MAPPING TEXTURE FROM MULTIPLE CAMERA VIEWS ONTO 3D–OBJECT MODELS FOR COMPUTER ANIMATION

Wolfgang Niem and Hellward Broszio

Universität Hannover
Institut für Theoretische Nachrichtentechnik und Informationsverarbeitung
Appelstr. 9A, 30167 Hannover
phone: ++49–511–762–5313, fax: ++49–511–762–5333
email: niem@tnt.uni–hannover.de

ABSTRACT

An algorithm for the mapping of texture from multiple camera views onto a 3D model of a real object is presented. The texture sources are images taken from an object rotating in front of a stationary calibrated camera. The 3D model is represented by a wireframe built of triangles and is geometrically adjusted to the camera views. The presented approach aims at the reduction of texture distortion associated with the boundaries between triangles mapped from different camera views and disturbance due to not visible parts of the object surface. For this purpose adjacent triangles describing the surface of the 3D model are grouped to homogenous surface regions which are textured with a common image followed by a local texture filtering at the region boundaries. For triangles not visible in any camera view a filter has been developed which uses the texture from adjacent visible triangles for the generation of synthetic texture. Experimental investigations with different real 3D objects have confirmed the suitability of the proposed technique for computer animation applications.

1 INTRODUCTION

An algorithm for texture mapping from multiple camera views is presented, which is applied to the automatic generation of 3D models of real objects. Those models are used in the field of computer animation like virtual studio production, trick film production, advertising, driving/flight simulators, tele–shopping, etc.. All these applications require the generation of realistic looking 3D models with simple equipment and in reasonable time to be cost efficient.

One way of automatically generating 3D models of real objects is the evaluation of camera images from multiple views. Therefore a turntable with the real object rotating in front of a stationary calibrated camera is used for image acquisition. Techniques have been described, which estimate the object shape from these images automatically [1][3][7]. The rendering of this 3D model represented by a wireframe does not satisfy the requirements of computer animation in terms of realism. Additional information about the surface texture has to be added to the 3D model to improve its appearance.

For that reason a technique called ”texture mapping” or ”texturing” has been studied in computer graphics [3][4][5], an image synthesis technique in which a texture image is mapped onto a surface of a 3D model and consecutively projected onto a 2D–screen. Applying this technique enhances the realism immensely by adding detail to the 3D model.

The use of multiple camera images for texture mapping needs special attention and has been studied only little in previous work. Hoffman [6] presented a technique for adjusting single camera images to a 3D model by hand and Busch [1] applied a method for texturing the surface model automatically from multiple camera images. In his approach he assigns texture to a surface polygon from that camera image, which ensures the highest texture resolution.

The approach presented in this paper is based on Busch’s technique but looks more closely to the following problems:

- Distortion of texture at the boundaries of triangles textured from different images; this can occur in case of errors in model shape, camera parameters or not ideal illumination conditions.
- Not visible surface parts of the real object, i.e. parts which can not be observed by the camera from any viewpoint.

In the second section the approach for the automatic 3D–modelling from multiple views is introduced, while the third section gives some basics of texture mapping. The fourth section is describing the new algorithm for mapping texture from multiple camera views and in the fifth section some results are presented.

2 AUTOMATIC 3D MODELLING FROM MULTIPLE VIEWS

Measurement environment

The environment used for this work consists of a stationary CCD TV camera in front of a turntable, which can be rotated in controlled steps (Fig 1a). The background is of uniform colour in order to facilitate the segmentation of the real object against the background by colour keying. Before the image acquisition of a real object can be started, the camera has to be calibrated in order to obtain position, orientation and focal length. For that purpose, a test pattern with precisely known control points is placed on the turntable. The detection of the control points in the image and the estimation of all parameters describing the acquisition process is performed using a calibration method as proposed by Pedersini et al. [8].

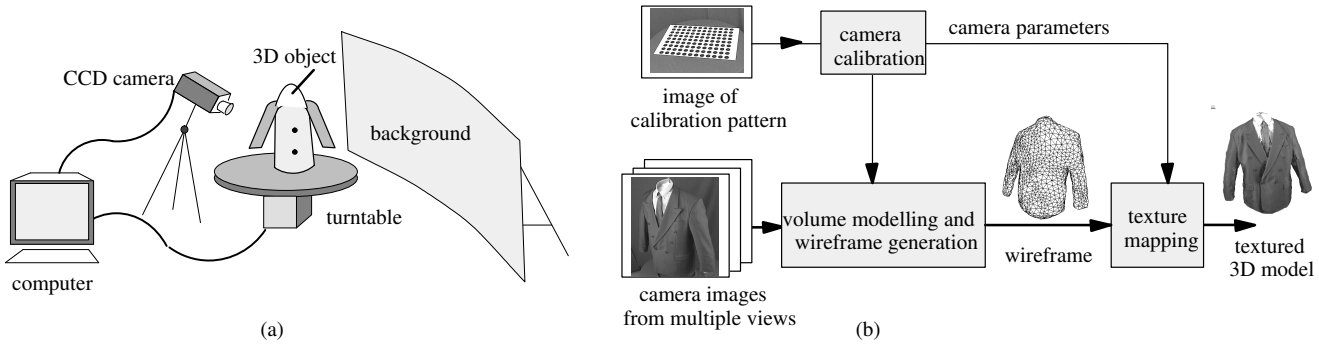

Fig. 1 (a) Measurement environment and (b) Block diagram of the automatic generation of 3D models.

Processing Pipeline

The automatic modelling process estimates a 3D model from the calibrated images in two main steps (Fig. 1b): In the first step the method of occluding contours is used to estimate a volume model from the object silhouettes [7]. The silhouettes are obtained by segmentation of the input images using a colour keying technique. The amount of data required for the representation of the 3D shape is reduced by approximating the surface of the volume model with a wireframe model consisting of triangular polygons.

In the second modelling step texture obtained from the multiple camera images is mapped onto the wireframe model. This ”texture mapping” enhances the realism significantly by adding detail to the 3D model and is the topic of this paper.

3 BASICS OF TEXTURE MAPPING

Texture mapping is an image synthesis technique in which a texture image is mapped onto a surface of a 3D model and consecutively projected onto a 2D screen. In this paper the texture image is obtained from a camera image while the 3D model surface is represented by a set of polygons (”wireframe model”). Three different coordinate systems will be used:

- *texture space* is labeled with *(u, v)* coordinates
- *3D model space* is labeled with *(x, y, z)* coordinates
- *screen space* is labeled with *(I, J)* coordinates

Following this definitions, the mapping from texture space to screen space can be subdivided into two steps [6]: In a first step called *parameterization,* the texture space is mapped onto the 3D model space, i.e. each point of the texture array is attached to a point of the polygonal surface by a mapping function. The second step is called the *viewing projection* and maps the 3D model space onto the screen space. By varying the viewing positions, arbitrary views of the parameterized model can be obtained. The parameterization is part of the 3D–modelling, while the viewing projection is part of the image synthesis and will therefore not be further investigated within his paper.

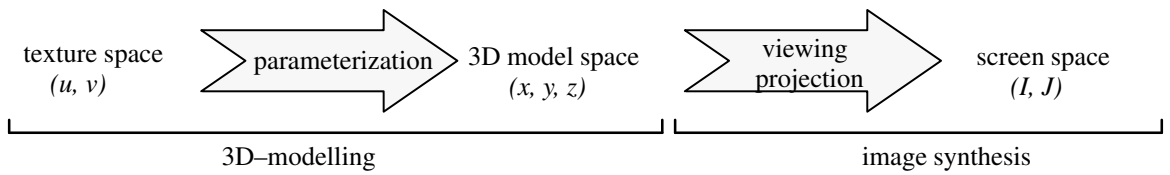

Fig. 2 Mapping from texture space to screen space using parameterization and the viewing projection.

Parameterization

Because the given 3D models are represented by triangular polygons the parameterization of polygonal surfaces will be investigated. A linear parameterization of a triangular polygon is easily performed by specifying the related texture space coordinates $\{[u_i v_i], i = 1, 2, 3\}$ of each triangle vertex $\{[x_i y_i z_i], i = 1, 2, 3\}$. This information is sufficient to determine the parameters $\{a_i, i = 1.....9\}$ of an *affine mapping* from texture space to 3D model space which is given in homogenous matrix notation by:

$$[x \;\; y \;\; z] = [u \;\; v \;\; 1] \cdot \begin{bmatrix} a_1 & a_4 & a_7 \\ a_2 & a_5 & a_8 \\ a_3 & a_6 & a_9 \end{bmatrix} \tag{1}$$

A nonlinear parameterization is the *perspective mapping* from texture space to 3D model space. This mapping is specified by the related texture space coordinates $\{[u_i v_i], i = 1, 2, 3, 4\}$ of the vertices of a planar quadrilateral $\{[x_i y_i z_i], i = 1, 2, 3, 4\}$ which is sufficient to determine the parameters $\{a_i, i = 1.....11\}$. The homogenous coordinate w is divided through to calculate the true object space coordinates:

$$[x \cdot w, \; y \cdot w, \; z \cdot w, \; w] = [u \;\; v \;\; 1] \cdot \begin{bmatrix} a_1 & a_4 & a_7 & a_{10} \\ a_2 & a_5 & a_8 & a_{11} \\ a_3 & a_6 & a_9 & 1 \end{bmatrix} \tag{2}$$

This operation can also be understood as a rectification of the original texture to a texture taken from a viewing direction perpendicular to the polygon. Note, that the affine mapping is a subset of the perspective mapping for $a_{10} = a_{11} = 0$

The choice of a suitable parameterization depends on the model describing the real camera taken for texture acquisition. In this work, the used camera can be described by the projective geometry of a pinhole camera. Therefore only the projective mapping ensures the proper perspective foreshortening while the linear parameterization of the affine mapping leads to errors. Those errors depend on (a) the triangle size, (b) the location of texture space coordinates of each triangle vertex, (c) the angle between camera viewing direction and the triangle normal, and (d) the focal length of the camera with respect to the CCD–chip size.

If the triangles are sufficiently small, the error can be kept small as follows: The texture space coordinates $\{[u_i v_i], i = 1, 2, 3\}$ are specified by a perspective projection of each triangle vertex $\{[x_i y_i z_i], i = 1, 2, 3\}$ into the texture space using the model of the real camera. Afterwards affine mapping of each point inside the triangle is applied according to equation (1).

4 TEXTURE MAPPING FROM MULTIPLE CAMERA IMAGES

Whereas the texture mapping of 3D models from single images is already described in various papers [4][5], the texture mapping from *multiple* camera images has only be paid little attention in literature and is investigated within this paper. Therefore, each polygon of the wireframe model has to be assigned to one camera image as explained in the previous section. This leads to the following problems:

- Distortion of texture at the boundaries of triangles which are textured from different images. This can occur in case of errors in model shape, camera parameters or not ideal illumination conditions.
- Not visible surface parts of the real object, i.e. parts which can not be observed by any of the camera views.

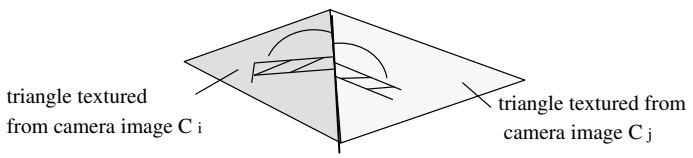

Fig. 3 Distortion of texture at the boundaries of triangles which are textured from different images

The proposed algorithm addresses those problems by applying three steps:

- Assignment of surface triangles to a camera image
- Local texture filtering at distorted boundaries between triangles assigned to different camera images
- Synthesis of texture for triangles not visible from any camera view

4.1 Assignment of surface triangles to a camera image

In a first processing step each triangle is assigned to that camera image, which ensures the highest *texture resolution* defined by the ratio of texture elements per surface area. If the distance of all camera positions to the surface triangle is almost the same, that camera image is selected, for which viewing direction the smallest angle with the surface normal of the triangle[1] is enclosed. Applying this leads to groups of triangles textured from a common camera image which triangles are further referred to as *surface regions*. Because those surface regions are quite inhomogenous, this results in a large number of distorted boundaries between them (Fig. 5a).

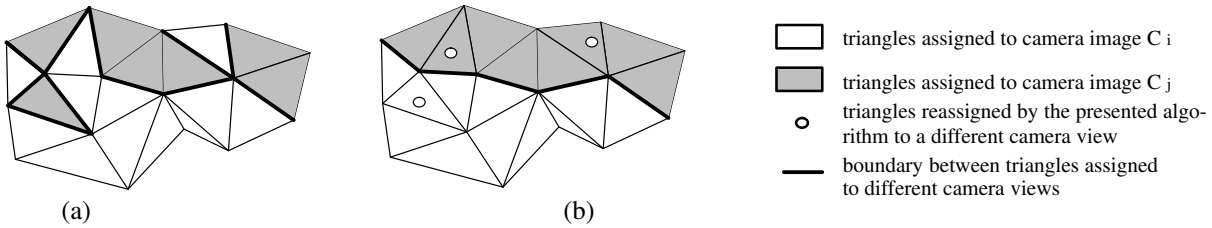

Fig. 4 Surfaces regions assigned to camera images C_i and C_j: (a) Assignment of triangles applying the criterion of maximal texture resolution, (b) assignment of triangles after applying regrouping

In order to reduce this number, a second processing step is applied which regroups the triangles applying the following algorithm:

1. *Detection of a triangle assigned to a camera image C_i which is bounded with at least two sides to triangles assigned to a different camera image C_j.*
2. *Calculation of the texture resolution for the detected triangles applying the texture from camera image C_j.*
3. *Assignment of the triangle to the camera C_j if the reduction of the texture resolution is within a predefined tolerance.*

The effect of regrouping is exemplary shown in Fig.4b: A surface composed of triangles is shown for which triangles are assigned to different camera images C_i and C_j (marked with different greylevels). Three of the triangles are detected by the algorithm and regrouped. The resulting number of distorted boundaries therefore is reduced from 11 to 5 in this example and surface regions are more homogenous.

4.2 Local texture filtering of distorted boundaries

The local texture filtering considers only two adjacent triangles textured from different camera images C_i and C_j. A possible distortion of luminance values is shown on a scanline over the surface built by the two triangles D_1 and D_2 in Fig. 5. Typically a step edge in the texture luminance can be observed at the boundary.

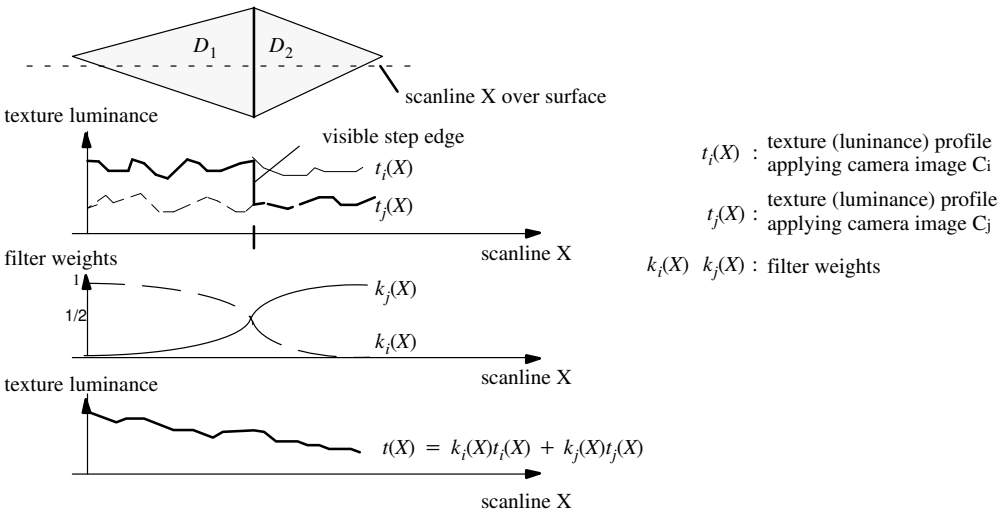

Fig. 5 Local texture filtering by averaging texture from different camera views

This distortion can be reduced by applying a filtering of the textures as follows:

1. Computing the mapping of the texture t_i from the camera image C_i onto the triangles D_1 and D_2
2. Computing the mapping of the texture t_j from the camera image C_j onto the triangles D_1 and D_2
3. Computing a weighted average texture $t = k_i(x,y,z) \cdot t_i + k_j(x,y,z) \cdot t_j$ for the triangles D_1 and D_2

The space variant filter weights $k_i(x,y,z)$ and $k_j(x,y,z)$ are chosen so that a blending of the two textures is obtained from triangle D_1 to D_2 as shown in Fig. 5.

4.3 Synthesis of texture for not visible triangles

With the applied measurement environment some triangles of the wireframe are not visible from any viewing direction. In order to obtain 3D models for which viewing projections look natural from *any* direction, reasonable texture has to be synthezised for those triangles.

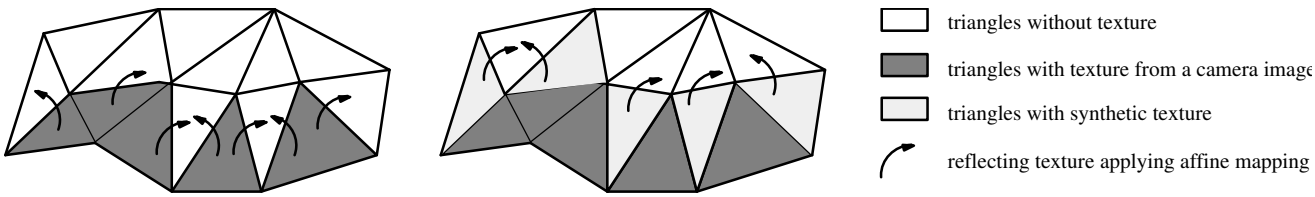

Fig. 6 Two consecutive phases of the algorithm for texture synthesis

The applied algorithm solves this by using the texture from adjacent visible triangles. It starts with those triangles, which have at least one visible adjacent triangle and proceeds as follows:

1. Reflecting texture of an already textured triangle across the common edge into the visible triangle using an affine mapping.
2. Computing a weighted average texture according to 4.2 if the triangle has two visible adjacent triangles.
3. Repeating of this procedure until all triangles are textured.

The applied reflection ensures that the triangle boundaries are not visible.

5 RESULTS

The described algorithm was tested on a number of real 3D objects. 36 images have been acquired from each object with a constant rotation of the turntable of 10^0 between consecutive views. Each image has a resolution of 720x576 pel. The effect of the applied regrouping algorithm is shown in Fig. 7a. Experiments with different real objects show, that with the applied regrouping the total length of surface region boundaries is reduced by 50 % tolerating a 5% reduction of texture resolution for the regrouped triangles.

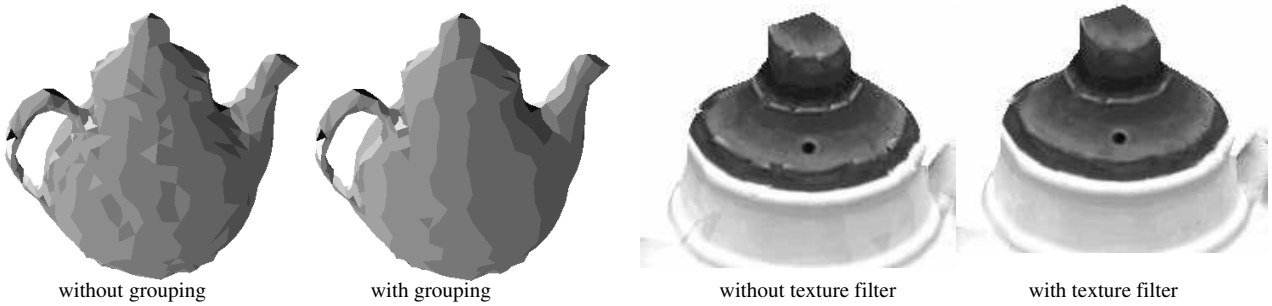

Fig. 7 (a) Effect of grouping triangle to homogenous surface regions (marked with a common greylevel) (b) effect of the texture filter

The effect of the applied texture filter is shown in Fig. 7b. Obviously the remaining errors of the viewing projection in the image with texture filtering is less disturbing.

In Fig. 8a a viewing projection of a model object with and without synthetic texture is shown. The projection of the model without using synthetic texture is disturbed from this viewing direction, because some triangles without any texture are visible. The projection of the model with synthetic texture looks reasonable from this direction.

Two viewing projections of a model ”antilope” in Fig. 8 emphasizes the applicability of the algorithm for complex objects.

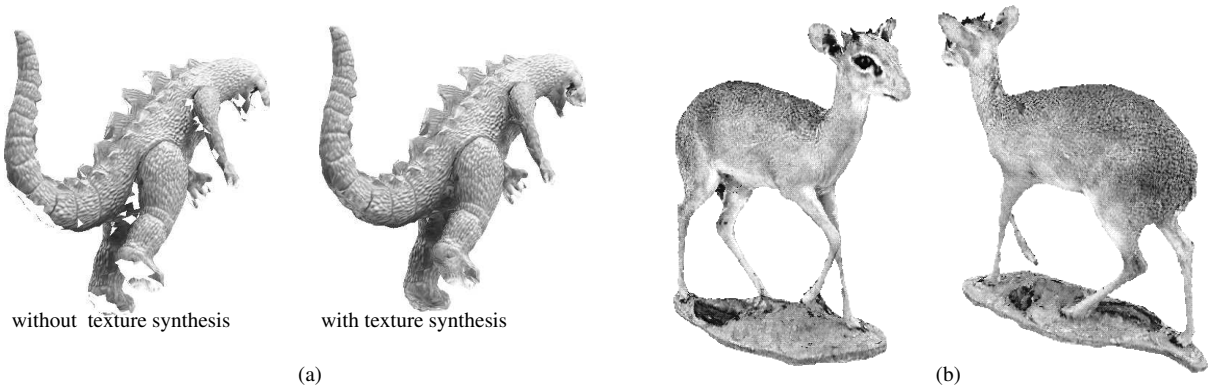

Fig. 8 (a) Effect of the texture synthesis (b) two viewing projections of a model ”antilope”

6 CONCLUSIONS

A new algorithm for the mapping of texture from multiple camera views onto 3D models has been presented. It addresses the problems of :

- Distortion of texture at the boundaries of triangles textured from different images; this can occur in case of errors in model shape, camera parameters or not ideal illumination conditions.
- Not visible surface parts of the real object, i.e. parts which can not be observed by the camera from any viewpoint.

The implemented approach can be subdivided into three steps, where step one and two are related to the first problem and step three to the second problem.

In a first step adjacent triangles describing the surface of the 3D model are grouped to a surface region which is textured with a common image. For that purpose a high texture resolution as well as a small lenght of boundaries between those surface regions are considered. In a second step the boundaries between adjacent surface regions are filtered with a local texture filter. Finally the texture for surface regions which are invisible in any image is synthesized from the texture of adjacent surface regions in a last step.

Results with real image sequences have confirmed the improvement in model quality compared to former proposed algorithm. The total length of distorted boundaries could be reduced by a factor of two and the local texture filter reduces the distortion at the remaining boundaries effectively. The applied texture synthesis filter ensures that natural looking texture is applied to triangles not visible from any camera view.

The overall quality of the models emphasizes their usefulness for a variety of applications in the field of computer animation.

7 ACKNOWLEDGEMENTS

This work was supported by the RACE II project R2052 ”MONA LISA”. Partners in this project are Thomson Broadband Systems, BBC, Daimler Benz, DVS Digitale Videosysteme GmbH, Queen Mary & Westfield College, Siemens AG, University of Balearic Islands, University of Hannover and VAP Video Art Production GmbH.

8 REFERENCES

[1] H. Busch, ”Ein Verfahren zur Oberflächenmodellierung dreidimensionaler Objekte aus Kamerabildfolgen”, *Ph.D. thesis, Universität Hannover*, 1991.

[2] E. Catmull, ”A Subdivision Algorithm for Computer Display of Curved Surfaces”, PhD Thesis, Report UTEC–CSc–74–133, Computer Science department, University of Utha, Salt Lake City, 1974.

[3] C. H. Chien and J. K. Aggarwal, "Identification of 3D objects from multiple silhouettes using quadtrees/octrees ", *Comp. Vision, Graphics, and Image Processing* 36, 1986, pp. 256–273.

[4] J. D. Foley et al., "Computer Graphics: Principles and Practice", Addison Wesley, 1992.

[5] P. S. Heckbert, "Fundamentals of Texture Mapping and Image Warping", Dept. of Electrical Engineering and Computer Science, University of California, Berkley, CA 94720, 1989.

[6] G. R. Hofmann, "Non–planar Polygons and Photographic Components for Naturalism in Computer Graphics", *Proceedings of the 10th Eurographics*, Hamburg 1989, ISBN 0–444–88–013–5.

[7] W. Niem, "Robust and Fast Modelling of 3D Natural Objects from Multiple Views", *SPIE Proceedings "Image and Video Processing II"*, Vol. 2182, 1994, pp. 388–397.

[8] F. Pedersini, S. Tubaro, F. Rocca, "Camera Calibration and Error Analysis. An Application to Binocular and Trinocular Stereoscopic System", *4th Int'l Workshop on Time–varying Image Processing and Moving Object Recognition*, June 1993, Florence, Italy.