Proceedings of the 11th KACC, October 1996

A Self-Localization Algorithm for a Mobile Robot Using Perspective Invariant

° Kyoung Sig Roh *, Wang Heon Lee *, Sang Sam Jung*, In So Kweon**

*Dept. of Automation and Design Eng., KAIST (Tel:958-3455; Fax:968-1638; E-mail: nks@design.kaist.ac.kr)
**Dept. of Electrical and Electronic Eng.,KAIST (Tel:958-3415; Fax:968-1638; E-mail: kweon@design.kaist.ac.kr)

Abstract This paper presents a new algorithm for the self-localization of a mobile robot using perspective invariant(Cross Ratio). Most of conventional model-based self-localization methods have some problems that data structure building, map updating and matching processes are very complex. Use of the simple cross ratio can be effective to the above problems. The algorithm is based on two basic assumptions that the ground plane is flat and two parallel walls are available. Also it is assumed that an environmental map is available for matching between the scene and the model. To extract an accurate steering angle for a mobile robot, we take advantage of geometric features such as vanishing points(V.P). Point features for computing cross ratios are extracted robustly using a vanishing point and the intersection points between floor and the vertical lines of door frames. The robustness and feasibility of our algorithms have been demonstrated through experiments in indoor environments using an indoor mobile robot, KASIRI-II(KAist SImple Roving Intelligence).

Keywords Mobile Robot, Vision, Self-Localization, Projective Invariant

1. INTRODUCTION

Self-localization of a mobile robot in indoor environment is one of the basic capabilities for an autonomous system. There are many methods for self-localization. Among these methods, a vision-based approach is simple and very flexible. Kanbara et. al.[5] classify image features to category using a defined geometric relationship, and determine whether there is a vertical line or horizontal line and the direction of a boundary only within a specific window. This approach is sensitive to image noise and cannot provide the global information. Kosaka and Kak[1] implement a system for a given environment using a CAD model based expectation map. This method constructs a complex database and requires additional analysis for handling uncertainty. In most of visual-based approaches, the database for an environment becomes complex because the scene is perspectively projected and the geometric properties are not invariant.

We present a new efficient self-localization algorithm using 1-D perspective invariant[7] and vanishing point. Since this method uses a simple cross ratio, the data structure building, map updating and matching processes become trivial tasks.

The algorithm for navigation in corridors and similar indoor environments is based on two basic assumptions that the ground plane is flat and two locally parallel side-lines are formed by floor and two side walls in corridor environments. Also we assume that the environmental map database is available for matching between the scene and the model. Intersection points between floor and the vertical lines of door frames are used as point features to compute cross ratios.

As an off-line process, we construct a database consisting of the cross ratios of point features, and the moving direction of a mobile robot. Using cross ratios in the constructed database, the correspondences between the model and scene features can be found. The corresponding feature points in the database of a real environment and in the image are used to compute the position of the mobile robot.

We demonstrate the robustness and feasibility of our algorithms through real world experiments in indoor environments using an indoor mobile robot.

The paper is organized as follows: In Section 2, we explain the basic theory of cross ratio and its error modeling and present a self-localization algorithm. In section 3, we explain the image preprocessing for the extraction of feature points and vanishing points. In section 4, we present experimental results for the real scene.

2. BASIC THEORY

2.1 Cross Ratio and Error Modeling

Let (X_c, Y_c, Z_c) denote the camera coordinate system whose origin is at the optical center and its Z-axis aligned with the optical axis of lens(See Fig.1). Let (X_g, Y_g, Z_g) be an object coordinate system. (Q1,Q2,Q3,Q4) are points on a line in an object plane and (P1,P2,P3,P4) are the corresponding points on the image plane.

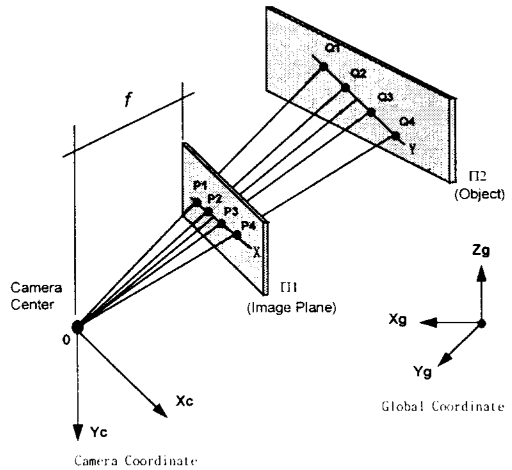


Fig.1.Camera Coordinate and Global Coordinate

X_i is the coordinate of Qi with respect to the object coordinate system and x_i is the corresponding coordinate of Pi in the camera coordinate system. Then we define the cross-ratio,1-D invariant, as follows:

$$C(x_1,x_2;x_3,x_4)=\frac{(x_3-x_1)(x_2-x_4)}{(x_2-x_1)(x_3-x_4)}=\frac{(X_3-X_1)(X_2-X_4)}{(X_2-X_1)(X_3-X_4)}\ . \quad (1)$$

In Eq.(1), the cross ratio is a function of four variables :

$$C=C(x_1,x_2,x_3,x_4)\ . \quad (2)$$

Let x_i be the true and $\tilde{x}_i$ be the noisy observation of x_i. Then we can write the relationship between these as

$$\tilde{x}_i = x_i + \xi_i,\quad i=1\sim 4 \quad (3)$$

where ξ_i denotes an independently distributed noise term with the having mean "0", and the variance σ_i^2.

$$E[\xi_i]=0,\ V[\xi_i]=\sigma_i^2, E[\xi_i\xi_j]=\begin{cases}\sigma_0^2 & \text{if } i=j\\ 0 & \text{otherwise}\end{cases} \quad (4)$$

Using these noisy observations; we define a noisy invariant,

$$\tilde{C}=\tilde{C}(\tilde{x}_1,\tilde{x}_2,\tilde{x}_3,\tilde{x}_4). \quad (5)$$

To determine the expected value and the variance of $\tilde{C}$, we represent $\tilde{C}$ as a Taylor series around x_1,x_2,x_3,x_4,

$$\tilde{C}\approx C+\sum_{i=1}^{5}\left[(\tilde{x}_i-x_i)\frac{\partial\tilde{C}}{\partial\tilde{x}_i}\right]. \quad (6)$$

From Eq.(3), we can write Eq.(6) as

$$\tilde{C}\approx C+\sum_{i=1}^{5}\left[\xi_i\frac{\partial\tilde{C}}{\partial\tilde{x}_i}\right]. \quad (7)$$

To obtain the expected value $E[\tilde{C}]$, we take expectation on both sides of Eq.(7) :

$$E[\tilde{C}]=E[C]+E\left[\sum_{i=1}^{5}[\xi_i\frac{\partial\tilde{C}}{\partial\tilde{x}_i}]\right]=E[C]. \quad (8)$$

In a similar way, we can determine the variation,

$$V(\tilde{C})=E[(\tilde{C}-C)^2]=E\left[\left(\sum_{i=1}^{5}[\xi_i\frac{\partial\tilde{C}}{\partial\tilde{x}_i}]\right)^2\right]=\sigma_0^2\sum_{i=1}^{5}\left[\left(\frac{\partial\tilde{C}}{\partial\tilde{x}_i}\right)^2\right]. \quad (9)$$

where

$$\frac{\partial\tilde{C}}{\partial\tilde{x}_1}=C\left\{\frac{1}{(x_2-x_1)}-\frac{1}{(x_3-x_1)}\right\},\ \frac{\partial\tilde{C}}{\partial\tilde{x}_2}=C\left\{\frac{1}{(x_2-x_4)}-\frac{1}{(x_2-x_1)}\right\} \quad (10)$$

$$\frac{\partial\tilde{C}}{\partial\tilde{x}_3}=C\left\{\frac{1}{(x_3-x_1)}-\frac{1}{(x_3-x_4)}\right\},\ \frac{\partial\tilde{C}}{\partial\tilde{x}_4}=C\left\{\frac{1}{(x_3-x_4)}-\frac{1}{(x_2-x_4)}\right\}$$

We define a threshold for the cross ratio as follows :

$$\text{Threshold}=3\sqrt{V[\tilde{C}]}. \quad (11)$$

We use this threshold to search for the corresponding cross ratios in the constructed database.

2.2 Database Construction and Matching Algorithm

In this section, we explain how to construct a database using the cross ratio. Fig.2 shows a top-down view of a typical corridor scene. In this figure, point features, which are the intersection points between floor and door frame, are represented by L_K and R_K.

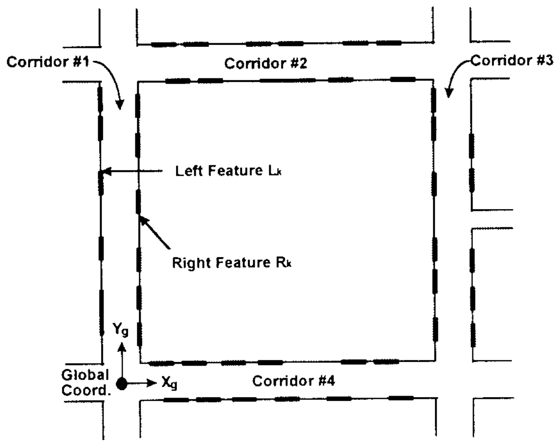


Fig. 2. Scene model to setup a database

Construction of Database

We compute the cross-ratio and its variance for a set of point features , ($L_K, L_{K+1}, L_{K+2}, L_{K+3}$) or ($R_K, R_{K+1}, R_{K+2}, R_{K+3}$) in a corridor i. We store {i, L_K(or R_K)} into the entries of a hash table, whose range is determined by $C\pm V(C)$.

Fig.3 shows an example of the hash table, where H(C) is a hash function for the cross-ratio C.

Using this database, it is relatively easy to generate hypotheses and to update the entries.

Entry(H(C))	Corridor #	Start Feature #	Direction
:	:	:	:
80	1 3	5 2	R R
90	2 2	10 11	L R
:	:	:	:

Fig.3 Example of Database

Matching(or Hypotheses Generation)

Given an image, we extract a set of point features and compute the cross-ratio for the set. Then, the correspondences between the model and the scene features are hypothesized by indexing the entries of the hash table with a similar cross-ratio.

Verification

From the results of the matching process, we obtain multiple promising correspondence sets. Therefore, we need a verification process to determine a unique corresponding feature set which is used to compute the position of robot.

We propose a Bayes rule based method that is implemented with a sequence of matching(or hypotheses).

Fig.4 shows the correspondences generated from consecutive images. $H_{i,j}$ represents the j-th correspondence in the i-th image and $P_{i,j}$ represents the path of robot from the i-th correspondence in the previous image to the j-th correspondence in the current image.

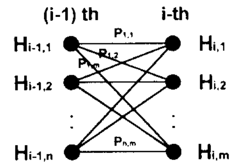


Fig.4 Hypotheses generated from two consecutive images.

Then, the probability for each path is

$$\text{Pro}(be|P_{i,j})=\frac{\text{Pro}(P_{i,j}|be)\text{Pro}(be)}{\text{Pro}(P_{i,j}|be)\text{Pro}(be)+\text{Pro}(P_{i,j}|not\ be)\text{Pro}(not\ be)}. \quad (12)$$

where $\text{Pro}(be|P_{i,j})$ means a probability that a robot has been "being"on the path based on the fact that the robot is observing $H_{i-1,j}$ and $H_{i,j}$.

$\text{Pro}(P_{i,j}|be)$ can be expressed as the following equation.

$$\text{Pro}(P_{i,j}|be)=k\times\text{Pro}(\Delta d-l). \quad (13)$$

where k is a normalizing factor and Δd is an estimated distance by the self-localization algorithm described in the next section and l is a measured distance by an odometer of the robot.

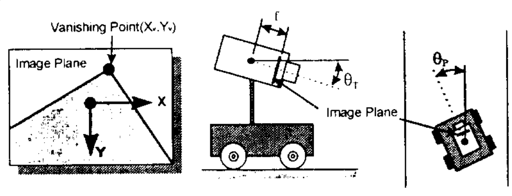
In Eq.(13), we assume that $\text{Pro}(\Delta d-l)$ has Gaussian distribution whose mean and variance are experimentally determined[16].

2.3 Self-Localization

In this section, we explain a two-step self-localization algorithm. In the first step, we compute the pan and tilt angles of camera by using the vanishing point[4]. In the second step, we compute the position of robot with the corresponding points obtained by the previous matching and pan, tilt angles obtained by the previous matching step .

The First step : (Pan and Tilt Angles)

Fig.5 shows the geometric relationship between the pan, tilt angles and the position of the vanishing point on an image.



(a) V.P on an image plane (b) Side View (c) Top View

Fig.5 The relationship between the V.P and angles

From Fig.5, if we extract the vanishing point, it is possible to compute the pan and tilt angle as follows :

$$\theta_P = \tan^{-1}\left(\frac{X_V}{f}\right),\ \theta_T = \tan^{-1}\left(\frac{Y_V}{f}\right), \tag{14}$$

where (X_V, Y_V) and f represent the coordinate of the vanishing point and the focal length, respectively.

The Second step : (Position)

Fig.6 shows the coordinate systems of camera and the scene. Let (Xg,Yg,Zg), (Xc,Yc,Zc) represent the global coordinate system and the camera coordinate system, respectively. Let (T_X,T_Y,T_Z) denote a translation from the global to the camera coordinate system.

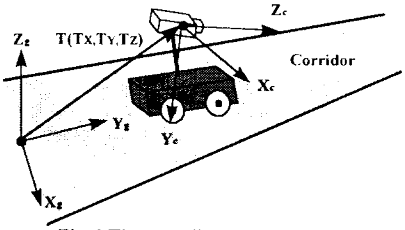


Fig.6 The coordinates systems

From Fig.6, we can write the relationship between the camera and the global coordinate systems as follows :

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} \cos\theta_P & -\sin\theta_P & 0 \\ \sin\theta_T \sin\theta_P & \sin\theta_T \cos\theta_P & -\cos\theta_T \\ \cos\theta_T \sin\theta_P & \cos\theta_T \cos\theta_P & \sin\theta_T \end{bmatrix} \begin{bmatrix} X_g - T_X \\ Y_g - T_Y \\ Z_g - T_Z \end{bmatrix}$$

$$\text{or } \mathbf{X}_c = \mathbf{A}(\mathbf{X}_g - \mathrm{T}). \tag{15}$$

By the perspective projection, we obtain

$$x = f\frac{X_c}{Z_c},\ y = f\frac{Y_c}{Z_c}. \tag{16}$$

In Eq.(1.15), we assume that the height from floor to the optical center, T_Z, is fixed to a constant.

Fom Eq.(1.15) and (1.16), we can obtain the equation,

$$\begin{bmatrix} (a_{11} - a_{31}x) & (a_{12} - a_{32}x) \\ (a_{21} - a_{31}y) & (a_{22} - a_{31}y) \end{bmatrix} \begin{bmatrix} T_X \\ T_Y \end{bmatrix} = \begin{bmatrix} (a_{11} - a_{31}x)X_g + (a_{12} - a_{32}x)Y_g + a_{33}xT_Z \\ (a_{21} - a_{31}y)X_g + (a_{22} - a_{32}y)Y_g + (a_{33}y - a_{23})T_Z \end{bmatrix}. \tag{17}$$

By solving Eq.(17), we can obtain T_x and T_y.

3. PRE-PROCESSING

We must extract point features to obtain the correspondences between the model and an image using the cross ratio. In this paper, we select point features that are the intersections between parallel lines on floor, which form a vanishing point, and vertical lines on the wall. Thus, vanishing point detection, vertical line detection, and the camera calibration are carried out.

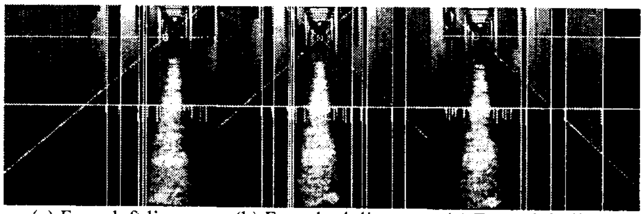
We first implement the camera calibration in order to obtain the camera parameters such as the image center, the scale factor, the focal length using Tsai's method[6].

For detecting the vanishing point, we use the Hough transformation method. In order to reduce the detecting time, we limited the range of the Hough space. We develop a method to detect the vanishing point by using the fact that the vanishing point is always projected on a constant horizontal line of the image plane, although one of two parallel lines is detected. We detect the vertical lines by using the ranged Hough transformation.

Fig.7 shows the extracted vanishing point with left(a), right(c), and both(b) of two parallel lines. Fig.8 shows the extracted vertical lines for each scene in Fig.7.

(a) From left line (b) From both lines (c) From right line

Fig.7 Vanishing point extraction



(a) From left line (b) From both lines (c) From right line

Fig.8 Vertical lines extraction

4. EXPERIMENTS

Experiments have been carried out in an indoor corridor environment using a mobile robot KASIRI-II. The mechanism of KASIRI-II consisting of wheels for conventional running and an infinite path wheel for running on unflat floor such as stairs. We use a 585 pentium as the master controller. It also includes a motion control board, a vision processing board(MVB-02), IR sensors, servo motors and drivers, and sonars.

4.1 Hypotheses Generation and Verification

Fig.9 shows two consecutive images from which we cannot uniquely determine the correspondence and the position of robot.

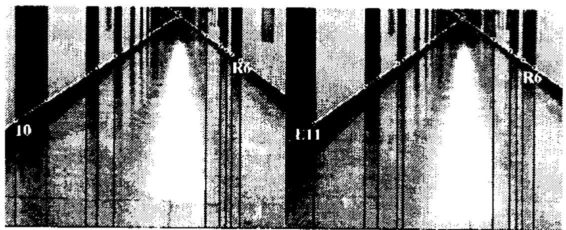


Fig.9 Two sequence images

Table1 shows the generated consecutive hypotheses and the corresponding positions of the robot. **Ci** and **Li** represent a corridor number and the i-th feature point on the left wall of the corridor.

Table 1 Results of hypotheses generation and self-localization.

Image	*Hypotheses*	*Position(Xg)*
1	*C2L2*	*1735*
	C2L10	*3051*
2	*C2L3*	*1832*
	C2L11	*3154*

For finding s most similar matching, we implement the verification algorithm described in section 2.2. Fig.10 shows the probability of each path from Eq.(12).

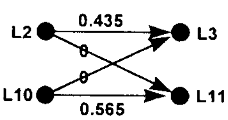


Fig.10 The result of verification

We select the path from L10 to L11 to be the best matching path.

As denoted in Fig.9, door frame on the left wall are correctly identified as L10 and L11. In a similar way, a door frames on the right wall is found to be R6.

4.2 Self-Localization

Fig.11 shows as experiment scenario for testing the accuracy of self-localization algorithm. The robot is commanded to navigate along the center of corridor #1 from Y_g=1230cm to Y_g=3740cm.

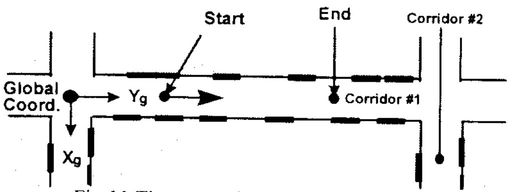


Fig.11 The run region of a mobile robot

Fig.12 shows the results of the self-localization of the mobile robot.

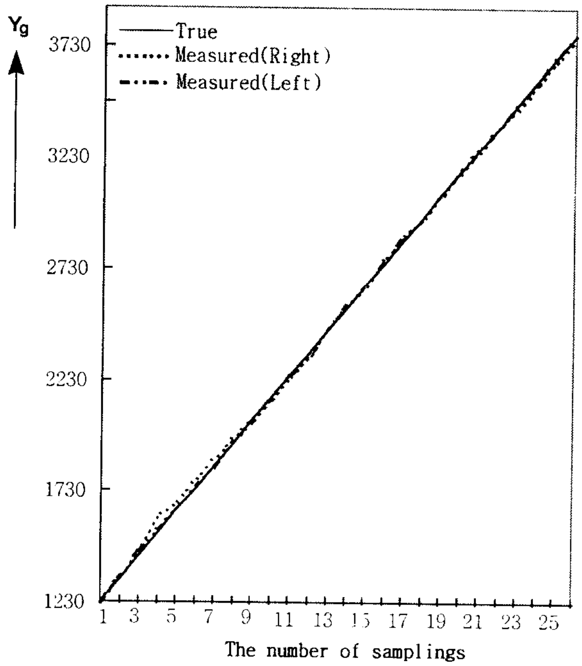


Fig.12 The result of self-localization

Fig.13 shows the error between the true position, measured by a tape measure, and the extracted Y_g position of the robot.

The proposed self-localization has successfully computed the robot position with a maximum accuracy of ±20cm.

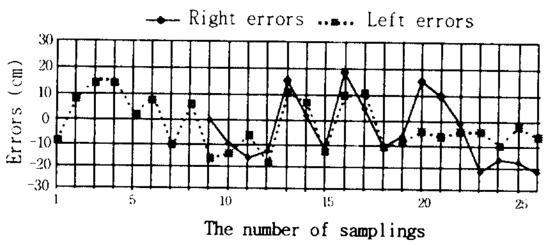


Fig.13 The error of self-localization

5. DISCUSSION

In this paper, we present a self localization algorithm for a mobile robot by using the cross ratio that is invariant for the perspective transformation.

The algorithm solves easily the self localization problem by using the cross ratio and is efficient to construct the database and to search for the corresponding points, and it is also easy to update and modify the database.

Proposed algorithms are verified through real experiments with KASIRI-II in indoor environments(shown in Fig.2).

In this experiments, the generated rates of the positive hypotheses is 100% whenever the point features are detected.

The maximum number of multiple promising correspondence is ten and occurred in corridor 2 where point features are distributed with same ratio. But, we are able to compute the robot position by the proposed the verification algorithm.

6. REFERENCE

[1] A.Kosaka and A.C.Kak, "Fast Vision-Guided Mobile Robot Navigation Using Model-Based Reasoning and Prediction of Uncertainties," *CVGIP:Image Understanding*, Vol. 56, No.3, November, pp. 271~329, 1992

[2] Z. Zhang et. al., "Automatic Calibration and Visual Servoing For A Robot Navigation System," *IEEE Int. Conf. on Robotics & Automation*, pp. 14 ~ 19,1993

[3] J.M.Badcock et al., "Autonomous Robot Navigation System - Integrating Environmental Mapping, Path Planning, Localization and Motion Control ," *Robotica*, Vol.11, pp. 97 ~ 103, 1993

[4] I.S.Kweon et al., "Vision-based Behaviors for Indoor Mobile Robots," *IEICE Technical Report*, Vol. 91, no. 437, 1992

[5] T.Kanbara et.al., "Selection of efficient Landmarks for an Autonomous Vehicle ," *Proc. of the 1993 IEEE/RSJ Int. Conf. on Intel. Robots and Systems*, Yokohama, Japan, July 26-30, 1993

[6] 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. of Robotics and Auto.*, Vol. RA-3, No. 4, August, 1987

[7] J.L.Mundy and A.Zisserman, "Geometric Invariance in Computer Vision," *MIT Press*, 1992

[8] R.O.Duda and P.E.Hart, "Pattern Classification and Scene Analysis," *John Wiley & Sons*, 1973

[9] B.Burns et.al., " Extracting Straight Lines," *IEEE Transactions on Patter. Anal. And Mach. Intel.*, Vol. PAMI-8, No.4, July, 1986

[10] J.G.Semple, "Algebraic Projective Geometry," *Oxford Press*, 1952

[11] M.Betke et al., "Mobile Robot Localization Using Landmarks," pp 135 ~ 142, *Proc. of the IEEE/RSJ/GI Int. Conf. on Intelligent Robots and Systems(IROS'94)*

[12] J.D.Courtney et al.,"Mobile Robot Localization via Classification of Multiple Maps," *IEEE Int. Conf. on Robotics & Automation*, pp 1672 ~ 1678, 1994

[13] A.Curren et al., "Sensor-Based Self-Localization for Wheeled Mobile Robots," *IEEE Int. Conf. on Robotics & Automation*, pp 662 ~ 667, 1994

[14] A.M.Sabatini et al., "Towards a Robust Methodology for Mobile Robot Localization Using Sonar," *IEEE Trans. on Robotics & Automation*, pp 3142 ~ 3147, 1994

[15] John J.Leonard et al., "Mobile Robot Localization by Tracking Geometric Beacons," *IEEE Trans. on Robotics & Automation*, Vol. 7, No. 3, June 1991

[16] S.S.Jung., "Robust Self-Localization of Indoor Mobile Robots Using a Single CCD Image," *KAIST MS. Theses*, 1996.