

# **OMNIDIRECTIONAL VISUAL TRACKING**

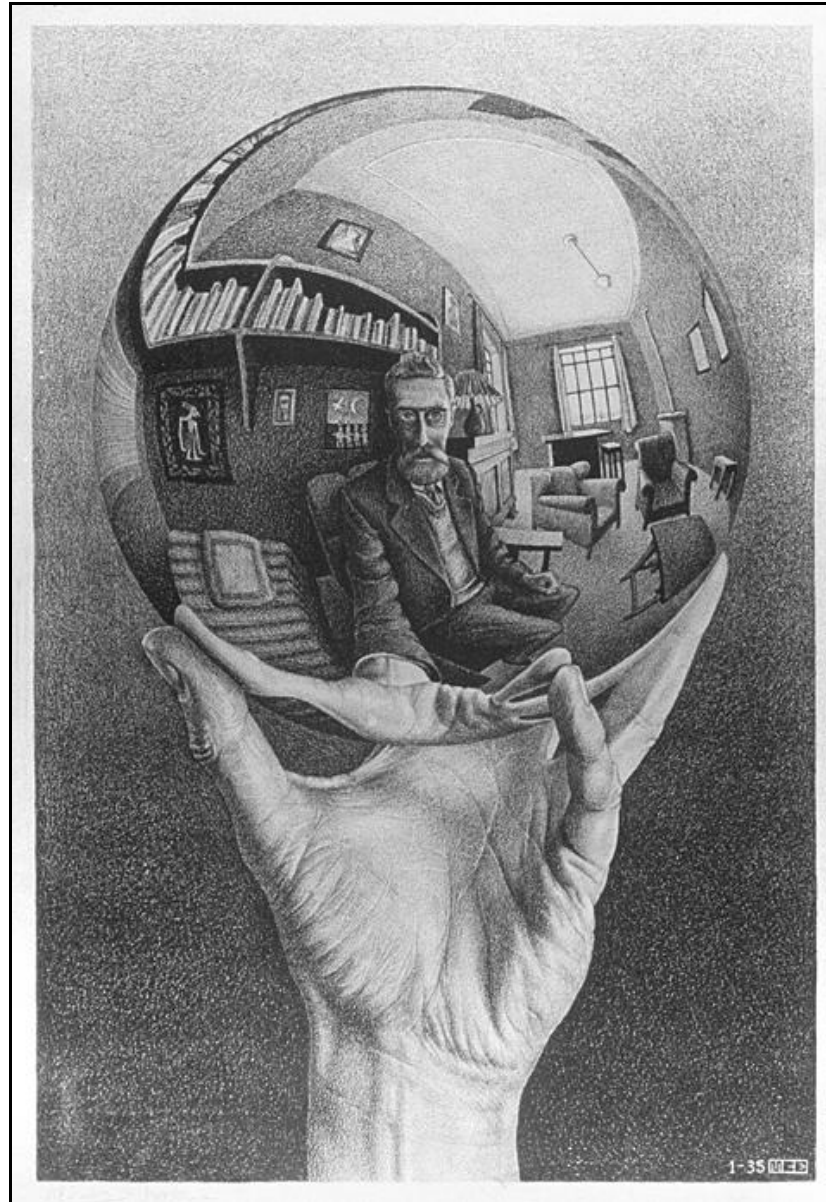
By  
**Mark Borg**

Project Supervisor  
**Dr. James M. Ferryman**

**September 2003**

A dissertation in partial fulfilment of the  
requirements for the degree of  
Master of Science in  
Engineering and Information Sciences

**The University of Reading  
School of Systems Engineering  
Department of Computer Science**



*Hand with Reflection Sphere (Self-Portrait in Spherical Mirror)*  
M. C. Escher, 1935 lithograph.

*“The picture shows a spherical mirror, resting on a left hand. Such a globe reflection collects almost one’s whole surroundings in one disk-shaped image. The whole room, four walls, the floor, and the ceiling, everything, albeit distorted, is compressed into that one small circle. Your own head, or more exactly the point between your eyes, is the absolute centre. No matter how you turn or twist yourself, you can’t get out of that central point. You are immovably the focus, the unshakable core, of your world.”*

M. C. Escher (1898 – 1972).

# Abstract

Omnidirectional vision is the ability to see in all directions at the same time. Sensors that are able to achieve omnidirectional vision offer several advantages to many areas of computer vision, such as that of tracking and surveillance. This area benefits from the unobstructed views of the surroundings acquired by these sensors and allows objects to be tracked simultaneously in different parts of the field-of-view without requiring any camera motion.

This thesis describes a system that uses an omnidirectional camera system for the purpose of detecting and tracking moving objects. We describe the methods used for detecting objects, by means of a motion detection technique, and investigate two different methods for tracking objects – a method based on tracking groups of moving pixels, commonly referred to as blob tracking methods, and a statistical colour-based method that uses a mixture model for representing the object’s colours. We evaluate these methods and show the robustness of the latter method for occlusion and other problems that are normally encountered in tracking applications.

For this thesis, we make use of a catadioptric omnidirectional camera based on a paraboloidal mirror, because of its single viewpoint and its flexibility of calibration and use. We also describe the methods used to generate virtual perspective views from the non-linear omnidirectional images acquired by the catadioptric camera. This is used in combination with the tracking results to create virtual cameras that produce perspective video streams while automatically tracking objects as they move within the camera’s field-of-view.

Finally, this thesis demonstrates the advantage that an omnidirectional visual tracking system provides over limited field-of-view systems. Objects have the potential of being tracked for as long as they remain in the scene, and are not lost because they exit the field-of-view as happens for the latter systems. This should ultimately lead to a better awareness of the surrounding world – of the objects present in the scene and their behaviour.

# Acknowledgements

I would like to express my sincere gratitude to Dr. James M. Ferryman for his continuous guidance, valuable advice and helpful discussions. I would also like to thank him for giving me the opportunity to work on a related project during my studies, from which I learned a lot about computer vision.

I also wish to thank the company I work for, Mosaic Software, for allowing me to take extended periods of leave to pursue my studies, and my colleagues at work for having to share the extra workload during my absence from the office.

Finally, many thanks to my family for all the encouragement and support they have always given me.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Aim . . . . .	2
1.2	Thesis Outline . . . . .	3
<b>2</b>	<b>Omnidirectional Vision</b>	<b>5</b>
2.1	Omnidirectional Vision . . . . .	5
2.2	Motivation . . . . .	6
2.3	Omnidirectional Camera Systems . . . . .	7
2.3.1	Rotating Camera Systems . . . . .	7
2.3.2	Multiple Camera Systems . . . . .	8
2.3.3	Fish-Eye Lens Systems . . . . .	9
2.3.4	Catadioptric Systems . . . . .	10
2.3.5	Others . . . . .	10
2.4	Comparison of the Omnidirectional Camera Systems . . . . .	11
2.4.1	Rotating Cameras . . . . .	11
2.4.2	Multiple Cameras . . . . .	12
2.4.3	Fish-Eye Cameras . . . . .	13
2.4.4	Catadioptric Cameras . . . . .	15
2.5	The Single Viewpoint Constraint . . . . .	16
2.5.1	Non-Central Omnidirectional Cameras . . . . .	18
2.6	Conclusion . . . . .	18
<b>3</b>	<b>Catadioptric Systems</b>	<b>20</b>
3.1	Catadioptric Cameras . . . . .	20
3.2	Different types of Catadioptric Systems . . . . .	21
3.2.1	Planar mirror . . . . .	21
3.2.2	Conical mirror . . . . .	22

3.2.3	Spherical mirror . . . . .	23
3.2.4	Ellipsoidal mirror . . . . .	24
3.2.5	Hyperboloidal mirror . . . . .	25
3.2.6	Paraboloidal mirror . . . . .	26
3.3	Achieving a Full Sphere of View . . . . .	27
3.4	Simulations . . . . .	29
3.5	Applications using Catadioptric Cameras . . . . .	34
3.6	Conclusion . . . . .	37
<b>4</b>	<b>Application Scenarios</b>	<b>38</b>
4.1	Indoor and Outdoor Environments . . . . .	38
4.2	Datasets . . . . .	39
4.3	Omnidirectional Cameras – Advantages for the Program . . . . .	40
4.4	Omnidirectional Cameras – Restrictions for the Program . . . . .	41
4.5	Objectives . . . . .	42
<b>5</b>	<b>Implementation Aspects</b>	<b>43</b>
5.1	Requirements of Computer Vision Applications . . . . .	43
5.2	MMX Technology . . . . .	44
5.3	OpenCV Library . . . . .	45
5.4	Windows and Threads . . . . .	47
5.5	Application Structure . . . . .	49
5.6	Conclusion . . . . .	54
<b>6</b>	<b>Catadioptric Geometry and Virtual View Generation</b>	<b>55</b>
6.1	Catadioptric Mirror Equations . . . . .	55
6.2	Paraboloidal Mirror Equation . . . . .	57
6.3	Resolution of the Paraboloidal Mirror . . . . .	59
6.4	Catadioptric Projective Geometry . . . . .	60
6.5	Re-Projection Equations for a Paraboloidal Mirror . . . . .	63
6.6	Implementation . . . . .	65
6.6.1	Camera Models . . . . .	65
6.6.2	Re-Projection Optimisation . . . . .	68
6.6.3	Interactive Control . . . . .	70
6.7	Conclusion . . . . .	75

<b>7</b>	<b>Catadioptric Camera Calibration</b>	<b>76</b>
7.1	Camera Calibration . . . . .	76
7.2	Catadioptric Camera Calibration . . . . .	77
7.2.1	Paraboloidal Mirror Calibration . . . . .	78
7.3	Review of Existing Methods . . . . .	79
7.4	Implementation . . . . .	81
7.4.1	Method Chosen . . . . .	81
7.4.2	Boundary Detection . . . . .	82
7.4.2.1	Low-Variance Method (first attempt) . . . . .	83
7.4.2.2	Iterative Thresholding Method (second attempt) . . . . .	83
7.4.2.3	Edge Detection . . . . .	85
7.4.2.4	Circle Detection . . . . .	87
7.4.3	Calculating the Paraboloid's Parameter $H$ . . . . .	89
7.5	Results and Conclusions . . . . .	90
<b>8</b>	<b>Moving Object Detection</b>	<b>94</b>
8.1	Optical Flow . . . . .	95
8.2	Frame Difference . . . . .	95
8.3	Background Subtraction . . . . .	96
8.3.1	Simplest Form – Mean and Global Threshold Method . . . . .	97
8.3.2	Handling Noise – Normal Distribution Method . . . . .	98
8.3.3	Handling Background Motion – Mixture of Gaussians Method . . . . .	99
8.3.4	Post-Processing . . . . .	100
8.4	Background Adaptation . . . . .	100
8.4.1	Types of Background Update . . . . .	102
8.5	Review of Applications using Background Subtraction . . . . .	102
8.6	Types and Sources of Detection Errors . . . . .	105
8.7	Implementation . . . . .	109
8.7.1	Choosing a Background Model . . . . .	109
8.7.1.1	A Test using Mixture Models . . . . .	109
8.7.1.2	The Chosen Model . . . . .	111
8.7.2	Robustness to Shadow . . . . .	113
8.7.2.1	HSV Colour Space and Shadow Detection . . . . .	114
8.7.2.2	Low Chromaticity Conditions . . . . .	115

8.7.3	Background Initialisation . . . . .	116
8.7.4	Background Subtraction Algorithm . . . . .	117
8.7.5	Thresholding with Hysteresis . . . . .	119
8.7.6	Background Model Adaptation . . . . .	120
8.7.6.1	Background Model Failure . . . . .	122
8.8	Results . . . . .	123
<b>9</b>	<b>Object Tracking – an Overview</b>	<b>126</b>
9.1	Object Tracking . . . . .	126
9.2	Tracking Problems . . . . .	129
9.3	Different Tracking Methods . . . . .	131
9.4	Tracking for Omnidirectional Applications . . . . .	133
9.5	Chosen Methods – an Introduction . . . . .	135
<b>10</b>	<b>Blob Tracking</b>	<b>136</b>
10.1	Connected Components . . . . .	136
10.1.1	Size Filtering . . . . .	137
10.1.2	Blob Clustering . . . . .	137
10.2	Object Features . . . . .	141
10.3	Similarity Measures . . . . .	142
10.3.1	The Overlap Measure . . . . .	143
10.3.2	The Centroid Distance Measure . . . . .	144
10.3.3	The Area Ratio Measure . . . . .	144
10.3.4	The Colour Similarity Measure . . . . .	144
10.4	Object Tracking . . . . .	146
10.4.1	Match Scoring . . . . .	148
10.4.2	Matching Blob Clusters to Objects . . . . .	149
10.4.3	Updating the Object . . . . .	151
10.5	Results . . . . .	152
10.6	Conclusion . . . . .	155
<b>11</b>	<b>Colour-based Tracking</b>	<b>156</b>
11.1	Colour Models . . . . .	156
11.2	Mixture Models . . . . .	158
11.2.1	Gaussian Mixture Models . . . . .	158
11.3	Tracking with Colour Models . . . . .	161



11.4	Colour Model Learning . . . . .	163
11.4.1	The Expectation-Maximisation (EM) Algorithm . . . . .	165
11.4.1.1	Selecting the Initial Parameters . . . . .	168
11.5	Colour Model Update . . . . .	170
11.6	Implementation Issues . . . . .	173
11.7	Results . . . . .	176
11.8	Conclusion . . . . .	179
<b>12</b>	<b>3D Localisation and High-Level Processing</b>	<b>180</b>
12.1	3D Localisation . . . . .	180
12.2	Automatic Target Tracking with Virtual Cameras . . . . .	182
12.2.1	Automatic Target Tracking . . . . .	183
12.2.2	Manually-Initiated Automatic Target Tracking . . . . .	184
12.2.3	Automatic Camera Control . . . . .	184
12.3	Tracking History . . . . .	185
12.4	Conclusion . . . . .	186
<b>13</b>	<b>Conclusions and Future Work</b>	<b>187</b>
13.1	Conclusions . . . . .	187
13.2	Future Work . . . . .	190
	<b>Bibliography</b>	<b>191</b>

## List of Figures

2.1	A 360° panorama created by rotating a camera around its focal point . .	8
2.2	Multi-Camera systems . . . . .	9
2.3	Fish-Eye camera systems . . . . .	9
2.4	Catadioptric camera systems . . . . .	10
2.5	Single Centre of Projection and Single Viewpoint . . . . .	17
2.6	Generating virtual perspective views . . . . .	18
3.1	Planar mirror . . . . .	22
3.2	Conical mirror . . . . .	23
3.3	Spherical mirror . . . . .	24
3.4	Ellipsoidal mirror . . . . .	25
3.5	Hyperboloidal mirror . . . . .	25
3.6	Paraboloidal mirror . . . . .	27
3.7	Achieving 360° by 360° with two paraboloidal mirrors . . . . .	28
3.8	Simulations – the room . . . . .	30
3.9	Simulations – Spherical and Conical mirrors . . . . .	30
3.10	Simulations – Ellipsoidal mirror . . . . .	31
3.11	Simulations – Paraboloidal mirror . . . . .	32
3.12	Simulations – Hyperboloidal mirror . . . . .	32
3.13	Two paraboloidal systems to achieve a full view . . . . .	33
3.14	Panorama captured by two paraboloidal systems . . . . .	33
4.1	Two different scenarios . . . . .	39
4.2	Omnidirectional and conventional (linear) cameras . . . . .	41
5.1	Libraries . . . . .	46
5.2	Pipeline model . . . . .	50
5.3	Node thread execution timeline . . . . .	51
5.4	OmniTracking application pipeline . . . . .	53

5.5	System diagram . . . . .	54
6.1	Geometry of catadioptrics with single viewpoint . . . . .	56
6.2	Geometry of the 2D parabolic profile of the mirror . . . . .	58
6.3	Geometry of the 3D paraboloidal mirror . . . . .	59
6.4	Non-uniform mirror gain of the paraboloidal mirror . . . . .	60
6.5	Equivalence of parabolic projection to stereographic projection . . . . .	61
6.6	The horizon and vertical lines under parabolic projection . . . . .	62
6.7	Lines projected into circles and their vanishing points . . . . .	62
6.8	Re-projection . . . . .	63
6.9	Generating perspective and panoramic views . . . . .	65
6.10	Virtual perspective camera model . . . . .	66
6.11	Virtual panoramic camera model . . . . .	67
6.12	Optimising the distance factor in the re-projection equations . . . . .	69
6.13	Algorithm for calculating the re-projection look-up table . . . . .	71
6.14	A screenshot of the OmniTracking system with virtual panoramic and perspective video cameras . . . . .	72
6.15	Different examples of perspective and panoramic view with camera control parameters applied to them . . . . .	73-74
6.16	Performance of generating virtual camera views by the OmniTracking program . . . . .	75
7.1	Calibration of the catadioptric internal parameters . . . . .	77
7.2	Circle-based self-calibration . . . . .	82
7.3	JPEG noise affecting pixel variance and boundary detection . . . . .	84
7.4	Iterative thresholding for the PETS2001 dataset . . . . .	85
7.5	Iterative thresholding for the PETS-ICVS dataset . . . . .	86
7.6	Results of the edge detection and external contour extraction phase . . .	87
7.7	Hierarchical Hough transform, with multi-resolution image pyramid and reduction of parameter space search . . . . .	88
7.8	Calculating the paraboloidal mirror's parameter $H$ . . . . .	90
7.9	Calibration results for the PETS datasets . . . . .	91
7.10	Calibration algorithm . . . . .	92
7.11	Calibration tests under various simulated light, noise and visibility conditions . . . . .	92-93

8.1	Frame differencing technique . . . . .	96
8.2	Background subtraction technique – the basic idea . . . . .	97
8.3	Background subtraction technique – normal distribution model . . . . .	98
8.4	Background subtraction technique – Gaussian mixture model . . . . .	100
8.5	Exponential forgetting for background update . . . . .	101
8.6	Mixture models for pixels of the PETS2001 dataset . . . . .	112
8.7	Detection errors caused by shadow . . . . .	113
8.8	Low chromaticity thresholds for HSV colour space . . . . .	116
8.9	Background mean and standard deviation maps . . . . .	117
8.10	Thresholding with hysteresis – some results . . . . .	121
8.11	Results: Detection of shadows . . . . .	123
8.12	Results: Noise due to camera movements . . . . .	124
8.13	Results: Camouflage and object fragmentation . . . . .	124
8.14	Results: The ghost problem . . . . .	125
8.15	Results: Foreground aperture problem . . . . .	125
8.16	Results: Moved background elements . . . . .	125
9.1	Bottom-up and top-down object tracking . . . . .	128
9.2	World objects and image objects . . . . .	129
9.3	Non-rigid object boundary changes . . . . .	130
9.4	Corner tracking and its affine and linear translation model . . . . .	134
10.1	Connected components and blob clustering . . . . .	138
10.2	Blob clustering algorithm . . . . .	140
10.3	Object features . . . . .	142
10.4	Overlap and centroid distance measures . . . . .	143
10.5	Colour similarity measure . . . . .	146
10.6	Blob tracking algorithm . . . . .	147
10.7	Matching as a graph search problem . . . . .	151
10.8	Results: Blob clustering . . . . .	152
10.9	Results: Occlusion and merging . . . . .	153
10.10	Results: Colour similarity . . . . .	155
11.1	Non-parametric statistical colour model based on a HSV-colour histogram . . . . .	158
11.2	Parametric statistical colour model based on a Gaussian mixture	

model . . . . .	160
11.3 Locating an object by finding the region with highest colour probability . . . . .	162
11.4 Combining colour tracking with motion detection . . . . .	162
11.5 Using temporal constraints for the search window . . . . .	163
11.6 Learning a colour model using the EM algorithm . . . . .	169
11.7 Convergence of the log-likelihood value of the colour model . . . . .	170
11.8 Colour tracking algorithm . . . . .	173
11.9 Probability map used as a look-up table . . . . .	174
11.10 Results: Tracking through partial occlusion . . . . .	177
11.11 Results: Tracking through occlusion . . . . .	178
11.12 Results: Poor colour model . . . . .	179
12.1 3D localisation by means of back-projection from omnidirectional images. . . . .	181
12.2 Results of 3D localisation for the PETS2001 dataset . . . . .	182
12.3 Different ways of automatically tracking objects . . . . .	184
12.4 Tracking history results generated by the application and viewed in a web browser . . . . .	186

## List of Tables

3.1	Different mirror types and applications using them . . . . .	34
6.1	Solutions to the mirror equations . . . . .	57
7.1	Volume searched in Hough transform parameter search for the PETS-ICVS dataset . . . . .	89
10.1	Match score matrix . . . . .	150

