Geometric Invariants for Verification in 3-D Object Tracking

U. Uenohara
R&D Center
Toshiba Corporation
Kawasaki, Japan 210

T. Kanade
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

We demonstrate that geometric invariants are useful tools and can be used for verification of feature tracking. In visual tracking of objects, feature tracking is the first part of the process, and reliable feature tracking is essential for realizing robust object tracking. Feature tracking of many features on the target object, verification of feature tracking and selection of successfully tracked features, enables robust object tracking even if there is some occlusion and change of illumination. We show how geometric invariants from five coplanar points and affine moment invariants can be used for verification of feature tracking. The important point we have to consider is the sensitivity of invariants. The sensitivity of invariants is dependent on the configuration of feature points. Dynamic threshold setting is proposed for dealing with observation errors in tracking in any situation.

1 Introduction

Real-time visual tracking of objects will enable realization of many useful applications, e.g. automatic supervision system, human computer interface, and automatic handling of moving objects by robots. Despite a great deal of research in this area over many years[1-4], it is still difficult to realize robust real-time visual tracking of 3-D objects. Here we present ways of increasing the robustness of tracking with geometric invariants.

The essential element for realizing robust visual tracking is the feature tracking. When robust feature tracking is realized, the remaining element is the object pose computation from tracking results. There are three main cues: spatiotemporal continuity of feature locations in the image, similarity of pattern between frames, and geometrical constraints between features. In most previous work, the first two cues are used by limiting the search area of each feature to the region around the location of each feature in the previous frame: and, by searching for features most similar to the reference, which is the feature pattern in the previous frame, or the pattern adjusted to the feature in the previous frame. However, the last cue, the geometrical constraint between features, has not been commonly used.

Feature locations change during tracking due to view changes. There is, however, some geometrical constraint between feature locations. One way to check if the feature locations from tracking in this current frame meet the constraint is to compute object position and orientation from feature locations and to project each feature onto the image plane with the computed object pose. If feature locations meet the constraint, there should be an object position and orientation explaining all the feature locations, and therefore, all the projected features will match the feature locations obtained from tracking results. However, projected features would not match well if the feature locations did not meet the constraint. The difficulty here is its high computation cost. Object pose calculation is in most cases an expensive operation.

Geometric invariants [7,8], popular in object recognition as useful descriptors of objects, are properties in the image that stay invariant under some transformation. A simple example is the length of a rod. We can identify a particular rod by measuring its length on the image and comparing it to a database of rod lengths. The length of a rod is the invariant under Euclidean transformation. In computer vision, projective invariants and affine invariants are widely used. Projective transformation models the situation where a plane curve is subject to rigid motion in space, and projected using perspective. Affine transformation is a reasonable linear approximation of it when the camera is far away from the scene. Two major projective invariants are the cross ratios of four collinear points, and the cross ratios of four areas of triangles from five coplanar points. They remain constant in any view change.

Affine moment invariants are algebraic functions of the moments that are invariant to affine transformations. Taubin and Cooper introduced a new framework for generating affine moment invariants [10]. The particular moment invariants developed are the eigenvalues of certain matrices, whose coefficients are algebraic functions of the locations of 2D data. The computational cost of computing these invariants is low. They are used for matching corresponding regions in two images.

In this paper, we demonstrate that geometric invariants are useful tools and can be used for checking if features are successfully tracked with a low computational cost. We will show that both geometric invariants and moment invariants can be used, and show how they should be handled. The important point we have to consider is the sensitivity of invariants. The sensitivity of invariants is dependent on the configuration of feature points. Dynamic threshold

Proc. IROS 96
0-7803-3213-X/96/ $ 5.00 © 1996 IEEE

setting is presented for dealing with observation errors in any situation.

2 Invariants

Although any functions that are invariant to view change can be used for tracking verification, we will introduce two invariants that can be widely used.

2.1 Five Coplanar Points

Five coplanar points give rise to two invariants which have found important applications in machine vision. The invariants are useful for recognizing surfaces that are approximately planar. We can prove the invariance of the cross-ratios of either areas Sijk of triangles or the corresponding determinants in homogeneous coordinates [7].

$$I_1 = \frac{S_{423}S_{125}}{S_{124}S_{523}} \quad I_2 = \frac{S_{143}S_{125}}{S_{124}S_{153}}$$

2.2 Affine Moment Invariants

Affine invariants are functions of geometric structure which remain unchanged under affine transformation. Taubin and Cooper introduced affine moment invariants which are the eigenvalues of certain matrices, whose coefficients are algebraic functions of the locations of 2D image data [10]. The elements of the matrices involved are functions of moments such as

$$\frac{1}{n}\Sigma_{i=1}^{n}(\alpha_{i1}-\bar{\alpha}_{i1})^{j}(\alpha_{i2}-\bar{\alpha}_{i2})^{k}$$

where n is the number of image data points in the region. The simplest invariants of 2D shapes are the eigenvalues of the symmetric 2 x 2 matrix $M'_{[1,2]}M'^{t}_{[1,2]}$ where $M'_{[1,2]}$ is

$$M'_{[1,2]} = \begin{bmatrix} \frac{1}{\sqrt{2}}M'_{(3,0)} & M'_{(2,1)} & \frac{1}{\sqrt{2}}M'_{(1,2)} \\ \frac{1}{\sqrt{2}}M'_{(2,1)} & M'_{(1,2)} & \frac{1}{\sqrt{2}}M'_{(0,3)} \end{bmatrix}$$

$$M'_{(i,j)} = \frac{1}{n}\sum_{k=1}^{n}[L_{11}(x_k-\bar{x})]^{i}[L_{21}(x_k-\bar{x})+L_{22}(y_k-\bar{y})]^{j}$$

$$L_{11} = 1\Big/\sqrt{M_{(2,0)}}$$

$$L_{21} = -M_{(1,1)}\Big/\sqrt{M_{(0,2)}-M_{(1,1)}^{2}/M_{(2,0)}}$$

$$L_{22} = 1\Big/\sqrt{M_{(0,2)}-M_{(1,1)}^{2}/M_{(2,0)}}$$

$$M_{(2,0)} = \frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2$$

$$M_{(1,1)} = \frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})$$

$$M_{(0,2)} = \frac{1}{n}\sum_{i=1}^{n}(y_i-\bar{y})^2$$

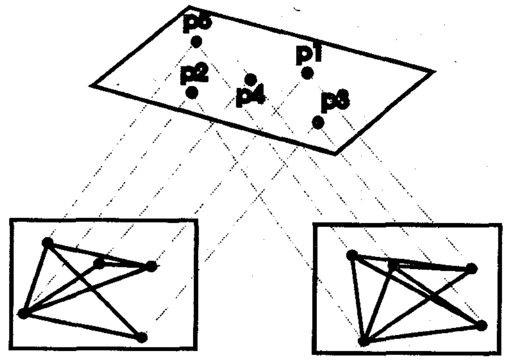


Fig. 1 Five Coplanar Points.

3 Tracking Verification by Invariants

Geometric invariants have been used for object recognition as useful descriptors. The process of object recognition with invariants proceeds as follows: the generation of list of values of invariants for each object, the computation of invariants for input image, and the matching of the invariant pairs.

In visual tracking, we have good candidates for feature correspondences; the nearest feature. Geometric invariants can be used to check if the candidate is really the corresponding feature by computing values in the initial image, storing them, computing values in the input image, and comparing them. Good correspondence will lead to the same value as the initial one.

The process whereby many features are tracked and successfully tracked features are selected, realizes robust 3-D object tracking even if there is some occlusion and change of illumination. Features can be edges, corners, etc.

In object tracking of a known 3D object, we have object centered coordinates of features. The object recognition is the process of finding the correspondence of some of the features on the object, and to compute the object pose. When the number of features to be tracked is *n*, and the number of features used for object pose calculation is *p*, the problem is to find successfully tracked p features with minimum observation errors. The number of combinations of p features out of n features is

$$C(n,p) = \binom{n}{p}$$

The process of computing *C(n,p)* invariants, and the selection of the best combination with the minimum variation from the initial value, gives us the best combination with low computation cost, in the sense that the error from the geometrical constraint is the minimum.

Any invariants can be used for the process. One of the most useful invariants is the affine moment invariant in case that the process can be approximated by affine transformation. We can

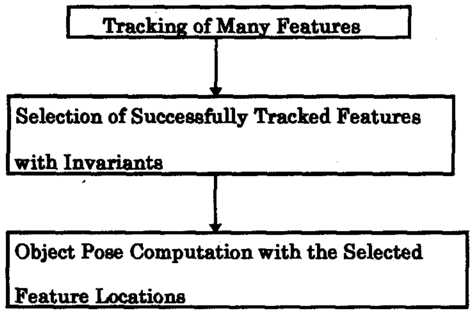


Fig. 2 Robust Object Tracking by Invariants.

compute the invariants with any number of features. Affine moments are usually used for region matching, the limitation is the necessity of segmentation before matching. For tracking verification, the segmentation process is carried out implicitly at the feature tracking phase by limiting the area of search and outputting only the best matched point.

4 Sensitivity of Invariants

One important point we have to consider is the sensitivity of geometric invariants [17]. Some observation errors are included in tracking feature positions (typically 0.5 pixel), which cause invariants computed from them to vary. The sensitivity of invariants is also dependent on the configuration of feature points. It makes it difficult to set constant thresholds to judge whether or not invariants are violated and thus feature points are successfully tracked. Instead of using constant thresholds, the thresholds need to be adjusted by the standard deviation of each invariant. Assume that observation errors of each feature position have Gaussian distributions. Invariants then have the distribution with a covariance matrix up to second order [13]

$$\sigma_I^2 = J\Lambda_x J^T$$

$$= \begin{bmatrix} \frac{\partial I}{\partial x_1} & \frac{\partial I}{\partial y_1} & \cdots & \frac{\partial I}{\partial x_p} & \frac{\partial I}{\partial y_p} \end{bmatrix} \begin{bmatrix} \sigma_x^2 & 0 & \cdots & 0 \\ 0 & \sigma_x^2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \sigma_x^2 \end{bmatrix} \begin{bmatrix} \partial I/\partial x_1 \\ \partial I/\partial y_1 \\ \cdots \\ \partial I/\partial y_p \end{bmatrix}$$

where σ_I^2 is the variance of the invariant, σ_x^2 is the variance of the feature locations, p is the number of features, and Λ_x is the covariance matrix of feature positions in the image. J is the Jacobian matrix. We assume that there is no correlation between the observation error of each point so that Λ_x becomes a diagonal matrix and the covariance matrix of the invariant becomes scalar, which is variance of the invariant.

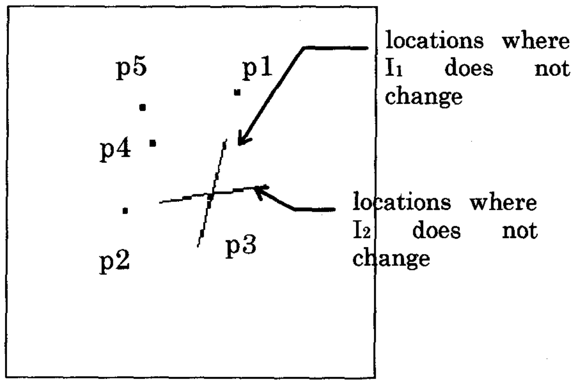


Fig. 3 Direction where Invariants do not change.

Sensitivity of invariants varies not only with the configuration of points but also with the direction of position change of each point. Invariants do not change their values in some directions. Fig.3 shows locations of the point p3 where the values of five-coplanar invariants I_1 and I_2 do not change. We need to check more than one invariant value to avoid missing some situations of false tracking results.

Verification of feature tracking with invariants is carried out as follows. m is the number of invariants, n is the number of tracked features, p is the number of selected features. m invariants $I_1, I_2, \cdots, I_m$ are computed for all combinations of p points out of n feature points. They are compared with their thresholds and discarded unless deviations of I_1 to I_2 from initial values are all below their thresholds. The threshold for each invariant is set to the standard deviation of invariants multiplied by some constant c. When there is more than one combination of p features satisfying the condition, we select the p features that minimize the maximum of I_1 to I_m divided by their respective standard deviation.

5 Experimental Results

The effectiveness of verification of feature tracking with invariants is demonstrated with experimental results. We will present an example of the deviation of

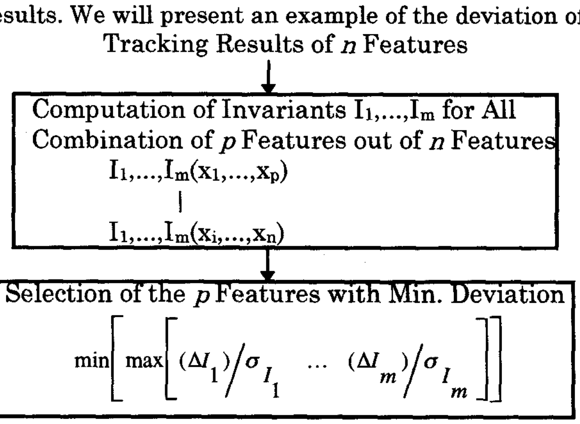


Fig. 4 Selection of Successfully Tracked Features.

the invariant computed from real tracking results, and show that our algorithm is effective for selecting only successfully tracked features. We will then present an experimental result of tracking a desktop PC as a 3-D object.

5.1 Invariant with Five-Coplanar Points

Five feature points on the same flat plane are tracked: ΔI_1, the difference between the initial value and the current value of invariant I_1, and σ_I, the standard deviation of I_1, are computed. The variance of each feature point in the image is assumed to be 1.0. The result is shown in Fig. 5. The figure shows that the tracking result of the no.1 feature point is sometimes wrong, around 2.6 sec, 3.2 sec, etc. The corresponding part of ΔI_1 becomes larger than σ_I, which correctly indicates the tracking result at that time is not good.

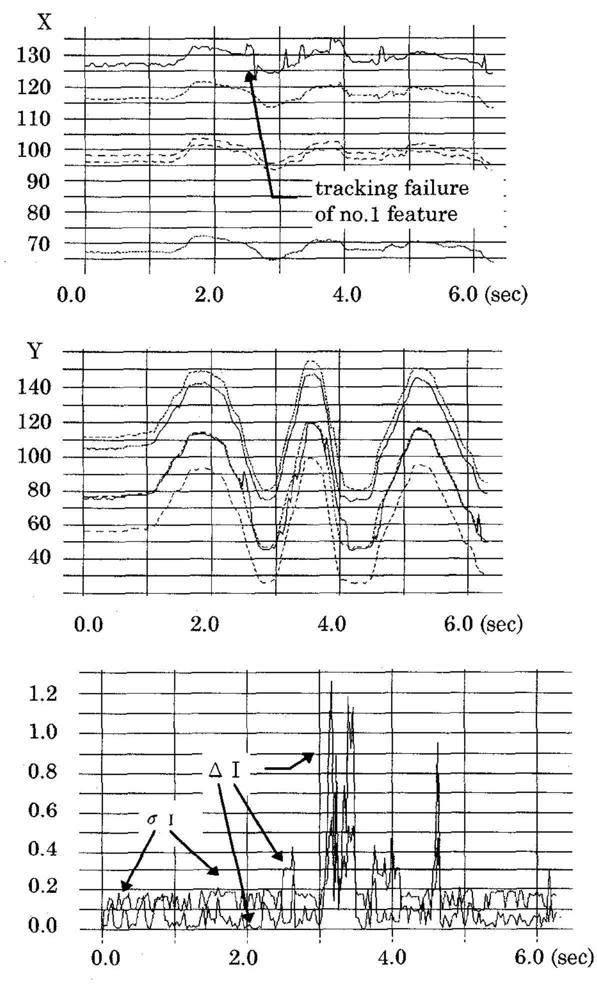


Fig. 5 Five Coplanar Points, and $\sigma_I, \Delta I$.

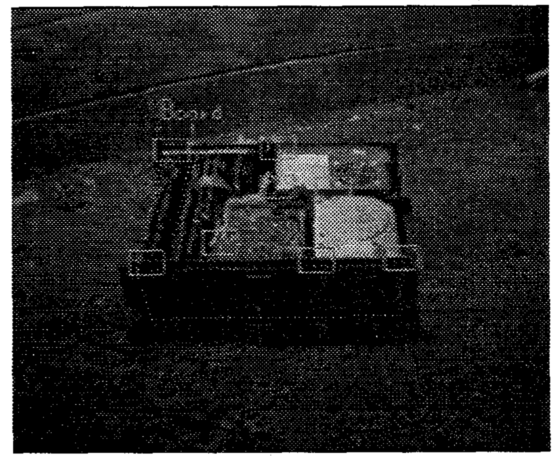


Fig. 6 Eight Features on the PC.

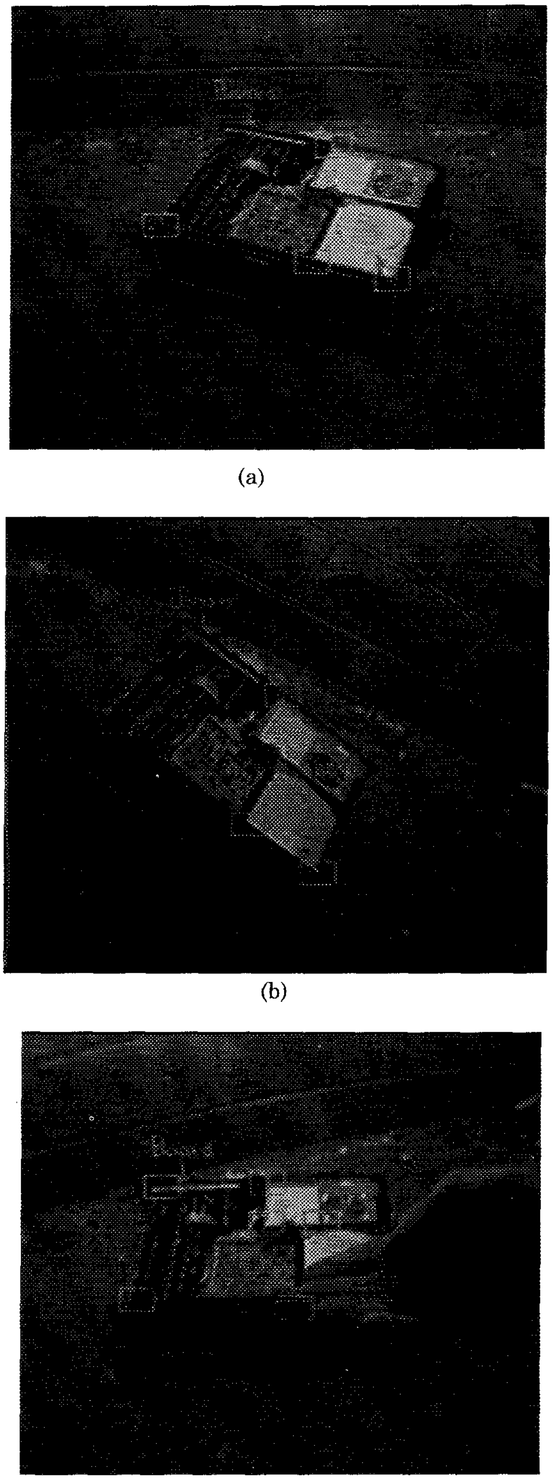
5.2 Object Tracking with Invariant Verification

3-D object tracking of a desktop PC is carried out[15][16]. Eight feature points are tracked by computing the normalized correlation to the reference image pattern and the points with the highest scores are selected as the current feature locations. Tracking results are checked by calculating invariants with five coplanar points that are described above. All eight feature points are on the same plane as Fig. 6. The number of combinations of eight points taken five points at a time is fifty-six. Therefore, fifty-six invariants are calculated and differences from the initial values divided by standard deviations are computed. The best five points which have the minimum change are selected, and the pose of the PC is computed with these five points by Newton's method. Fig. 7 shows that the system tracks the PC successfully in 3-D motion even when there is some occlusion by superimposing the frame of the PC and the IO board. It is capable of continuing to track and superimpose even when up to three feature points are occluded. The verification process with affine moment invariants works well also. A video will be shown in the presentation.

The tracking of the PC is executed at frame rate (30 Hz). The system is implemented on three TMS320C40 (C40), and the low latency vision hardware developed at CMU[14] is used. The latency of the system is only two frames, which is one of the main reason for the low computation cost of the verification process.

6 Conclusion

We have demonstrated that geometric invariants are useful tools for verification of feature- tracking. Calculation cost for computing invariants is relatively small compared to the correlation in feature tracking. The big difference when we use geometric invariants in object tracking is the existence of candidates of correspondence. The role of geometric invariants in



(a)

(b)

(b)

Fig. 7 Tracking of the PC.

object tracking is just the verification, while the role in object recognition is the generation of hypotheses. Geometric invariants depend on the order of the features in the computation. The number of invariants we should compute in object recognition is the number of permutations $P(n,p)$ where n is the number of features in the image and p is the number of features in the model. In object tracking, we have candidates of correspondence and the role of invariants is just verification. Therefore, the number of invariants to be computed is the number of combinations $C(n,p)$. This is a big difference and enables real-time tracking of objects.

The important point we have to consider is the sensitivity of invariants. Dynamic threshold setting has been presented for checking if those feature locations meet the geometrical constraint. The threshold is set to be the standard deviation of the invariant multiplied by some constant.

We have shown two invariants, five coplanar points and affine moment invariants, although any invariants can be used for verification. The use of affine moment invariants for recognition has a difficulty with regard to segmentation. However, in our formulation, segmentation is carried out in feature tracking so that there is no such difficulty.

Acknowledgments

The work was done while one of the authors (Uenohara) was at the Vision and Autonomous Systems Center of Carnegie Mellon University. The authors would like to thank the many researchers at VASC from whose stimulating discussion they benefited, and are especially grateful to Dr. Luc Robert for his advice.

References

[1] D. Gennery, "Tracking known three-dimensional objects", Proc. 2nd Nation. Conf. Artif. Intell., Pittsburgh, pp.13-17, 1982.

[2] D.G. Lowe, "Robust model-based motion tracking through the integration of search and estimation", Int. J. Computer Vision, Vol. 8, No.2, pp. 113-122, 1992.

[3] D.G. Lowe, "Fitting parameterized three-dimensional models to images", IEEE Trans. Patt. Anal. Mach. Intell. Vol. 13, No.5, pp. 441-450, 1991.

[4] D.B. Gennery, "Visual tracking of known three-dimensional objects", Int. J. Computer Vision, Vol. 7, No. 3, pp. 243-270, 1992.

[5] A. Rosenfeld and A. Ka, "Digital picture processing", New York Academic, 1982.

[6] H. Inoue, T. Tachikawa and Masayuki Inaba, "Robot vision system with a correlation chip for real-time tracking, optical flow and depth map generation", Proc. IEEE Int. Conf. Robotics and Automation, Nice, France, 1992.

[7] I. Weiss, "Geometric invariants and object recognition", Int. J. Computer Vision, Vol.10, No.3, pp. 207-231, 1993.

[8] J. Munday and A. Zisserman, "Introduction-towards a new framework for vision. In Geometric Invariance in Machine Vision", MIT Press, Cambridge, MA, 1992.

[9] D. Forsyth, J. Mundy, A. Zisserman, and C. Brown, "Projectively invariant representations using implicit algebraic curves", IEEE Trans. Patt. Anal. Mach. Intell. Vol.13, pp.971-991, 1990.

[10] G. Taubin, D. Cooper, "Object recognition based on moment(or algebraic) invariants", In Geometric Invariance in Computer Vision, Edited by J. Mundy and Z. Zisserman, MIT Press, pp.375-397, 1992.

[11] C. Lee, D. Cooper, D. Keren, "Computing correspondence based on regions and invariants without feature extraction and segmentation", CVPR'93, pp.655-656, 1993.

[12] J. Burns, R. Weiss, and E. Riseman, "The non-existence of general-case view-invariants", in Geometric Invariance in Computer Vision, Edited by J. Mundy and Z. Zisserman, MIT Press, pp.120-131, 1992.

[13] H. F. Durrant-Whyte, Uncertain geometry in robotics, IEEE J. Robotics and Automation, Vol.4, No.1, pp.23-31, 1988.

[14] O. Amidi, Y. Mesaki, T. Kanade, M. Uenohara, "Research on autonomous vision guided helicopter", RI/SME Fifth World Conf. on Robotics Research, Cambridge, MA, 1994.

[15] M. Uenohara and T. Kanade, "Vision-based object registration for real-time image overlay", Int. Conf. Comp. Vision, Virtual Reality and Robotics in Medicine , Nice, France, 1995.

[16] M. Uenohara and T. Kanade, "Vision-based object registration for real-time image overlay", Int. Journal of Computers in Biology and Medicine, Vol.25, No.2, pp.249-260, 1995.

[17] D. Sinclair, A. Blake, "Quantitative planar region detection", Int. Journal of Computer Vision, Vol.18, No.1, pp.77-91, 1996.