Image-Based Geometrically-Correct Photorealistic Scene/Object Modeling (IBPhM): A Review

Zhengyou Zhang[1,2]

[1] INRIA, 2004 route des Lucioles, BP 93, F-06902 Sophia-Antipolis Cedex, France
[2] ATR HIP, 2-2 Hikaridai, Seika-cho Soraku-gun, Kyoto 619-02 Japan
e-mail: zzhang@sophia.inria.fr, zzhang@hip.atr.co.jp

Abstract. There are emerging interests from both computer vision and computer graphics communities in obtaining photorealistic modeling of a scene or an object from real images. This paper presents a tentative review of the computer vision techniques used in such modeling which guarantee the generated views to be geometrically correct. The topics covered include mosaicking for building environment maps, CAD-like modeling for building 3D geometric models together with texture maps extracted from real images, image-based rendering for synthesizing new views from uncalibrated images, and techniques for modeling the appearance variation of a scene or an object under different illumination conditions. Major issues and difficulties are addressed.

Keywords: photorealistic modeling, image-based rendering, multiple-view geometry, photometric models, CAD, camera calibration, 3D reconstruction, uncalibrated images, domain knowledge, illumination variation.

1   Introduction

Considerable effort in computer graphics has been devoted, on the one hand, to the development of complex computer aided design (CAD) systems which aim at modeling the geometric and material attributes of the objects in the environment, and on the other hand, to the development of systems which try to reproduce the light propagation under physical laws in order to generate photorealistic renderings. Unfortunately, besides it is a labor-intensive process, this traditional approach has difficulty in creating realistic photographs because the geometry of objects found in the real world is very complex and the subtle light effect is difficult, if impossible, to model.

Computer vision, although it may promise much more, can be considered as an inverse process of computer graphics, namely the process of recovering 3D structure from 2D images. Furthermore, real images directly capture the material properties of objects under real world illumination. Therefore, the combination of computer vision and computer graphics allows us to create, directly from real images, realistic photographs of the environment from viewpoints different from the original ones. Indeed, there are emerging interests in applying the computer vision techniques to the interdisciplinary field of virtual reality, such as human-computer interface, 3D scene/object reconstruction and image-based rendering.

Both vision and graphics technologies have difficulties in producing complex geometric models in great details, but by appropriately using the visual details contained

in the original images, we can achieve photorealistic modeling without replying on a complex model. Therefore, compared with the traditional computer graphics modeling, image-based photorealistic modeling (IBPhM) has a number of advantages including

- a weaker dependency of scene complexity because many details need not to be explicitly modeled,
- a simpler geometric model because of the previous reason, which also implies the ease of model acquisition,
- no need of physical simulation because the realism is in the original images,
- less computational requirement because of all previous reasons.

There are two approaches to IBPhM:

CAD-like modeling: A scene is represented by a 3D model together with texture maps extracted from real images.

Image-based rendering (IBR): A scene is represented as a collection of images. New images are generated from the original images.

It seems that several researchers prefer IBR to CAD-like modeling, but I argue that it depends on applications and on the available input information. Among the techniques reviewed below, those *based on camera geometry* are usually more appropriate to use 3D models rather than a collection of images. This is for three reasons:

- CAD-like modeling is less memory demanding than IBR, because the data redundancy in the original images are used in building 3D models and are later discarded.
- The conventional rendering pipeline can be used once a CAD-like model is available, while it is not designed for IBR. One can of course expect that special IBR-dedicated hardware will be developed in the near future.
- Both approaches require the establishment of feature correspondences between images. IBR is not easier in this task.

However, IBR does have an advantage over CAD-like modeling at the current stage of development. That is, uncalibrated images can be used in IBR to generate new views. Use of uncalibrated images implies that only a 3D *projective* model can be built. Lack of metric measures in such models makes it difficult to use the conventional rendering pipeline.

The well-known Apple QuickTimeVR system [1] creates a series of environment maps at key locations in the scene. An environment map records the light arriving from all directions at a point. The user is then able to look around a scene from these *fixed* locations. The fixed location constraint can be relaxed in four ways. The first tries to model the apparent motion of pixels (i.e., optical flow) from one camera location (viewpoint) to another, which allows a smooth view interpolation [2, 3]. The second goes further by trying to capture the complete flow of light in a region of the environment [4–6]. In free space, the light field is a 4D function, and an image is a 2D slice of the 4D light field. The success of this technique depends on having a high sample rate, which implies to acquire and save (after proper compression) a large number of images. The third approach relies on the depth value (or disparity) at each pixel [7, 8]. The depth values are usually obtained with user interaction. Given the depth value, a point can be reprojected from

vantage points. The fourth approach is called image transfer, which uses point correspondences between images to synthesize a new one without explicitly reconstructing the structure [9, 10]. The last two are however mathematically equivalent. Except the second approach, all the other three require the establishment of correspondence of pixels or feature points between images, which can be a very difficult task and is usually done manually or semi-manually. The second approach has, however, its own limit. It requires the acquisition and storage of a large number of images from known camera viewpoints. It is mainly studied in the computer graphics community, and will not be reviewed in this paper. In Sect. 2, mosaicking techniques to build environment maps are reviewed. It is a 3D representation of a 3D scene. In Sect. 3, we describe the multiple-camera geometry, and how a CAD-like model is built. In Sect. 4, techniques to synthesize new views from original images are reviewed. In reviewing these techniques, pixel or feature point correspondences are assumed to be given. Appendix A will review the commonly used techniques for solving the correspondence problem.

When creating a *dynamic* virtual environment, illumination variation must be considered because it has an important effect on the person navigating in it. This has many applications including product advertisement on the Web where a user wants to examine the appearance of the product under various illumination conditions from different viewpoints. Section 5 reviews techniques which model illumination variations from real images.

2 Mosaicking

When the scene is a planar surface or when the images are taken from the same point of view (i.e., the camera undergoes a pure rotation around the optical center), images are related by a linear projective transformation called a collineation or a homography. More precisely, if $\mathbf{m} = [u, v]^T$ and $\mathbf{m}' = [u', v']^T$ are respectively the projected point of the same space point in the first and second image, then they are related by

$$\rho \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad \text{or more compactly, } \rho \widetilde{\mathbf{m}}' = \mathbf{H} \widetilde{\mathbf{m}} \qquad (1)$$

where ρ is an arbitrary scale factor, and $\mathbf{H}$ is a non-singular matrix defined up to a scale factor. Here, we have used the following notation: for any vector $\mathbf{x} = [x_1, x_2, \dots]^T$, $\widetilde{\mathbf{x}} = [x_1, x_2, \dots, 1]^T$ (i.e., 1 is added as the last element).

Because of relation (1), images can be stitched or mosaicked into a larger and/or higher resolution image. A full-view panorama is possible if images cover the whole viewing space. This is important because the full-view panorama allows us to create an environment map, which can be used to quickly generate new views within the environment. A nice survey of the mosaicking techniques is already available [11]. Several automatic techniques have recently been developed [12–14].

If one of the conditions mentioned at the beginning of this section is not satisfied, there will be misregistration caused by motion parallax. Shum and Szeliski [15] have developed a technique called *deghosting*, which divides each image into small patches, estimates patch alignment, and finally warps each image locally.

Another possibility to obtain a panorama is to use special hardware, such as a camera system with conic mirror [16], hyperboloidal mirror [17], or paraboloidal mirror [18]. However, such a panoramic image is limited by the resolution.

3 CAD-like modeling

Building a CAD-like model from an image sequence consists of the following steps:

Calibration: Recover the external (position and orientation) and internal parameters of the camera for each image in the sequence.

Shape modeling: Build, usually with manual interaction, a 3D geometric model of the scene/object, which is usually a polyhedral approximation.

Texture modeling: Build a texture map for each face of the polyhedral model from original images.

Different calibration techniques will be reviewed shortly. Automatic shape modeling is difficult, and at the current stage of development, one is usually contented with manual initialization followed by automatic refinement by computer vision techniques. Techniques for texture modeling follow much the same way as for image mosaicking, as described in Sect. 2. A face of the polyhedral model is planar, which is seen in several images. These image patches are related by a homography. Possibly different parts of each face are seen from different viewpoints. Mosaicking techniques can therefore be used to build a complete representation of the appearance of the face. Sometimes, image affine transformation, as an approximation to homography, can be used.

Regarding the camera calibration problem, there are essentially three paradigms:

Photogrammetry. The calibration of each camera is performed by observing a calibration object whose geometry in 3-D space is known with a very good precision, and can be done very efficiently [19]. In the stereo setup, the calibration object should be seen simultaneously from all cameras, which is quite problematic when we need to capture a complete view of a scene or an object using many cameras. The solution to this is to follow the structure-from-motion path, either using only one camera [20–22] or using a stereo rig [23–25]. The idea is to recover the rigid displacements of the cameras from visual information by establishing correspondences of features (points, line segments, lines, curves, etc.).

Self-calibration. Techniques in this category do not use any calibration object. Just by moving a camera in a static scene, the rigidity provides in general two constraints [26, 27] on the cameras' internal parameters from one camera displacement by using image information alone (image point or line correspondences). Therefore, if images are taken by the same camera with constant internal parameters, point correspondences between three images are sufficient to recover both the internal and external parameters which allow us to reconstruct 3-D structure up to a similarity [28, 29]. Self-calibration can also be done for uncalibrated stereo rig, where the internal parameters and the relative orientation of the cameras and the motion of the stereo rig are all unknown [30–32]. More knowledge about the camera internal parameters and camera motion will simplify the computation, and more precise and robust results can be obtained.

Domain knowledge. If there is no constraint at all on either the internal parameters or the external parameters of different cameras, we can only achieve a projective reconstruction of the scene [33–35]. This is not enough for many applications which require 3D

Euclidean modeling. However, we usually have domain knowledge of the imaged scene. Such knowledge includes Euclidean location of a point, parallelism, distances between two points, angles between two lines, ratios of distances, etc. This allows us to compute the projective transformation which brings the projective structure back to a Euclidean coordinate system [36–38].

4   Image-based rendering

As mentioned in the introduction, I do not consider here techniques based on interpolation from dense image sequences. Furthermore, I do not consider the case where cameras' internal parameters are known, because I believe that CAD-like modeling is usually more appropriate than IBR. We consider here only the case of uncalibrated images without using explicit 3D models. The main steps in image-based rendering with uncalibrated images are listed below:

1. Establish point correspondences between images
2. Estimate the epipolar geometry between images
3. Build a representation of the scene using matched points
4. Specify the desired position of the new image
5. Transfer the scene representation into the new image
6. Map textures (colors) from the original images to the new images

The most crucial and difficult part is Step 1 and Step 2. Step 1 will be reviewed in Appendix A. The second step consists in estimating the fundamental matrix between two images [39], the trifocal tensor between three images [40], or the $\mathbf{P}$-matrices (camera projection matrices) between N images [41]. A complete review of techniques for estimating the fundamental matrix and projective reconstruction is done by Zhang [42]. A good technique for estimating the trifocal tensor is developed in [43]. The PhD thesis of Laveau [44] addresses the issues of estimating $\mathbf{P}$-matrices between N images.

The representation of the scene depends on the number of matched points. If full pixel correspondences are available, the scene is probably better to be represented by the original images, and the later *texture mapping* step can use linear or bilinear interpolation to find the color of a point in the new image. If only a sparse set of point matches are available, we can divide each image into a set of triangular patches using points as vertices. Texture mapping can then be realized through affine transformation or even better through plane projective transformation (homography).

Given a set of point/pixel correspondences and the epipolar geometry, we have *implicitly* a 3D projective description of the scene with respect to a projective coordinate system. Step 4 is then to specify the desired position of the new image (i.e., its $\mathbf{P}$-matrix with respect to the projective coordinate system). There are 15 degrees of freedom, and one can imagine how difficult the task is. This really limits the usefulness of uncalibrated images. There are two possibilities to get around this difficulty: use a reference image if it is available for the desired position; use domain knowledge to obtain a quasi-Euclidean structure [38].

Image transfer is simply a shortcut of 3D reconstruction followed by projection onto the new image according to the $\mathbf{P}$-matrix. There are however a number of problems in practice, the occlusion problem in particular. There are probably several points which are transfered to the same location of the new image. One should decide which point is

visible. If the structure is Euclidean, z-buffer technique can be used. This is however not useful for uncalibrated images, because a 3D point is projective and defined up to a scale factor. The solution is to use *oriented* projective geometry [44].

The transfered point in the new image is in general a nonlinear function of the points in the original images. It is, however, linear if affine camera model (including orthographic, weak-perspective and paraperspective projections) is used. Another case where the transfer is linear is the following: if the original views have parallel optical axis which is orthogonal to the line joining the optical centers, then a new view on the same line with parallel optical axis is a linear combination of the original views. The view morphing technique described in [10] is based on this observation. It first rectifies the original images to have aligned scan lines, then produces an intermediate rectified image through linear interpolation, and finally postwarps it to obtain the desired image. With this technique, one is able to generate an image sequence corresponding to a camera moving along a line joining the optical centers of the two original images. However, if postwarping is not chosen appropriately, there will be a significant projective distortion in the generated views.

5   Modeling the illumination variation

All the techniques described up to now assume a static scene with *fixed illumination*. Illumination variation must be considered when creating a dynamic virtual environment because it has an important effect on the person navigating in it. Traditionally, techniques based on some reflectance models such as ray tracing are used. The most widely used model is the Lambertian where surface radiance depends only on the irradiance of the surface and not on the observer's viewpoint. One usually observes a significant deviation of this model from reality. Recently, there is an important effort on the estimation of surface reflectance of natural materials using more general functions including the BRDF (Bidirectional Reflection Distribution Function) [45–47]. The state-of-the-art, however, does not yet allow us to estimate the BRDF reliably enough to be used in practice. Within computer vision, researchers follow a different approach to model the illumination variation for object recognition. The idea is to capture illumination and object reflectance directly from real images. For Lambertian surfaces of arbitrary texture without self-occlusion and self-shadows, three images taken with non-collinear light sources are enough to completely determine the structure of the illumination manifold [48, 49]. In other cases, more (but usually only a few) images are required [50–52]. The Karhunen-Loéve transform or Singular-Value Decomposition (SVD) [53] is typically used to compute the basis images which capture the essential information relevant to the reflectance and illumination variations. Finally, we observe that, except Shashua's work [48], all other works do not consider 3D geometric information. Shashua represents 3D geometry of an object as a linear combination of two images because an approximate (affine) camera model is used, and uses only three images taken under different illumination conditions to represent the photometric property. However, three images are usually not sufficient to represent the photometric property of a complex object. Recently, Zhang [54] developed a system which uses full perspective camera model and as many images under different illumination conditions as possible. The scene/object is represented by a 3D geometric model together with a set of basis images which capture the essential photometric property of the scene/object under different illumination condition. He is then

able to generate photorealistic views simulating both changes in point of view and in illumination condition.

Instead of recovering shape from 2D images, Sato et al. [55] use a light-stripe range finder and a color CCD camera to acquire a sequence of range images and a sequence of color images of an object. Range images are merged to reconstruct the surface shape of the object. The reconstructed shape and the color images are then used to determine the surface reflectance properties of the object. Finally, they are able to synthesize images with realistic shading effects under arbitrary illumination conditions.

6 Concluding remarks

There are emerging interests from both computer vision and computer graphics communities in obtaining photorealistic modeling of a scene or an object from real images. This paper has presented a tentative review of the computer vision techniques used in such modeling which guarantee the generated views to be geometrically correct. The topics covered include mosaicking for building environment maps, CAD-like modeling for building 3D geometric models together with texture maps extracted from real images, image-based rendering for synthesizing new views from uncalibrated images, and techniques for modeling the appearance variation of a scene or an object under different illumination conditions.

There are a number of open issues:

– What is the difference in rendering mechanism between CAD-like modeling and IBR? Which one is more efficient? There is probably no universal answer, and it is likely task-dependent (processing time, hardware accelerators, memory requirement, field of view of the scene to be modeled, etc).
– How to obtain a coherent representation of a large scale scene from a large number of images? Geometric reasoning should play a fundamental role in this regard.
– The intensity/color is almost always different from one image to another because of dynamic change in camera gain, even if images are perfectly aligned. How to fuse different observations in order to avoid intensity/color discontinuity in the final texture maps? This is known as *image blending*.
– How to compute the reflectance properties of a complex scene from real images?
– Matching, the eternal problem in computer vision, although alleviated in IBPhM through user interaction, still needs considerable work. Before having a perfect image matching, we need to answer the following question: How to tolerate false matches in generating photorealistic views?

And there are many more.

A Image matching techniques

When the epipolar geometry is unknown, there are mainly two categories of techniques. The first is optical-flow based. The representative work is the pyramidal hierarchical motion estimation framework proposed by Bergen et al. [56]. The second is feature-based. The representative work is the robust image matching technique through the recovery of the unknown epipolar geometry proposed by Zhang et al. [57].

When the epipolar geometry is known or recovered, there are many stereo matching techniques available, such as correlation, relaxation and dynamic programming. See, e.g., textbook [19] for a general discussion.

References

1. S. Chen, "QuickTime VR - an image-based approach to virtual environment navigation," in *Computer Graphics, Annual Conference Series*, pp. 29–38, ACM SIGGRAPH, 1995.
2. S. Chen and L. Williams, "View interpolation for image synthesis," in *Computer Graphics, Annual Conference Series*, pp. 279–288, ACM SIGGRAPH, 1993.
3. T. Werner, R. Hersch, and V. Hlavac, "Rendering real-world objects using view intepolation," in *Proc. Fifth International Conference on Computer Vision*, (Cambridge, Massachusetts), pp. 957–962, June 1995.
4. A. Katayama, K. Tanaka, T. Oshino, and H. Tamura, "A viewpoint independent stereoscopic display using interpolation of multi-viewpoint images," in *Stereoscopic displays and virtual reality systems II* (S. Fisher, J. Merritt, and B. Bolas, eds.), vol. 2409 of *Proc. SPIE*, pp. 11–20, 1995.
5. S. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen, "The Lumigraph," in *Computer Graphics, Annual Conference Series*, pp. 43–54, ACM SIGGRAPH, 1996.
6. M. Levoy and P. Hanraham, "Light field rendering," in *Computer Graphics, Annual Conference Series*, pp. 31–42, ACM SIGGRAPH, 1996.
7. L. McMillan and G. Bishop, "Plenoptic modeling: An image-based rendering system," in *Computer Graphics, Annual Conference Series*, pp. 39–46, ACM SIGGRAPH, 1995.
8. P. Debevec, C. Taylor, and J. Malik, "Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach," in *Computer Graphics, Annual Conference Series*, pp. 11–20, ACM SIGGRAPH, 1996.
9. O. Faugeras and S. Laveau, "Representing three-dimensional data as a collection of images and fundamental matrices for image synthesis," in *Proc. International Conference on Pattern Recognition*, (Jerusalem, Israel), pp. 689–691, Computer Society Press, Oct. 1994.
10. S. Seitz and C. Dyer, "View morphing," in *Computer Graphics, Annual Conference Series*, pp. 21–30, ACM SIGGRAPH, 1996.
11. S. Kang, "A survey of image-based rendering techniques," Tech. Rep. CRL 97/4, Digital Equipment Corporation, Cambridge Research Lab, Aug. 1997.
12. R. Szeliski and H.-Y. Shum, "Creating full view panoramic image mosaics and environment maps," in *Computer Graphics, Annual Conference Series*, pp. 251–258, ACM SIGGRAPH, 1997.
13. I. Zoghlami, O. Faugeras, and R. Deriche, "Using geometric corners to build a 2d mosaic from a set of images," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, (San Juan, Puerto Rico), pp. 420–425, IEEE Computer Society, June 1997.
14. S. Peleg and J. Herman, "Panoramic mosaics by manifold projection," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, (San Juan, Puerto Rico), pp. 338–343, IEEE Computer Society, June 1997.
15. H.-Y. Shum and R. Szeliski, "Construction and refinement of panoramic mosaics with global and local alignment," in *Proc. 6th International Conference on Computer Vision*, (Bombay, India), IEEE Computer Society Press, Jan. 1998.
16. Y. Yagi and S. Kawato, "Panorama scene analysis with conic projection," in *Proc. IEEE International Workshop on Intelligent Robots and Systems*, pp. 181–187, July 1990.
17. K. Yamazawa, Y. Yagi, and S. Kawato, "Omnidirectional imaging with hyperboloidal projection," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1029–1034, July 1993.
18. S. Nayar, "Catadioptric omnidirectional camera," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (G. Medioni, R. Nevatia, D. Huttenlocher, and J. Ponce, eds.), (San Juan, Puerto Rico), pp. 482–488, IEEE Computer Society, June 1997.

19. O. Faugeras, *Three-Dimensional Computer Vision: a Geometric Viewpoint*. MIT Press, 1993.
20. J. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images — a review," *Proc. IEEE*, vol. 76, pp. 917–935, Aug. 1988.
21. T. Huang and A. Netravali, "Motion and structure from feature correspondences: A review," *Proc. IEEE*, vol. 82, pp. 252–268, Feb. 1994.
22. Z. Zhang, "Motion and structure from two perspective views: From essential parameters to euclidean motion via fundamental matrix," *Journal of the Optical Society of America A*, vol. 14, no. 11, 1997. In Press.
23. Z. Zhang and O. Faugeras, *3D Dynamic Scene Analysis: A Stereo Based Approach*. Springer-Verlag, Berlin, New York, 1992.
24. Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *The International Journal of Computer Vision*, vol. 13, no. 2, pp. 119–152, 1994. also Research Report No.1658, INRIA Sophia-Antipolis, 1992.
25. Z. Zhang, "Motion of a stereo rig: Strong weak and self calibration," in *Recent Developments in Computer Vision* (S. Li, D. Mital, E. Teoh, and H. Wang, eds.), vol. 1035 of *Lecture Notes in Computer Science*, pp. 241–254, Springer-Verlag, Berlin, 1996.
26. S. J. Maybank and O. D. Faugeras, "A theory of self-calibration of a moving camera," *The International Journal of Computer Vision*, vol. 8, pp. 123–152, Aug. 1992.
27. Q.-T. Luong, *Matrice Fondamentale et Calibration Visuelle sur l'Environnement-Vers une plus grande autonomie des systèmes robotiques*. PhD thesis, Université de Paris-Sud, Centre d'Orsay, Dec. 1992.
28. Q.-T. Luong and O. Faugeras, "Self-calibration of a moving camera from point correspondences and fundamental matrices," *The International Journal of Computer Vision*, vol. 22, no. 3, pp. 261–289, 1997.
29. R. Hartley, "An algorithm for self calibration from several views," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, (Seattle, WA), pp.908–912, 1994.
30. Z. Zhang, Q.-T. Luong, and O. Faugeras, "Motion of an uncalibrated stereo rig: self-calibration and metric reconstruction," *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 103–113, Feb. 1996. Short version appeared in the *Proc. International Conference on Pattern Recognition*, volume I, pages 695–697, Jerusalem, Israel, Oct. 1994.
31. A. Zisserman, P. A. Beardsley, and I. D. Reid, "Metric calibration of a stereo rig," in *Proc. Workshop on Visual Scene Representation*, (Boston, MA), June 1995.
32. F. Devernay and O. Faugeras, "From projective to euclidean reconstruction," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, (San Francisco, CA), pp. 264–269, IEEE, June 1996.
33. O. Faugeras, "What can be seen in three dimensions with an uncalibrated stereo rig," in *Proc. 2nd European Conference on Computer Vision* (G. Sandini, ed.), vol. 588 of *Lecture Notes in Computer Science*, (Santa Margherita Ligure, Italy), pp. 563–578, Springer-Verlag, May 1992.
34. R. Hartley, "Projective reconstruction and invariants from multiple images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 10, pp. 1036–1040, 1994.
35. G. Xu and Z. Zhang, *Epipolar Geometry in Stereo, Motion and Object Recognition*. Kluwer Academic Publishers, 1996.
36. R. Hartley, R. Gupta, and T. Chang, "Stereo from uncalibrated cameras," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, (Urbana Champaign, IL), pp. 761–764, IEEE, June 1992.
37. R. Mohr, B. Boufama, and P. Brand, "Understanding positioning from multiple images," *Artificial Intelligence*, vol. 78, pp. 213–238, 1995.
38. Z. Zhang, K. Isono, and S. Akamatsu, "Euclidean structure from uncalibrated images using fuzzy domain knowledge: Application to facial images synthesis," in *Proc. 6th International Conference on Computer Vision*, (Bombay, India), IEEE Computer Society Press, Jan. 1998.

39. Q.-T. Luong and O. D. Faugeras, "The fundamental matrix: Theory, algorithms and stability analysis," *The International Journal of Computer Vision*, vol. 17, pp. 43–76, Jan. 1996.

40. A. Shashua, "Algebraic functions for recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 779–789, 1995.

41. Q.-T. Luong and T. Viéville, "Canonical representations for the geometries of multiple projective views," *Computer Vision and Image Understanding*, vol. 64, pp. 193–229, Sept. 1996.

42. Z. Zhang, "Determining the epipolar geometry and its uncertainty: A review," *The International Journal of Computer Vision*, 1997. In Press. Updated version of INRIA Research Report No.2927, 1996.

43. P. Torr and A. Zisserman, "Robust parameterization and computation of the trifocal tensor," *Image and Vision Computing*, vol. 15, pp. 591–605, 1997.

44. S. Laveau, *Géométrie d'un système de N caméras. Théorie, estimation et applications.* PhD thesis, École Polytechnique, May 1996.

45. M. Oren and S. Nayar, "Generalization of the lambertian model and implications for machine vision," *The International Journal of Computer Vision*, vol. 14, pp. 227–251, Apr. 1995.

46. L. Wolff, "Generalizing Lambert's law for smooth surfaces," in *Proc. 4th European Conference on Computer Vision* (B. Buxton, ed.), vol. II, (Cambridge, UK), pp. 40–53, Apr. 1996.

47. J. Koenderink, A. van Doorn, and M. Stavridi, "Bidirectional reflection distribution function expressed in terms of surface scattering modes," Research Report UU-PAhp-046, Utrecht State University, 1995.

48. A. Shashua, *Geometry and Photometry in 3D Visual Recognition.* PhD thesis, Massachusetts Institute of Technology, 1992.

49. S. Nayar and H. Murase, "Dimensionality of illumination in appearance matching," in *Proc. IEEE International Conference on Robotics and Automation*, (Minneapolis, Minnesota), pp. 1326–1332, Apr. 1996.

50. R. Epstein, P. Hallinan, and A. Yuille, "5±2 eigenimages suffice: An empirical investigation of low-dimensional lighting models," in *Proc. IEEE Workshop on Physics Based Modeling in Computer Vision*, (Cambridge, Massachusett), pp. 108–116, June 1995.

51. P. Belhumeur and D. Kriegman, "What is the set of images of an object under all possible lighting conditions?," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 270–277, June 1996.

52. G. Hager and P. Belhumeur, "Real-time tracking of image regions with changes in geometry and illumination," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 1996.

53. G. Golub and C. van Loan, *Matrix Computations.* The John Hopkins University Press, 1989.

54. Z. Zhang, "Modeling geometric structure and illumination variation of a scene from real images," in *Proc. 6th International Conference on Computer Vision*, (Bombay, India), IEEE Computer Society Press, Jan. 1998.

55. Y. Sato, M. Wheeler, and K. Ikeuchi, "Object shape and reflectance modeling from observation," in *Computer Graphics, Annual Conference Series*, pp. 379–387, ACM SIGGRAPH, 1997.

56. J. Bergen, P. Anandan, K. Hanna, and R. Hingorani, "Hierarchical model-based motion estimation," in *Proc. 2nd European Conference on Computer Vision* (G. Sandini, ed.), vol. 588 of *Lecture Notes in Computer Science*, (Santa Margherita Ligure, Italy), pp. 237–252, Springer-Verlag, May 1992.

57. Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry," *Artificial Intelligence Journal*, vol. 78, pp. 87–119, Oct. 1995.