SSIP 2011

19th Summer School on Image Processing

7 July - 16 July 2011 • Szeged, Hungary

3D Reconstruction from two views

Presented by **Radu Orghidan**

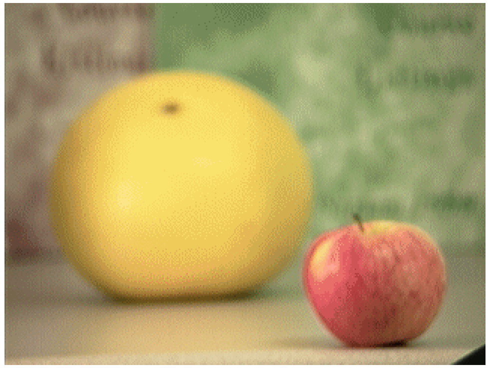
Contents

3.1 Shape from *X*

Techniques based on:

- Modifying the intrinsic camera parameters

  i.e. Depth from Focus/Defocus and Depth from Zooming



Shape from Focus/Defocus

- Considering an addi al source e

  i.e. Shape from Structu ar ric Stereo

- Considering additional surface in

  i.e. Shape from Shading, Shape fr om Geometric Constraints

- Multiple views

  i.e. Shape from Stereo and Shape from Motion

3.1 Shape from *X*

Techniques based on:

- Modifying the intrinsic camera parameters

  i.e. Depth from Focus/Defocus and Depth from Zooming

- Considering an additional source of light onto the scene

  i.e. Shape from Structured Light and Shape from Photometric Stereo

- Considering additiona… in…

  i.e. Shape from Shading, Shape fr… om Geometric Constraints

Shape from Structured Light

- Multiple views

  i.e. Shape from Stereo and Shape from Motion

3.1 Shape from *X*

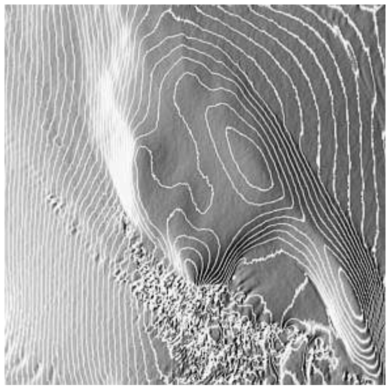
Techniques based on:

- Modifying the intrinsic camera parameters
  i.e. Depth from Focus/Defocus and Depth from Zooming

- Considering an additional source of light onto the scene
  i.e. Shape from Structured Light and Shape from Photometric Stereo

- Considering additional surface information
  i.e. Shape from Shading, Shape from Texture and Shape from Geometric Constraints

- Multiple views
  i.e. Shape from Stereo and Shape



Shape from Shading

3.1 Shape from *X*

Techniques based on:

- Modifying the intrinsic camera parameters

  i.e. Depth from Focus/Defocus and Depth from Zooming

- Considering an additional source of light onto the scene

  i.e. Shape from Structured Light and Shape from Photometric Stereo

- Considering additional surface in

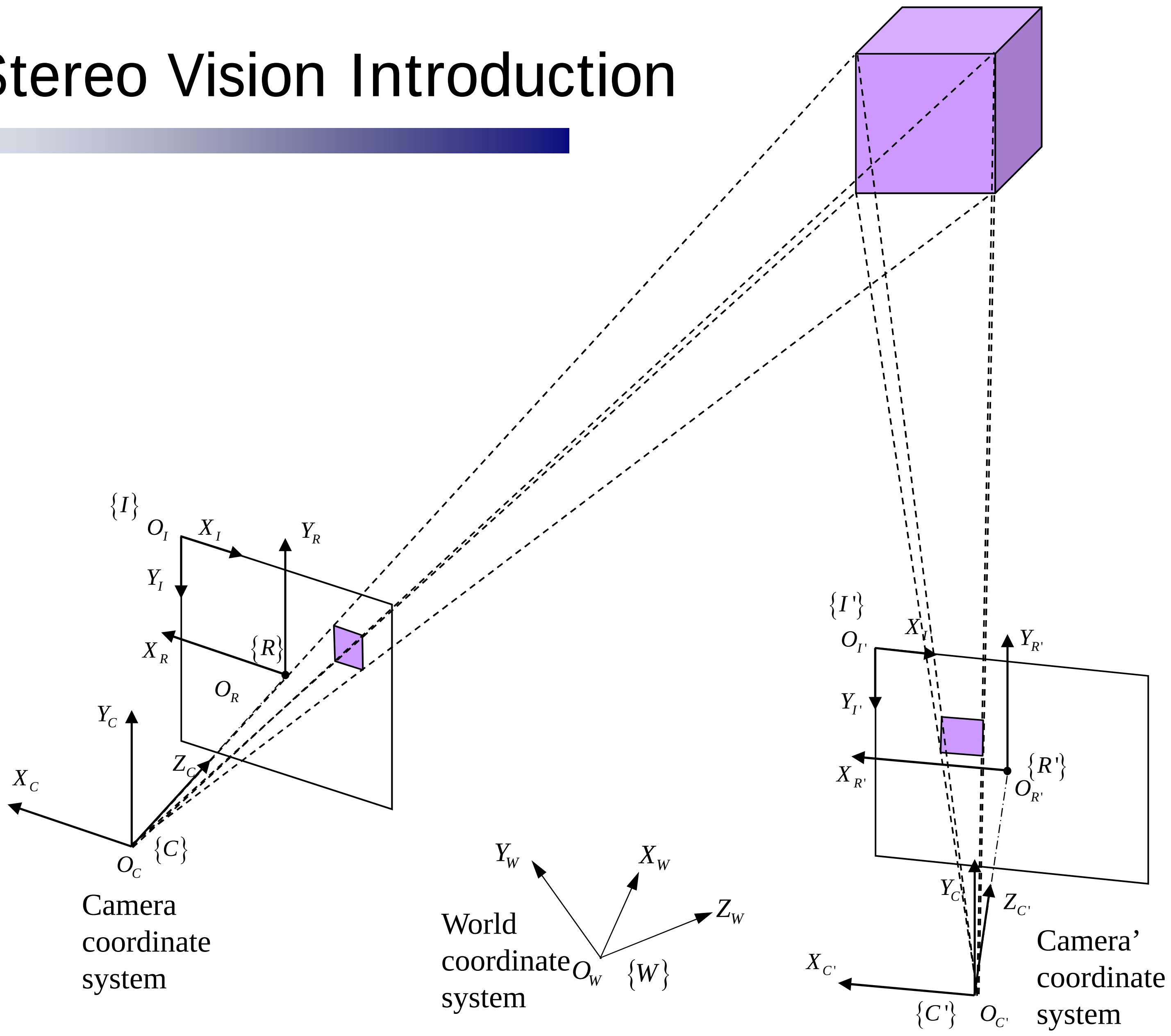
  i.e. Shape from Shading, Shape fr om Geometric Constra

- Multiple views

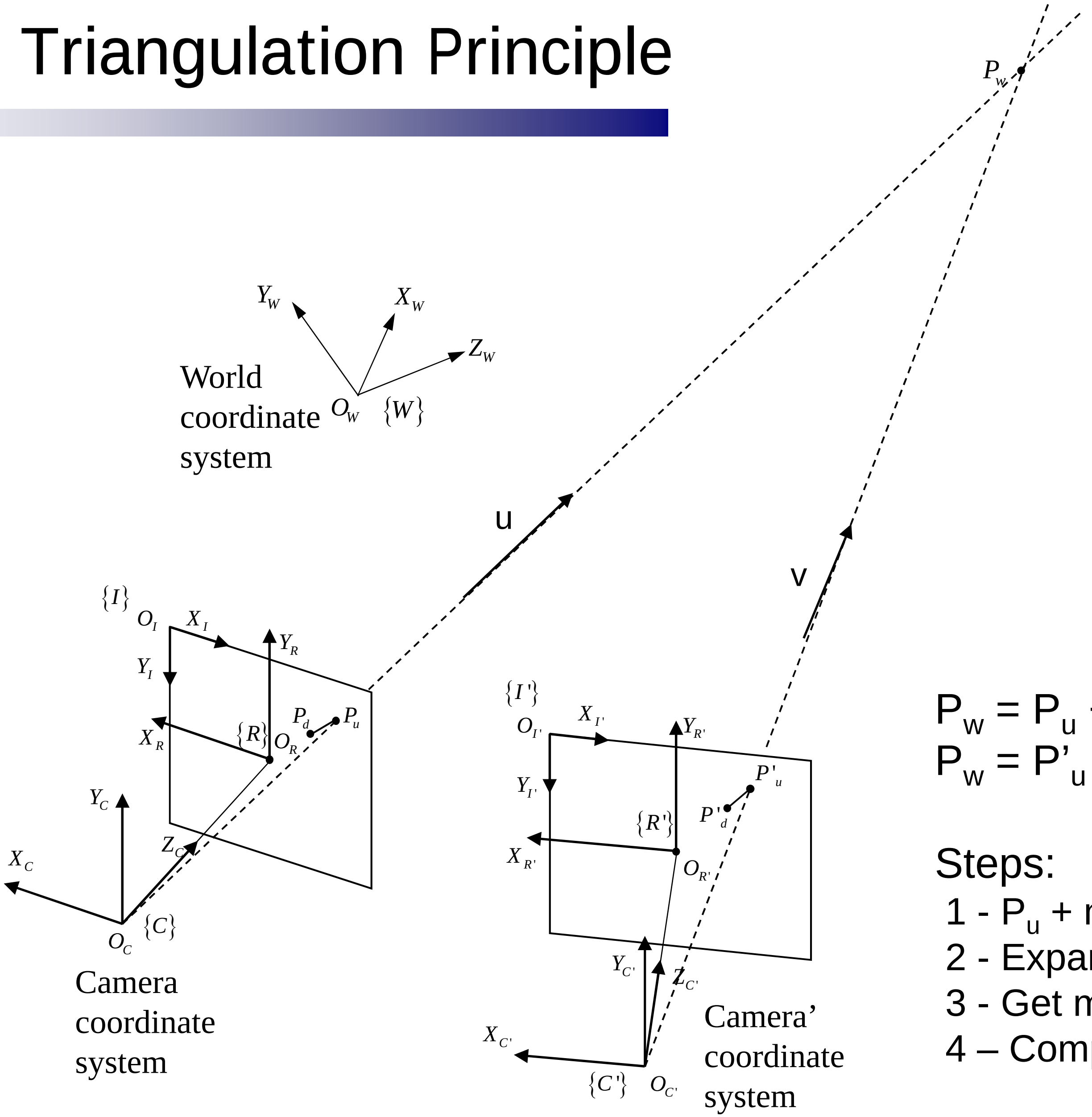
  i.e. Shape from Stereo and Shape from Motion

Shape from Stereo

3.2 Stereo Vision Introduction



3.3 Triangulation Principle



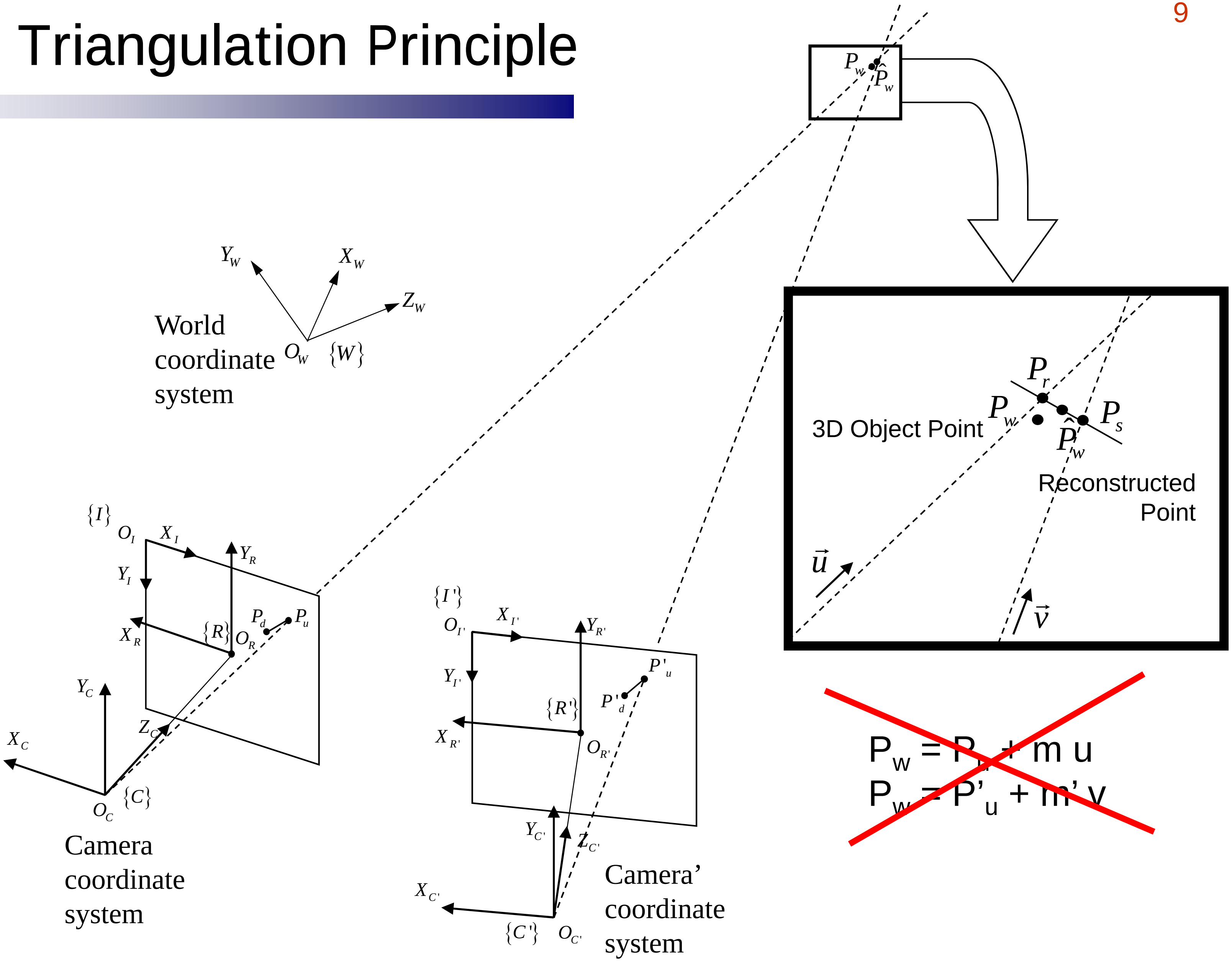
$P_w = P_u + m\,u$

$P_w = P'_u + m'\,v$

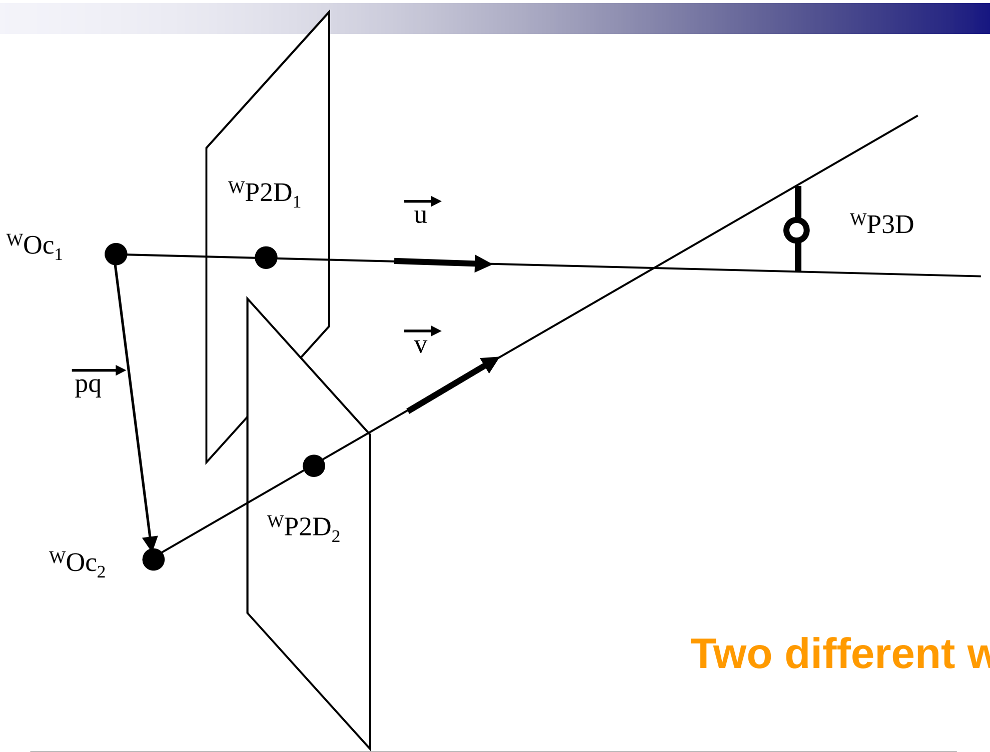
Steps:

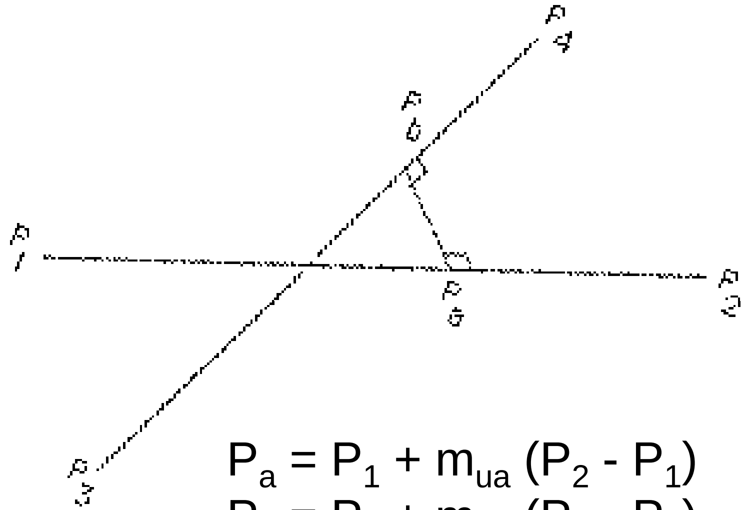
1 - $P_u + m\,u = P'_u + m'\,v$
2 - Expand to x,y,z
3 - Get m and m'
4 – Compute Pw

3.3 Triangulation Principle



3.3 Triangulation Principle





$P_a = P_1 + m_{ua}\ (P_2 - P_1)$
$P_b = P_3 + m_{ub}\ (P_4 - P_3)$

Two different ways:

Minimize the distance between points:

$\text{Min}\ \|\ P_b - P_a\ \|^2$

$\text{Min}\ \|\ P_1 + m_{ua}\ (P_2 - P_1) - P_3 - m_{ub}\ (P_4 - P_3)\ \|^2$

Finding m_{ua} and m_{ub} once expanded to (x,y and z)

Compute the dot product between vectors:

$(P_a - P_b)^T\ (P_2 - P_1) = 0$

$(P_a - P_b)^T\ (P_4 - P_3) = 0$

Because they are perpendicular.

Finding m_{ua} and m_{ub} once expanded to P_a, P_b and (x,y and z)

http://astronomy.swin.edu.au/~pbourke/geometry/lineline3d/

3.3 Constraints in Stereo Vision

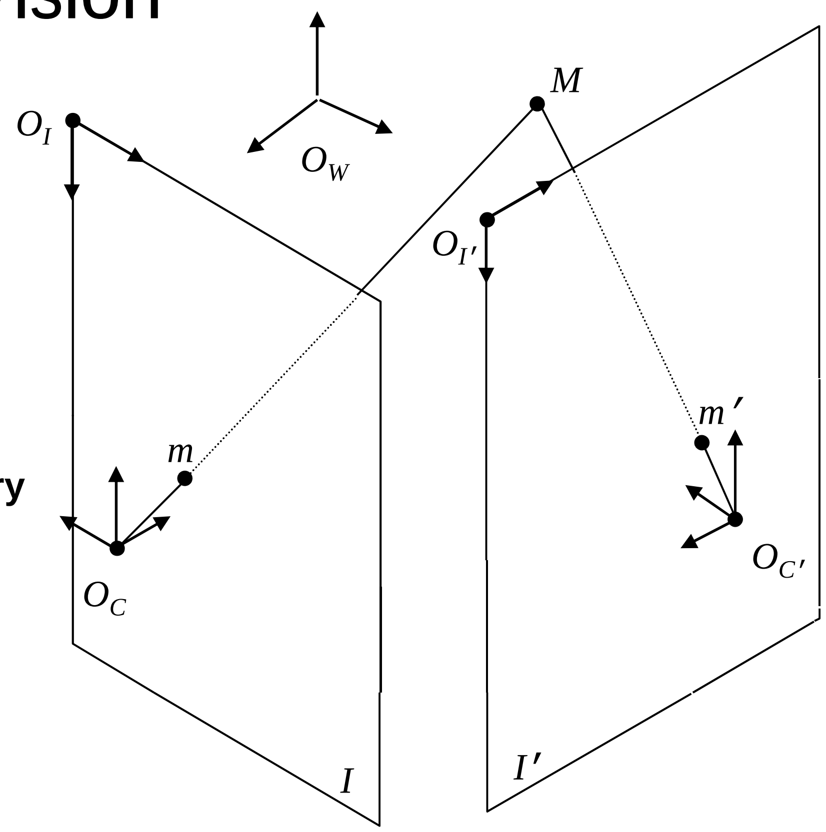
3D Reconstruction:

$$s\,{}^{I}m = {}^{I}A_{C}\,{}^{C}K_{W}\,{}^{W}M$$

$$s'\,{}^{I'}m' = {}^{I'}A'_{C'}\,{}^{C'}K'_{W'}\,{}^{W}M$$

${}^{I}A_{C}\ ;\ {}^{I'}A'_{C'}$ **Intrinsics: Optics & Internal Geometry**

${}^{C}K_{W}\ ;\ {}^{C'}K'_{W'}$ **Extrinsics: Camera Pose**



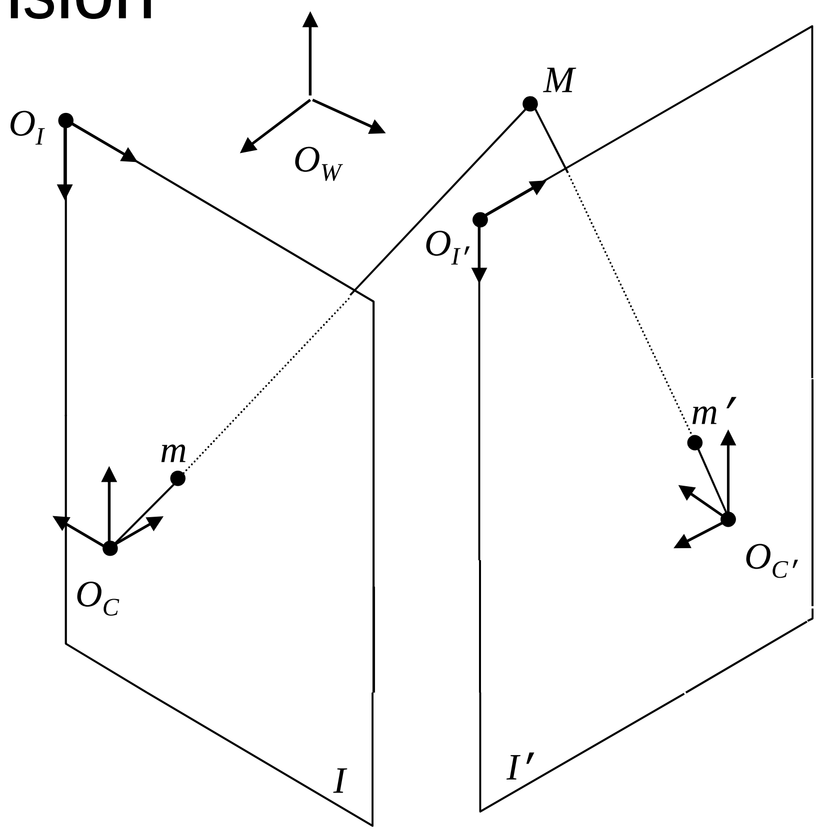
Constraints:

- The Correspondence Problem → F/E matrix

3.3 Constraints in Stereo Vision

Calibrated 3D Reconstruction process:

1. The **Internal Parameters** are known (by camera calibration)
2. Calculate the **Fundamental Matrix**
3. Determine the **External Parameters** (rotation and translation from one camera to the other) from the Fundamental Matrix
4. Determine **3D point locations,** i.e. perform the **3D reconstruction.**



3.3 Constraints in Stereo Vision

3D Reconstruction:

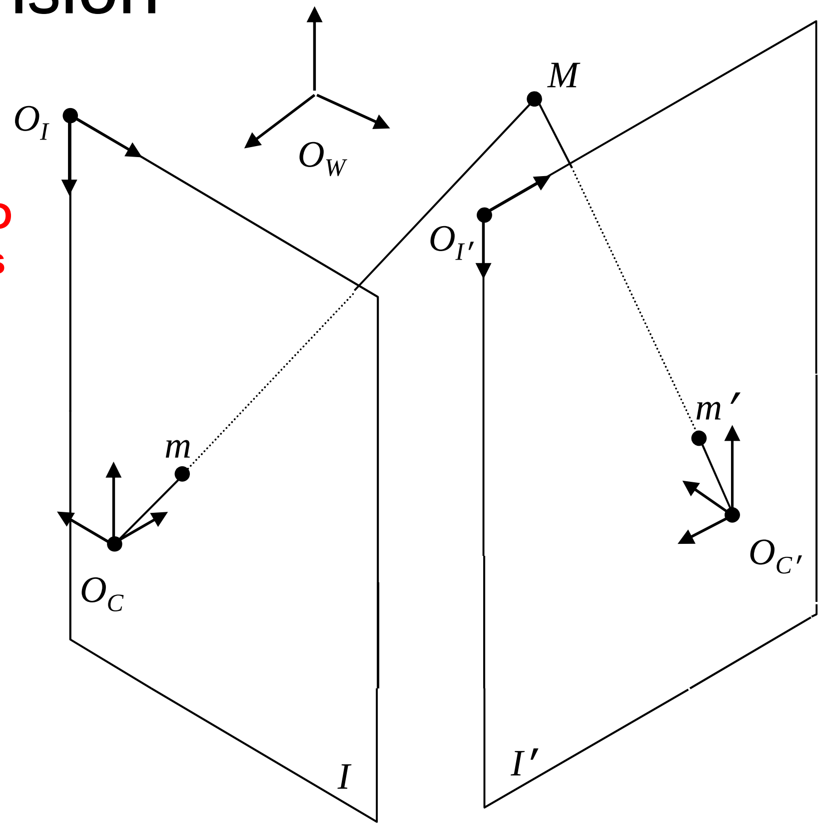
$$s\,{}^{I}m = {}^{I}A_{C}\,{}^{C}K_{W}\,{}^{W}M$$

$$s'\,{}^{I'}m' = {}^{I'}A'_{C'}\,{}^{C'}K'_{W'}\,{}^{W}M$$

Unknown 3D coordinates

2D image coordinates

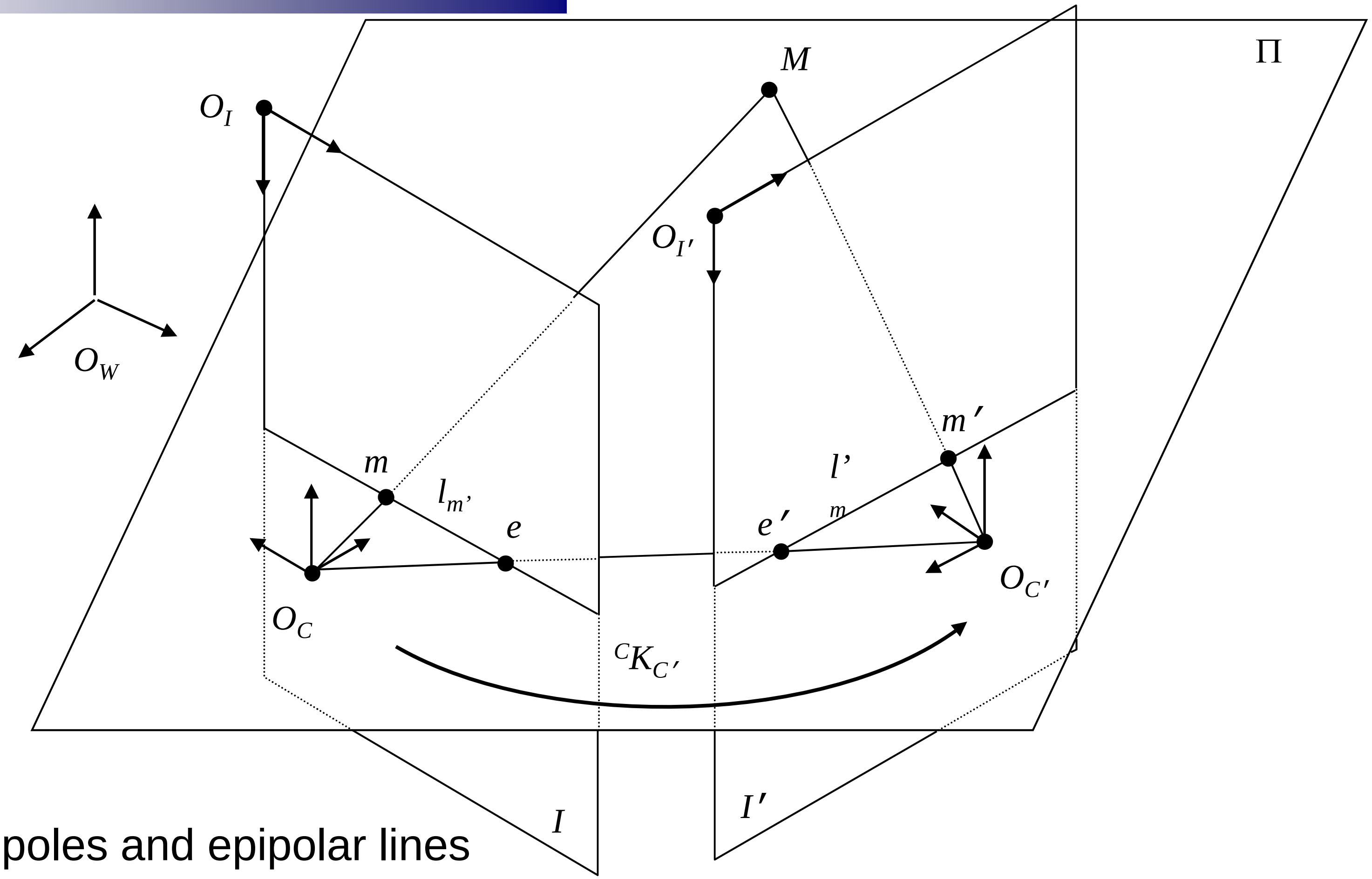
Intrinsic and extrinsic parameters



Obtained by **Feature extraction techniques**

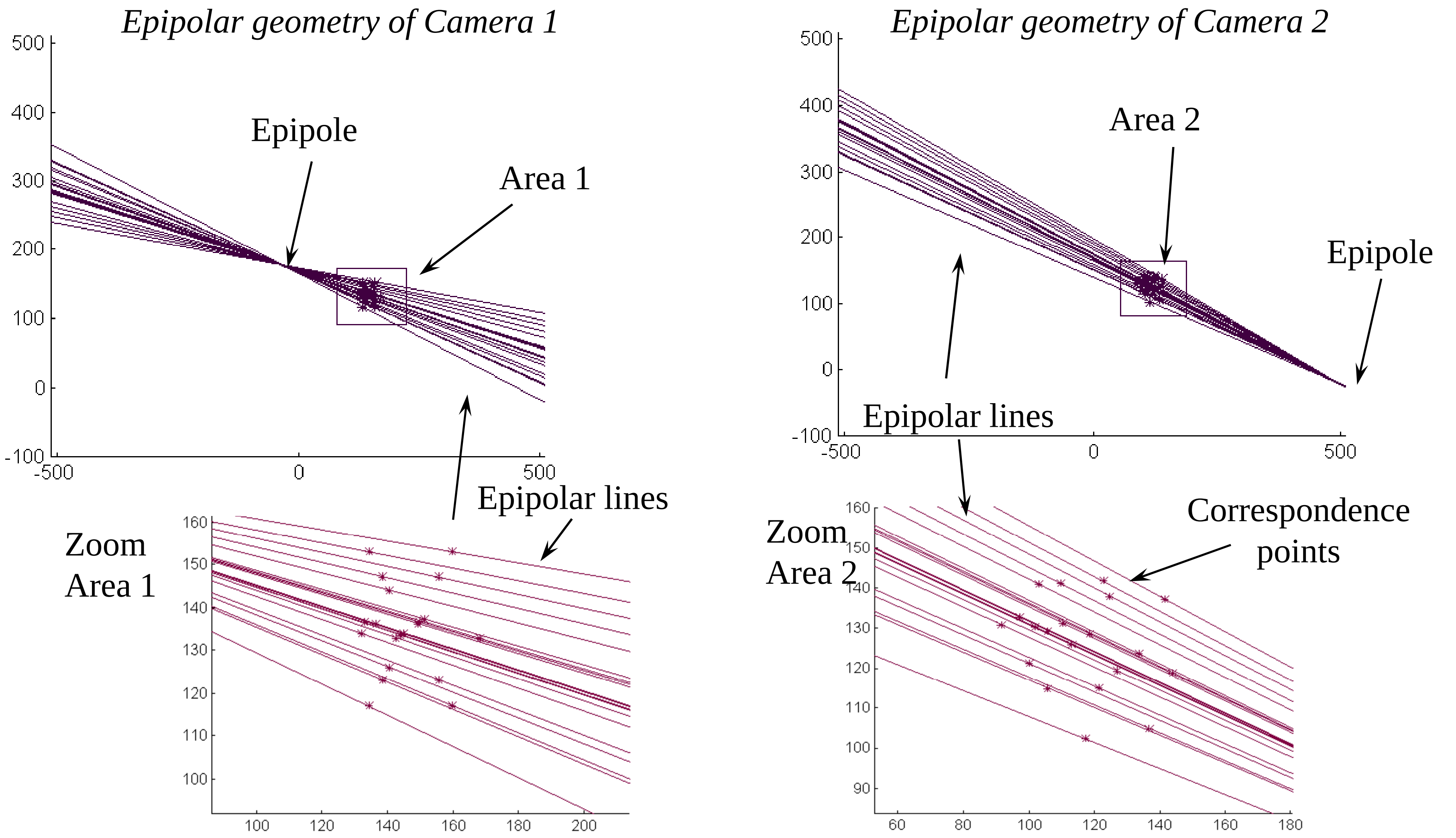
The **intrinsic parameters** must be obtained from camera calibration.
The **extrinsic parameters** are obtained from the **FUNDAMENTAL MATRIX**

3.4 Epipolar Geometry (I) - Modelling

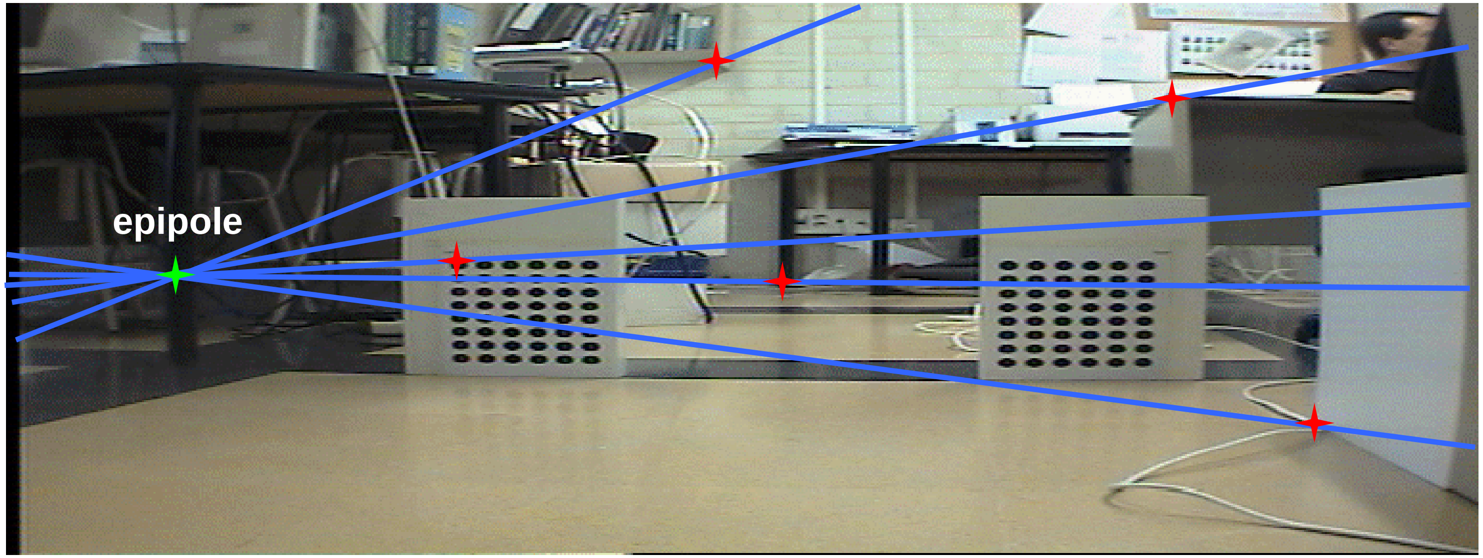


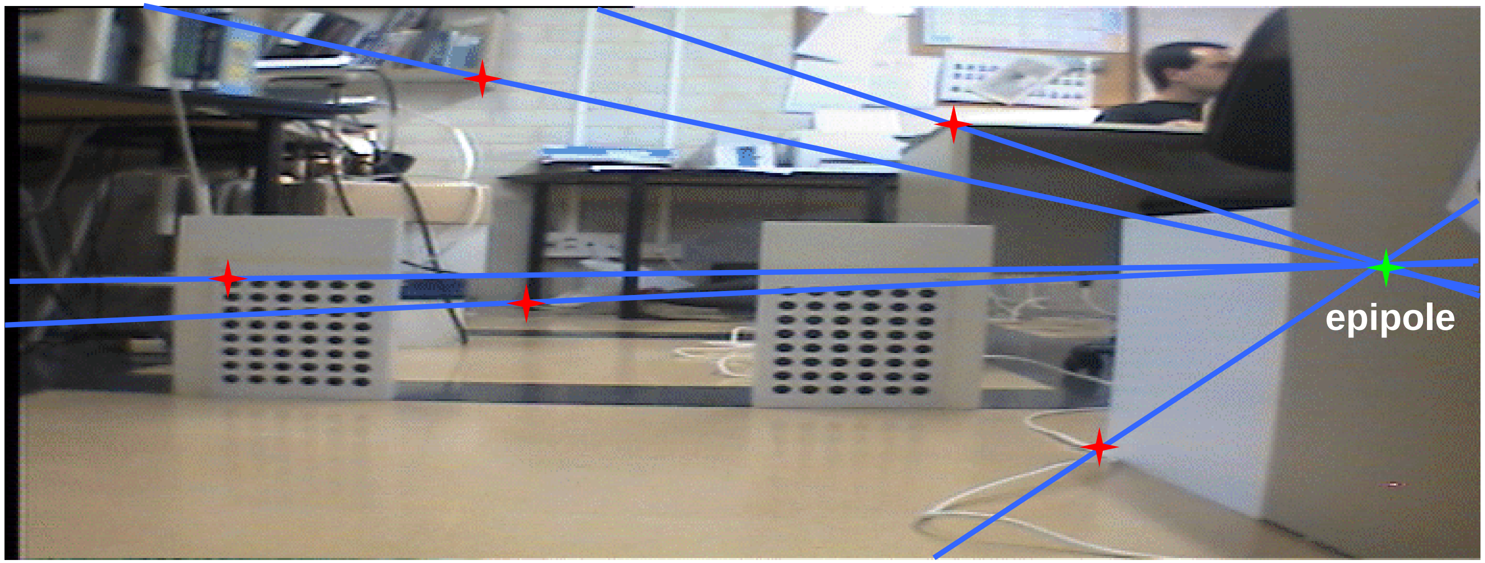
- Focal points, epipoles and epipolar lines
- e is defined by $O_{C'}$ in {I}, e' is defined by O_C in {I'}
- m defines an epipolar line in {I'}; m' defines an epipolar line in {I}
- All epipolar lines intersect at the epipole

3.4 Epipolar Geometry (II) - Modelling

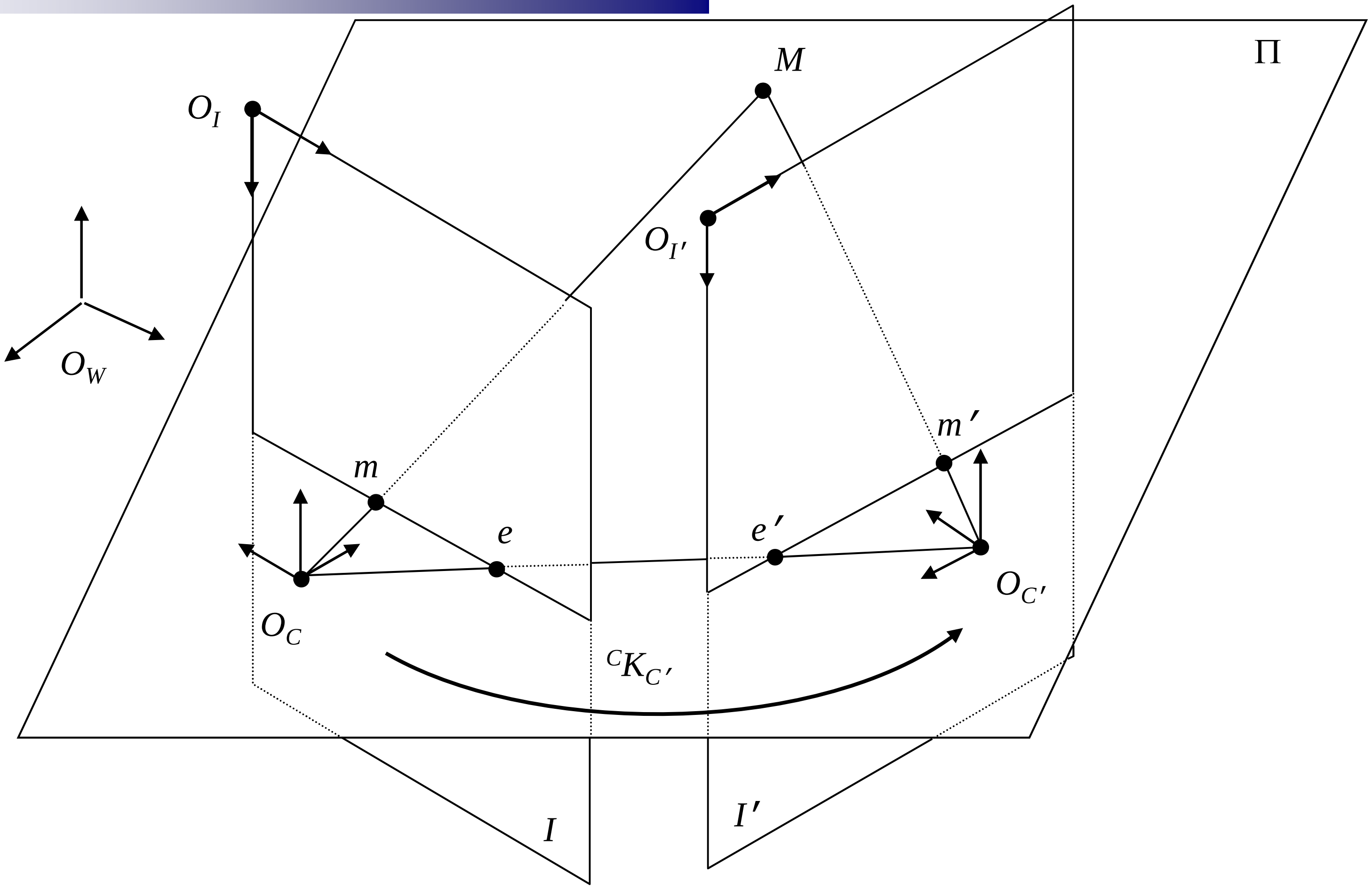


3.4 Epipolar Geometry (III) - Modelling



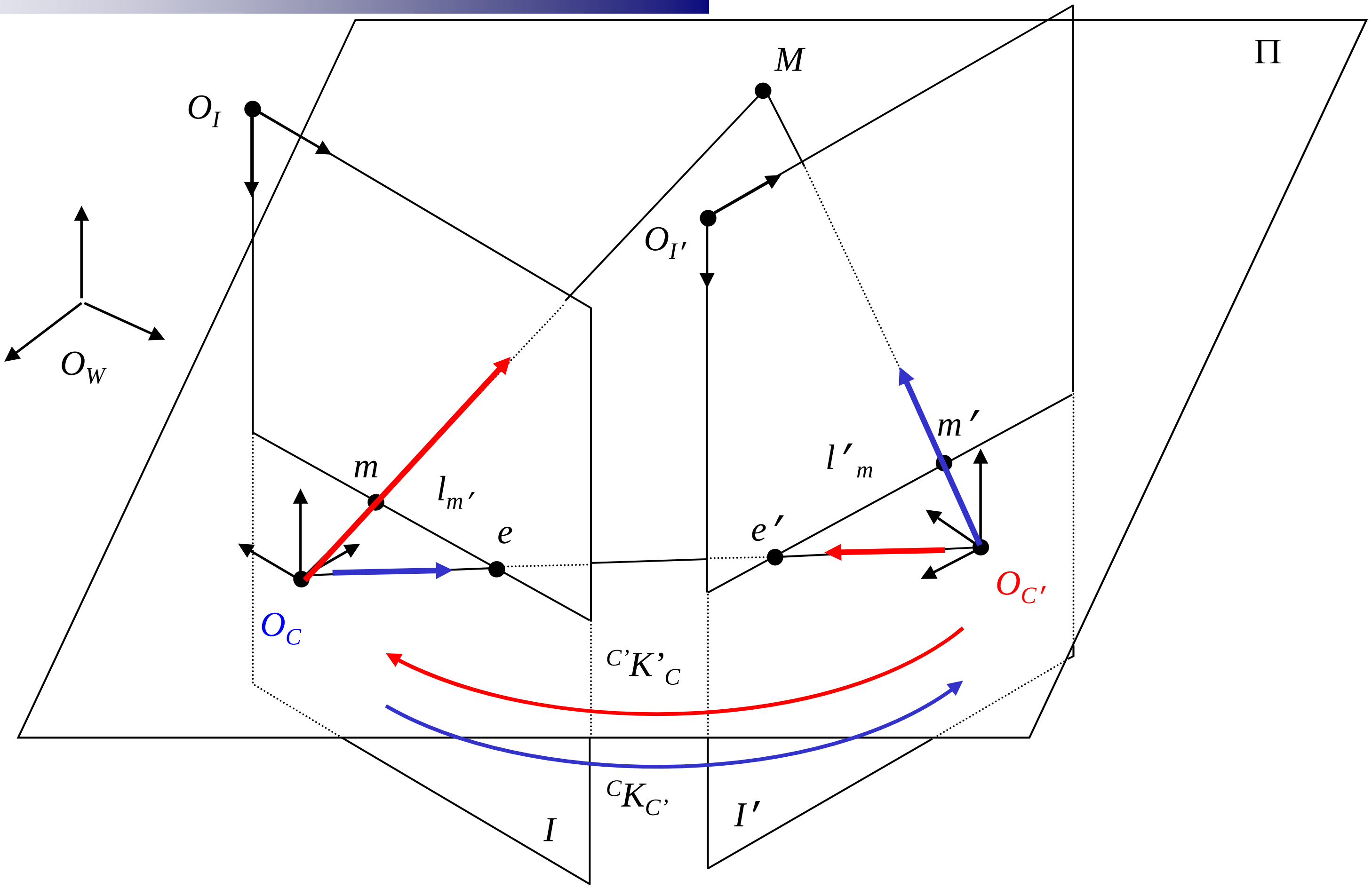


3.4 Epipolar Geometry (IV) - Modelling



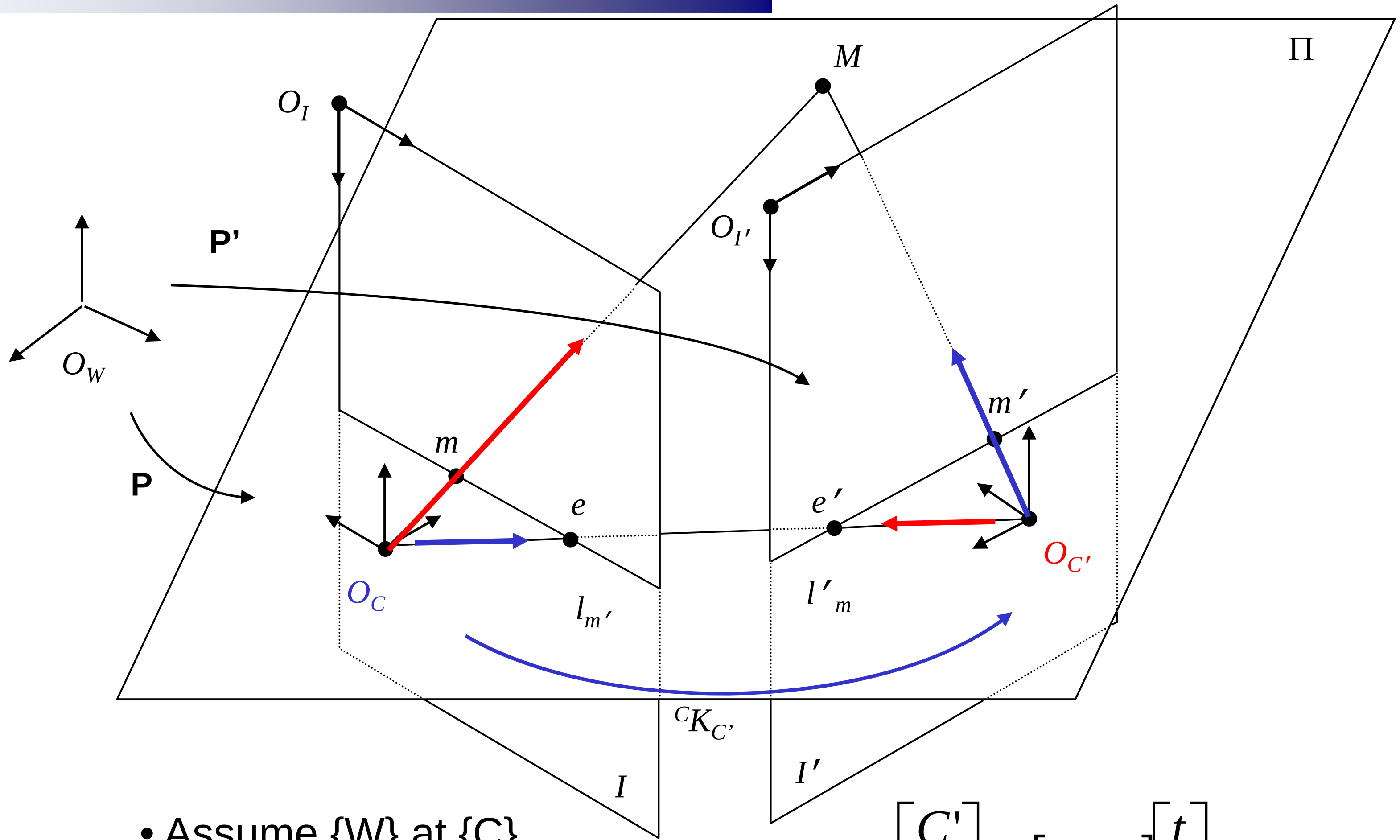
- The Epipolar Geometry concerns the problem of computing the plane Π.
  - A plane is defined by the cross product between two vectors
  - M is unknown, m and m' are known
  - {W} is located at {C} or {C'} and Π can be computed at {C} or {C'} → 4 solutions

3.4 Epipolar Geometry (IV) - Modelling



- The Epipolar Geometry concerns the problem of computing the plane Π.
  - A plane is defined by the cross product between two vectors
  - M is unknown, m and m' are knowns
  - {W} is located at {C} or {C'} and Π can be computed at {C} or {C'} → 4 solutions

3.4 Epipolar Geometry (V) - Modelling



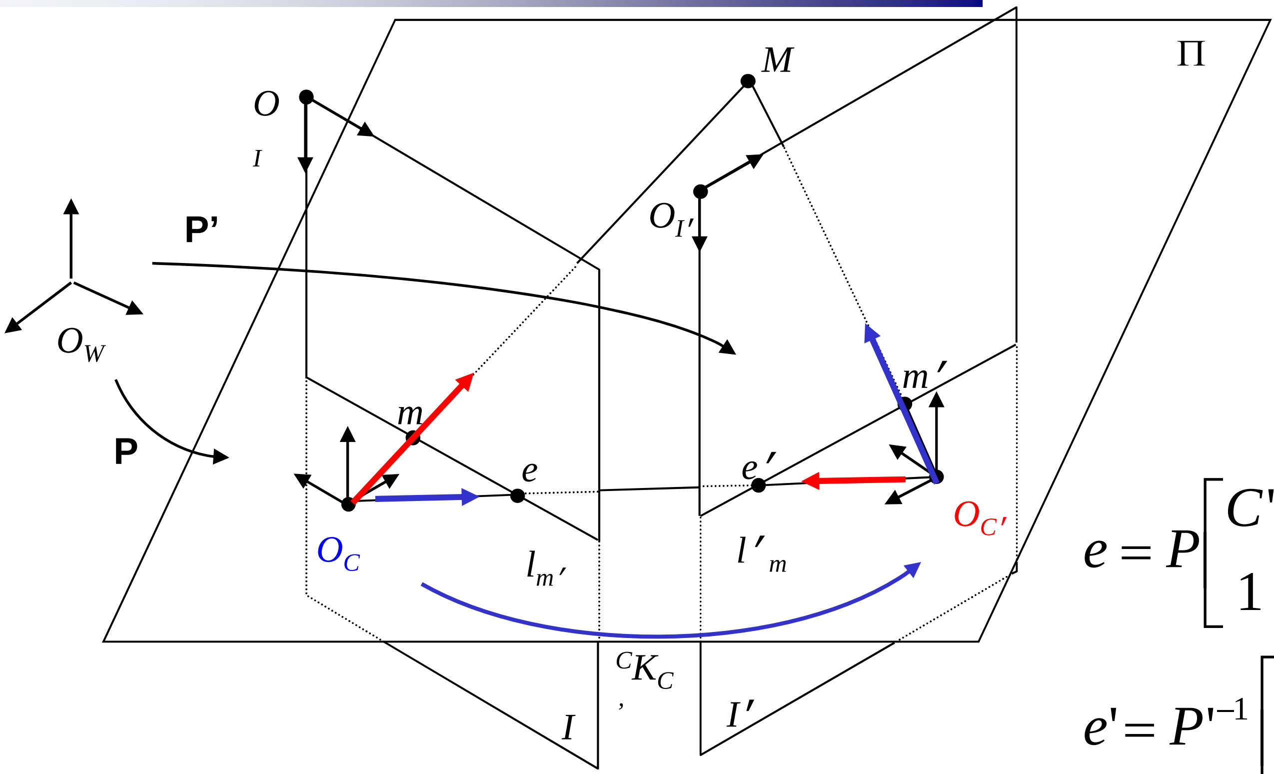
- Assume {W} at {C}

$$P = [I \quad 0]$$
$$K = [R \quad t]$$
$$P' = PK = [R \quad t]$$

$$e = P\begin{bmatrix} C' \\ 1 \end{bmatrix} = [I \quad 0]\begin{bmatrix} t \\ 1 \end{bmatrix} = t$$

$$e' = P'^{-1}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = [R^t \quad -R^t t]\begin{bmatrix} 0 \\ 1 \end{bmatrix} = -R^t t$$

3.4 Epipolar Geometry (VI) - Modelling



- Assume {W} at {C}

$$P = \begin{bmatrix} I & 0 \end{bmatrix}$$

$$K = \begin{bmatrix} R & t \end{bmatrix}$$

$$P' = PK = \begin{bmatrix} R & t \end{bmatrix}$$

$$e = P\begin{bmatrix} C' \\ 1 \end{bmatrix} = \begin{bmatrix} I & 0 \end{bmatrix}\begin{bmatrix} t \\ 1 \end{bmatrix} = t$$

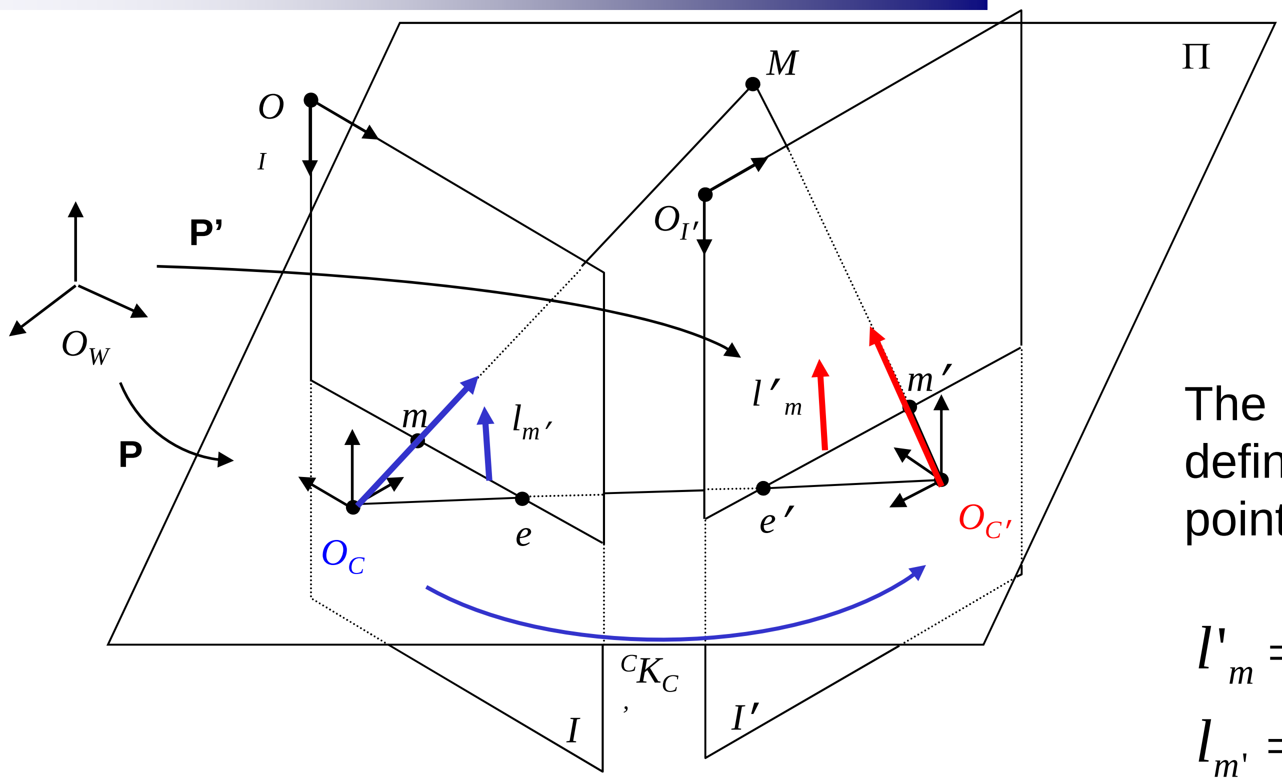
$$e' = P'^{-1}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} R^t & -R^t t \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = -R^t t$$

Since epipolar lines are contained in the plane Π, we can define the line by a cross product of two vectors, obtaining the orthogonal vector of the line.

$$l'_m = e' \times P'^{-1} m = -R^t t \times R^t m = -R^t (t \times m) = -R^t [t]_x m$$

$$l_{m'} = e \times P' m' = t \times R m' = [t]_x R m'$$

3.4 Epipolar Geometry (VII) - Modelling



The Fundamental matrix is defined by inner product of a point with their epipolar line.

$$l'_m = -R^t [t]_x m$$

$$l_{m'} = [t]_x Rm'$$

$$m' \cdot l'_m = m'^t \, l'_m = -m'^t \, R^t [t]_x m$$

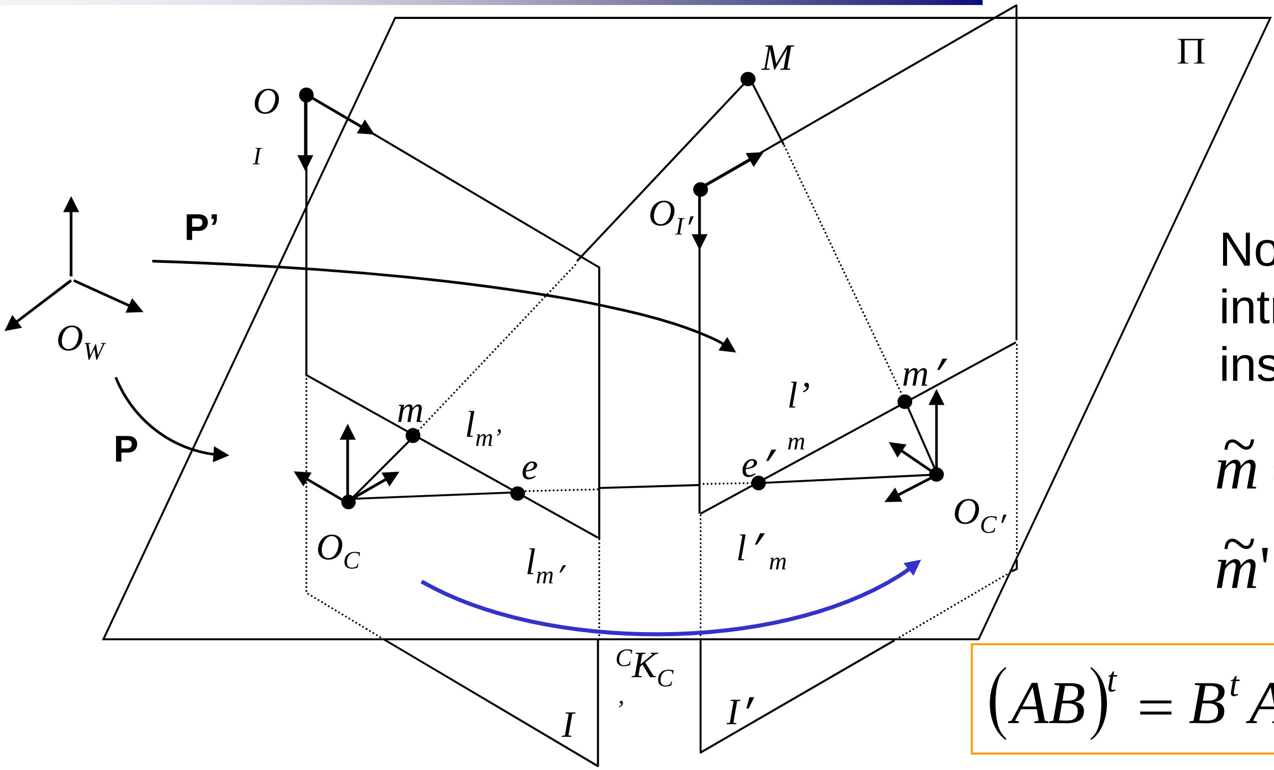
$$m \cdot l_{m'} = m^t l_{m'} = m^t [t]_x Rm'$$

$$0 = -m'^t \, R^t [t]_x m$$

$$0 = m^t [t]_x Rm'$$

Orthogonal, their cosinus is 0

3.4 Epipolar Geometry (VIII) - Modelling



Now we consider the intrinsics. Points in pixels instead of metrics

$$\tilde{m} = \mathbf{A} m \qquad m = \mathbf{A}^{-1} \tilde{m}$$

$$\tilde{m}' = \mathbf{A}' m' \qquad m' = \mathbf{A}'^{-1} \tilde{m}'$$

$$(AB)^t = B^t A^t$$

$$\left(A^{-1}\right)^t = \left(A^t\right)^{-1} = A^{-t}$$

$$0 = -m'^t R^t [t]_x m = \left(\mathbf{A}'^{-1} \tilde{m}'\right)^t R^t [t]_x \mathbf{A}^{-1} \tilde{m} = \tilde{m}'^t \mathbf{A}'^{-t} R^t [t]_x \mathbf{A}^{-1} \tilde{m}$$

$$0 = m^t [t]_x R m' = \left(\mathbf{A}^{-1} \tilde{m}\right)^t [t]_x R \mathbf{A}'^{-1} \tilde{m}' = \tilde{m}^t \mathbf{A}^{-t} [t]_x R \mathbf{A}'^{-1} \tilde{m}'$$

$$F = \mathbf{A}'^{-t} R^t [t]_x \mathbf{A}^{-1} \qquad \tilde{m}'^t F \tilde{m} = 0$$

$$F' = \mathbf{A}^{-t} [t]_x R \mathbf{A}'^{-1} \qquad \tilde{m}^t F' \tilde{m}' = 0$$

3.4 Epipolar Geometry (IX) - Modelling

F and F' are related by a transpose. So,

$$F = F'^t \qquad F = \mathbf{A}'^{-t} R^t [t]_x \mathbf{A}^{-1}$$

$$F' = F^t \qquad F' = \mathbf{A}^{-t} [t]_x R \mathbf{A}'^{-1}$$

Demonstration:

$$F^t = \left(\mathbf{A}'^{-t} R^t [t]_x \mathbf{A}^{-1}\right)^t = \mathbf{A}^{-t}\left(\mathbf{A}'^{-t} R^t [t]_x\right)^t = \mathbf{A}^{-t}[t]_x\left(\mathbf{A}'^{-t} R^t\right)^t = \mathbf{A}^{-t}[t]_x R \mathbf{A}'^{-1} = F'$$

$$F'^t = \left(\mathbf{A}^{-t}[t]_x R \mathbf{A}'^{-1}\right)^t = \mathbf{A}'^{-t}\left(\mathbf{A}^{-t}[t]_x R\right)^t = \mathbf{A}'^{-t} R^t \left(\mathbf{A}^{-t}[t]_x\right)^t = \mathbf{A}'^{-t} R^t [t]_x \mathbf{A}^{-1} = F$$

The same dissertation can be made assuming the origin at {C'}, obtaining two more fundamental matrices that are also related to F and F'.

3.4 Epipolar Geometry (X) - Modelling

The Essential Matrix is the calibrated case of the Fundamental matrix.

- The Intrinsic parameters are known: A and A' are known

The problem is reduced to estimate E or E'.

$$F = \mathbf{A}'^{-t} R^t [t]_x \mathbf{A}^{-1}$$

$$F' = \mathbf{A}^{-t} [t]_x R \mathbf{A}'^{-1}$$

$$E = R^t [t]_x$$

$$E' = [t]_x R$$

The monocular stereo is a symplified version of F where A = A', reducing the complexity of computing F.

$$F = \mathbf{A}^{-t} R^t [t]_x \mathbf{A}^{-1}$$

$$F' = \mathbf{A}^{-t} [t]_x R \mathbf{A}^{-1}$$

3.5 Computing F: The Eight Point Method

The epipolar geometry is defined as:

$$m^T \mathbf{F}' m' = 0 \qquad \begin{bmatrix} x_i & y_i & 1 \end{bmatrix} \mathbf{F}' \begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix} = 0$$

Operating, we obtain:

$$U_n f = 0$$

$$U_n = (u_1, u_2, \ldots, u_n)$$

$$u_i = (x_i' x_i, y_i' x_i, x_i, x_i' y_i, y_i' y_i, y_i, x_i', y_i', 1)$$

$$f = (F_{11}, F_{12}, F_{13}, F_{21}, F_{22}, F_{23}, F_{31}, F_{32}, F_{33})^t$$

3.5 Computing F: The Eight Point Method and LS

$$U_n f = 0$$

First solution is : $f = 0$ NOT WANTED

F is defined up to a scale factor, so we can fix one of the component to 1.
Let's fix $\mathbf{F_{33} = 1}$.

$$U'_n f' = -1_n$$

$$U'_n = (u'_1, u'_2, \ldots, u'_n)$$

$$u'_i = (x'_i x_i, y'_i x_i, x_i, x'_i y_i, y'_i y_i, y_i, x'_i, y'_i)$$

$$f = (F_{11}, F_{12}, F_{13}, F_{21}, F_{22}, F_{23}, F_{31}, F_{32})^t$$

Then:

$$U_n'^{-1} U'_n f' = -U_n'^{-1} 1_n$$

$$f' = -U_n'^{-1} 1_n \quad \Longrightarrow \quad f' = -\left(U_n'^t U'_n\right)^{-1} U_n'^t 1_n$$

Least-Squares

3.5 Computing F: The Eight Point Method and Eigen Analysis

$$U_n f = 0$$

First solution is : $f = 0$ NOT WANTED

F has to be rank-2 because $[t_x]$ is rank-2.

$$F = \mathbf{A}'^{-t} R^t [t]_x \mathbf{A}^{-1} \qquad [t]_x = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix}$$

Any system of equations:

$$U_n f = 0 \qquad f = (F_{11}, F_{12}, F_{13}, F_{21}, F_{22}, F_{23}, F_{31}, F_{32}, F_{33})^t$$

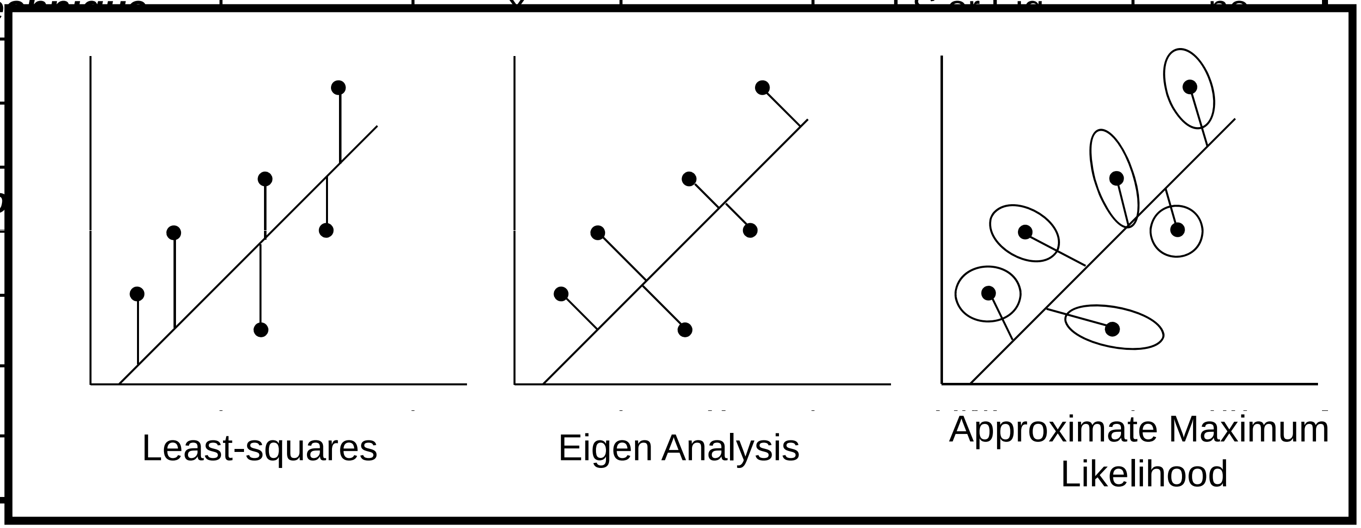
can be **solved by SVD** so that f lies in the nullspace of $U_n = UDV^T$.

[U,D,V] = svd (U_n)

Hence f corresponds to a multiple of the column of V that belongs to the unique singular value of D equal to 0.
Note that f is only known up to a scaling factor.

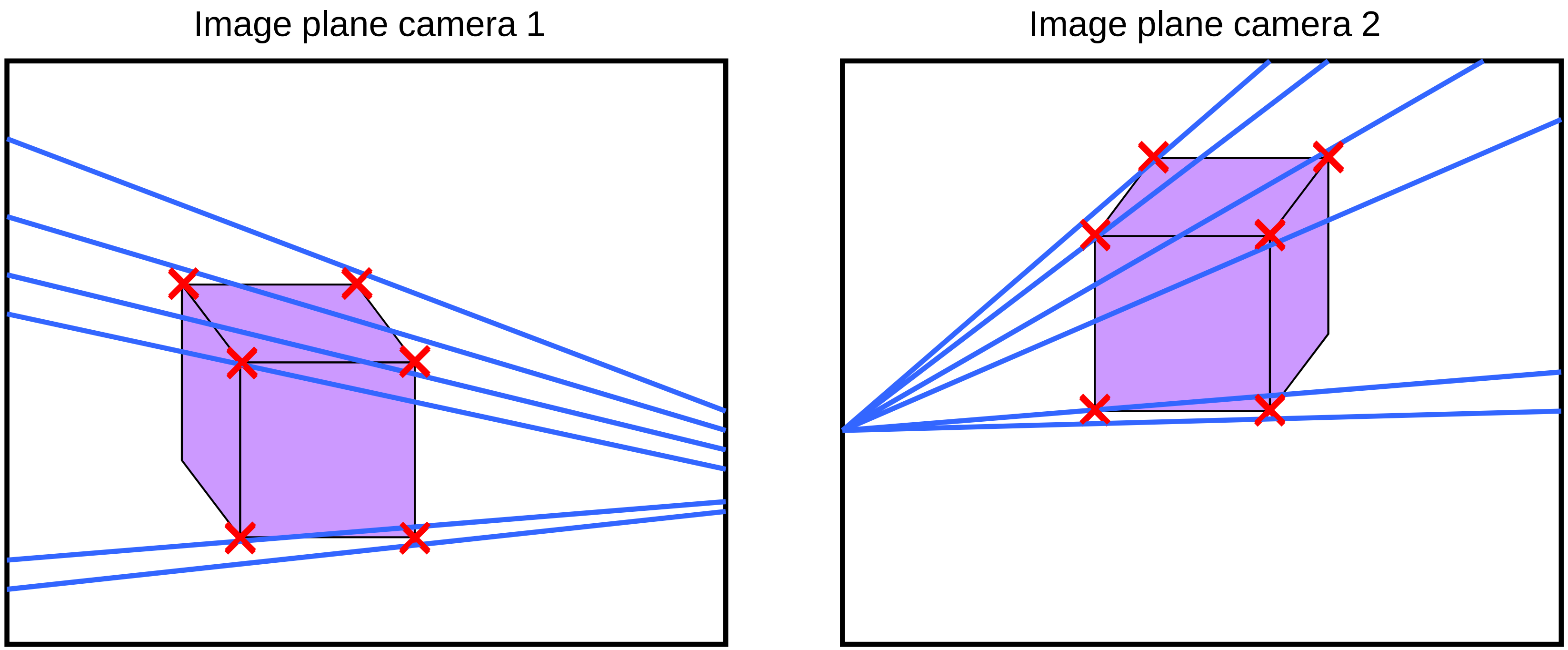
3.5 Computing the Fundamental Matrix: A Survey

	Linear	*Iterative*	*Robust*	*Optimisation*	*Rank-2*
Seven point (7p)	X			—	yes
Eight point (8p)	X			LS or Eig.	no
Rank-2 constraint	X			LS	yes
Iterative Newton-Raphson		X		LS	no
Linear iterative		X		LS	no
Non-linear minimization in parameter space		X		Eig.	yes
Gradient te		X		LS or Eig	no
FNS					
CFNS					
M-Estimato					
LMedS					
RANSAC					
MLESAC					
MAPSAC					



LS: Least-Squares   Eig: Eigen Analysis   AML: Approximate Maximum Likelihood

3.6 Accuracy Evaluation



3.7 Experimental Results: Synthetic Images (I)

Linear methods: Good results if the points are well located and no outilers

Methods*	Linear			
	1	2	3	4
$\sigma = 0.0$ outliers 0%	14.250 13.840	0.000 0.000	0.000 0.000	1.920 1.143
$\sigma = 0.0$ outliers 10%	25.370 48.428	339.562 433.013	17.124 31.204	30.027 59.471
$\sigma = 0.1$ outliers 0%	135.775 104.671	1.331 0.788	0.107 0.088	0.120 0.091
$\sigma = 0.1$ outliers 10%	140.637 104.385	476.841 762.756	19.675 46.505	70.053 63.974
$\sigma = 0.5$ outliers 0%	163.839 178.222	5.548 3.386	0.538 0.362	0.642 0.528
$\sigma = 0.5$ outliers 10%	140.932 109.427	507.653 1340.808	19.262 49.243	26.475 54.067
$\sigma = 1.0$ outliers 0%	65.121 58.184	21.275 12.747	1.065 0.744	1.319 0.912
$\sigma = 1.0$ outliers 10%	128.919 100.005	429.326 633.019	21.264 53.481	61.206 64.583

mean
std

* Mean and Std. in pixels

Methods: 1.- 7-Point; 2.- 8-Point with Least-Squares;
3.- 8-Point with Eigen Analysis 4.- Rank-2 Constraint

3.7 Experimental Results: Synthetic Images (I)

Iterative methods: Can cope with noise but inefficient in the presence of outliers

Methods*	Iterative						
	5	6	7	8	9	10	11
$\sigma = 0.0$ outliers 0%	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000
$\sigma = 0.0$ outliers 10%	161.684 117.494	20.445 30.487	∞ ∞	187.474 197.049	18.224 36.141	17.124 31.204	16.978 29.015
$\sigma = 0.1$ outliers 0%	1.328 0.786	0.107 0.088	1.641 0.854	1.328 0.786	0.112 0.092	0.107 0.088	0.110 0.091
$\sigma = 0.1$ outliers 10%	158.961 124.202	32.765 67.308	146.955 94.323	183.961 137.294	15.807 40.301	14.003 38.485	14.897 39.388
$\sigma = 0.5$ outliers 0%	5.599 3.416	0.538 0.361	7.017 3.713	5.590 3.410	0.554 0.361	0.538 0.362	0.543 0.368
$\sigma = 0.5$ outliers 10%	161.210 136.828	31.740 59.126	∞ ∞	217.577 368.061	19.409 51.154	22.302 59.048	22.262 59.162
$\sigma = 1.0$ outliers 0%	20.757 12.467	1.068 0.772	345.123 294.176	21.234 12.719	1.071 0.745	1.065 0.744	1.066 0.748
$\sigma = 1.0$ outliers 10%	158.849 120.461	37.480 52.762	∞ ∞	152.906 120.827	18.730 38.644	18.374 39.993	19.683 42.112

Methods: 5.- Iterative Linear; 6.- Iterative Newton-Raphson;
7.- Minimization in parameter space;
8.- Gradient using LS; 9.- Gradient using Eigen;
10.- FNS; 11.- CFNS

* Mean and Std. in pixels

3.7 Experimental Results: Synthetic Images (I)

Robust methods: Cope with both noise and outliers

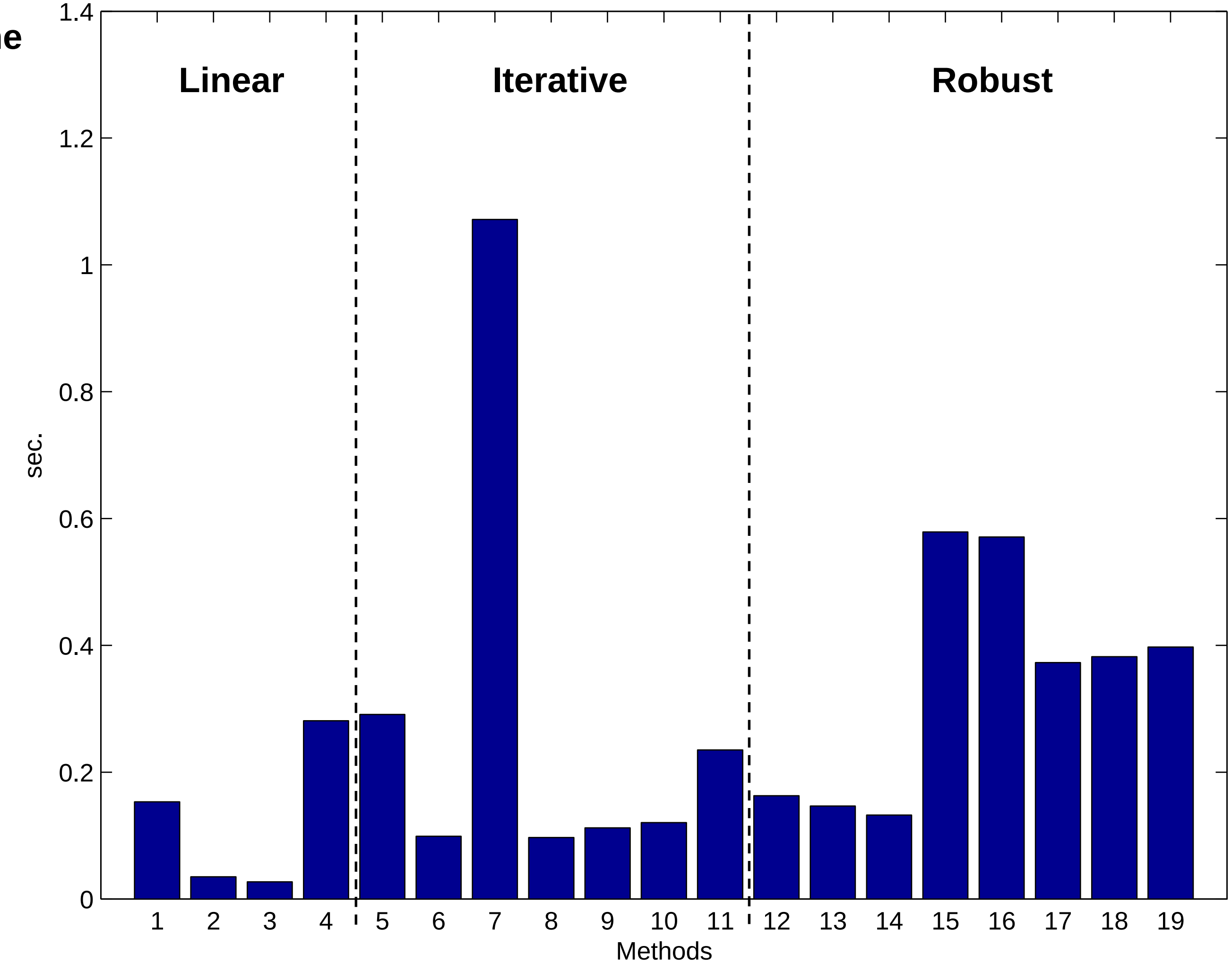
Methods	Robust							
	12	**13**	**14**	**15**	**16**	**17**	**18**	**19**
$\sigma = 0.0$ outliers 0%	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000	0.100 0.079	0.011 0.009
$\sigma = 0.0$ outliers 10%	273.403 360.443	4.909 4.493	4.714 2.994	0.000 0.000	0.000 0.000	16.457 26.923	19.375 70.160	0.115 0.115
$\sigma = 0.1$ outliers 0%	0.355 0.257	0.062 0.042	0.062 0.041	1.331 0.788	0.107 0.088	0.107 0.088	0.139 0.123	0.168 0.155
$\sigma = 0.1$ outliers 10%	73.354 59.072	4.876 4.808	4.130 2.997	0.449 0.271	0.098 0.077	2.389 5.763	21.784 97.396	0.701 0.740
$\sigma = 0.5$ outliers 0%	2.062 1.466	0.392 0.237	0.367 0.207	5.548 3.386	0.538 0.362	0.538 0.362	0.550 0.377	0.762 0.618
$\sigma = 0.5$ outliers 10%	143.442 111.694	3.887 3.969	3.147 2.883	47.418 29.912	0.586 0.434	18.942 53.098	23.859 79.890	0.629 0.452
$\sigma = 1.0$ outliers 0%	8.538 6.306	0.794 0.463	0.814 0.463	21.275 12.747	1.065 0.744	1.065 0.744	1.089 0.768	1.072 0.785
$\sigma = 1.0$ outliers 10%	120.012 122.436	3.921 3.752	4.089 4.326	25.759 15.217	1.052 0.803	14.076 30.274	19.298 65.149	1.041 0.822

Methods: 12.- M-Estimator using LS; 13.- M-Estimator using Eigen;
14.- M-Estimator proposed by Torr;
15.- LMedS using LS; 16.- LMedS using Eigen;
17.- RANSAC; 18.- MLESAC; 19.- MAPSAC.

* Mean and Std. in pixels

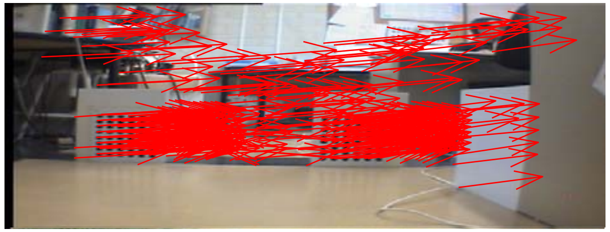
3.7 Experimental Results: Synthetic Images (II)

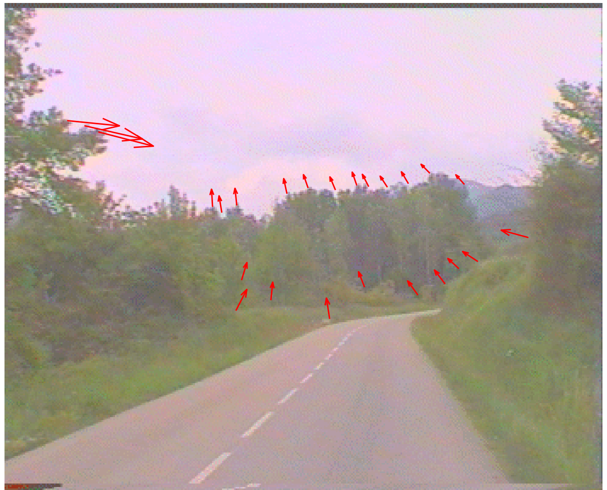
Computing Time

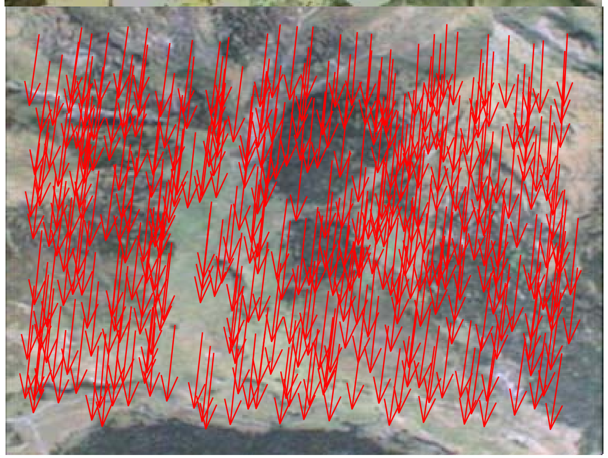


1.- 7-Point; 2.- 8-Point with Least-Squares; 3.- 8-Point with Eigen Analysis; 4.- Rank-2 Constraint; 5.- Iterative Linear; 6.- Iterative Newton-Raphson; 7.- Minimization in parameter space; 8.- Gradient using LS; 9.- Gradient using Eigen; 10.- FNS; 11.- CFNS; 12.- M-Estimator using LS; 13.- M-Estimator using Eigen; 14.- M-Estimator proposed by Torr; 15.- LMedS using LS; 16.- LMedS using Eigen; 17.- RANSAC; 18.- MLESAC; 19.- MAPSAC.

3.7 Experimental Results: Real Images (I)

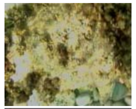






3.7 Experimental Results: Real Images (II)






Methods*	Robust							
	12	13	14	15	16	17	18	19
Urban Scene	1.668 0.935	0.309 0.228	0.279 0.189	1.724 1.159	0.319 0.269	0.440 0.334	0.449 0.373	0.440 0.348
Mobile Robot Scene	5.775 50.701	0.274 0.192	0.593 0.524	24.835 38.434	1.559 2.715	3.855 6.141	2.443 5.629	1.274 2.036
Underwater Scene	0.557 0.441	0.650 0.629	0.475 0.368	2.439 2.205	0.847 0.740	1.725 2.138	3.678 12.662	1.000 0.761
Road Scene	0.373 0.635	0.136 0.113	0.310 0.256	0.825 1.144	0.609 0.734	0.609 0.734	0.427 0.410	0.471 0.403
Aerial Scene	0.099 0.063	0.085 0.058	0.161 0.106	0.179 0.158	0.149 0.142	0.149 0.142	0.216 0.186	0.257 0.197
Kitchen Scene	0.584 0.425	0.280 0.207	0.263 0.191	1.350 1.200	0.545 0.686	2.623 3.327	0.864 3.713	0.582 0.717

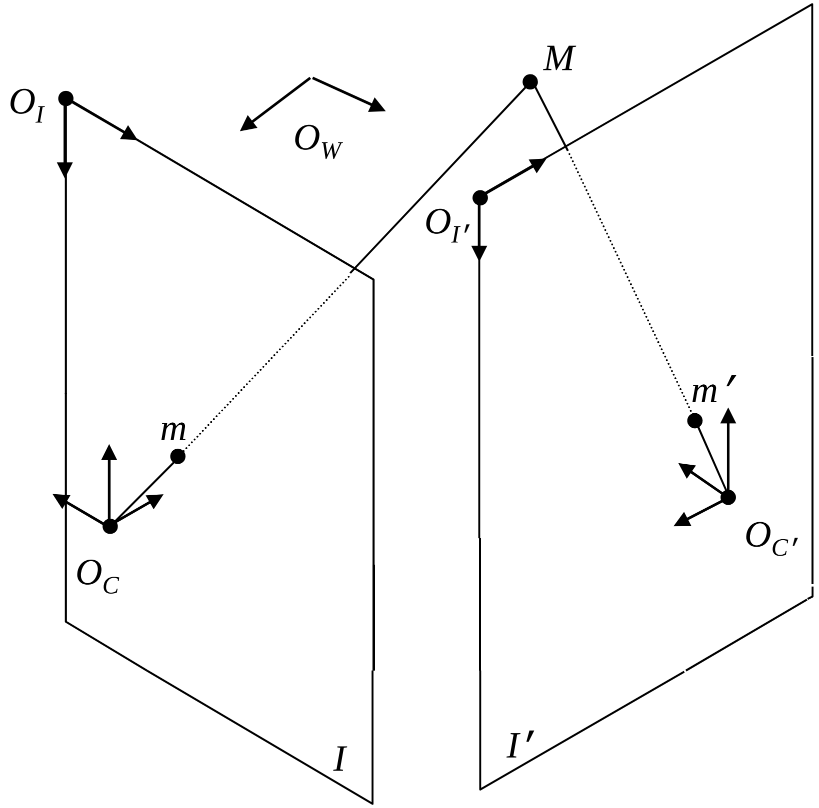
Methods: 12.- M-Estimator using LS; 13.- M-Estimator using Eigen;
14.- M-Estimator proposed by Torr;
15.- LMedS using LS; 16.- LMedS using Eigen;
17.- RANSAC; 18.- MLESAC; 19.- MAPSAC.

* Mean and Std. in pixels

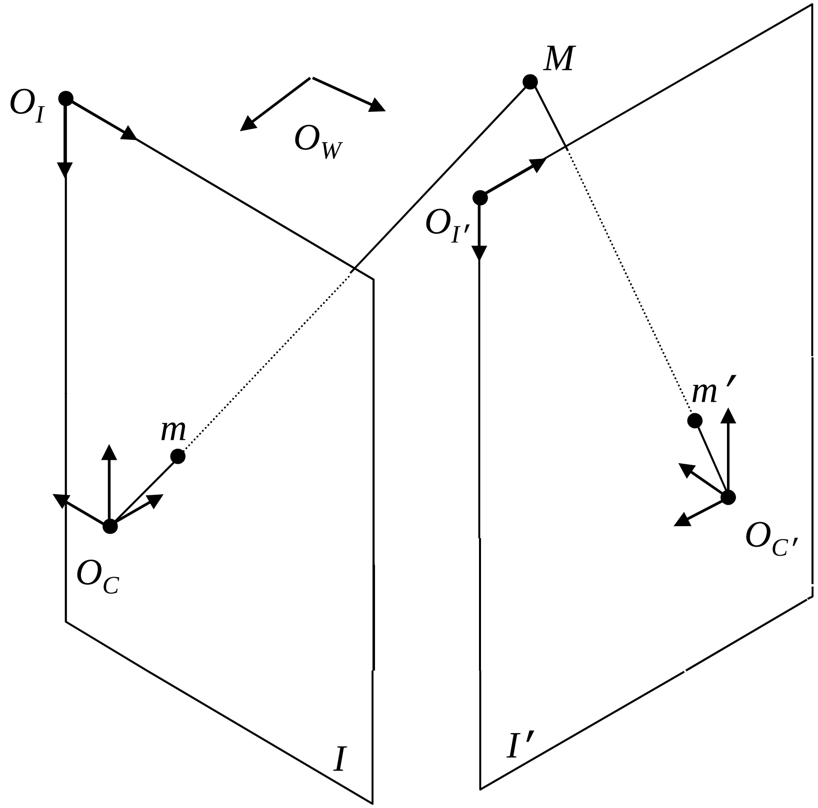
3.8 Calibrated reconstruction

Calibrated 3D Reconstruction process:

1. The **Internal Parameters** are known (by camera calibration)
2. Calculate the **Fundamental Matrix**
3. Determine the **External Parameters** (rotation and translation from one camera to the other) from the Fundamental Matrix
4. Determine **3D point locations,** i.e. perform the **3D reconstruction.**



3.8 Calibrated reconstruction



Determine the External Parameters:

1. Determine the **Essential matrix** E from the fundamental matrix F and the camera calibration matrix K, since we know the Intrinsic Parameters of the camera.

$$F = \mathbf{A}'^{-t} R^t [t]_x \mathbf{A}^{-1} \qquad E = R^t [t]_x$$

$$F' = \mathbf{A}^{-t} [t]_x R \mathbf{A}'^{-1} \qquad E' = [t]_x R$$

2. Calculate the **External Parameters**: **R** and **t**

   a) SVD of E: E = USV^T

R = UWV^T or UW^TV^T

t = u3 or -u3, where u3 is the last column of U

There are 4 combinations of translations and rotations.

3.8 Calibrated reconstruction

Determine the External Parameters:

1. Determine the **Essential matrix** E from the fundamental matrix F and the camera calibration matrix K, since we know the Intrinsic Parameters of the camera.

$$F = \mathbf{A}'^{-t} R^t [t]_x \mathbf{A}^{-1} \qquad E = R^t [t]_x$$

$$F' = \mathbf{A}^{-t} [t]_x R\mathbf{A}'^{-1} \qquad E' = [t]_x R$$

$$s\,{}^{I}m = {}^{I}A_C\,{}^{C}K_W\,{}^{W}M$$

$$s'\,{}^{I'}m' = {}^{I'}A'_{C'}\,{}^{C'}K'_{W'}\,{}^{W}M$$

2D image coordinates

Unknown 3D coordinates

2. Calculate the **External Parameters**: **R** and **t**

   a) SVD of E: E = USV^T

$R = UWV^T$ or UW^TV^T

t = u3 or -u3, where u3 is the last column of U

There are 4 combinations of translations and rotations.

The correct pair will produce points **in front of both cameras**.

3.10 Stereo vision using ...

White Box model

Explicit model: the physical parameters of the camera are known

As close as possible to a full description of the real system

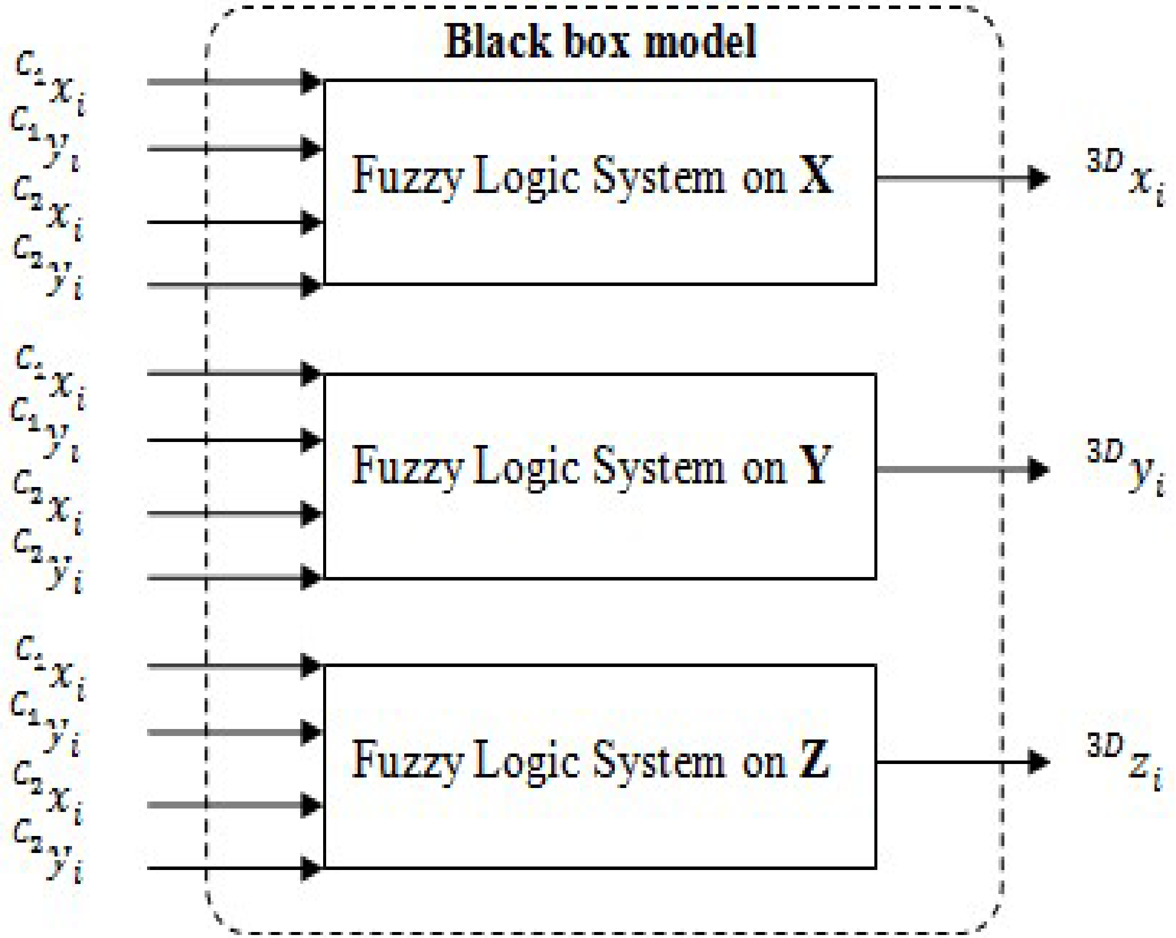
Black Box model

Implicit model: emulates the camera behavior without actually knowing the camera parameters.

A set of transfer functions and parameters that do not describe any internal physics.

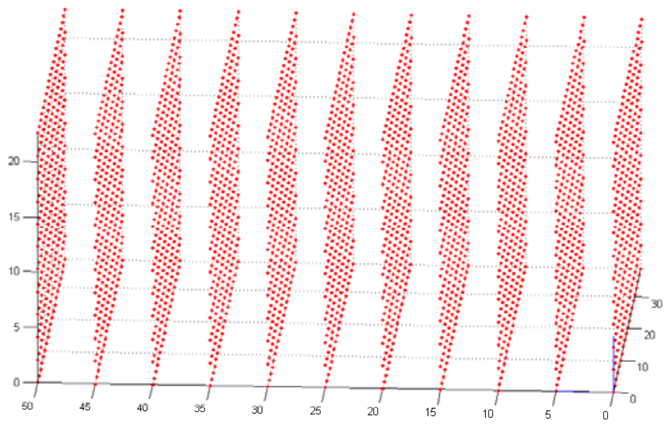
- Do you think that is possible to build a black box model of a stereo configuration of cameras?
- What are the inputs? / Outputs?

3.10 Stereo vision using fuzzy systems

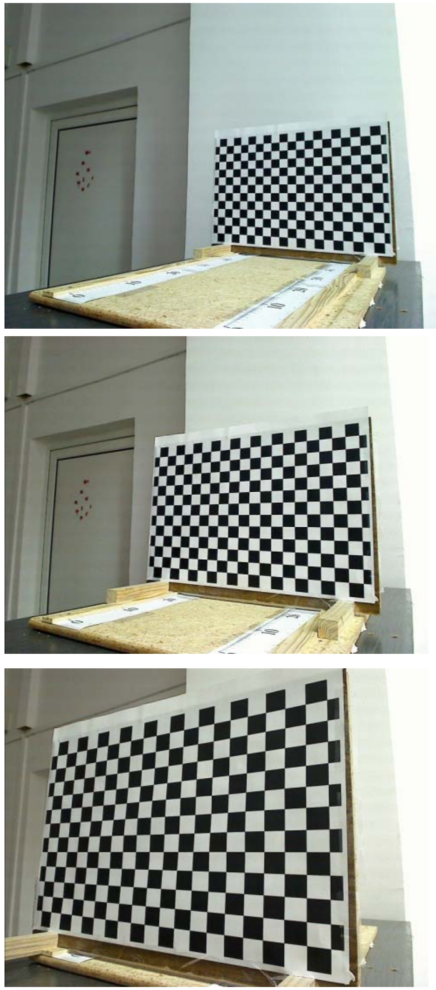


3.10 Stereo vision using fuzzy systems

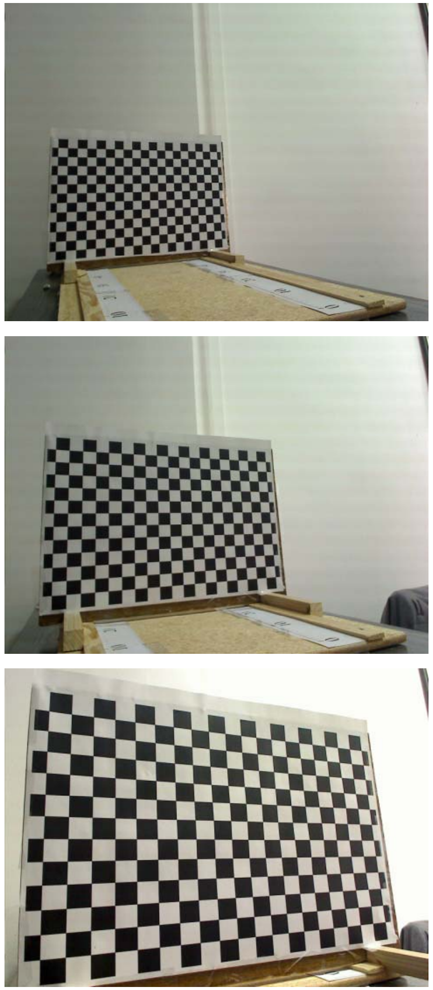
3D points



Camera 1



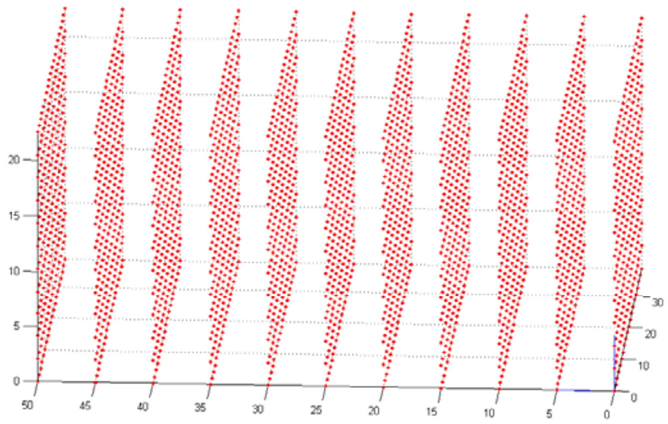
Camera 2

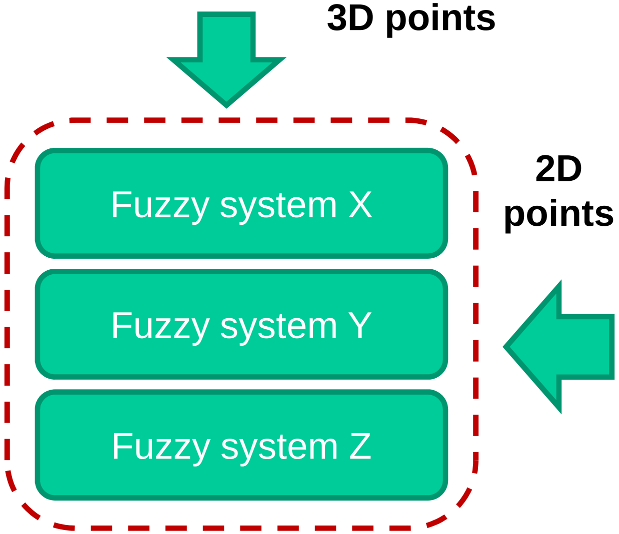


The 3D points are calculated taking as origin of the coordonate system the lower left corner of the calibration pattern.

The pattern is moved at different known distances from the camera and the stereo images are captured.
The 2D points are identified and matched in all the images.

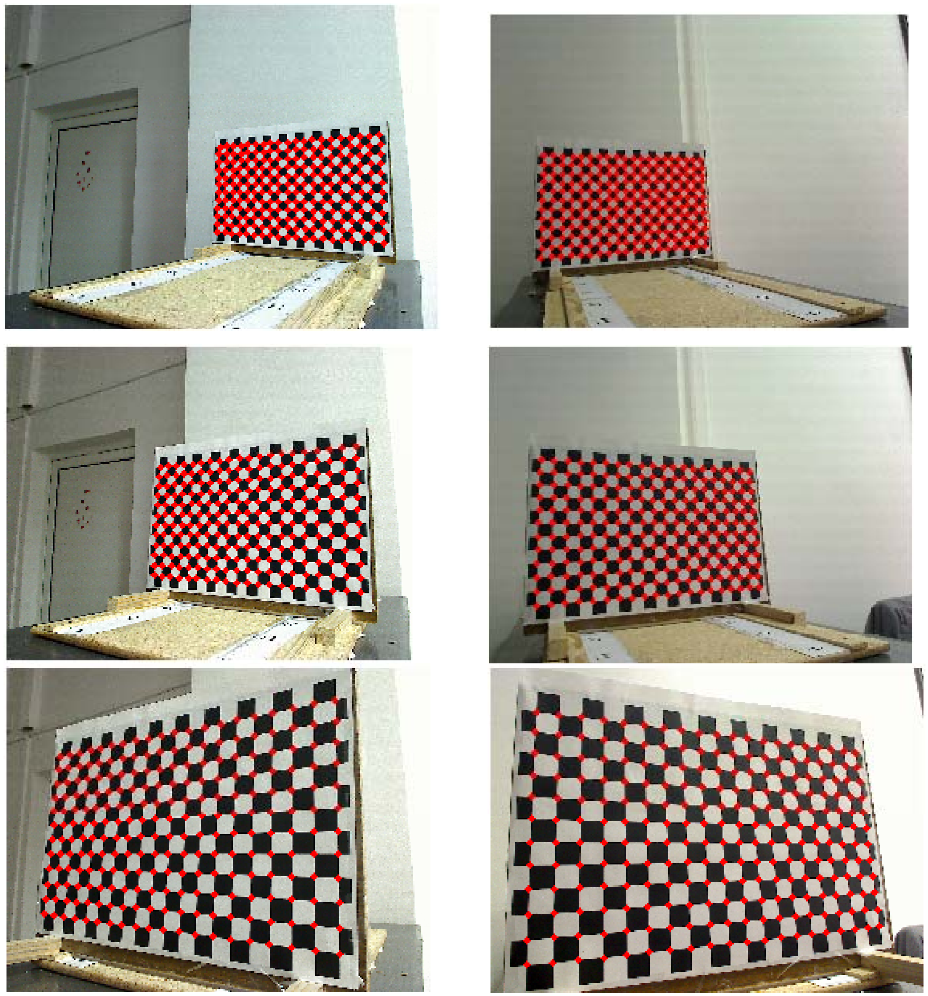
3.10 Stereo vision using fuzzy systems



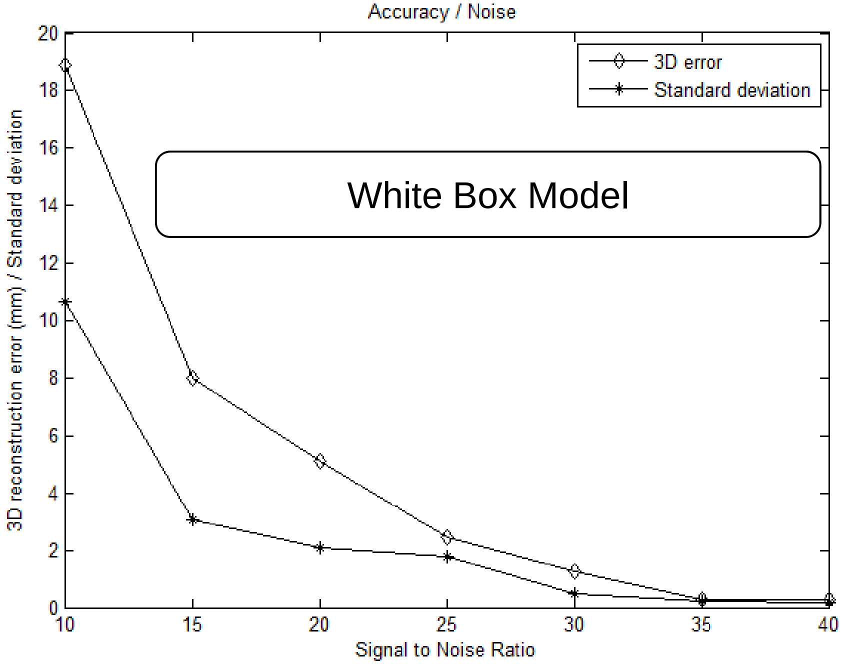


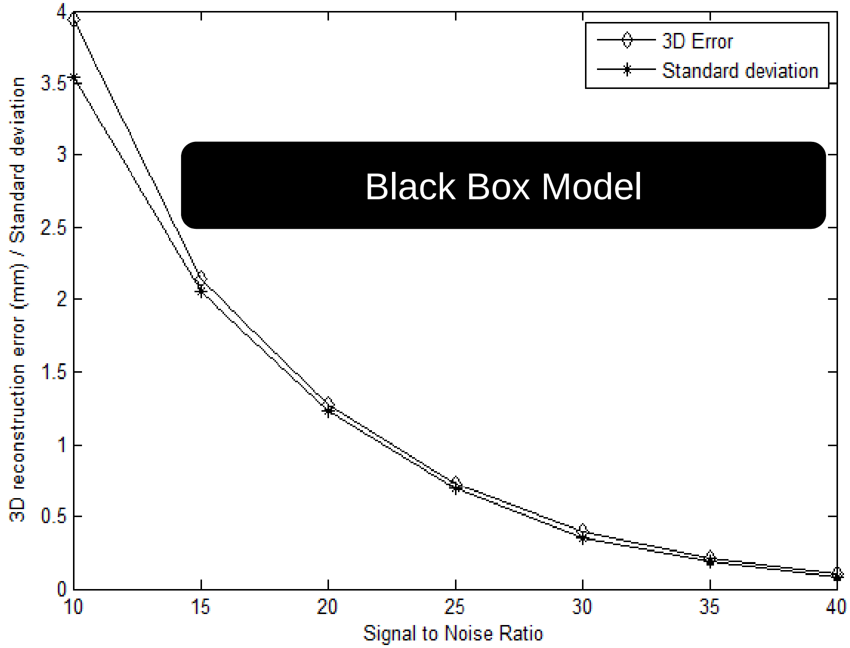
Calibration of the fuzzy system using the 3D and 2D points

Camera 1 **Camera 2**



3.10 Stereo vision using fuzzy systems





Features	Crisp (White Box)	Fuzzy (Black Box)
Robustness to noise	Low	High
Accuracy outside of a bounding box	High	Low
Model simplicity	Low	High
Adaptability to new camera configurations	Low	High

End of the SSIP presentation

Thank you for your time.

Any questions?