A Hough Transform-based method for Radial Lens Distortion Correction

R. Cucchiara, C. Grana, A. Prati, R. Vezzani

Dipartimento di Ingegneria dell'Informazione - Università di Modena e Reggio Emilia

{cucchiara.rita,grana.costantino,prati.andrea,vezzani.roberto}@unimore.it

Abstract

The paper presents an approach for a robust (semi-)automatic correction of radial lens distortion in images and videos. This method, based on Hough Transform, has the characteristic to be applicable also on videos from unknown cameras that, consequently, can not be a priori calibrated. We approximated the lens distortion by considering only the lower-order term of the radial distortion. Thus, the method relies on the assumption that pure radial distortion transforms straight lines into curves. The computation of the best value of the distortion parameter is performed in a multi-resolution way. The method precision depends on the scale of the multi-resolution and on the Hough space's resolution. Experiments are provided for both outdoor, uncalibrated camera and indoor, calibrated one. The stability of the value found in different frames of the same video demonstrates the reliability of the proposed method.

1. Introduction

In computer vision the basic camera model for image formation is the pin-hole camera model [1]. It assumes that each image point is generated as a direct projection of a world point through the optical center. In practice, the model of real cameras is not so ideal, and many factors contribute to introduce different types of distortion that can slightly change the object's appearance.

The more common non-linear distortion is due to the lens (in particular wide-angle or low-cost lenses) and causes rectilinear (straight) lines that do not pass through the optical center to be represented as curved lines (i.e., the radial distortion does not change the position of optical center). This problem can be sometimes neglected but becomes critical in many applications of computer vision where shape recognition, localization and tracking (in case of videos) is essential. Therefore, a mathematical formulation of the distortion has been proposed, together with camera calibration methods to correct the problem (as the plumb line technique proposed in [2]).

The image distortion can be decomposed in two factors: radial and tangential distortion. Tsai [1] has demonstrated that for many computer vision applications, tangential distortion can be neglected. In addition, several works showed that typically only lower-order term of radial distortion k_1 must be taken into account [1,3].

In this paper we propose a simple but reliable method to compute the best estimation of this parameter to correct the radial distortion in an automatic or semiautomatic way. This method can be applied to single images or to videos and can be adopted as an initial step for numerous computer vision processes. We tested the algorithm on different situations (both indoor and outdoor) and in particular on video from uncalibrated camera and unknown source. In these cases, a chessboard-based approach is unsuitable. Moreover, we compared the results with manual corrections to demonstrate the effectiveness of the method.

2. The radial distortion

The pin-hole camera model maps a 3D point P whose coordinates in the camera-centered coordinate system are (X,Y,Z) to an image point $p=(x,y)$ on the image plane:

$$\begin{aligned} x &= f\frac{X}{Z} \\ y &= f\frac{Y}{Z} \end{aligned} \tag{1}$$

As aforementioned, this model is not satisfactory for real cameras [4]. A more general model of distortion has been proposed [5]. Given a "distorted" image point $p_d=(x_d\,,\,y_d)$ we can obtain the "undistorted" image point $p_u=(x_u\,,\,y_u)$ as follows:

$$\begin{aligned} x_u &= c_x + \left(x_d - c_x\right)\left(1 + k_1 r_d^2 + k_2 r_d^4 + \ldots\right) + \\ &+ p_1\left[2\left(x_d - c_x\right) + r_d^2\right] + 2p_2\left(x_d - c_x\right)\left(y_d - c_y\right) \end{aligned} \tag{2}$$

$$\begin{aligned} y_u &= c_y + \left(y_d - c_y\right)\left(1 + k_1 r_d^2 + k_2 r_d^4 + \ldots\right) + \\ &+ p_2\left[2\left(y_d - c_y\right) + r_d^2\right] + 2p_1\left(x_d - c_x\right)\left(y_d - c_y\right) \end{aligned} \tag{3}$$

where (c_x,c_y) are the coordinates of the center of distortion and $r_d = \sqrt{x_d^2 + y_d^2}$.

Proceedings of the 12th International Conference on Image Analysis and Processing (ICIAP'03)
0-7695-1948-2/03 $17.00 © 2003 **IEEE**

As above mentioned, previous works [1,3] have demonstrate that we can neglect the tangential components:

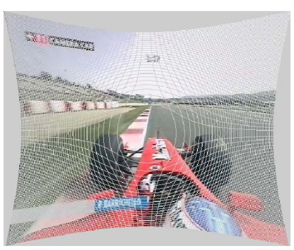
$$
\begin{aligned}
x_u &= c_x + \left(x_d - c_x\right)\left(1 + k_1 r_d^2 + k_2 r_d^4 + \ldots\right) \\
y_u &= c_y + \left(y_d - c_y\right)\left(1 + k_1 r_d^2 + k_2 r_d^4 + \ldots\right)
\end{aligned}
\tag{4}
$$

In practice, the lens distortion model can be written as an infinite series. Again, several tests have demonstrated that approximating the series with only the lower-order component corrects more than the 90% of the radial distortion [4,6].

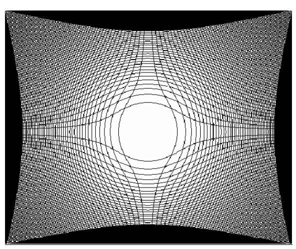
Therefore, equations (4) can be written as:

$$
\begin{aligned}
x_u &= x_d + \left(x_d - c_x\right)\left(k_1 r_d^2\right) \\
y_u &= y_d + \left(y_d - c_y\right)\left(k_1 r_d^2\right)
\end{aligned}
\tag{5}
$$

From equation (5) we can map points from distorted to undistorted image. Unfortunately, not all the points of the undistorted image are covered due to the quantization of the coordinates. Fig. 1 reports an example: in particular, Fig. 1(b) shows the locus of points for which the undistorted coordinates are defined. To overcome to this problem, what we need is a function that maps between undistorted coordinates and the corresponding distorted coordinates.



(a)



(b)

Fig. 1. Example of incomplete mapping on the undistorted image. (b) shows the locus of points for which the coordinates are defined.

Considering the center of distortion coincident to the position (0,0), this equation can be re-written as a polynomial of degree three in r_d, and thus solved by using the Cardano method. If k_1 is positive, this method provides the only real solution of the polynomial, that is:

$$
r_d = \sqrt[3]{\frac{r_u}{2k_1} + \sqrt{\left(\frac{1}{3k_1}\right)^3 + \left(\frac{r_u}{2k_1}\right)^2}} + \sqrt[3]{\frac{r_u}{2k_1} - \sqrt{\left(\frac{1}{3k_1}\right)^3 + \left(\frac{r_u}{2k_1}\right)^2}} \tag{6}
$$

where $r_u = \sqrt{x_u^2 + y_u^2}$.

Once r_d is computed, we can obtain the distorted coordinates with the following equations:

$$
\begin{aligned}
x_d &= c_x + \left(x_u - c_x\right)\frac{r_d}{r_u} \\
y_d &= c_y + \left(y_u - c_y\right)\frac{r_d}{r_u}
\end{aligned}
\tag{7}
$$

Different automatic procedures to correct radial distortion have been proposed in the previous works. Authors in [3] propose a global optimization method that aim at maximizing the number of straight lines in the whole image. As evident, this method is not suitable in images with few straight lines and many real curves that act as distractors. Other methods rely on the classical chessboard-based calibration technique by Zhang [7] to compute the distortion parameters. Unfortunately, this technique requires to acquire a set of images containing the chessboard in different positions and orientations. This is not possible if the aim is to remove lens distortion from pre-registered videos from unknown source. If the camera is unknown and can not be a priori calibrated, image-based methods must be exploited. Similarly to [3], our method is based on Hough transform and it will be detailed in the next section.

3. The HTRDC method

As stated in previous section, the proposed method does not require any previous calibration and thus can be applied directly on the acquired or stored videos, independently on the camera source. The basic idea relies on the aforementioned definition of radial distortion, i.e. on its characteristic to distort straight lines that do not pass through the optical center.

Accordingly, our method is based on two step:

1. automatic or semi-automatic identification of candidate line(s);
2. iterative variation of the distortion parameter so as to maximize the *straightness* of the candidate line(s).

Since the most known and well assessed method to detect lines and to measure their straightness is the Hough Transform (HT) [8], we adopt it for both steps. As a consequence, we called our method HTRDC, that stays for HT-based Radial Distortion Correction.

The first step of the method aims at finding a region of interest (ROI) where the distortion is evident. The ROI can be provided to the system in a manual way by asking to the user to select a zone of the image where a line should be found (Fig. 2(a)). This method with manual mask selection is particularly useful as an initial fast tool when camera's parameters have to be set. The manual selection can be replaced by an automatic, iterative process that will be detailed in the following.

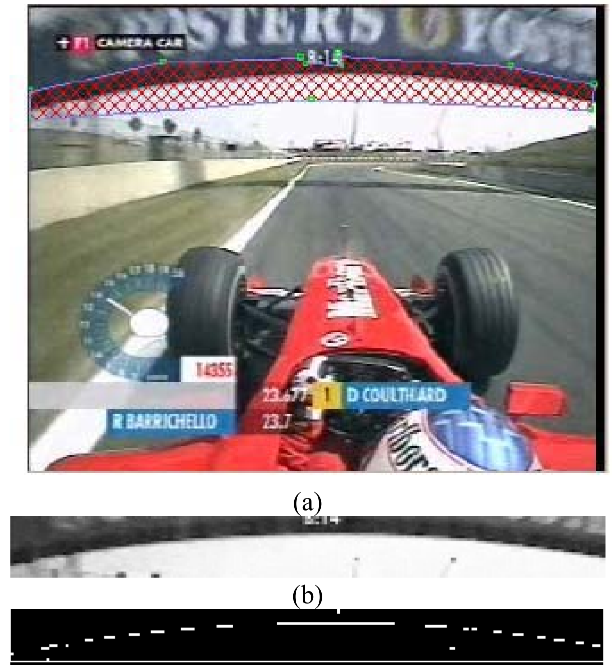
Some parameters must be a priori defined:

Proceedings of the 12th International Conference on Image Analysis and Processing (ICIAP'03)
0-7695-1948-2/03 $17.00 © 2003 **IEEE**

A. k^{min}, k^{max}: the minimum and maximum value of the distortion parameter k_1. Since k_1 is defined in the range [0,1], initially k^{min}=0 and k^{max}=1. The range can be limited if the computational load is excessive;

B. c_x , c_y: the coordinates of the center of radial distortion. If no other information are provided and assuming a symmetrical error, we can set them to (0,0);

C. the number n of samples of k_1 for each iteration and the acceptable error ε on the value of k_1. These values must be set in dependence on the required precision and taking into account that there is an unavoidable approximation due to the quantization of the image space.



(a)

(b)

(c)

Fig. 2. Example of manual ROI extraction (a) and (b), and application of Canny's edge detector (c).

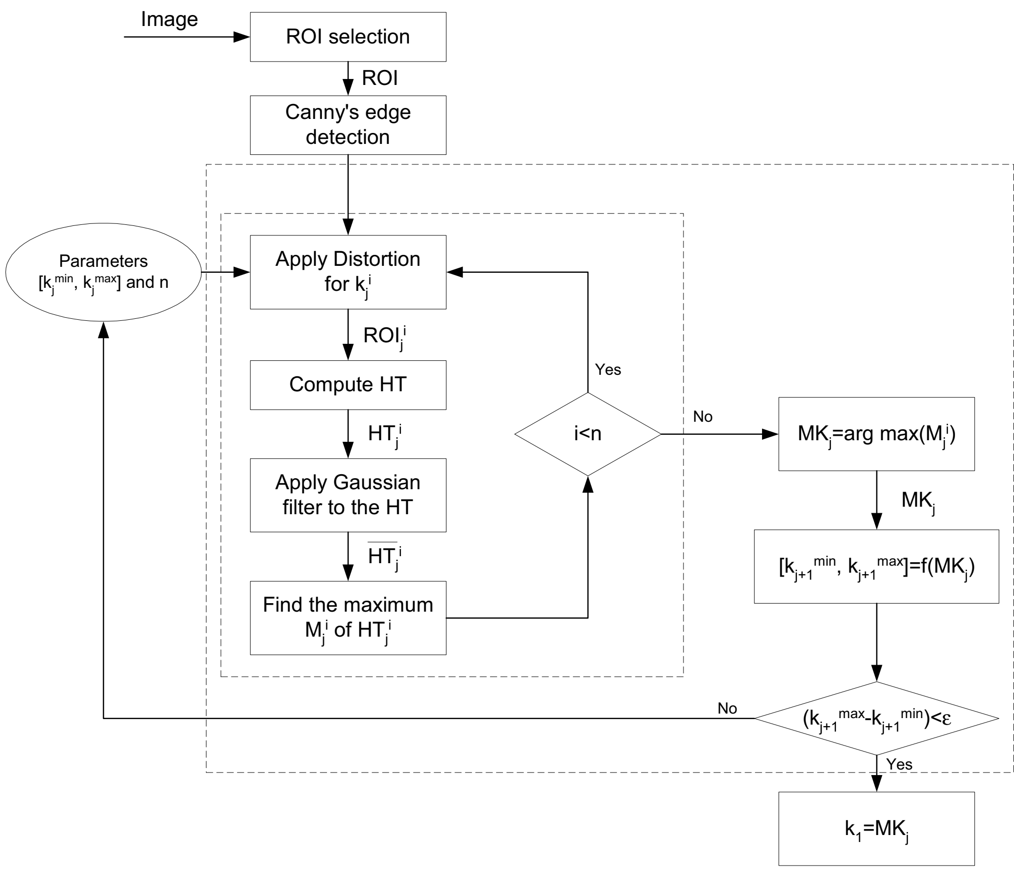


Fig. 3. Data flow of the HTRDC proposed system.

A Canny's edge detector [9] is applied to the extracted ROI (Fig. 2(b) and an hysteresis non-maxima suppression method is exploited to extract "strong" edges only (Fig. 2(c)). Authors in [3] claim that there is a need for a sub-pixel accuracy edge detector in this type of problems. Although we agree with them, Canny's method is enough precise working on a ROI instead of the whole image.

After edge detection we iteratively apply to the ROI a distortion in a multi-resolution way. As depicted in Fig. 3, starting from the initial k^{min} and k^{max} we divide the range in n steps such that for each step i the distortion parameter is computed as:

$$k^i = k^{\min} + i\frac{k^{\max} - k^{\min}}{n} \tag{8}$$

By using this value as k_1 in equation (6) we obtain the "undistorted" version of the ROI for the j-th iteration and the i-th step (ROI_j^i). In particular, the system first sets to 0 each point of ROI_j^i; then, for each edge point found in the original ROI the corresponding coordinates (x_d, y_d) in the undistorted ROI_j^i are computed and the value of ROI_j^i in that position incremented. By doing this, we assure that the number of edge points in the undistorted image remains equal to the number of edge points in the original ROI.

Then, the Hough Transform for straight lines is computed. The Hough space is defined as:

$$HT_j^i = \left\{ h_j^i(\rho,\theta), \rho \in [-R,+R], \theta \in [0,\pi[\right\} \tag{9}$$

where $R = \sqrt{w_{ROI}^2 + h_{ROI}^2}$ with w_{ROI} the width of the selected ROI and h_{ROI} the height. Thus, R is the maximum ρ distance in the ROI area. The standard HT with parametric line equation is slightly modified, with a weight proportional to the ROI_j^i (x,y) value, that is:

```
for each (x,y)∈ROIj^i and
for each θ=a•Δθ with a=0,…,⌈2π/Δθ⌉
{    ρ←⌈x•cosθ+y•sinθ⌉
     HTj^i(ρ,θ)←HTj^i(ρ,θ)+ROIj^i(x,y)  }
```

The choice of the quantization values (Δρ,Δθ), i.e. the resolution of the Hough space, must be correlated with the desired precision of the k_1 parameter. In [10], the authors defined the relation between the image size (in our case the ROI dimension, where the HT is computed) and the acceptable quantization of the Hough space. In particular, if we fix the acceptable precision of k_1 to the second decimal digit, acceptable value for (Δρ,Δθ) are (2,π/90) or (1, π/180).

As shown in Fig. 3, the next step is to post-process the HT space with a Gaussian filter in order to smooth possible oscillations due to the quantization of both the image and the HT space.

To evaluate the straightness of the undistorted ROI, we can found the global maximum M_j^i of the HT space after the post-processing:

$$M_j^i = \max_{(\rho,\theta)}\left(\overline{HT_j^i}(\rho,\theta)\right) \tag{10}$$

As above said, the method is very simple but effective. It allows us to draw a graph of the straightness in function of the value of k_1. This process is repeated for each of the n steps in which the range of k_1 has been divided. The value of k_1 corresponding of the maximum of these maxima is the candidate MK_j for the j-th iteration.

The next iteration begins by computing the new range of values of k_1 as a function of MK_j:

$$\begin{aligned} k_{j+1}^{\min} &= \max\left(0, MK_j - \frac{k_j^{\max} - k_j^{\min}}{n}\right) \\ k_{j+1}^{\max} &= \min\left(MK_j + \frac{k_j^{\max} - k_j^{\min}}{n}, 1\right) \end{aligned} \tag{11}$$

and re-iterating the above described process. In addition, we provide the possibility to change the number n of samples at each iteration. The recursion ends when the newly computed range of k_1 value is smaller than a fixed threshold ε. When this conditions is satisfied, the value MK_j of the last iteration is chosen as best k_1 value.

From our experiments, the best choice is to start with a number n of samples greater than 10 (from 10 to 100) and to increase n at each iteration. With this parameter we are able to estimate the best k_1 value with an error of ε=0.001.

Obviously, the computed value of k_1 depends on the initially selected ROI. If there are strong edges in the ROI the effect of changing the radial distortion is more evident, and, assuming that the edges belong to a "supposed straight" part, the proposed method is effective.

When the ROI is not manually selected, the system can provide an automatic procedure that uses a method also based on Hough transform to select an area, possibly far from the center of distortion, with a high global straightness. In particular, the system concentrates the search only on quasi-horizontal or quasi-vertical lines, that are $\theta \in [-\tau_\theta, +\tau_\theta] \cup \left[\frac{\pi}{2} - \tau_\theta, \frac{\pi}{2} + \tau_\theta\right]$. The HT is computed on the whole image iteratively for a certain number s (for example s=10) of samples of the whole range of k_1. The HT spaces obtained are correlated to compute the space $\mathrm{H} = \prod_s HT_s$. The parameter pair (ρ_0, θ_0) that maximizes H can be used to apply an antitransformation and thus obtain the approximated ROI in the image plane.

4. Experiments and discussion

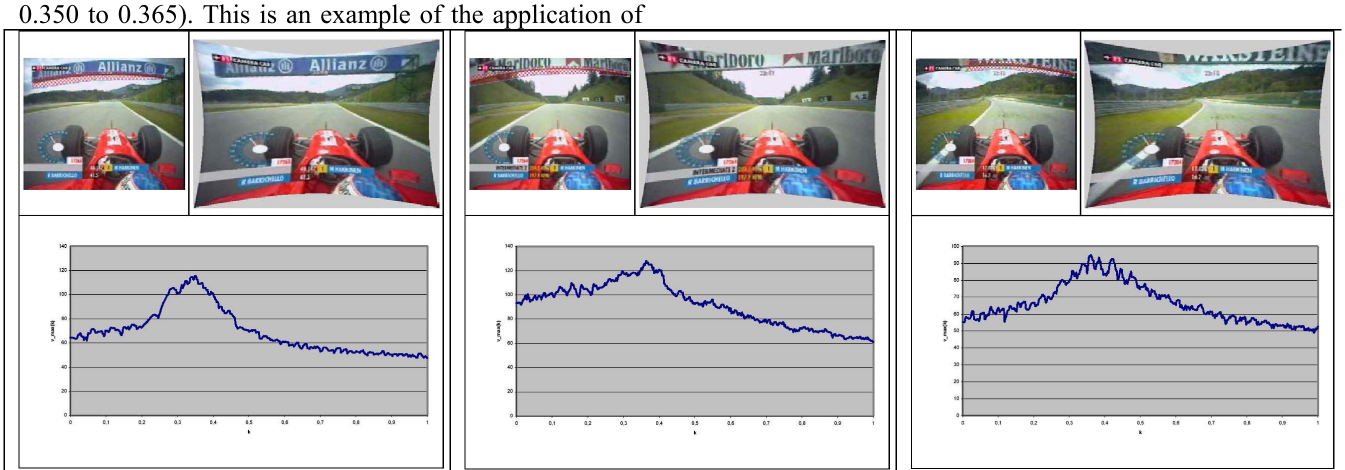
The HTRDC method relies heavily on the concept and definition of radial distortion. Since the radial distortion is independent on the time, on the camera motion or on the scene, it can be used to find the best distortion parameter at the beginning of the process and only once for each camera. The only requirement is that the scene contains some strong-contrast straight lines, as frequently happens. Therefore, we adopted this initial step for numerous applications, such as indoor surveillance, domotics, human body 3D modeling, vision-based traffic control and management, lane detection in videos from camera-car, and so on.

When a video is provided instead of a single image, we can repeat the distortion parameter computation on different frames, provide a more robust result as the average value computed on them. Fig. 4 shows an example of distortion correction in camera-car videos: in this case the camera mounted in-vehicle has a wide-angle lens that causes a strong distortion. This distortion considerably affects both further lane detection and speed computation (based on homography and, thus, affected by lens distortion). In fact, the distortion modifies the position of key-features (as corners) used for correlating successive frames. In Fig. 4, the graphs of the maximum values of k_1 are shown using a single step of multi-resolution with n=500 (for descriptive purpose only). Note that in the three different frames approximately the same ROI has been selected. The best k_1 value found changes among the three frames of less than the second decimal digit (in the overall experiment k_1 varies between 0.350 to 0.365). This is an example of the application of our method to a video taken with an unknown camera so that no manual calibration can be provided.

Instead, the second example (reported in Fig. 5) shows an indoor scene taken with a PTZ (Pan-Tilt-Zoom) camera. In this case, we have the possibility to manually calibrate the camera (by using the well known method by Zhang [7] and the MATLAB toolbox provided in [11]) and we are able to compare the result provided by our HTRDC method with that of the calibration, that is 0.025.

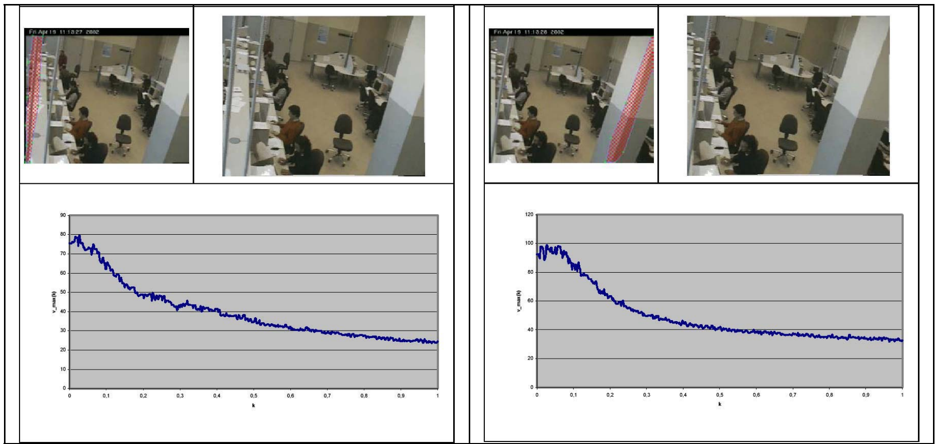
At the end, we would like to remark that this problem, often neglected, is very important in many different applications: the example of Fig. 6 demonstrates that the trajectory of a moving object in an indoor scene is distorted, affecting the tracking phase.

Fig. 6. An example of trajectory displacement due to lens distortion (green=distorted, red=undistorted).



(a) Frame 1232, computed k_1=0.352	(a) Frame 2066, computed k_1=0.364	(a) Frame 3587, computed k_1=0.362

Fig. 4. Examples of the HTRDC method in different frames of a video sequence taken from a camera-car in a Formula One race.



(a) Frame 166, computed k_1=0.028 (b) Frame 171, computed k_1=0.025

Fig. 5. Examples of the HTRDC method in different frames of an indoor video sequence.

5. Conclusions

The proposed method is a robust approach to detect and correct radial lens distortion. It can be applied to different types of videos, stored or directly acquired by cameras, and can be considered a powerful initial step for many computer vision applications. It is particularly valuable if the manual calibration is not possible since no information about the camera is available. The method's precision depends on the number of samples, the Hough space's resolution, and the straightness of the cues present in the scene. An human observer often perceives the distortion phenomenon because straight lines are distorted: our proposal relies on this and thus gives us a good approximated solution. Experiments confirm this statement and, thus, are very promising.

6. References

[1] R.Y. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses", *IEEE Journal of Robotics and Automation*, vol. 3, no. 4, Aug. 1987, pp. 323-344

[2] D.C. Brown, "Close-range camera calibration", *Photogrammetric Engineering*, vol. 37, no. 8, 1971, pp. 855-866

[3] F. Devernay, O. Faugeras, "Straight lines have to be straight", *Machine Vision and Applications*, vol. 13, no. 1, 2001, pp. 14-24

[4] R. Mohr, B. Triggs, "Projective Geometry for Image Analysis – Tutorial at ISPRS, Vienna", July 1996, http://www.dai.ed.ac.uk/CVonline/LOCAL_COPIES/MOHR_TRIGGS/node10.html

[5] C.C. Slama, *Manual of Photogrammetry, Fourth* Edition, American Society of Photogrammetry and Remote Sensing, Falls Church, Virginia, USA, 1980.

[6] O. Faugeras, G. Toscani, "Structure from Motion using the Reconstruction and Reprojection Technique", in *Proceedings of IEEE Workshop on Computer Vision*, Miami Beach, Nov.Dec. 1987, pp. 345-348

[7] Z. Zhang, "A flexible new technique for camera calibration", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, Nov. 2000, pp. 1330-1334

[8] J. Illingworth, J. Kittler, " A Survey of the Hough transform" *Computer Vision, Graphics, Image Processing*, vol 43, 1988, pp. 221-238

[9] J. Canny, "A Computational Approach to Edge Detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, Nov. 1986, pp. 679-697

[10] T.M. Van Veen, F.C.A. Groen, "Discretization errors in the Hough Transform", *Pattern Recognition,* vol 14, 1981, pp 137-145

[11] "Camera Calibration Toolbox for Matlab", http://www.vision.caltech.edu/bouguetj/calib_doc/index.html

Proceedings of the 12th International Conference on Image Analysis and Processing (ICIAP'03)
0-7695-1948-2/03 $17.00 © 2003 **IEEE**