EUROGRAPHICS 2003 / M. Chover, H. Hagen and D. Tost

Short Presentations

An approach to construct augmented CAD models using acquired digital images.

F. Bianconi and P. Conti

Department of Industrial Engineering, University of Perugia, Perugia, Italy

---

Abstract

An approach for integrating CAD models and information associated with digital images is discussed in this paper. The main objective of the proposed framework is the construction of augmented CAD models using digital images acquired on real components of that models. An approach to associate general shape surfaces with related images is discussed. The proposed method can be used to associate engineering information retrieved through optical methods (thermoelasticity, photoelasticity, surface analysis and similar) with CAD models. The overall architecture and the first outcomes of the implementation of the framework are presented.

Categories and Subject Descriptors (according to ACM CCS): I.4.1 [Computer Graphics]: Digitization and Image Capture

---

1. Introduction

In product design and manufacturing digital images represent an important source of information. A wide literature is available on how digital images are used to retrieve information on distribution of temperature[16], stress[15, 18], strain[14] and surface roughness[1, 2] on mechanical components. In many cases both CAD and physical prototypes of the objects exist, and it would be useful to link the information embedded in the digital images to the 3D CAD model.

The main problem investigated in this paper is the following: given a CAD model of a part and a set of digital images of that component carrying some kind of information about it, define a framework to associate such information to the CAD model. In other words the problem can be stated as the construction of augmented CAD models given a set of digital images. The CAD model is assumed to be available in any form: it can be the result of a design process or it can be obtained through reverse engineering tasks.

It has been written [7] that augmented reality is the process of integrating virtual objects into real images. Otherwise the process described in this paper can be considered as the integration of real images into virtual objects. With reference to fig. 1, we can see that the proposed procedure involves the following main aspects:

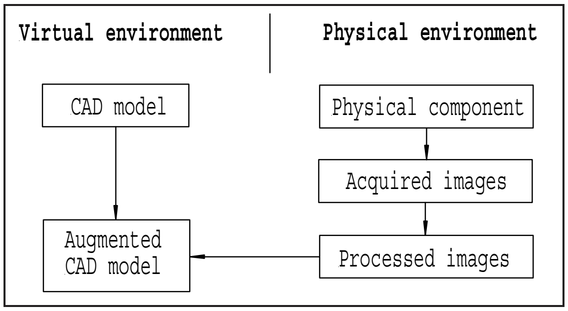

Figure 1: *Overview of the proposed framework.*

1. acquisition of digital images;
2. digital image processing;
3. integration between images and CAD model to get the augmented model.

These aspects are discussed in this paper with particular emphasis on topics 2 and 3. The remainder of this paper is organized as follows: after a brief review of related works, the overall architecture of the method is presented (sec. 3). Section 4 describes problems and solutions related to image acquisition such as camera calibration and space resection. After a short description of the implementation (sec. 5), some results are presented (sec. 6) and further developments are discussed (sec. 8).

2. Related research

The problem of associating the digital representation of the 3D shape of an object with pictorial information has been investigated by several authors. It has been noted[12] that the critical points are how to acquire *pictorial information* and how to link them to the shape description.

The problem of augmenting CAD models with extra information has been discussed by Byne and Anderson[6] for vision programs development. In their approach CAD models that need to be augmented are converted into polygonal meshes stored according to the *Winged Edge Structure - WEP*. The elements of the WEP model are then associated with symbolic information describing material types and surface properties. This is used to support visual processing in their CAD-based computer vision system. The data structure adopted to associate WEP elements and material/surface properties is not described in detail.

Rocchini[12] et. al proposed a system for the acquisition of pictorial details of 3D free-form objects and their integration with the 3D shape through standard texture mapping. The shape of the object is assumed to be available as a polyhedral approximation. An algorithm is defined in order to assign, to each vertex of the mesh specific texture data given a set of digital images and a polyhedral mesh.

Hendrik et al. developed a system[11] which automatically registers and stitches textures obtained from digital photographic images onto the surfaces of a 3D model. Given an acquired digital image, using the recovered camera transformations, the image is stitched onto the surface, which is assumed to be available as a triangular mesh. Each vertex of the mesh is then projected into the image plane, using the recovered camera transformation, in order to compute the texture coordinate.

Interesting results have been obtained in the reconstruction of a single texture map from multiple views. Wang[13] et al. proposed an algorithm to determine spatially distributed weights to be used to get an optimal “average” texture from multiple reference images. The proposed approach takes into account the effects of anisotropy and non uniform spatial image sampling. Also Rocchini et al.[12] introduced, in their algorithm, a routine to generate a single texture triangle mesh from multiple images where particular attention is paid to prevent discontinuities in the representation of pictorial details.

Useful hints for the problem discussed in this paper can be retrieved from texture map construction problems that have been investigated by a fairly good number of authors in Architecture. Among them Debevec et. al[10] described an approach to render architectural scenes from a sparse set of photographs. In their approach texture maps are formed by projecting the original photos onto the recovered geometry, considering visibility.

It is possible to recognize, from literature, that, in general, the majority of the approaches have been focused on getting augmented images, mainly for rendering purposes, using acquired images. Most of the approaches have been applied on 3D polyhedral meshes.

3. Our approach in detail

From the above discussed references it is possible to point out that the majority of the proposed methods have been focused on rendering purposes, or, in other words, on obtaining augmented *images* given the shape of the model and one or more acquired pictures. Anyway in some cases pictorial data carry some physical meanings (strain, temperature, roughness) which are not only for rendering purposes. In such situations information acquired from digital images usually need to be processed and compared with topological and geometrical information, such as normal and curvature values, closeness from an edge/vertex and similiar. Such information typically pertain to CAD models. For such reasons the overall aim of this work is to define a framework to associate engineering information gathered from digital images with CAD models.

The approach proposed here, and described below, presents two peculiar distinctive features, if compared with other approaches proposed in literature:

- pictorial information are mapped to CAD models composed of generic shape surfaces, not limited to polyhedral models;
- information related to digital images are stored as data members of the surfaces which make up the CAD model.

The overall aim of the procedure is to associate each surface of the model with a function $D(u,v)$, defined on the parametric space of the surface, which returns a floating point value determined from digital images. In order to achieve this result, information contained in digital images are associated to each surface of a CAD model as a set of values computed along a grid of equally spaced points defined on the u,v curves of the surface at a given resolution. Data are stored in a matrix $M(n,m)$ of `SurfaceDataPoint` objects, which, as described in the following section, stores whether each point of the grid is active or not. This is needed since, in general, not all the points of a surface have a related pixel with an associated value, because some or all of them may be hidden given a certain pose of the digital image. A grid point may also be not active when, in trimmed surfaces, it pertains to those areas of the (u,v) domain, delimited by trimming curves, which are marked as invalid or invisible[17]. When a point is active, it is assumed that it is possible to define a function which maps the data of a pixel to a floating point value.

For each point of the grid a value is computed from a set of digital images. We assume that for all the digital images considered in this work there are enough information (such as the *pose* of the camera and the camera calibration data) that permit to associate each grid point to a specific pixel of the image. The value associated to a point different from the points of the grid is then computed using interpolating shape functions.

We assume that differences between the shape of the digital CAD model and the shape of the real component are negligible.

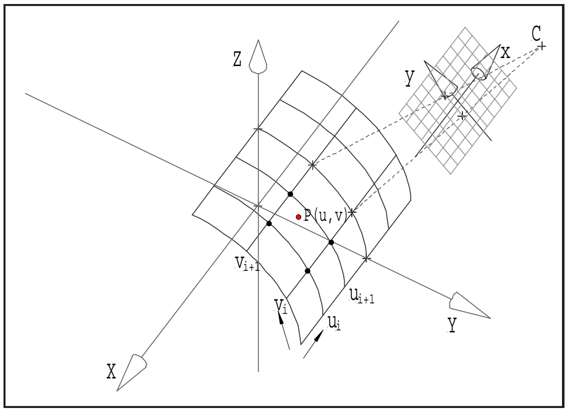

Figure 2: *Digital image and surface association.*

3.1. Logical model

The logical model is exemplified by the UML diagrams sketched in fig. 3 and 4. The `Face` class stands for a general surface patch that is an element of the boundary of a solid model. It can be interpreted as the part of the data structure which represents a solid model in B-rep format. In addition to other members (not reported in the figure), the `Face` class has a `Data` member that holds the information gathered from digital images.

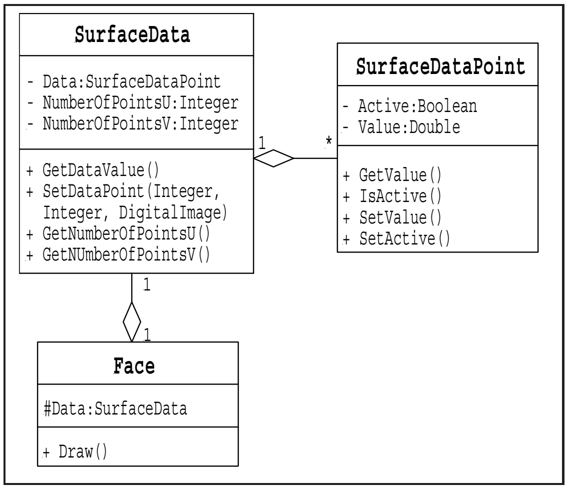

Figure 3: *UML diagram 1.*

The `SurfaceData` class (fig. 3) stores the information associated with each point of the grid and provides, for a generic couple of parameter values (u,v), the value of the function $D(u,v)$ through the `GetDataValue(double, double)` member. In general the domain of the $D(u,v)$ is a subset of the parametric space of the surface. This may occur either when a surface is represented as a trimmed patch, where the trimming curves define zones that have to be considered "blanked", or when a part or the whole surface has no associated pictorial information since, for instance, it is not visible from the available images. For such reasons the returned value $D(u,v)$ is an istance of the `SurfaceDataPoint` class which may be active or not. Given a couple of parameter values (u,v) and given $u_i, u_{i+1}, v_i, v_{i+1}$ the grid parameter values which bound (u,v) (fig. 2)if all of the four surface data points $M(i,j), M(i+1,j), M(i+1,j+1)$ and $M(i,j+1)$ are active, the function value $D(u,v)$ is computed as follows:

$$D(u,v) = \sum_{n=i}^{i+1} \sum_{m=j}^{j+1} w_{n,m} M(u_n, v_m); \tag{1}$$

where:

$$w_{n,m} = [1 + (-1)^{n-i+1}\overline{u}][1 + (-1)^{m-i+1}\overline{v}] \tag{2}$$

represent the weights and

$$\overline{u} = \frac{2u - u_i - u_{i+1}}{u_{i+1} - u_i}; \qquad \overline{v} = \frac{2v - v_i - v_{i+1}}{v_{i+1} - v_i} \tag{3}$$

are the values of the u,v parameters normalized to the $[u_i, u_{i+1}]$ and $[v_i, v_{i+1}]$ intervals. If not all the four bounding points are active, `GetDataValue(double ,double)` returns a non active `SurfaceDataPoint` instance. For each surface the number of points of the grid, or, alternatively, the minimum distance of consecutive points which determines the fineness of the grid can be set according to the required level of precision.

Digital images are wrapped around the `DigitalImage` class which, in addition to bitmap RGB or Color-Index data, stores specific photogrammetric information such as *intrinsic* and *extrinsic* camera calibration data[5] (fig. 4).

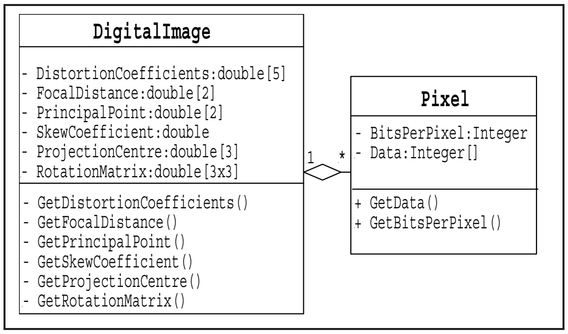

Figure 4: *UML diagram 2.*

4. Image acquisition and processing

This section describes problems and solutions related to image acquisition such as camera calibration and space resection.

4.1. Camera model

The basic theoretical model of photogrammetry is the *pinhole* camera model[5, 4, 3], where each point of the object is projected by a straight line through the projection centre onto the image plane (fig. 5).

Let (X,Y,Z) a co-ordinate system with its origin at O associated with the object (fig. 5) - *object co-ordinate system* -, and (x,y,z) a co-ordinate system associated with the camera - *camera co-ordinate system* with its origin at o. Let x and y the axes identifying the projection plane, and $C(X_0,Y_0,Z_0)$ the projection centre. The co-ordinates of a generic point $P_i(X_i,Y_i,Z_i)$ in the camera coordinate system are given by: 4:

$$\left\{ \begin{array}{c} x_i \\ y_i \\ z_i \end{array} \right\} = \left[\begin{array}{ccc} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{array} \right] \left\{ \begin{array}{c} X_i \\ Y_i \\ Z_i \end{array} \right\} + \left\{ \begin{array}{c} X_0 \\ Y_0 \\ Z_0 \end{array} \right\} \tag{4}$$

where

$r_{11} = cos(\phi)cos(\kappa); r_{12} = -cos(\phi)sin(\kappa); r_{13} - sin(\phi)$
$r_{21} = cos(\omega)sin(\kappa) - sin(\omega)sin(\phi)cos(\kappa);$
$r_{22} = cos(\omega)cos(\kappa) - sin(\omega)sin(\phi)sin(\kappa);$
$r_{23} = -sin(\omega)cos(\phi);$
$r_{31} = sin(\omega)sin(\kappa) + cos(\omega)sin(\phi)cos(\kappa);$
$r_{32} = sin(\omega)cos(\kappa) - cos(\omega)sin(\phi)sin(\kappa);$
$r_{33} = cos(\omega)cos(\phi).$

The Eulero angles ω, ϕ and κ define the relative orientation of the two co-ordinate systems in terms of rotations around the x, y and z axis respectively. The normalized[8] image projection coordinates of the point $p(x_i,y_i)$ are then given by:

$$\left\{ \begin{array}{c} \overline{x}_i \\ \overline{y}_i \end{array} \right\} = \left\{ \begin{array}{c} X_i/Z_i \\ Y_i/Z_i \end{array} \right\} \tag{5}$$

4.2. Camera calibration

Digital images acquired using real cameras usually differ from the theoretical geometry of central prospective projection. The co-ordinates of a projected point in the resulting image differ from their theoretical values due to a certain number of error that occur during acquisition. The most common source of error is due to lens distortion which introduces variations in angular magnification (*radial lens distortion*) and displacement of a point in the image caused by misalignment of the components of the lens (*tangential lens distortion*). In addition, due to misaligment among lens and CCD, the principal point usually is not coincident with the geometrical centre of the acquired image.

Table 1: *Calibration data of the camera used to acquire images.*

Camera calibration data.			
$K_i(i=1:5)$	$f_j(j=1:2)$	C_x,C_y	s
-0.19133	1381.62	634.54	0.00047
0.25197	1380.96	463.05	
-0.00500	-	-	-
0.00179	-	-	-
0.00000	-	-	-

The problem of determining sistematic errors which afflict acquired images is known as *camera calibration* process[4].Camera calibration plays a critical role in the framework described in this paper. In this work we adopted the camera calibration model introduced by Heikkilä and Silvén. [5, 8]. This model is based on the following parameters:

- lens distortion coefficients $K_i(i=1:5)$ used to compute radial and tangential distortion;
- co-ordinates of the principal point C_x,C_y in pixels;
- skew coefficient s which defines the real angle between the x and y axis;
- focal length $f_j(j=1:2)$ indicating the distance, expressed in units of horizontal and vertical pixels, between the origin of the camera coordinate system and the projection plane. These two values correspond[8] to a unique value in mm.

These parameters are also known as the *intrinsic* camera parameters.

The distorted co-ordinates $p(\overline{x}_{id},\overline{y}_{id})$ are then given by:

$$\left\{ \begin{array}{c} \overline{x}_{di} \\ \overline{y}_{di} \end{array} \right\} = a \left\{ \begin{array}{c} \overline{x}_i \\ \overline{y}_i \end{array} \right\} + \left\{ \begin{array}{c} b \\ c \end{array} \right\} \tag{6}$$

where a, b and c are scalar coefficients which depend on the distortion coefficients K_i through the following relations:

$$\left\{ \begin{array}{l} a = 1 + K_1 r^2 + K_2 r^4 + K_5 r^6; \\ b = 2K_3 \overline{x}_i \overline{y}_i + K_4(r^2 + 2\overline{x}_i^2); \\ c = K_3(r^2 + 2\overline{y}_i^2) + 2K_4 \overline{x}_i \overline{y}_i; \end{array} \right. \tag{7}$$

where $r^2 = \overline{x}_i^2 + \overline{y}_i^2$. The resulting coordinate values in pixel $p(n_i,m_i)$ can be computed as follows:

$$\left\{ \begin{array}{l} n_i = nint[f_1(\overline{x}_{id} + s\overline{y}_{id}) + C_x]; \\ m_i = nint[f_2 \overline{y}_{id} + C_y]; \end{array} \right. \tag{8}$$

where $nint(x)$ indicates the nearest integer function, sometimes denoted as $round(x)$.

4.3. Space resection

The problem of *space resection* is about determining the relative position of the centre of projection and of the projection plane with respect to the object which is being projected. These parameters are usually referred to as the *extrinsic* camera parameters.

Different numerical methods are available for determining the six unknowns of exterior orientation of digital images. All of them require a set of points (at least three) of which both the co-ordinates in object space and in the projection plane are known. The algorithm adopted in this work assumes that the correspondance of five points and their projections is known. The unknowns are then obtained using a least-square fitting method. Given $\mathbf{P} = \{P_1(X_1, Y_1, Z_1), \ldots, P_5(X_5, Y_5, Z_5)\}$ the array of the points in object space, $\mathbf{p} = \{P_1(x_1, y_1), \ldots, P_5((x_5, y_5)\}$ the array of the projected points, the six unknowns result from the minimization of the following function

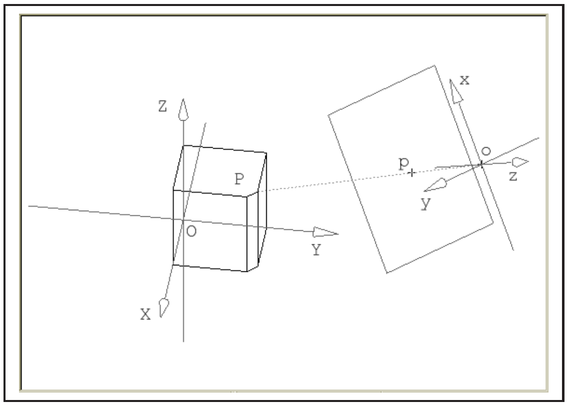

Figure 5: *The pinhole camera model.*

$$F = \sum_{i=1}^{5} (x_{it} - x_{im})^2 + \sum_{i=1}^{5} (y_{it} - y_{im})^2; \qquad (9)$$

where (x_{it}, y_{it}) represent the theorethical co-ordinate values of the projected points according to the (4) and (5), (x_{im}, y_{im}) the *measured* values of the projected points that can be computed iteratively using equations (6), (7) and (8).

5. Implementation

A prototype version of the framework has been implemented on WindowsNT/2000 platform. The core of the software has been coded in C++. The prototype is linked to a commercial CAD package (*Solid Edge*) through the *Geometry and Topology query interface V1.0*[9]. Such interface provides all the geometrical and topological data that are needed to implement the method. Camera calibration relies on an external Matlab toolbox called *Camera Calibration Toolbox for Matlab* developed by the California Institute of Technology Vision Group[8]. At present input images are submitted to the program in jpeg format. The routines devoted to digital images processing have been implemented on the top of *ImgSource v3.0* by *Smaller Animals Software Inc.*, a Win32 library which provides a set of features for reading, writing and processing of digital images in several formats.

6. Results and discussion

This section reports some tests and results about the implemented prototype. A simple mechanical component, shown in the following pictures has been used as a testbed. The geometry of the part is composed of 13 planar surfaces, 11 cylindrical surfaces and 2 spherical surfaces. Digital images have been acquired using a low cost digital camera (*Sony MVC-FD83*). The resolution of the images is 1216x912. A calibration procedure performed on the camera returned the parameters shown in table 4.2.

Figures 6 and 7 show the results of a preliminary test conducted to check the overall reliability of the framework. Letters have been applied on the faces of the physical component in order to verify their correct reconstruction in the augmented models. The chessboard which appears on acquired images is used for checking purposes, and its presence is not mandatory.

Figure 6: *A digital image of the test component with reference letters aplied on its faces.*

Fig. 8 shows an image where artificial fringes have been projected on the same component using a commercial LCD projector. Fig. 9 shows a snapshot of the prototype system where the CAD model of the component has been augmented and rendered with information contained in fig. 8. The parts of the model which are displayed in yellow corresponds to those zones which are not visible from the view point of picture 8.

7. Conclusions

This paper described the overall architecture and the first outcomes of the implementation of a framework which main objective is the construction of augmented CAD models using digital images. One of the advantages of the sysetm is that information retrieved from images are associated with generic shape surfaces. Such associaton can be at different levels of detail since the fineness of data associated to each

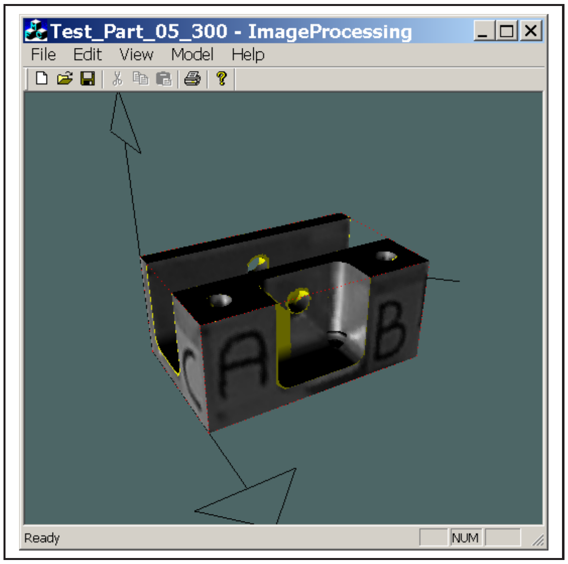

Figure 7: *A snapshot of the augmented CAD model obtained using information contained in the digital image shown in fig.6.*

Figure 8: *A digital image of a simple component with artificial fringes projected on it.*

surface can be set according to the required level of precision.

The discussed framework has been implemented in a prototype software and tested using a simple object and digital images acquired using a low cost camera. The first outcomes seem to be promising; further developments and investigations are in progress.

8. Further developments

The current protoype implementation of the proposed framework can manage a single image, since mutltiple images blending has not been introduced yet. The next forthcoming extension is the recovery of information from multiple

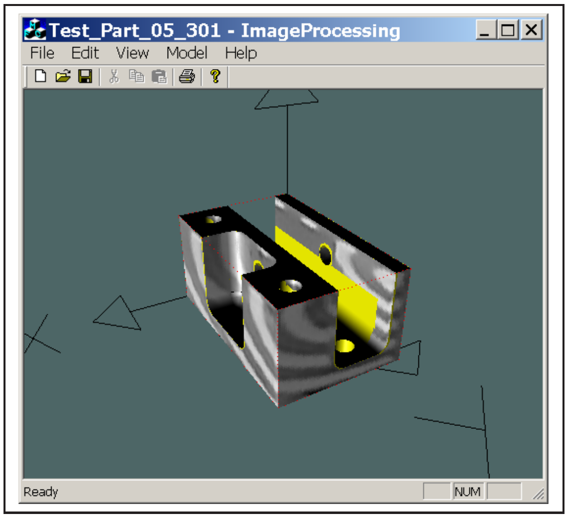

Figure 9: *A snapshot of the CAD model textured with information retreived from digital image in fig.8.*

images, using a suitable approach, starting from those proposed in literature[13, 12].

Another possible evolution, which is being studied, is about the development of a multi-layer framework where information related to a specific set of digital images pertaining to a peculiar situation (e.g. stress images related to a specific load condition) are stored in a specific layer.

A specific development we are currently working on is the simultaneous acquisition both of the geometry -by means of stereo vision techniques- and of the stress state of the component -through a thermoelastic stress measurement system- . The result should be an ”integral reverse engineering“ system capable of providing not only the shape of an object but even its stress state.

References

1. Hyosung Lee et al. Object’s Surface Roughness Measurement Using a High Resolution Digital Camera. http://www.fig.net/figtree/pub/proceedings/korea/full-papers/pdf/session4/lee-eo-kim-ahn.pdf, 1–9. 1
2. M.A. Younis On Line Surface Roughness Measurements Using Image Processing Towards an Adaptive Control. *Computers & Industrial Engineering*, **35**(1-2):49–52, 1998. 1
3. P.R. Wolf *Elements of Photogrammetry*. McGraw-Hill, 1983. 4
4. M.A.R. Cooper and S.Robson. Theory of close range photogrammetry. In *Close Range Photogrammetry and Machine Vision*. K.B. Atkinson Editor. Whittles Publishing, 9–51, 1996. 4

5. J. Heikkilä and O. Silvén. A Four-step Camera Calibration Procedure with Implicit Image Correction. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, 1106–1112, 1997. 3, 4
6. J.H.M. Byne and J.A.D.W Anderson A CAD-based computer vision system. *Image and Vision Computing*, **16**:533–539, 1998. 2
7. Chu-Song Chen et al. Integrating Virtual Objects into Real Images for Augmented Reality. *Proc. of the ACM symposium of Virtual reality software and technology*, 1–8, 1998. 1
8. J.Y. Bouguet. Camera Calibration Toolbox for Matlab. *http://www.vision.caltech.edu/bouguetj/calib_doc/*. 4, 5
9. The Design and Modeling Application Council (DMAC) Geometry and Topology Query Interfaces V1.0. Overview document. *http://www.dmac.org/tech/GandT/G&Tover01.htm*, 1997. 5
10. P.E. Debevec et al. Image-based modeling and rendering of architecture with interactive photogrammetry and view-dependent texture mapping. *Proc. of the IEEE int. symposium on Circuits and Systems*, 514–517, 1998 2
11. P.A. Hendrik, W. Heidrich and H.P. Seidel Automated Texture Registration and Stitching for Real World Models. *http://citeseer.nj.nec.com/lensch00automated.html*, 317–327, 2000 2
12. C. Rocchini, P. Cignoni and C. Montani Multiple Textures Stitching and Blending on 3D Objects. *Proc. of the Eurographics Rendering Workshop*, 1999 2, 6
13. L. Wang et al. Optimal Texture Map Reconstruction from Multiple Views. *Proc. of IEEE Comp. Society Conference on Computer Vision and Pattern Recognition*,347–354 (I), 2001 2, 6
14. A. Ajovalasit, S. Barone and G. Petrucci A Review of Automated Methods for the Collection and Analysis of Photoelastic Data. *The Journal of Strain Analysis for Engineering*, **33**(2):75–91, 1998. 1
15. B.R. Boyce Steps to Modern Thermoelasticity Stress Analysis *Proc. of the ATEM Conference*, 1999 1
16. G.C. Holst *Common Sense Approach to Thermal Imaging*. SPIE Optical Engineering Press, 2000. 1
17. G. Farin *Curves and Surfaces for CAGD. A Practical Guide*. Academic Press, 1993. 2
18. S.T. Lin and R.E. Rowlands Hybrid Stress Analysis. *Optics and Laser in Engineering*, **32**(3):257–298, 1999. 1