12)$$

where $\angle$A represents the direction of observer translation.

$$\mathbf{n} = (\sin\sigma\cos\tau, \sin\sigma\sin\tau, \cos\sigma)^T \quad (13)$$

where σ and τ represent the surface slant and tilt respectively.

According to Eqs.(11)-(13) we can construct the measurement function $h(\mathbf{x}_k)$ as follows:

$$h(\mathbf{x}_k) = (X_f, Y_f, \dot{z}, \omega_3, n_1, n_2, n_3)^T \quad (14)$$

ω_3 is measured from central moments of target. Finally, the measurement model of Kalman filter is obtained by adding the measurement random noise $\mathbf{v}_k \sim \mathrm{N}(0, \mathbf{R}_k)$ to $h(\mathbf{x}_k)$ as:

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k. \quad (15)$$

Because this model is also non-linear, the first order Taylor series expansion is necessary to obtain a measurement matrix, $\mathbf{H}_k$ at $\mathbf{x} = \hat{\mathbf{x}}_k^-$[3]. Then the filtered state estimate, $\hat{\mathbf{x}}_k^+$ and error covariance , $\mathbf{P}_k^+$ are obtained as follows :

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k[\mathbf{z}_k - h(\hat{\mathbf{x}}_k^-)] \quad (16)$$

$$\mathbf{P}_k^+ = [\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]\mathbf{P}_k^- \quad (17)$$

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^{\mathrm{T}}[\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k]^{-1} \quad (18)$$

where $\mathbf{H}_k = \left.\frac{\partial h}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_k^-}$

3. Surface orientation and velocity with respect to optical-axis

3.1. Computing shape moments

In this section a new method of computing shape moments based on a triangle is introduced. First. triangular grid is made by using extracted feature points through the triangulation algorithm. Then the general two-dimensional (p+q)$^{\text{th}}$ order moments of a continuous density function $f(x,y) = 1$ are defined as

$$m_{pq} = \iint_R x^p y^q f(x,y)dxdy \quad p,q = 0,1,2,\cdots. \quad (19)$$

We use Eq.(19) to represent the (p+q)$^{\text{th}}$ order moments of a triangle. The subdivision of a triangle into R_1 and R_2 is required as shown in Fig. 2. Then the (p+q)$^{\text{th}}$ order moments of a triangle are defined by

$$m_{pq} = \iint_{R1} x^p y^q dxdy + \iint_{R2} x^p y^q dxdy. \quad (20)$$

For R_1 and R_2, Eq.(20) is described as

$$\iint_{R_1} x^p y^q dxdy = \int_{x_1}^{x_2} x^p \left[\int_{f_2(x)}^{f_1(x)} y^q dy\right] dx. \quad (21)$$

$$\iint_{R_2} x^p y^q dxdy = \int_{x_2}^{x_3} x^p \left[\int_{f_2(x)}^{f_3(x)} y^q dy\right] dx. \quad (22)$$

where $f_1(x)$, $f_2(x)$, and $f_3(x)$ are the equations of each side of a triangle.

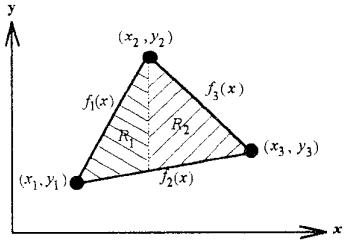


Fig. 2. Subdivision of a triangle

we calculate the shape moments of each triangle which is included in triangular grid of a target and then for all triangles we sum corresponding moments as follows :

$$M_{pq} = \sum_{i=1}^{n} m_{pq} \quad (23)$$

where n is the number of triangles of a target.

Let the center of the gravity of a target as $(\bar{x}, \bar{y})$

$$\bar{x} = \frac{M_{10}}{M_{00}}, \quad \bar{y} = \frac{M_{01}}{M_{00}}. \quad (24)$$

Then the central moments can be expressed as[4]

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q \quad x,y \in R \text{ and } p,q = 0,1\cdots. \quad (25)$$

3.2. Extraction of the first order differential invariants

Using the central moments we can extract the first-order spatial derivatives of image flow $\mathbf{v} = (u,v)$, at an image point $(\bar{x}, \bar{y})$ by[2]

$$\frac{d}{dt}\begin{bmatrix}\mu_{00}\\ \mu_{10}\\ \mu_{01}\\ \mu_{20}\\ \mu_{02}\\ \mu_{31}\end{bmatrix} = \begin{bmatrix}0 & 0 & \mu_{00} & 0 & 0 & \mu_{00}\\ \mu_{00} & 0 & 2\mu_{10} & \mu_{01} & 0 & \mu_{10}\\ 0 & \mu_{00} & \mu_{01} & 0 & \mu_{10} & 2\mu_{01}\\ 2\mu_{10} & 0 & 3\mu_{20} & 2\mu_{11} & 0 & \mu_{20}\\ 0 & 2\mu_{01} & \mu_{02} & 0 & 2\mu_{11} & 3\mu_{02}\\ 3\mu_{21} & \mu_{30} & 4\mu_{31} & 3\mu_{22} & \mu_{40} & 2\mu_{31}\end{bmatrix}\begin{bmatrix}u_0\\ v_0\\ u_x\\ u_y\\ v_x\\ v_y\end{bmatrix} \quad (26)$$

In this equation we can notice $u_0 = v_0 = 0$. After the extraction of u_x, u_y, v_x, and v_y divergence, curl, pure shear magnitude, and the direction of maximum expansion can be determined.

3.3. Recovery of the viewed surface normal vector

For any successive two image frames, we extract the center of gravity and the first order spatial derivatives of image velocity field and compute the first order differential invariants. And then we compute the surface normal vector as follows :

1)Let observer translation vector $\mathbf{A} = (A_1, A_2)$

2)Let the center of gravity of the first and second frame in consecutive images $(\overline{X}_f(1), \overline{Y}_f(1))$, $(\overline{X}_f(2), \overline{Y}_f(2))$ respectively, then

$$\Delta\overline{X}_f = \overline{X}_f(1) - \overline{X}_f(2), \quad \Delta\overline{Y}_f = \overline{Y}_f(1) - \overline{Y}_f(2) \quad (27)$$

3)Let u_0, v_0 of pure translation at the center of gravity as :

$$u_0 = \frac{\Delta X_f}{\Delta t} = -A_1 + \omega_2, \quad v_0 = \frac{\Delta Y_f}{\Delta t} = -A_2 - \omega_1 \quad (28)$$

4) Then we obtain

$$\angle\mathbf{A} = \tan^{-1}\frac{A_2}{A_1}, \quad |\mathbf{A}| = \sqrt{A_1^2 + A_2^2}. \quad (29)$$

4)The *slant* and *tilt* are given by

$$\tan\sigma = \frac{\text{defv}}{|\mathbf{A}|}, \quad \tau = 2\mu - \angle\mathbf{A}. \quad (30)$$

Therefore the unit surface normal vector **n** is calculated using Eq.(14).

3.4. Recovery of the translational and rotational velocity with respect to optical axis

In the camera coordinate system, z-axis is aligned with the optical axis. Therefore, the translational velocity of the target along the z-axis is

$$\dot{\mathbf{z}} = \frac{\text{divv} - \text{defv}\cos(2\mu - 2\angle\mathbf{A})}{2} z \quad (31)$$

The predicted estimate $\hat{z}_k^-$ of z is used in Eq.(31),. Ultimately, $\dot{\mathbf{z}}$ of Eq.(31) becomes measurement value in Kalman filtering. And the rotational velocity about the optical axis is computed with principal axis which is defined by

$$\theta_s = \frac{1}{2}\tan^{-1}\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}. \quad (32)$$

For any successive two image frames, let θ_s of the first and second frame be $\theta_s(1)$ and $\theta_s(2)$. Then the rotational velocity about the optical axis is

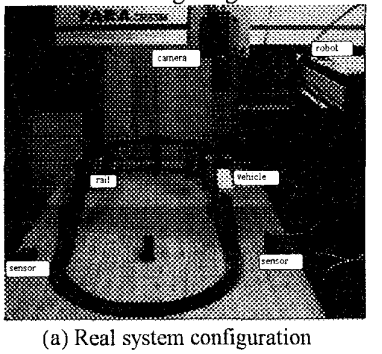
$$\omega_3 = \frac{\Delta\theta_s}{\Delta t}, \quad \Delta\theta_s = \theta_s(2) - \theta_s(1). \quad (33)$$

In the long run, ω_3 also becomes measurement value in Kalman filtering.

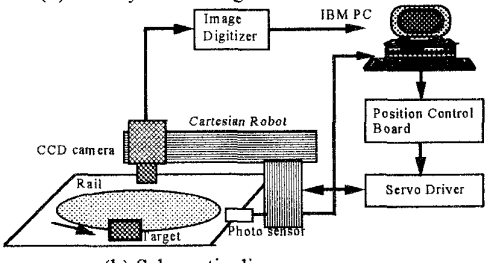
4 Experiments

4.1. System configuration

In this study, the system is composed of the Cartesian robot as a tracker, CCD camera attached to the robot arm to observe a target motion, two photo-electric switches which perceive a target and signal to the robot to start and finish the tracking, a toy vehicle which moves along the three dimensionally shaped rail, and a standard IBM PC-486 DX-33 with an image digitizer. Fig. 3 shows this configuration. The system starts as the robot goes to home-position and the tracking is beginning as soon as arriving the signal from the photo-electric switch-1. The robot continues to track according to the position vector estimated from the prediction phase of Kalman filtering as shown in Fig. 4. The tracking is continued until the other photo sensor-2 signals the target departs from the available working range of robot.



(a) Real system configuration



(b) Schematic diagram

Fig. 3. Conceptual system configuration

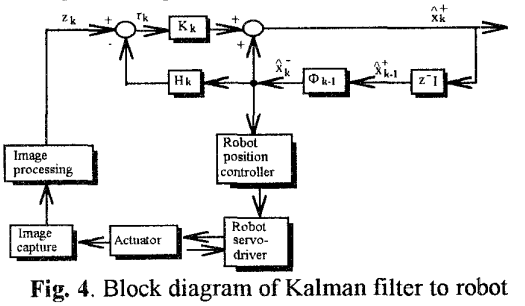
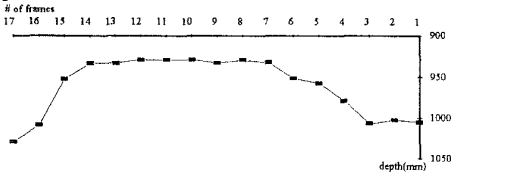


Fig. 4. Block diagram of Kalman filter to robot

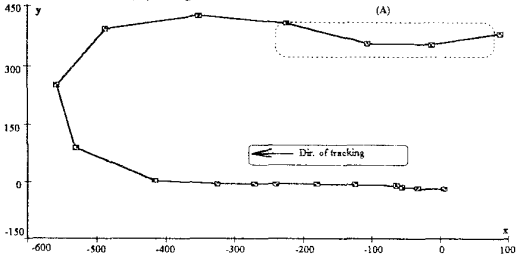
4.2.Synthetic experiments

In this section, we demonstrate the robustness and feasibility of the proposed tracking algorithm through the experiments using the target as shown

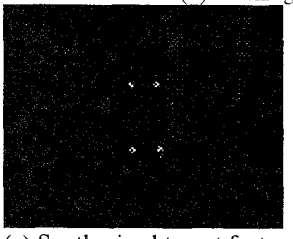
in Fig.5(c). We assume the lens parameter f is given by the calibration[5]. The deviation between the hypothesized ideal motion and real motion is captured by process noise covariance. Error covariance of initial state vector of Kalman filter are given by relatively large value. Covariance matrix of Gaussian random noise of the measurement model is specified through the experiment. We have repeated the image capturing and extracted the center of gravity$(\overline{X}_f, \overline{Y}_f)$ of a target at static environment and computed the variance of $(\overline{X}_f, \overline{Y}_f)$ using χ^2 distribution with n-1 degrees of freedom. Finally the initial state vector was determined by assigning a suitable value. These values are quite different from true ones. We have carried out the experiment in which the robot is presumed to track the toy vehicle moving with both linear and rotational motion. The initial state vector $\hat{\mathbf{x}}_0$= (-15,-50,1022,0,106,-10,0,0,0,0.0087,0.996,5°,90°) was given. Especially z and $\dot{z}$ of $\hat{\mathbf{x}}_0$ are quite different from real situation. In Fig. 5 we showed the tracking results for full tracking range. Fig. 5(a) shows the estimated depth and Fig. 5(b) indicates the resulting manipulator's tracking positions.



(a) Depth estimate



(b) Tracking position



(c) Synthesized target feature

Fig. 5. Tacking results for full tracking range

Even though the target changes the motion from linear to circular and vice versa the tracker adaptively tracks the target using previous motion parameters because of Kalman filter's predictive ability based on past data. In Fig. 5(b), the marked area (A) is a good example of this problem. The shape of rail of this area is originally close to straight line. But the tracking result is not so close to straight line. To overcome this problem, we need a faster sampling rate. Specially, in Fig. 6 we depicted the depth estimate using linear regression model on the linear tracking zone.

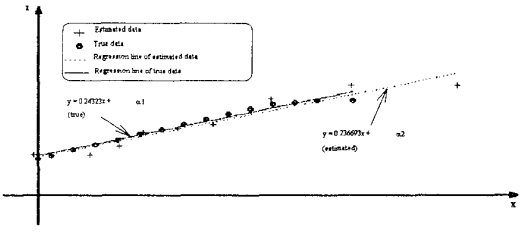
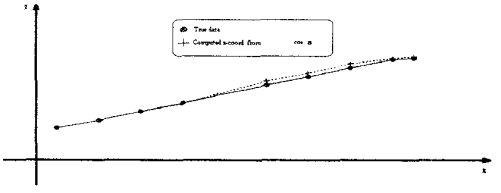
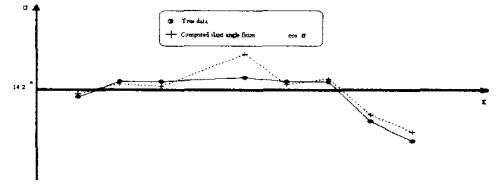


Fig.6.Depth estimation under linear and rotational motion

The viewed surface normal vector was estimated using the proposed scheme in section 3.3. Fig. 7a). shows the recovered z-coordinate based on the third component cosσ of the estimated normal vector using Eq. (14). And Fig. 7(b) shows the trend of slant angle.



(a) z-coordinate



(b) The slant angle

Fig. 7. Estimated surface normal vector

4.3. Real experiments

We illustrate the experimental results with real target under the same experimental setup as shown in Fig.3. The correspondence problem is solved using *Mahalanobis* distance and Kalman filter's prediction ability. The image processing results are presented in Fig. 8. In Fig.8 model

segments are obtained from prediction step of Kalman filtering and updated at filtering step, data segments are extracted line segments, matching candidate set(mcs) includes the all data segments with *Mahalanobis* distance less than threshold corresponding to each model segment, best fitted data segments are selected from mcs satisfying the adjacency condition of each model segment, and then triangulation is carried out using intersection points of best fitted lines. The principal axis, center of gravity, and higher order shape moments are produced according to Eqs.(20)-(25) and (32). Fig. 9 shows target tracking process through the extracted line segment matching. In this tracking the divergence change is well suited to real situation. Specially, in sixth frame the divergence change is bigger due to the correspondence problem.

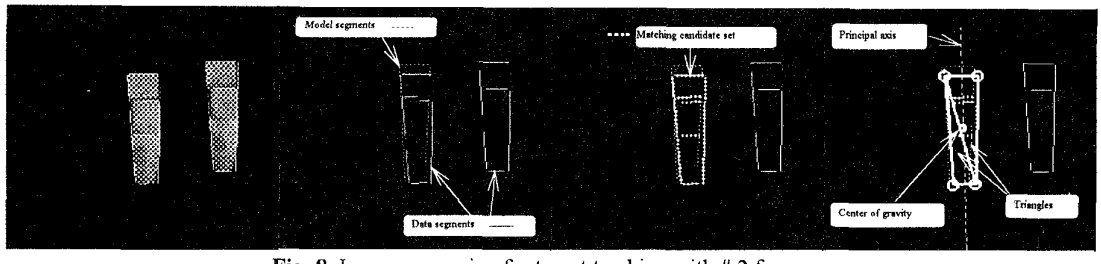
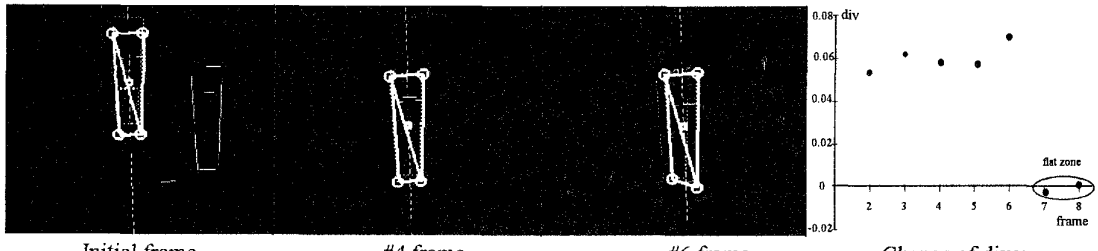


Fig. 8. Image processing for target tracking with # 2 frame



Initial frame #4 frame #6 frame Change of divergence

Fig. 9. Target tracking by extracted line correspondence

5. Conclusion

We developed a new tracking algorithm to track a target moving in 3D estimating a position, a linear and angular velocity, and an orientation of viewed surface in real-time with monocular visual sensory feedback. By using Kalman filter the tracking result was quite stable even in the presence of system and sensor noise.

In the extraction of the center of gravity, the principal axis, and the first order differential invariants of a target we used the triangulation of detected feature points of an image. Triangulation was very efficient method to reduce the processing time. To obtain the surface orientation and the depth in dynamic environment we took advantages of the image divergence and deformation principle.

We demonstrated that the first order differential invariants play the key role of estimating the depth from a viewer to a target and the surface normal vector of a target from monocular image under the situation in which target and viewer moves at the same time. The experiments showed that the proposed algorithm can be applicable satisfactorily to motion tracking in 3D on a 486-DX-33 PC. We expect this work will contribute in many areas of computer vision such as motion analysis, scene analysis, stereo vision and so on.

REFERENCES

[1] D.H. Ballard and C.M. Brown, Computer Vision, Prentice-Hall, Englewood Cliffs NJ(1982).John Wiley & Songs, Inc. 1976.

[2] R. Cipolla and A. Blake, Surface Orientation and Time to Contact from Image Divergence and Deformation, Computer-Vision-ECCV '92, pp.187-202, 1992.

[3] A. Gelb, Applied optimal estimation, 1984.

[4] R.G. Gonzalez and R.E. Woods, Digital Signal Processing, Addison Wesley, U.S.A., 1992.

[5] R. Y. Tsai, A versatile Camera calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses, *IEEE J. Robotics Automat.*, Vol. RA-3, No.4, pp.323-344, 1987.