Camera Calibration for Virtual Studio using Projective Invariance

SeoWon,HAN † Hiroki TAKAHASHI †† JoonWhaon,LEE † Masayuki NAKAJMA ††

† Graduate School of Engineering of Electronics, Chonbuk National University
664-14. 1-Ga, Duckjin-Dong Chonju, KOREA
hsw@idec.chonbuk.ac.kr jhlee@moak.chonbuk.ac.kr

†† Graduate School of Information Science & Engineering, Tokyo Institute of Technology
2-12-1, Ookayama, Meguro-ku, Tokyo 152, JAPAN
rocky@cs.titech.ac.jp nakajima@cs.titech.ac.jp

Abstract

Virtual Studio uses chromakey method in which an image is captured, and the blue portions of that image are replaced by computer graphics or real video. For increasing reality, the replaced image need to be changed according to camera parameters. This paper proposes a novel method to extract camera parameters, which include position, direction, focal length and lens distortion of the camera in virtual studio, using the recognition of pentagonal patterns that are painted on a blue screen. A pentagon has invariant features under projection, so that we can find corresponding points between two planes, a blue screen and a captured image, using these features. Then, calculate camera parameters based on corresponding points. Experimental results indicate that corresponding points and camera parameters were more easily calculated compared to the conventional methods. The proposed method can process about twelve frames of video per a second in Pentium-MMX processor with CPU clock of 166MHz.

Keywords : Camera calibration, Virtual studio, Pentagonal patterns, Projective invariance.

1 Inroduction

Chromakey method is one of the broadcasting techniques widely used in television program and in general studio to make viewers feel immersing in a news, a weather forecast and an election program. It is a kind of image merging technique in which an image is captured from studio with blue background and characters. And, the blue portions of a captured image are replaced by computer graphics images or real video. For increasing reality, the replaced image need to be changed according to the camera motion in studio. Therefore, it is necessary to extract camera parameters which represent the motion of camera. The extraction of camera parameters should be accurate and robust to various postures of the camera for genuine reality, and should be fast for real time processing. Also, the extraction method should be applicable to an unfixed camera like shoulder camera.

Several methods have been proposed to measure the camera parameters in the virtual studio[2][1]. One employs optical or mechanical sensors directly mounted on the camera or lens. In this method, however, sensors are noise-sensitive and inaccurate, and initial regulations are needed for accurate measurements. In addition to these problems, the sensor occupies large volume so that it may be obstacle to cameraman and cannot be used for handy camera and shoulder camera[2]. The other popular method uses the recognition of mesh patterns composed of two shades of blue color, in which two vanishing points of the mesh patterns are changed according to the camera motion. In this method, however, it is impossible to measure the parameters when pan or tilt of camera is near to zero because there exists only one vanishing point[1].

This paper proposes a novel method to extract camera parameters for measuring camera motion using the recognition of pentagonal pat-

terns that are painted on a blue screen. That means a blue screen of virtual studio consists of the pentagons and background composed of two different shades of blue color. Considering a case that objects are overlapped with some pentagons, we paint many pentagons uniformly distributed on a blue screen. Since the pentagonal patterns have a slightly different tone from the background, they are easily separable from the background image using simple color image processing, and that makes our scheme suitable to real time processing. It is well known that pentagonal patterns have invariant features to the projection of plane[3]. Therefore, the robust matching is possible for finding the corresponding points between a blue screen and a captured image using the invariant features. The obtained set of corresponding points are the vertices of pentagons, and it can be used to extract the camera parameters. The parameters considered in the paper include position, direction, focal length and lens distortion of the camera in virtual studio. In the proposed scheme, we modify Tasi's method to calculate camera parameters. In the modified method, there are no constraint in the location of corresponding points such as the points should not be located near the Y-axis. Also, the modified method does not need so many corresponding points comparing to the original Tasi's method[4]. Experiments were derived from images captured in a small studio. Experimental results indicate that the invariant of pentagons are stable to any posture of camera and it takes 82ms to extract the camera parameters from a captured image using the Pentium-MMX processor with CPU clock of 166MHz. Finally, we show the natural merging of a studio image and the virtual reality which consists of a real video and 3D graphics reflecting camera parameters.

2 Extraction of Camera Parameters

A captured image consists of three parts. One is a blue color region taken of a blue screen. The others are pentagonal patterns which are painted on the screen with some different tone and characters who act in front of the screen. In that image, the pentagonal patterns are used to extract parameters in the studio. And, the blue portions including pentagonal patterns are replaced by computer generated image or video

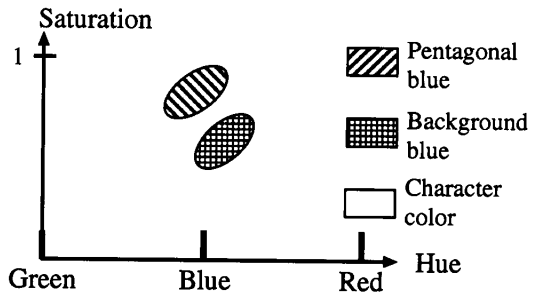


Figure 1: Color distribution of an image.

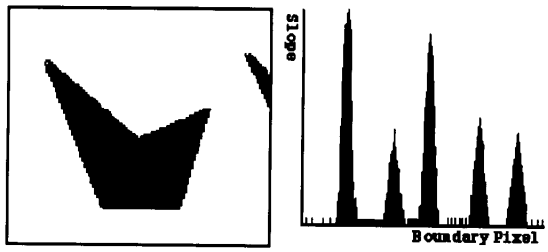
according to the camera motion.

The proposed algorithm for extracting camera parameters consists of four steps.

1. Simple color image processing is employed to separate pentagonal patterns from a captured image in virtual studio.
2. Corresponding points are found between a blue screen and a captured image using the invariant features of pentagon.
3. The projective transformation between two planes, a blue screen and a captured image, is constructed using these corresponding points.
4. We extract camera parameters by applying these transformation and corresponding points using the modification of Tasi's method[4].

2.1 Color Image processing

As shown in Fig.1, the color distribution of a captured image is divided into three regions: *pentagon blue* which indicates the pentagonal patterns, *background blue* of the blue screen except for pentagons and *character color* of the characters. Each color occupies a different bounded region in the HSI color space which consists of Hue, Saturation and Intensity component. Two principal properties for differentiating the region of pentagons are described as follow; (a) the hue and saturation values of pentagons near the predefined cluster center, and (b) the very small gradient value of intensity inside the pentagonal patterns. The properties stated in (a) and (b) can be defined by Eq.(1) and Eq.(2), respectively. Because the light sources of studio are located variously and a pentagon is defined by a bounded region that occupies small area, we divide the image pro-



(a) (b)

Figure 2: Extraction of pentagons.

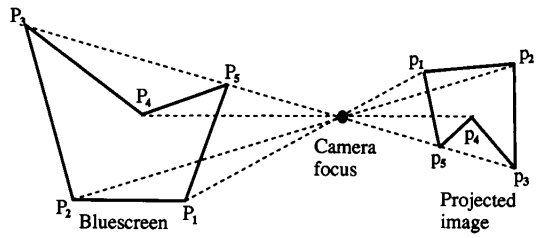


Figure 3: The projection of pentagon.

cessing for differentiating the portion of pentagons into two steps. At first, we select all possible pixels that satisfy the first property defined by Eq.(1), where (x, y) is the row and column number of a sampled pixel equally spaced by (N, N). Then, each selected pixel is treated as a seed and the region growing starts at the pixel by appending neighboring pixels that satisfy Eq.(1) and Eq.(2).

$$H_{T1} < H(x,y) < H_{T2} \quad and$$
$$S_{T1} < S(x,y) < S_{T2} \qquad (1)$$
$$\left|\frac{\partial I(x,y)}{\partial x}\right| + \left|\frac{\partial I(x,y)}{\partial y}\right| < G_T \qquad (2)$$

where $H_{T1}, H_{T2}, S_{T1}, S_{T2}$ and G_T are thresholds that were determined by preliminary experiments. After finding the region of a pentagon, we search the boundary to find five vertices using Boundary-Following algorithm. The variation of tangential slope at a vertex of pentagon is large so that an accurate pentagon can be defined by the five inflection points. Fig.2(a) shows an extracted region of pentagon by region growing and Fig.2(b) shows slopes of tangent line at the boundary pixels.

2.2 Corresponding points

In the pinhole camera model, the camera parameters characterize the formation of an image by the projection of 3D points onto an image plane. In other words, there is a projective transformation that maps the pentagons on the blue screen to the pentagons on a captured image plane. In order to determine this transformation, we must find the corresponding points between the two planes. Fig.3 shows the projection of a pentagon on the blue screen to a image plane. Here, the ratio of distance on a line and the ratio of area are not preserved under the projection, but the two functionally independent invariant features as in Eq.(3) can be constructed by using the five vertices. Note that three of them are not collinear in a plane[3]. To characterize the projective transformation, the matched pentagons are determined, and then the five pairs of corresponding vertices are found from a pair of pentagons.

$$\left(\frac{|\mathbf{M}_{431}||\mathbf{M}_{521}|}{|\mathbf{M}_{421}||\mathbf{M}_{531}|}, \frac{|\mathbf{M}_{421}||\mathbf{M}_{532}|}{|\mathbf{M}_{432}||\mathbf{M}_{521}|}\right)$$
$$= \left(\frac{|\mathbf{m}_{431}||\mathbf{m}_{521}|}{|\mathbf{m}_{421}||\mathbf{m}_{531}|}, \frac{|\mathbf{m}_{421}||\mathbf{m}_{532}|}{|\mathbf{m}_{432}||\mathbf{m}_{521}|}\right) \qquad (3)$$

where $\mathbf{M}_{ijk} = (\mathbf{P}_i, \mathbf{P}_j, \mathbf{P}_k)$ with $\mathbf{P}_i = (x_{wi}, y_{wi}, 1)^T$ and $\mathbf{m}_{ijk} = (\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k)$ with $\mathbf{p}_i = (X_i, Y_i, 1)^T$. And $|\mathbf{M}|$ is the determinent of matrix $\mathbf{M}$.

2.3 Projective transformation

A projective transformation between two projected planes can be represented by a generalized linear transformation denoted in Eq.(4) and Eq.(5) [3].

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & 1 \end{pmatrix} \begin{pmatrix} x_w \\ y_w \\ 1 \end{pmatrix} \qquad (4)$$

$$X = x/z \ , \ Y = y/z \qquad (5)$$

where (x_w, y_w) is the 2D coordinate of a blue screen and (X, Y) is that of a projected image plane.

The projective transformation matrix requires eight independent parameters to define a unique mapping. Since each pair of corresponding points provides two equations, it is necessary to

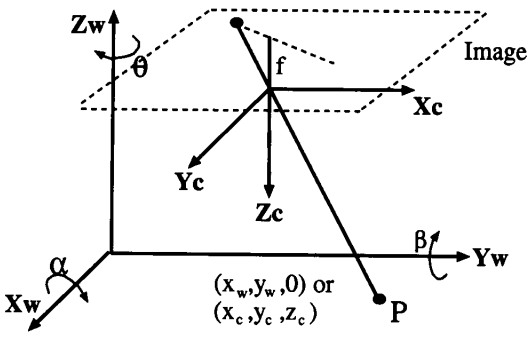


Figure 4: The Pinhole camera model.

find four pairs of points. But, we have more than four points available so that the least squares method is used to maximally satisfy the conditions.

2.4 Camera Calibration

In this paper, we consider nine camera parameters, which represent the motion of camera in virtual studio, as follows:

1. Extrinsic Parameters
   (a) The position of a camera in the virtual studio (X_0, Y_0, Z_0)
   (b) The direction of camera (α : $Tilt$, β : Pan, θ : $Role$)
2. Intrinsic Parameters
   (a) Horizontal focal length (f_x)
   (b) Vertical focal length (f_y)
   (c) Radial distortion of lens(κ_1).

where the lens distortions of a camera consist of three components: radial, decentering and thin prism. Note that the components of decentering and thin prism are negligible compared to that of radial.

Using the projective transformation acquired in the previous step, we calculate six parameters which include α,β,θ,X_0,Y_0 and the ratio of focal length, f_y/f_x, under the assumption of no lens distortions. Then, f_y, Z_0, κ_1 are calculated based on the corresponding points and previously determined six parameters.

The basic geometry of the camera can be modeled by Fig.4. where (x_w, y_w, z_w) is the coordinates of the object point P in the 3D world coordinate system and (x_c, y_c, z_c) is the coordinate of the object point P in the 3D camera coordinate system. Since object points exist on a plane, a blue screen of virtual studio can be represented as $z_w = 0$. The transformation from the world coordinate system to the camera coordinate system is given by Eq.(6) using the extrinsic camera parameters.

$$(x_c, y_c, z_c)^T = \mathbf{R}(x_w, y_w, 0)^T + \mathbf{T} \qquad (6)$$

where $\mathbf{R} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}$ is an orthonormal direction matrix expressed as a function of α, β and θ, and $\mathbf{T} = (X_0, Y_0, Z_0)^T$ is a translation vector. The projection of a point P in the camera coordinate system to image coordinates system is characterized by

$$X_f - X_c = \frac{f}{d_x}\frac{x_c}{z_c} = f_x\frac{x_c}{z_c} \qquad (7)$$

$$Y_f - Y_c = \frac{f}{d_y}\frac{y_c}{z_c} = f_y\frac{y_c}{z_c}. \qquad (8)$$

In Eq.(7) and Eq.(8), (X_f, Y_f) represent the row and column number in the image plane centered at (X_c, Y_c). Also, d_x and d_y[mm] is the center to center distance between adjacent elements of CCD sensor in the horizontal and vertical directions, respectively. But, dx/dy should be amended to reflect the characteristics of frame grabber so that we use f_x and f_y which absorbs d_x and d_y, respectively.

Using the transformation of coordinate system defined by Eq.(6), we can rewrite Eq.(7) and Eq.(8) as

$$X_f - X_c = f_x\frac{r_{11}x_w + r_{12}y_w + X_0}{r_{31}x_w + r_{32}y_w + Z_0} \qquad (9)$$

$$Y_f - Y_c = f_y\frac{r_{21}x_w + r_{22}y_w + Y_0}{r_{31}x_w + r_{32}y_w + Z_0}. \qquad (10)$$

Also, using the projective transformation in Eq.(4), we can rewrite Eq.(5) as

$$X_f - X_c = \frac{t_{11}x_w + t_{12}y_w + t_{13}}{t_{31}x_w + t_{32}y_w + 1} \qquad (11)$$

$$Y_f - Y_c = \frac{t_{21}x_w + t_{22}y_w + t_{23}}{t_{31}x_w + t_{32}y_w + 1}. \qquad (12)$$

Note that Eq.(9)-Eq.(12) has the similar form except that each equation has different coefficients. Therefore, we can calculate the relationships between the parameters of camera and the

coefficients of projective transformation as

$$\frac{r_{11}}{Y_0} = \frac{t_{11}}{t_{23}}\frac{f_y}{f_x}, \frac{r_{12}}{Y_0} = \frac{t_{12}}{t_{23}}\frac{f_y}{f_x}, \frac{X_0}{Y_0} = \frac{t_{13}}{t_{23}}\frac{f_y}{f_x}$$
$$\frac{r_{21}}{Y_0} = \frac{t_{21}}{t_{23}}, \frac{r_{22}}{Y_0} = \frac{t_{22}}{t_{23}}, \frac{r_{31}}{t_{32}} = \frac{t_{31}}{t_{32}}. \quad (13)$$

By applying six relations in Eq.(13) to Tasi's method, we can obtain X_0, Y_0, the direction matrix $\mathbf{R}$ and f_y/f_x which are invariant to the radial distortion of lens[4].

In the case of considering the radial distortion, Eq.(10) can be modified as

$$X_d + \kappa_1(X_d^2 + Y_d^2) = f_y \frac{r_{21}x_w + r_{22}y_2 + Y_0}{r_{31}x_w + r_{32}y_w + Z_0}. \quad (14)$$

where $X_d = X_f - X_c, Y_d = Y_f - Y_c$ is distorted coordinate of the object point P in the image plane and κ_1 is a factor of radial distortion. In order to calculate the f_y, Z_0, and κ_1, the approximations of f_y and Z_0 can be obtained from the overdetermined system of linear equations, that can be derived by Eq.(10) under the assumption of no distortion. The approximations of f_y and Z_0, and $\kappa_1 = 0$ are utilized to initiate the steepest decent method to refine the f_y, Z_0, and κ_1 in Eq.(14). In this process, the corresponding points are used to construct the system of linear equations from Eq.(10) and to define the optimization problem from Eq.(14).

3 Experimental results

We implemented the proposed method on color images captured in a small studio under different camera postures.

a) Experimental condition: Fig.6 shows captured images in virtual studio Considering a case that objects overlap with pentagons, we painted five pentagons on the screen. The color of pentagons, H_{T1}, H_{T2}, S_{T1}, and S_{T2} defined in section 2.1, were experimentally determined according to the illumination of studio. The images were acquired with a Kodak DC210 digital camera that has the spatial resolution of 640 × 480 pixels and the 24bit full color. Our method was implemented by Visual C++ in the Pentium-MMX processor with CPU clock of 166MHz.

b) The recognition of pentagons: To verify the robustness of the invariant feature of a pentagonal pattern under projection, we drew the euclidean distance of a pair of invariant features

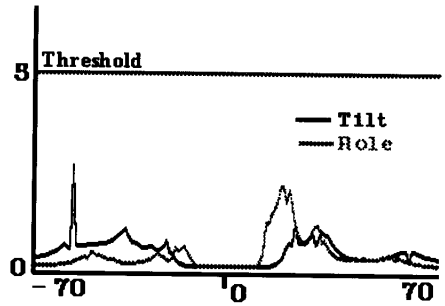


Figure 5: The difference of invariant feature.

which are calculated from two pentagons, that is , a pentagon on a blue screen and a corresponding pentagon on a captured image after camera moving. The result of this experiment is shown in Fig.5, the horizontal axis represents the angle of camera pan and tilt and the vertical axis represents the distance. In this graph, the distance is almost zero at the angle between -20 degree and 20 degree. For two pentagons, one doesn't correspond to the other, the distance of features was greater than 100. Therefore, when we used 5 as the threshold of distance, we could recognize all pentagons which were projected under any posture of camera. The accuracy of camera parameters according to the number of pentagons involved in the calibration process is derived by the average distance between calibration points of a captured image and reconstructed points of a blue screen using camera parameters. Because the average distance is consistent over the 4 pentagons, the proper number of pentagon is 4 or 5.

c) Chromakey merging: Fig.6 shows captured images after moving camera to the left and to the right respectively. In a captured image, the white boundaries represent extracted pentagons using the method described in section 2.1. Also, as we can see, the pentagon overlapped by the character(toy tank) were excluded from finding corresponding points. Fig.7 shows the natural merging of a captured image(Fig.6) and the virtual reality which consists of a real image (a scene of desert) and 3D graphics(a table model) according to the extracted camera parameters.

d) Processing time: Under the assumption that the HSI values of pixels were supplied by extra hardware, it takes about 52ms to find cor-