

Organizer/Lecturer  
Marc Pollefeys  
*Katholieke Universiteit Leuven*

# 54

Obtaining 3D Models  
With a Hand-Held Camera

## Course Notes

The 29th International Conference on  
Computer Graphics and Interactive Techniques

Henry B. Gonzalez Convention Center  
San Antonio, Texas USA

Conference: 21–26 July 2002  
Exhibition: 23–25 July 2002

SAN ANTONIO  
**SIGGRAPH**  
2002

# Obtaining 3D Models With a Hand-Held Camera

Marc Pollefeys  
Center for Processing of Speech and Images  
University of Leuven, Belgium

Lecture Notes SIGGRAPH Course  
Wednesday 24 July, 2002

# Acknowledgments

At this point I would like to thank the people who contributed directly or indirectly to the text for this tutorial. First of all I would like to express my gratitude towards professor Luc Van Gool, head of the Industrial Image Processing group (VISICS/PSI/ESAT) of the K.U.Leuven. I am very grateful to my colleagues Maarten Vergauwen and Kurt Cornelis who are not only doing great research work (of which you'll find many examples in this text), but also proofread the whole text. Next I would also like to thank Frank Verbiest, Jan Tops, Joris Schouteden, Reinhard Koch, Tinne Tuytelaars and Benno Heigl who have contributed to the work presented in this text. I would also like to acknowledge the Fund for Scientific Research - Flanders (Belgium) for granting me a Postdoctoral Fellowship. The financial support of the FWO project G.0223.01 and the European IST projects InViews, Attest, Vibes and Murale are also gratefully acknowledged.

# Notations

To enhance the readability the notations used throughout the text are summarized here.

For matrices bold face fonts are used (i.e.  $\mathbf{A}$ ). 4-vectors are represented by  $\mathbf{A}$  and 3-vectors by  $\mathbf{a}$ . Scalar values will be represented as  $a$ .

Unless stated differently the indices  $i$ ,  $j$  and  $k$  are used for views, while  $l$  and  $m$  are used for indexing points, lines or planes. The notation  $\mathbf{A}_{ij}$  indicates the entity  $\mathbf{A}$  which relates view  $i$  to view  $j$  (or going from view  $i$  to view  $j$ ). The indices  $i$ ,  $j$  and  $k$  will also be used to indicate the entries of vectors, matrices and tensors. The subscripts  $P$ ,  $A$ ,  $M$  and  $E$  will refer to projective, affine, metric and Euclidean entities respectively

$\mathbf{P}$	camera projection matrix ( $3 \times 4$ matrix)
$\mathbf{M}$	world point (4-vector)
$\Pi$	world plane (4-vector)
$\mathbf{m}$	image point (3-vector)
$\mathbf{l}$	image line (3-vector)
$\mathbf{H}_{ij}^{\Pi}$	homography for plane $\Pi$ from view $i$ to view $j$ ( $3 \times 3$ matrix)
$\mathbf{H}_{\Pi i}$	homography from plane $\Pi$ to image $i$ ( $3 \times 3$ matrix)
$\mathbf{F}$	fundamental matrix ( $3 \times 3$ rank 2 matrix)
$\mathbf{e}_{ij}$	epipole (projection of projection center of viewpoint $i$ into image $j$ )
$\mathbf{T}$	trifocal tensor ( $3 \times 3 \times 3$ tensor)
$\mathbf{K}$	calibration matrix ( $3 \times 3$ upper triangular matrix)
$\mathbf{R}$	rotation matrix
$\Pi_{\infty}$	plane at infinity (canonical representation: $W = 0$ )
$\Omega$	absolute conic (canonical representation: $X^2 + Y^2 + Z^2 = 0$ and $W = 0$ )
$\Omega^*$	absolute dual quadric ( $4 \times 4$ rank 3 matrix)
$\omega_{\infty}$	absolute conic embedded in the plane at infinity ( $3 \times 3$ matrix)
$\omega_{\infty}^*$	dual absolute conic embedded in the plane at infinity ( $3 \times 3$ matrix)
$\omega$	image of the absolute conic ( $3 \times 3$ matrices)
$\omega^*$	dual image of the absolute conic ( $3 \times 3$ matrices)
$\sim$	equivalence up to scale ( $A \sim B \Leftrightarrow \exists \lambda \neq 0 : A = \lambda B$ )
$\ \mathbf{A}\ _F$	indicates the Frobenius norm of $\mathbf{A}$ (i.e. $\sum_{ij} a_{ij}^2$ )
$\mathbf{F}(\mathbf{A})$	indicates the matrix $\mathbf{A}$ scaled to have unit Frobenius norm (i.e. $\frac{\mathbf{A}}{\ \mathbf{A}\ _F}$ )
$\mathbf{A}^T$	is the transpose of $\mathbf{A}$
$\mathbf{A}^{-1}$	is the inverse of $\mathbf{A}$ (i.e. $\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$ )
$\mathbf{A}^{\dagger}$	is the Moore-Penrose pseudo inverse of $\mathbf{A}$

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	3D from images . . . . .	2
1.2	Overview . . . . .	3
<b>2</b>	<b>Projective geometry</b>	<b>9</b>
2.1	Projective geometry . . . . .	9
2.1.1	The projective plane . . . . .	10
2.1.2	Projective 3-space . . . . .	10
2.1.3	Transformations . . . . .	10
2.1.4	Conics and quadrics . . . . .	11
2.2	The stratification of 3D geometry . . . . .	12
2.2.1	Projective stratum . . . . .	13
2.2.2	Affine stratum . . . . .	13
2.2.3	Metric stratum . . . . .	14
2.2.4	Euclidean stratum . . . . .	18
2.2.5	Overview of the different strata . . . . .	18
2.3	Conclusion . . . . .	18
<b>3</b>	<b>Camera model and multiple view geometry</b>	<b>21</b>
3.1	The camera model . . . . .	21
3.1.1	A simple model . . . . .	21
3.1.2	Intrinsic calibration . . . . .	21
3.1.3	Camera motion . . . . .	23
3.1.4	The projection matrix . . . . .	24
3.1.5	Deviations from the camera model . . . . .	25
3.2	Multi view geometry . . . . .	27
3.2.1	Two view geometry . . . . .	27
3.2.2	Three view geometry . . . . .	29
3.2.3	Multi view geometry . . . . .	31
3.3	Conclusion . . . . .	31
<b>4</b>	<b>Relating images</b>	<b>33</b>
4.1	Feature extraction and matching . . . . .	33
4.1.1	Comparing image regions . . . . .	33
4.1.2	Feature point extraction . . . . .	34
4.1.3	Matching using affinely invariant regions . . . . .	36
4.2	Two view geometry computation . . . . .	37
4.2.1	Eight-point algorithm . . . . .	37
4.2.2	Seven-point algorithm . . . . .	37
4.2.3	More points... . . . .	38
4.2.4	Robust algorithm . . . . .	38
4.2.5	Degenerate case . . . . .	39

4.3	Three and four view geometry computation . . . . .	40
<b>5</b>	<b>Structure and motion</b> . . . . .	<b>41</b>
5.1	Initial structure and motion . . . . .	41
5.1.1	Initial frame . . . . .	42
5.1.2	Initializing structure . . . . .	42
5.2	Updating the structure and motion . . . . .	42
5.2.1	projective pose estimation . . . . .	42
5.2.2	Relating to other views . . . . .	44
5.2.3	Refining and extending structure . . . . .	46
5.3	Refining structure and motion . . . . .	46
5.4	Dealing with dominant planes . . . . .	48
5.4.1	Detecting dominant planes . . . . .	48
5.4.2	Partial projective structure and motion recovery . . . . .	50
5.4.3	Combined metric structure and motion recovery . . . . .	50
5.5	Conclusion . . . . .	52
<b>6</b>	<b>Self-calibration</b> . . . . .	<b>55</b>
6.1	Calibration . . . . .	55
6.1.1	Scene knowledge . . . . .	56
6.1.2	Camera knowledge . . . . .	56
6.2	Self-calibration . . . . .	57
6.2.1	A counting argument . . . . .	57
6.2.2	Geometric interpretation of constraints . . . . .	57
6.2.3	The image of the absolute conic . . . . .	58
6.2.4	Self-calibration methods . . . . .	59
6.2.5	Critical motion sequences . . . . .	60
6.3	A practical approach to self-calibration . . . . .	61
6.3.1	linear self-calibration . . . . .	62
6.3.2	non-linear self-calibration refinement . . . . .	63
6.3.3	Metric bundle adjustment . . . . .	63
6.4	Conclusion . . . . .	64
<b>7</b>	<b>Dense depth estimation</b> . . . . .	<b>65</b>
7.1	Image pair rectification . . . . .	65
7.1.1	Planar rectification . . . . .	65
7.1.2	Polar rectification . . . . .	66
7.1.3	Examples . . . . .	69
7.2	Stereo matching . . . . .	71
7.2.1	Exploiting scene constraints . . . . .	71
7.2.2	Constrained matching . . . . .	73
7.3	Multi-view stereo . . . . .	76
7.3.1	Correspondence Linking Algorithm . . . . .	76
7.3.2	Some results . . . . .	78
7.4	Conclusion . . . . .	81
<b>8</b>	<b>Modeling</b> . . . . .	<b>83</b>
8.1	Surface model . . . . .	83
8.1.1	Texture enhancement . . . . .	83
8.1.2	Volumetric integration . . . . .	85
8.2	Lightfield model . . . . .	90
8.2.1	structure and motion . . . . .	91
8.2.2	Lightfield modeling and rendering . . . . .	91
8.2.3	Experiments . . . . .	93

8.2.4	conclusion	96
8.3	Fusion of real and virtual scenes	96
8.3.1	Augmenting video footage	96
8.4	Conclusion	97
<b>9</b>	<b>Some results</b>	<b>101</b>
9.1	Acquisition of 3D models from photographs	101
9.2	Acquisition of 3D models from pre-existing image sequences	107
9.3	Virtualizing archaeological sites	108
9.3.1	Virtualizing scenes	108
9.3.2	Reconstructing an overview model	109
9.3.3	Reconstructions at different scales	112
9.4	More applications in archaeology	112
9.4.1	3D stratigraphy	112
9.4.2	Generating and testing building hypotheses	113
9.5	Architecture and heritage conservation	114
9.6	Planetary rover control	115
9.7	Conclusion	116
<b>A</b>	<b>Bundle adjustment</b>	<b>119</b>
A.1	Levenberg-Marquardt minimization	119
A.1.1	Newton iteration	119
A.1.2	Levenberg-Marquardt iteration	120
A.2	Bundle adjustment	120