Visual Perception

Lecture 4

Reconstruction from two views

Joaquim Salvi
Universitat de Girona

Contents

Contents

4.1 Shape from *X*

Techniques based on:

- Modifying the intrinsic camera parameters

  i.e. <u>Depth from Focus/Defocus</u> 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 from Texture and Shape from Geometric Constraints

- Multiple views

  i.e. Shape from Stereo and Shape from Motion

4.1 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 fro om Geometric Constraints

- Multiple views

  i.e. Shape from Stereo and Shape from Motion

Shape from Structured Light

4.1 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

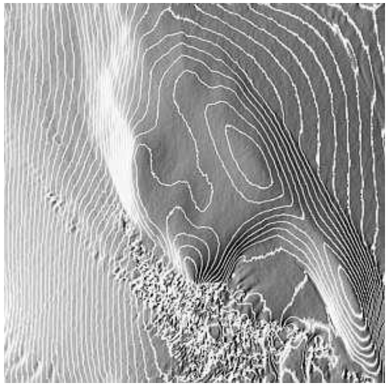
i.e. Shape from Structured Light and Shape from Photometric Stereo

– Considering additional surface information

i.e. Shape from Shading, Shape from Texture and Shape from Geometric Constr s

– Multiple views

i.e. Shape from Stereo and Shape



Shape from Shading

4.1 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 Constrai

- Multiple views

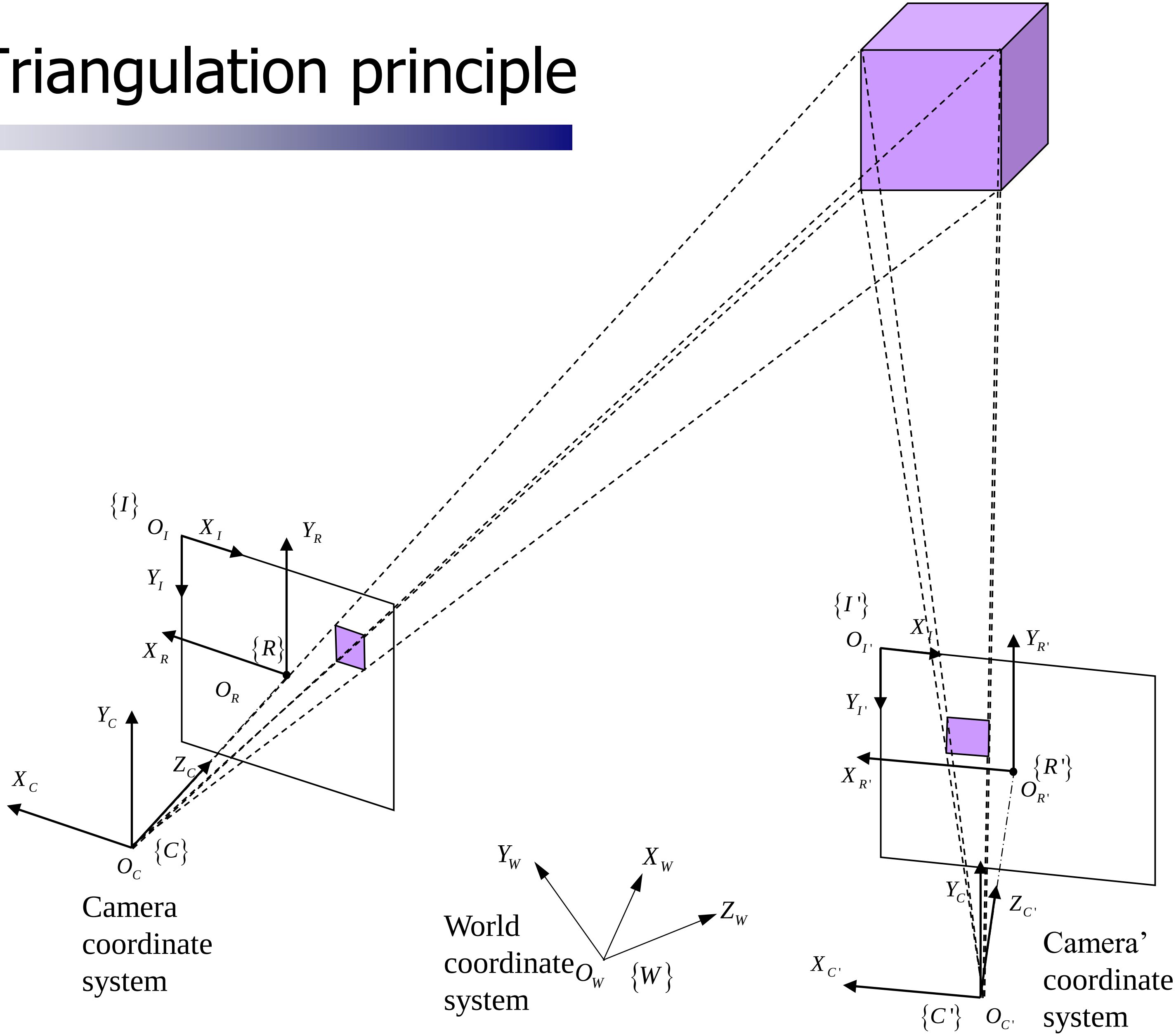
  i.e. Shape from Stereo and Shape from Motion

Shape from Stereo

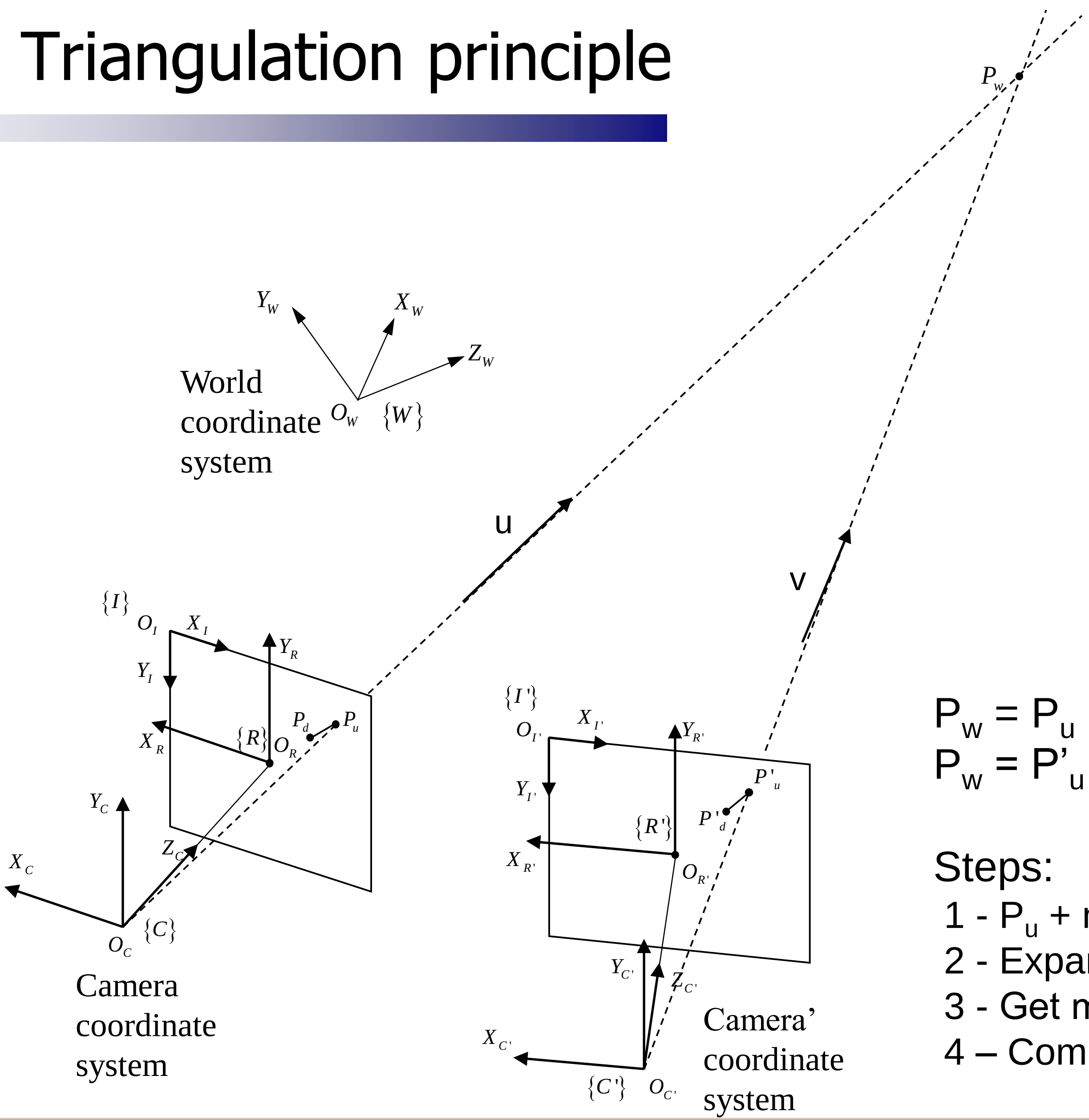
Contents

Contents

4.2 Triangulation principle



4.2 Triangulation principle

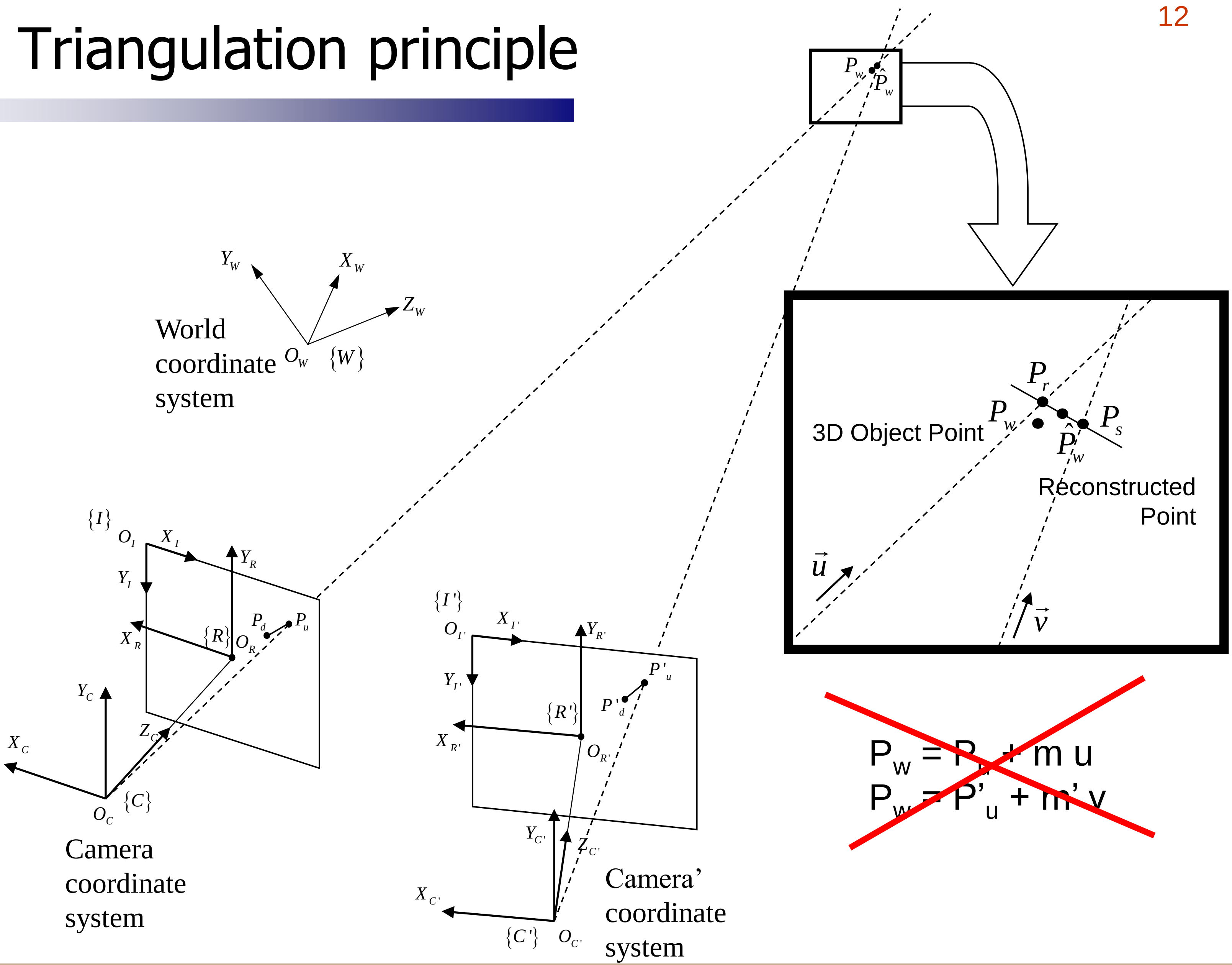


$$P_w = P_u + m\,u$$
$$P_w = P'_u + m'\,v$$

Steps:

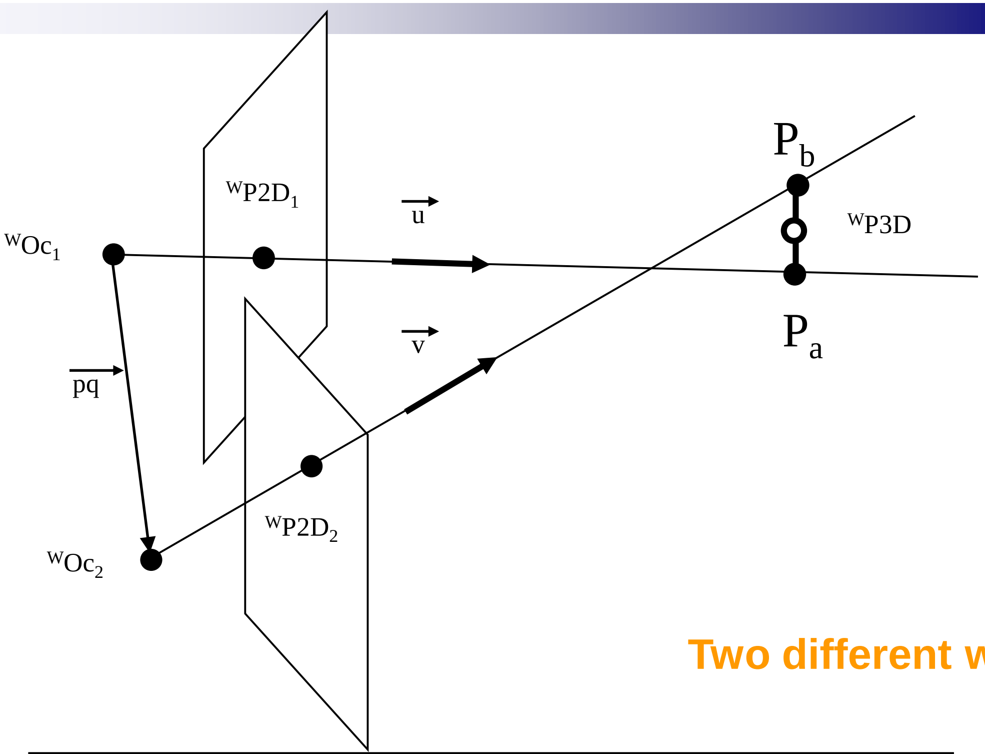
1 - $P_u + m\,u = P'_u + m'\,v$
2 - Expand to x,y,z
3 - Get m and m'
4 – Compute Pw

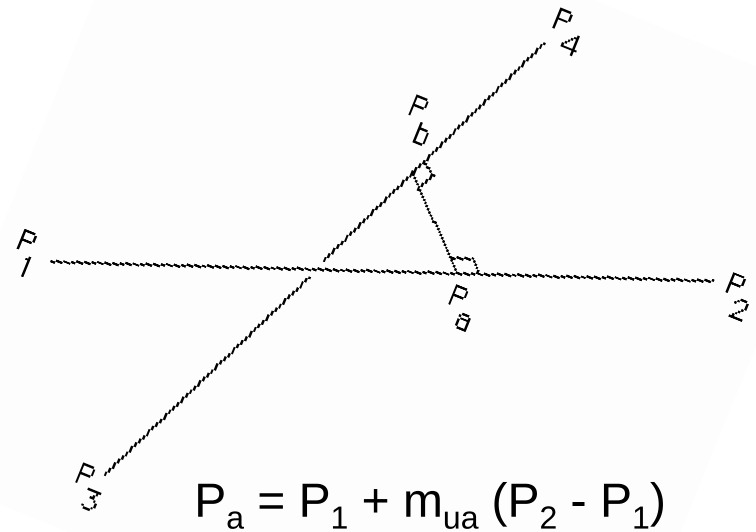
4.2 Triangulation principle



$$\cancel{P_w = P_u + m\,u}$$
$$\cancel{P_w = P'_u + m'\,v}$$

4.2 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/

4.2 Triangulation principle

In practice we can use Least-Squares:

$$\begin{bmatrix} s_1 u_1 \\ s_1 v_1 \\ s_1 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \qquad \begin{bmatrix} s_2 u_2 \\ s_2 v_2 \\ s_2 \end{bmatrix} = \begin{bmatrix} B_{11} & B_{12} & B_{13} & B_{14} \\ B_{21} & B_{22} & B_{23} & B_{24} \\ B_{31} & B_{32} & B_{33} & B_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$C = QX$$

$$X = Q^{-1} C$$

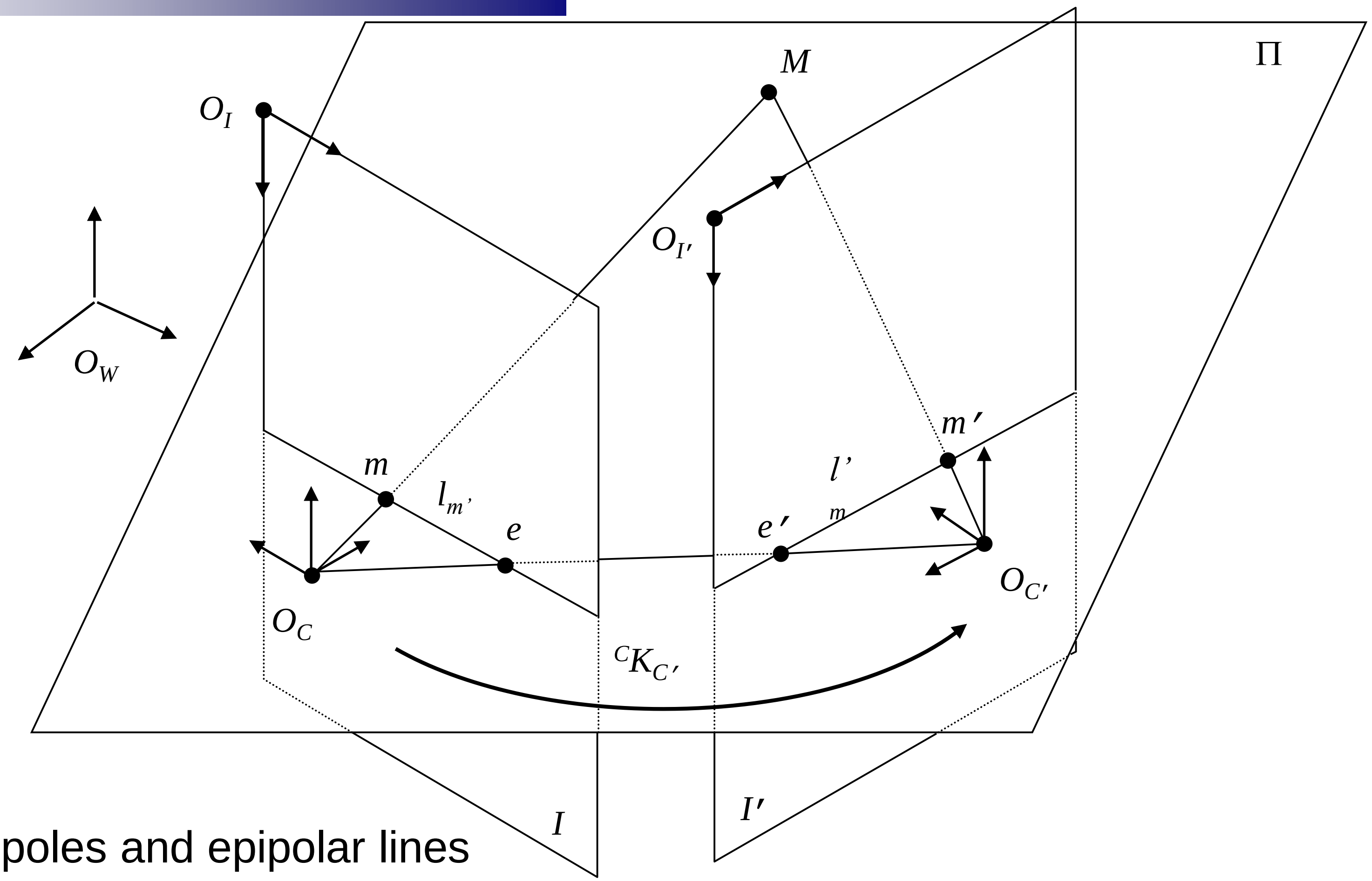
$$\begin{bmatrix} A_{14} - A_{34} u_1 \\ A_{24} - A_{34} v_1 \\ B_{14} - B_{34} u_2 \\ B_{24} - B_{34} v_2 \end{bmatrix} = \begin{bmatrix} A_{31} u_1 - A_{11} & A_{32} u_1 - A_{12} & A_{33} u_1 - A_{13} \\ A_{31} v_1 - A_{21} & A_{32} v_1 - A_{22} & A_{33} v_1 - A_{23} \\ B_{31} u_2 - B_{11} & B_{32} u_2 - B_{12} & B_{33} u_2 - B_{13} \\ B_{31} v_2 - B_{21} & B_{32} v_2 - B_{22} & B_{33} v_2 - B_{23} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Add additional rows if we have additiional views of the same point

Contents

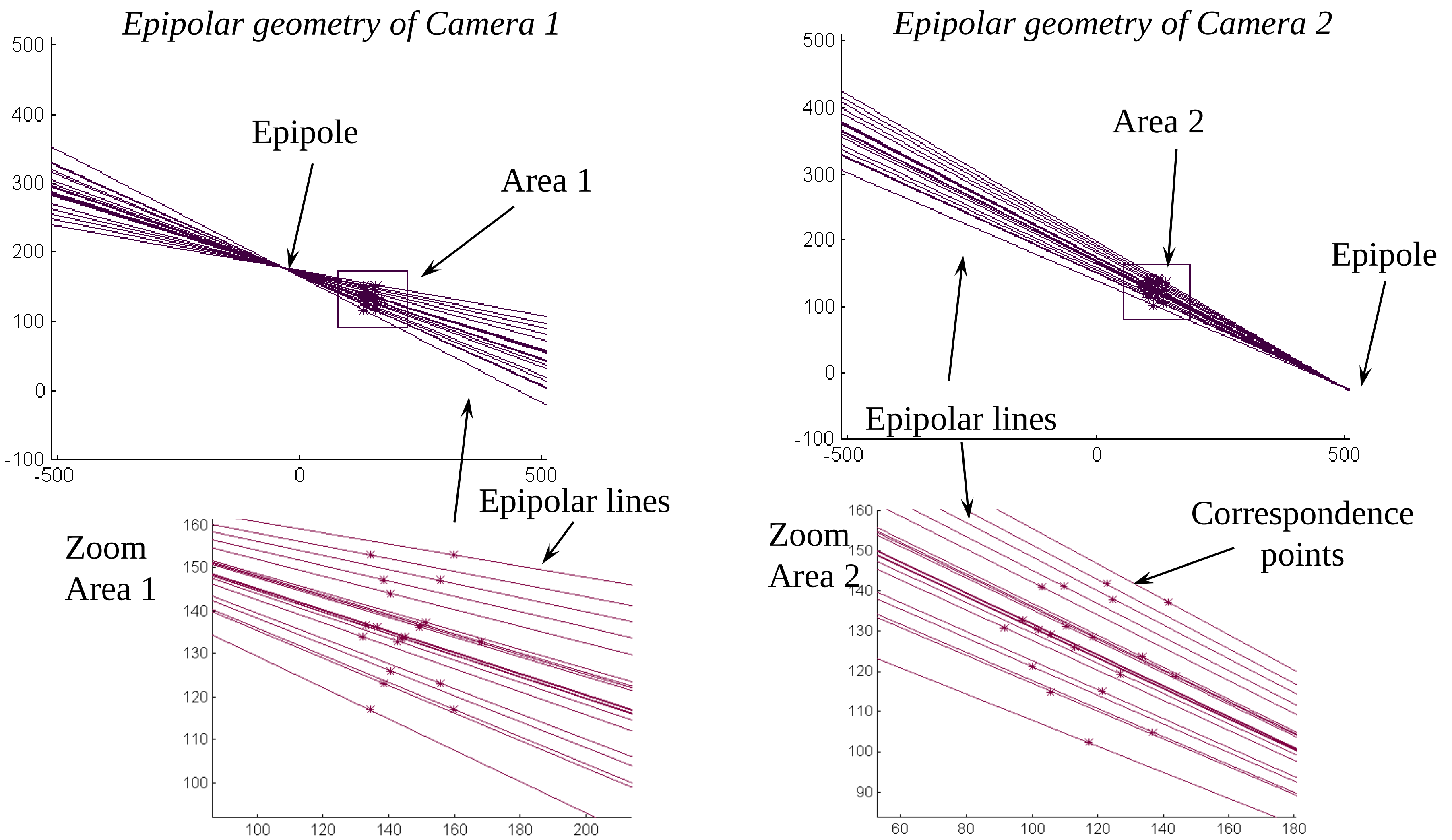
Contents

4.3 Epipolar Geometry – Modelling

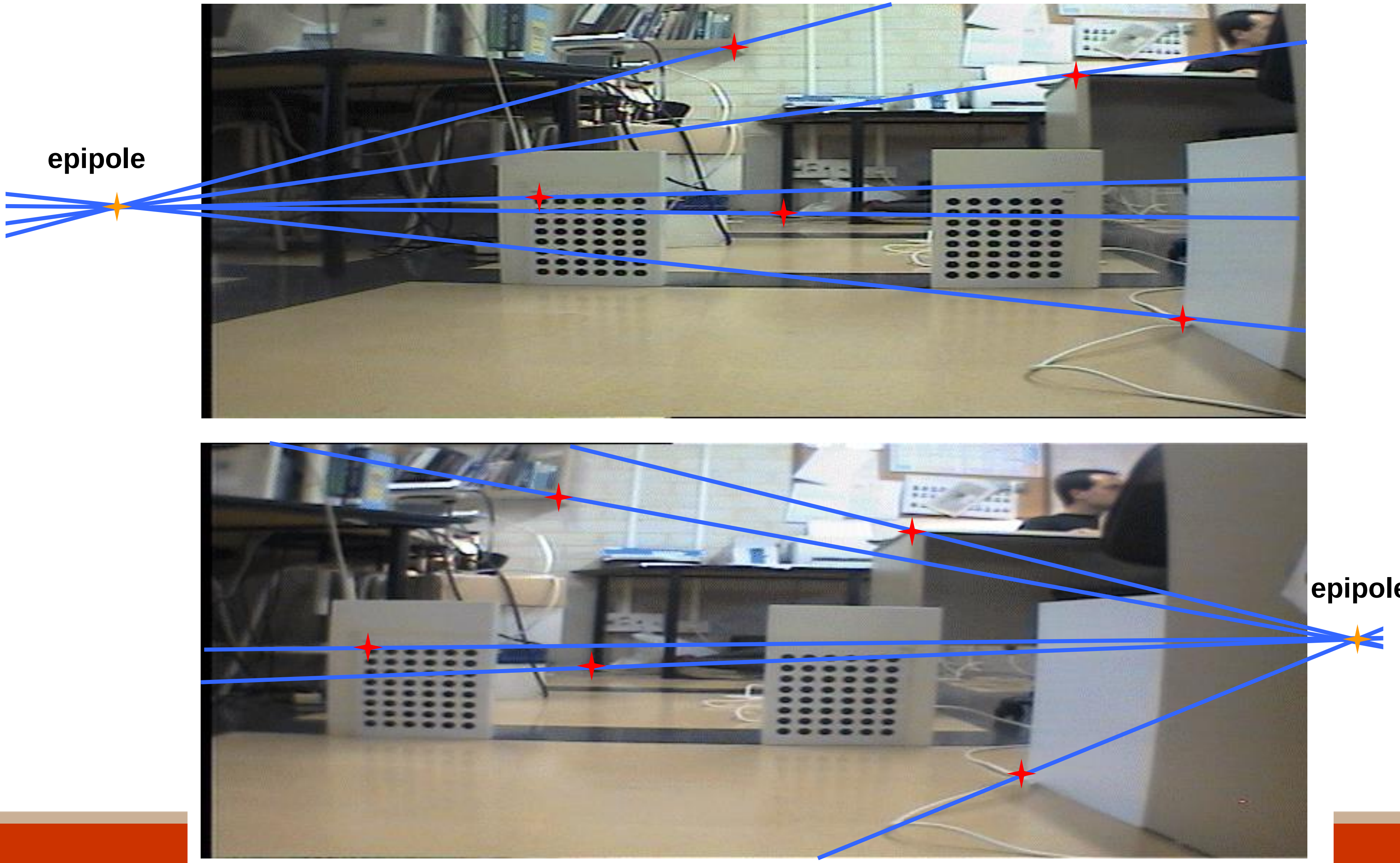


- Focal points, epipoles and epipolar lines
- e is defined by $O_{C'}$ in {I}, e’ is defined by O_C in {I’}
- m defines an epipolar line in {I’}; m’ defines an epipolar line in {I}
- All epipolar lines intersect at the epipole

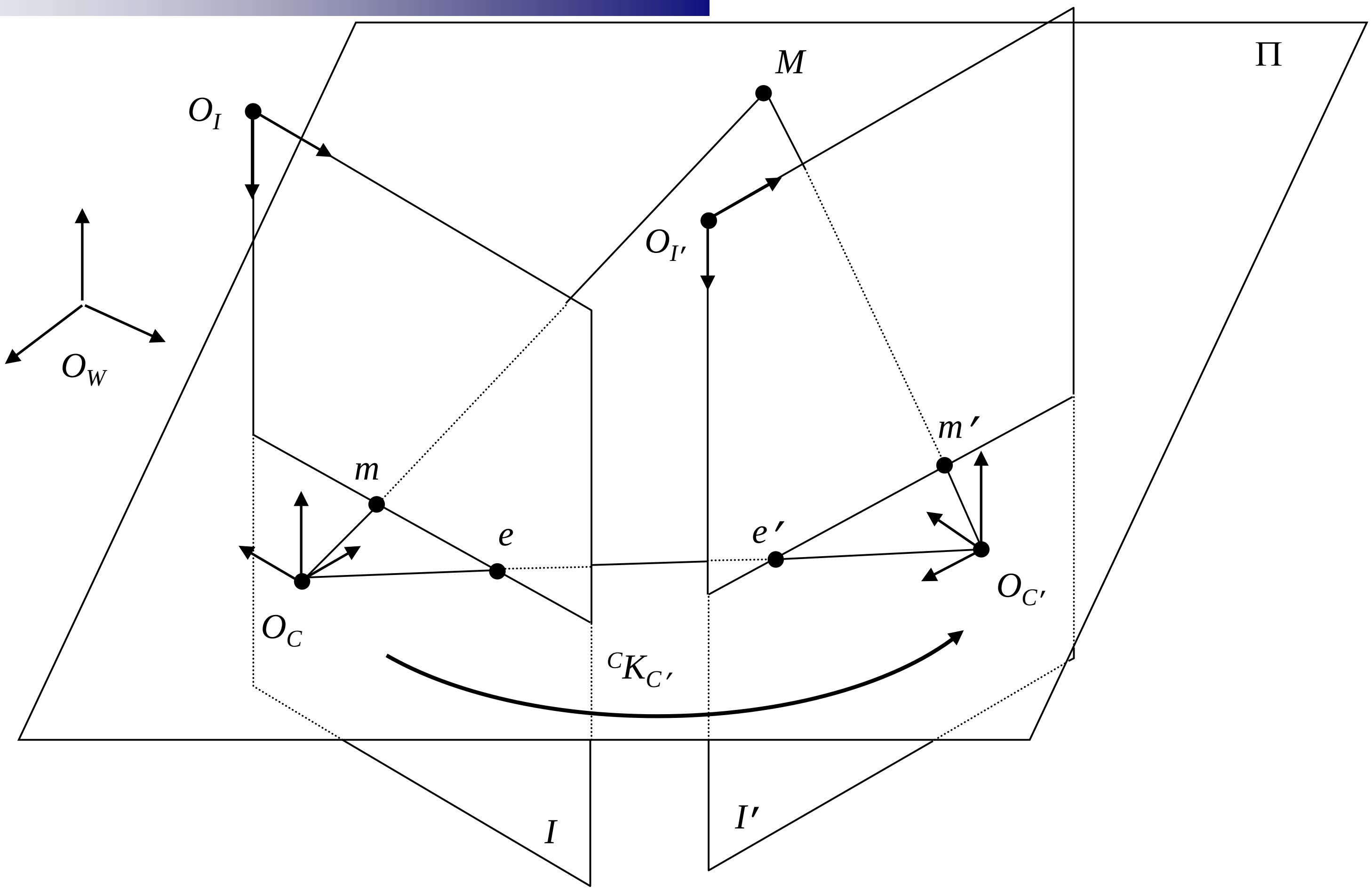
4.3 Epipolar Geometry – Modelling



4.3 Epipolar Geometry – Modelling

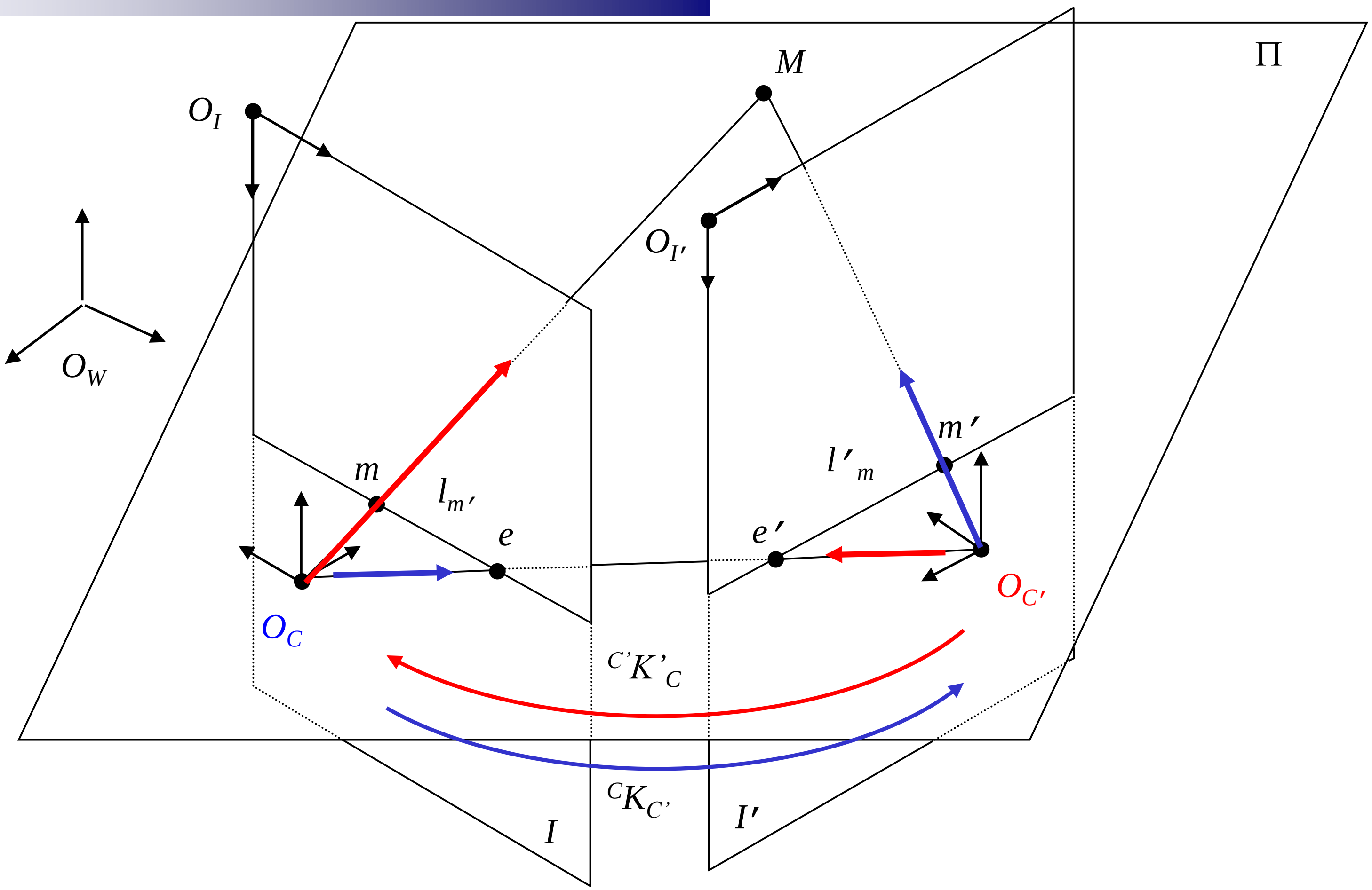


4.3 Epipolar Geometry – Modelling



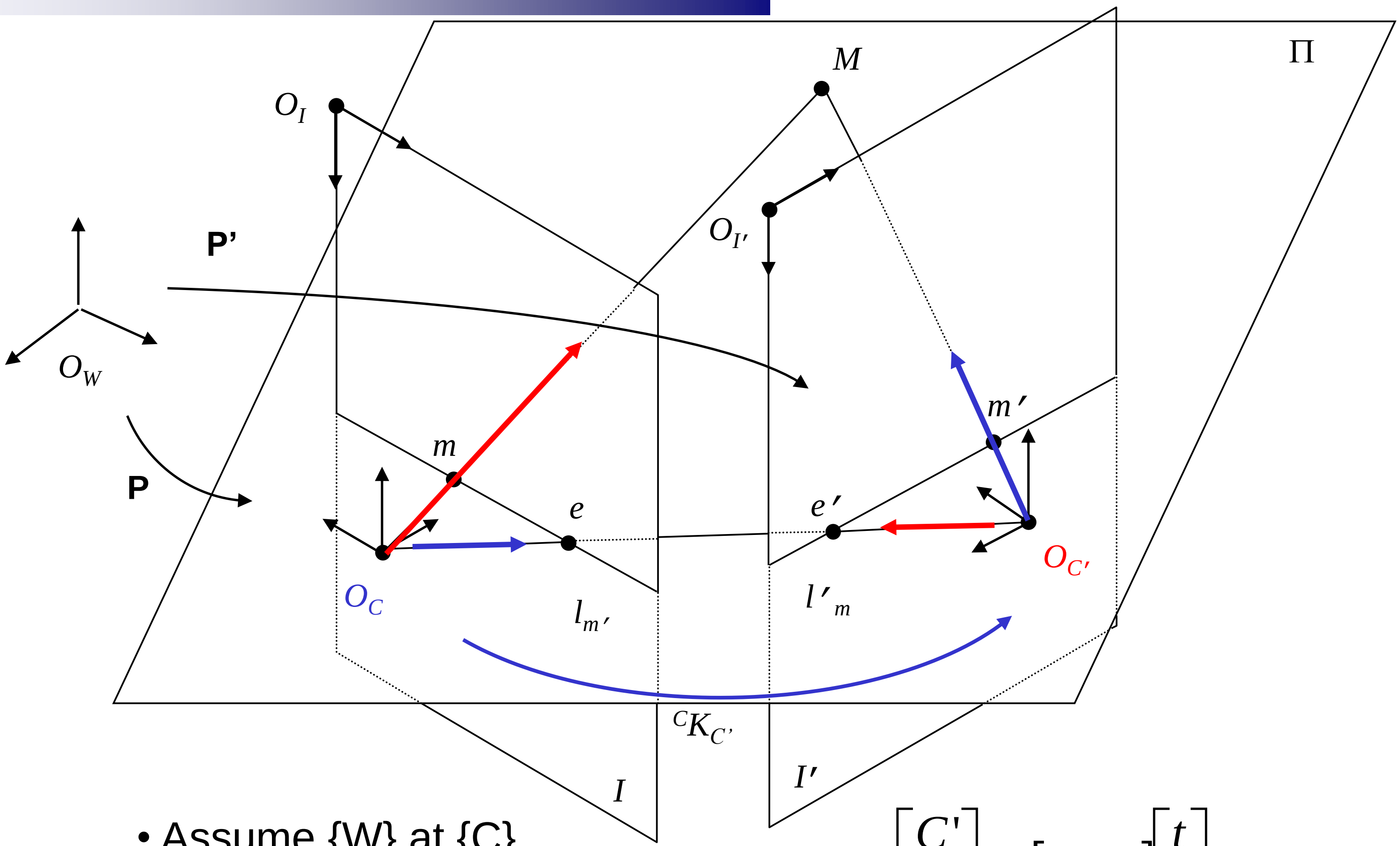
- The Epipolar Geometry concerns the problem of computing the plane Π.
  - a plane is defined by the cross product between two vectors
  - M is unknown, m and m' are knowns
  - {W} is located at {C} or {C'} and Π can be computed at {C} or {C'} → 4 solutions

4.3 Epipolar Geometry – 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

4.3 Epipolar Geometry – Modelling



- Assume {W} at {C}

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

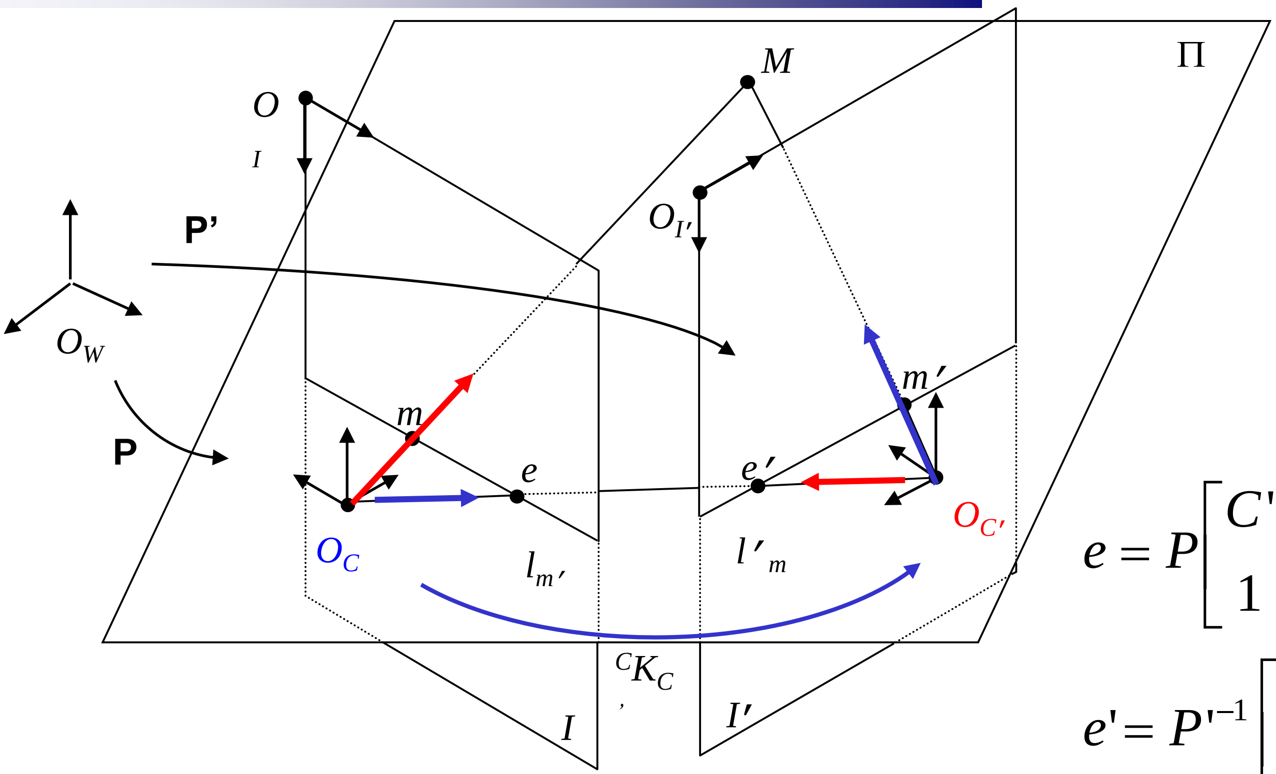
$$K = \begin{bmatrix} R & t \end{bmatrix}$$

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

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

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

4.3 Epipolar Geometry – Modelling



- Assume {W} at {C}

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

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

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

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

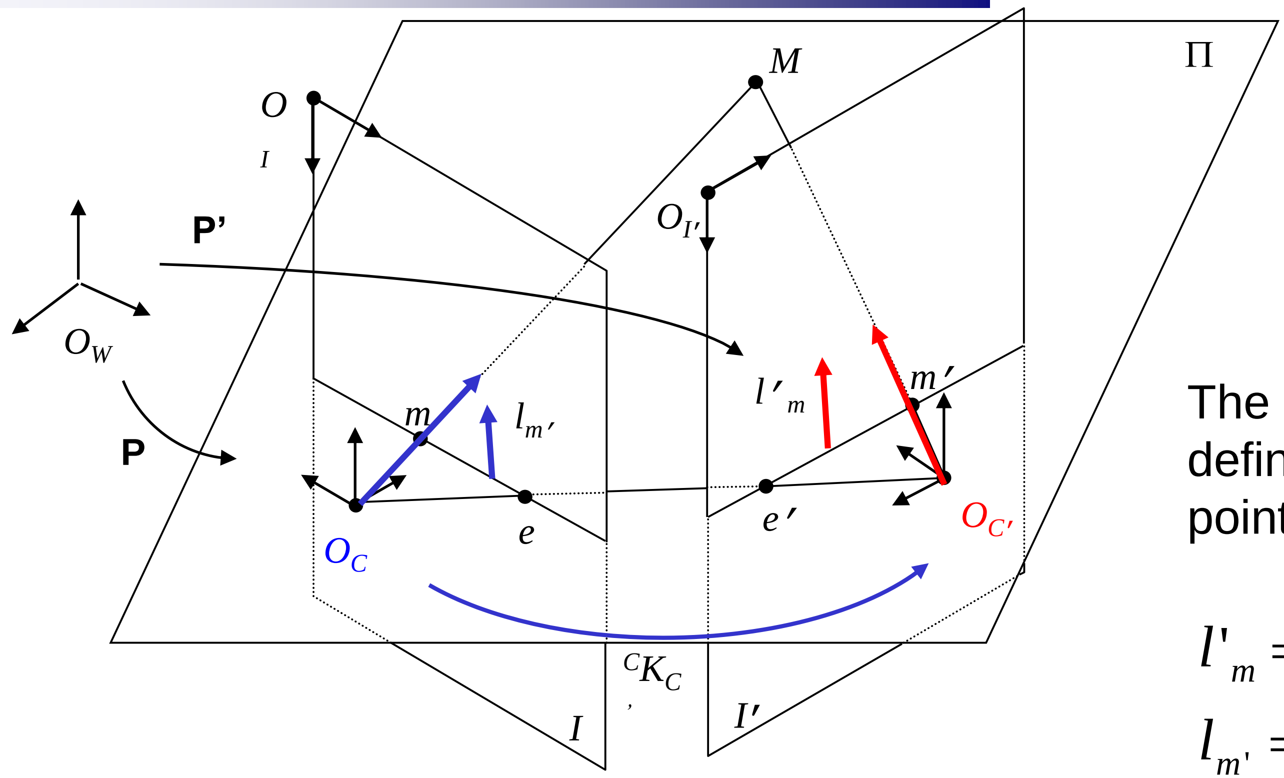
$$e' = P'^{-1}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} R^t & -R^t t \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = -R^t t$$

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.

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

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

4.3 Epipolar Geometry – Modelling



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

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

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

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

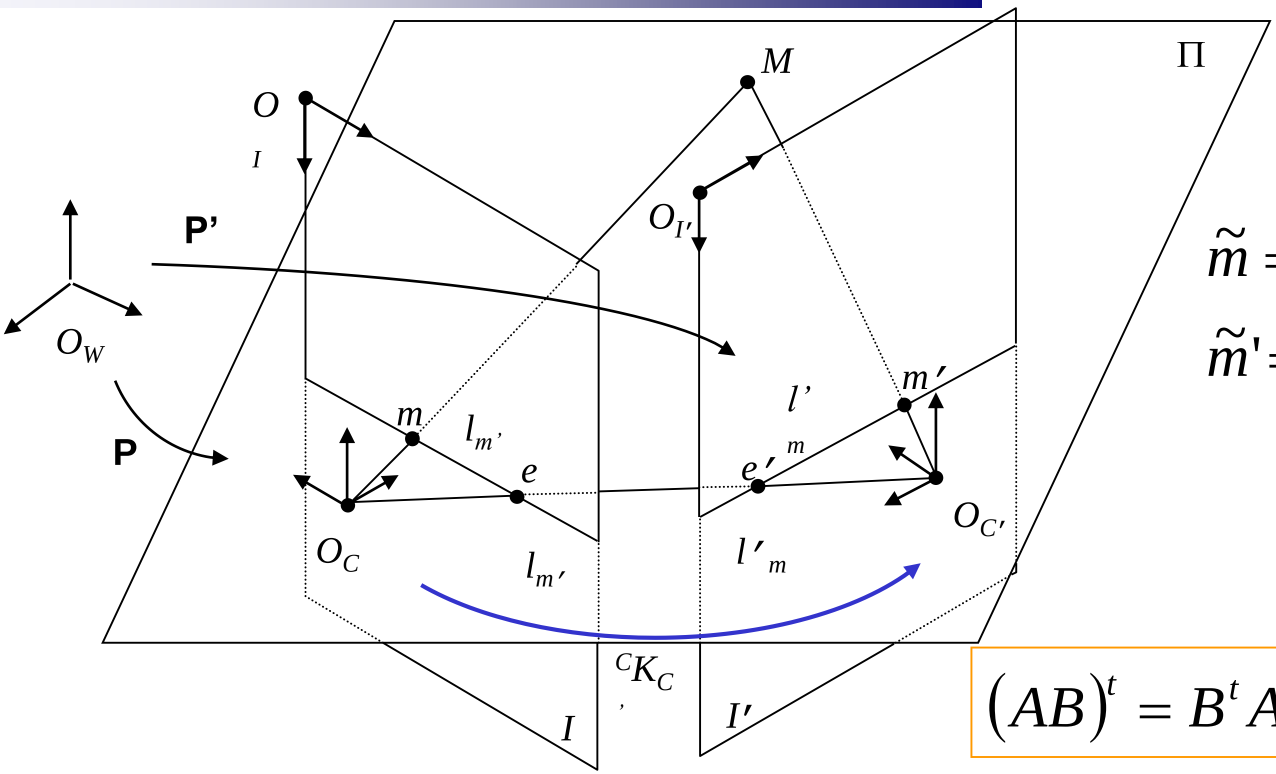
$$m \cdot l_{m'} = m^t l_{m'} = m^t [t]_x Rm'$$

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

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

Orthogonal, their cosinus is 0

4.3 Epipolar Geometry – Modelling



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

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

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

$$\mathbf{A} = \begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\boxed{(AB)^t = B^t A^t} \qquad \boxed{\left(A^{-1}\right)^t = \left(A^t\right)^{-1} = A^{-t}}$$

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

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

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

4.3 Epipolar Geometry – 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 equivalent to F and F'.

4.3 Epipolar Geometry – 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'.

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

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

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

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

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

Contents

Contents

4.4 Epipolar Geometry – Calibration

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:

$$U_n f = 0$$

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

$$u_i = \left(x_i' x_i, y_i' x_i, x_i, x_i' y_i, y_i' y_i, y_i, x_i', y_i', 1 \right)$$

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

4.4 Epipolar Geometry – Calibration

The Eight Point Method with Least Squares

$$U_n f = 0$$

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:

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

$$f' = -U'^{-1}_n 1_n \quad \Longrightarrow \quad f' = -\left(U'^t_n U'_n\right)^{-1} U'^t_n 1_n$$

Least-Squares

4.4 Epipolar Geometry – Calibration

The Eight Point Method with 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:

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

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

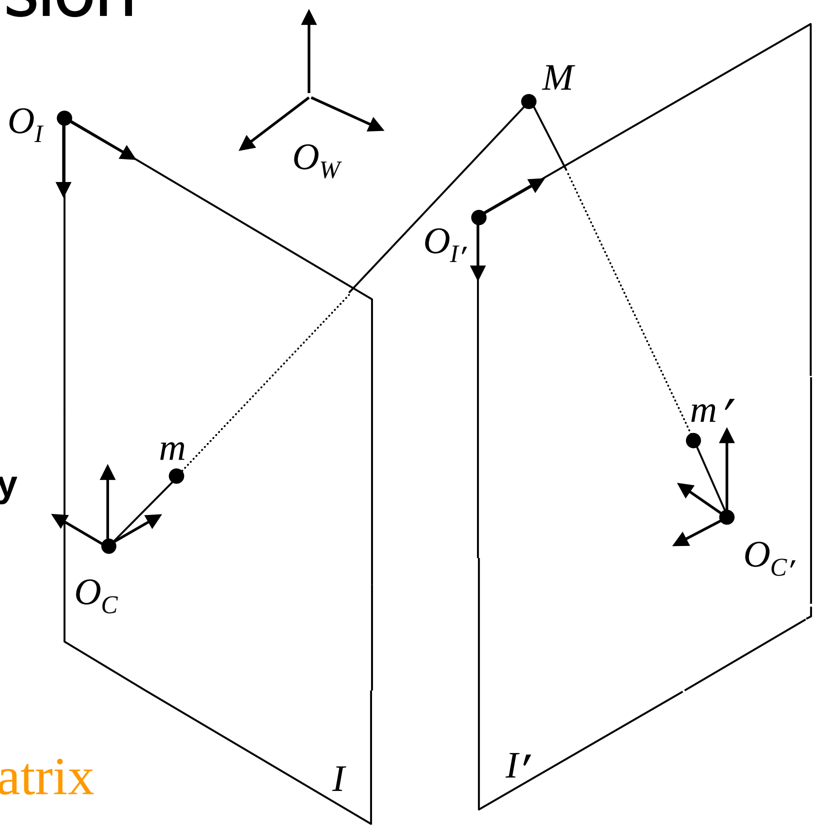
$$[U,D,V] = \mathrm{svd}\,(U_n)$$

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.

Contents

Contents

4.5 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:

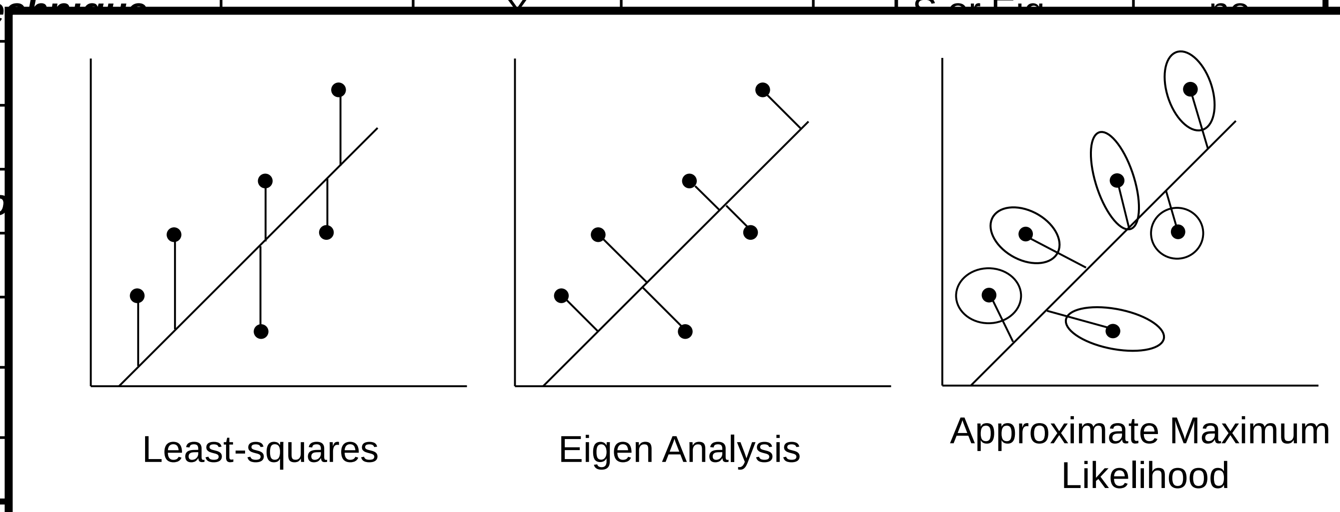
- The Correspondence Problem →F/E matrix
- Stereo Configurations:
  - Calibrated Stereo: Intrinsics and Extrinsics known → Triangulation!
  - Uncalibrated Stereo: Intrinsics and Extrinsics unknown → F matrix
  - Calibrated Monocular: Intrinsics known, Extrinsics unknown → E matrix
  - Uncalibrated Monocular: Intrinsics and Extrinsics unknown → F matrix

Contents

Contents

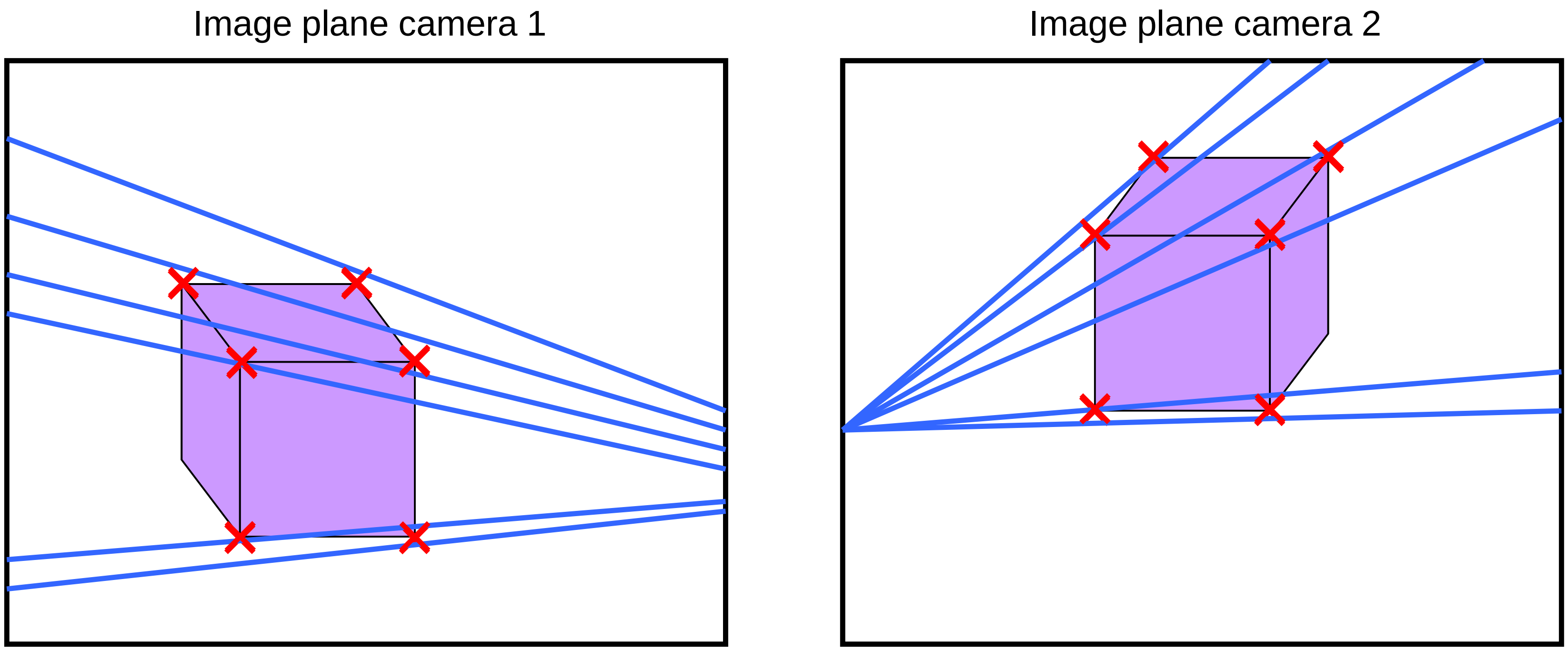
4.6 Experimental comparison – methods

	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 technique		X		LS or Eig.	no
FNS					
CFNS					
M-Estimato					
LMedS					
RANSAC					
MLESAC					
MAPSAC					



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

4.6 Experimental comparison – Methodology



4.6 Experimental comparison – Synthetic images

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

4.6 Experimental comparison – Synthetic images

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

4.6 Experimental comparison – Synthetic images

Robust methods: Cope with both noise and outliers

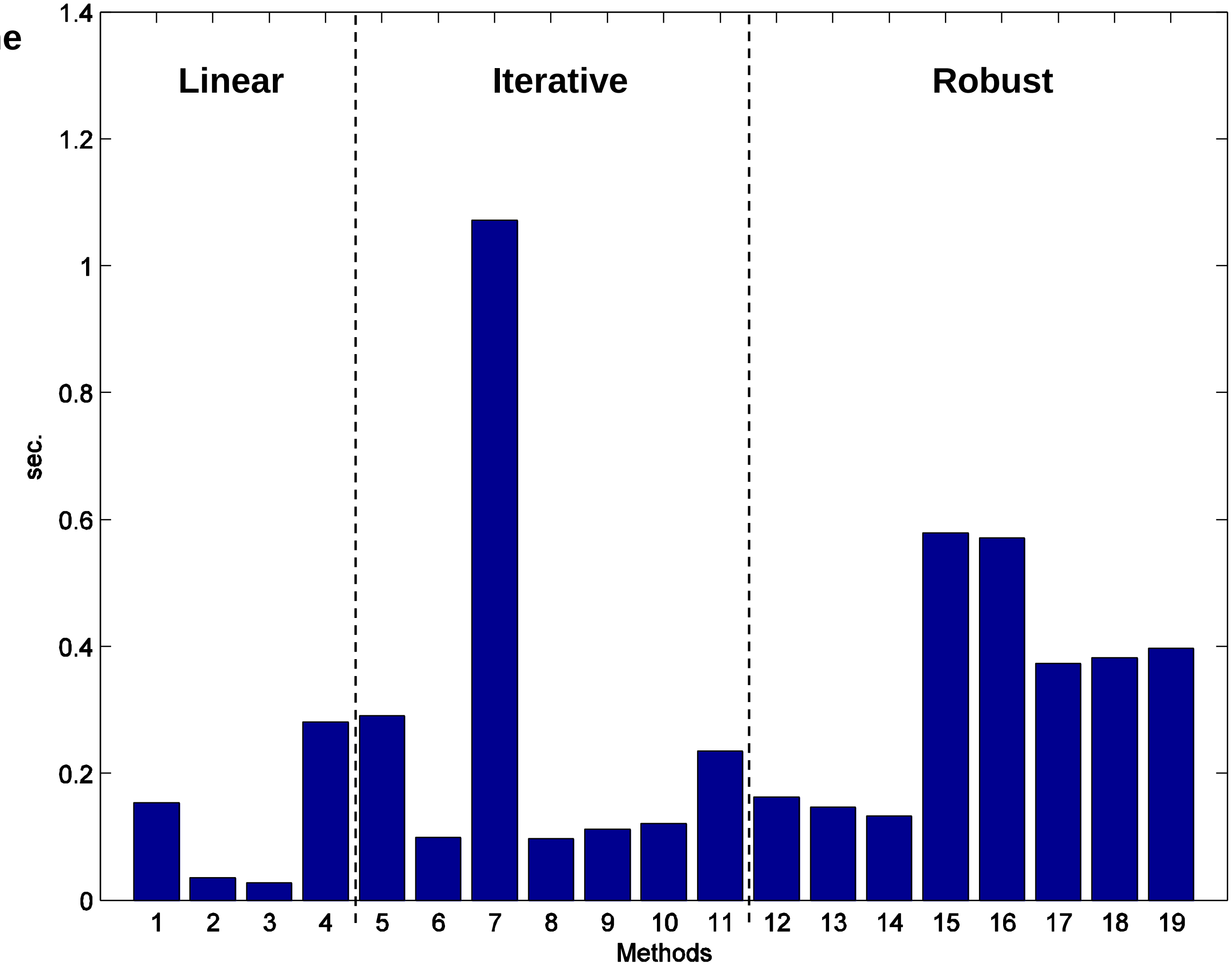
Methods	Robust							
	12	13	14	15	16	17	18	19
$\sigma = 0.0$ outliers 0%	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000	0.100 0.079	0.011 0.009
$\sigma = 0.0$ outliers 10%	273.403 360.443	4.909 4.493	4.714 2.994	0.000 0.000	0.000 0.000	16.457 26.923	19.375 70.160	0.115 0.115
$\sigma = 0.1$ 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

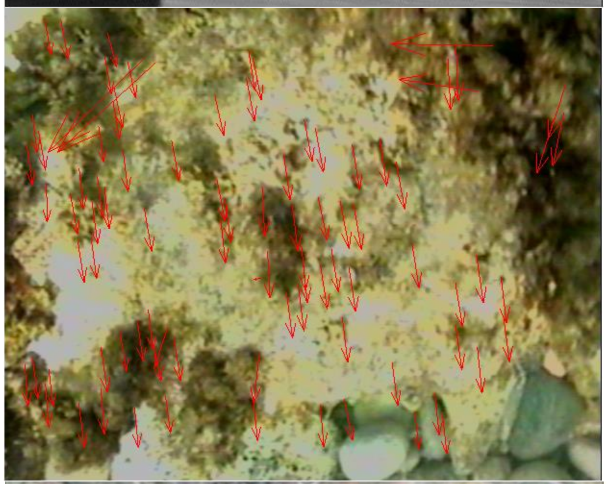
4.6 Experimental comparison – Synthetic images

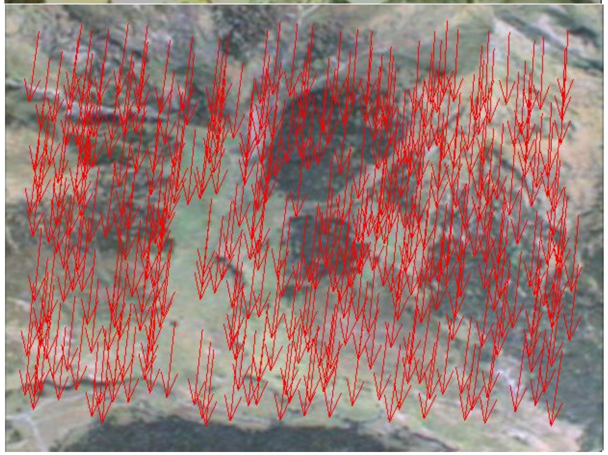
Computing Time

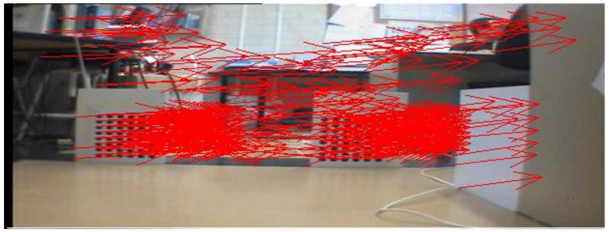


1.- 7-Point; 2.- 8-Point with Least-Squares; 3.- 8-Point with Eigen Analysis; 4.- Rank-2 Constraint; 5.- Iterative Linear; 6.- Iterative Newton-Raphson; 7.- Minimization in parameter space; 8.- Gradient using LS; 9.- Gradient using Eigen; 10.- FNS; 11.- CFNS; 12.- M-Estimator using LS; 13.- M-Estimator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

4.6 Experimental comparison – Real images

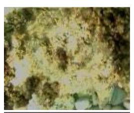






4.6 Experimental comparison – Real images






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

4.6 Experimental comparison – Conclusions

- Survey of 15 methods of computing $\mathbf{F}$ and up to 19 different implementations
- Description of the estimators from an algorithmic point of view
- Conditions: Gaussian noise, outliers and real images
  - **Linear methods**: Good results if the points are well located and the correspondence problem previously solved (without outliers)
  - **Iterative methods**: Can cope with noise but inefficient in the presence of outliers
  - **Robust methods**: Cope with both noise and outliers
- Least-squares is worse than eigen analysis and approximate maximum likelihood
- Rank-2 matrices are preferred if a good geometry is required
- Better results when data are previously normalized

4.6 Experimental comparison – Conclusions

Publications

- *X. Armangué and J. Salvi. Overall View Regarding Fundamental Matrix Estimation. Image and Vision Computing, IVC, pp. 205-220, Vol. 21, Issue 2, February 2003.*
- *J. Salvi. An approach to coded structured light to obtain three dimensional information. PhD Thesis. University of Girona, 1997. Chapter 3.*
- J. Salvi, X. Armangué, J. Pagès. *A survey addressing the fundamental matrix estimation problem*. IEEE International Conference on Image Processing, **ICIP 2001**, Thessaloniki, Greece, October 2001.

More Information: http://eia.udg.es/~qsalvi/

Contents

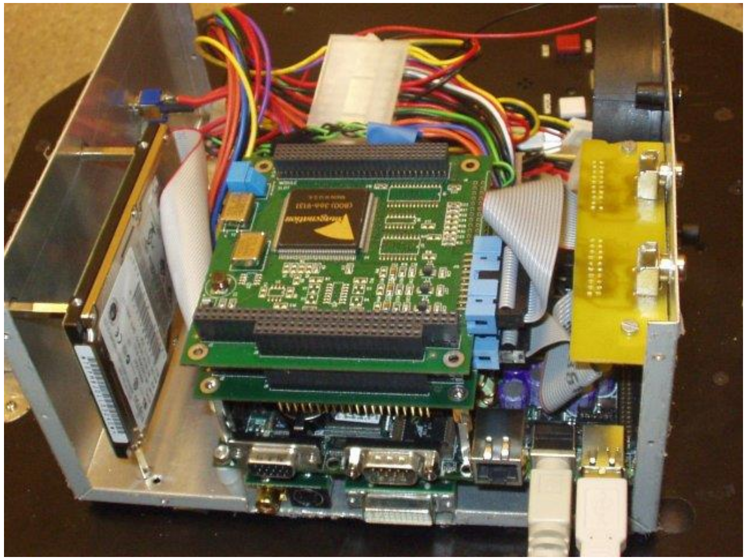
Contents

4.7 Sample: Mobile robot performing 3D mapping

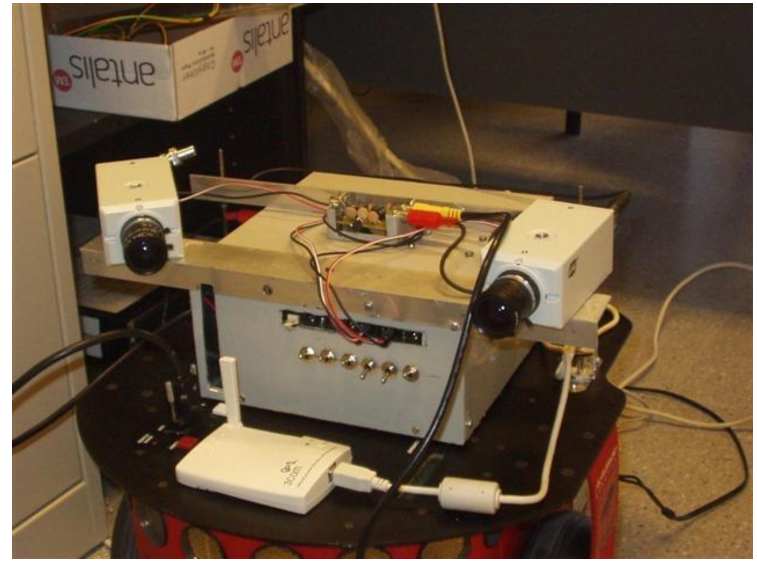
- Building a 3D map from an unknown environment using a stereo camera system
- Localization of the robot in the map
- Providing a new useful sensor for the robot control architecture

GRILL Mobile robot with a stereo camera system

4.7 3D mapping – Robot components

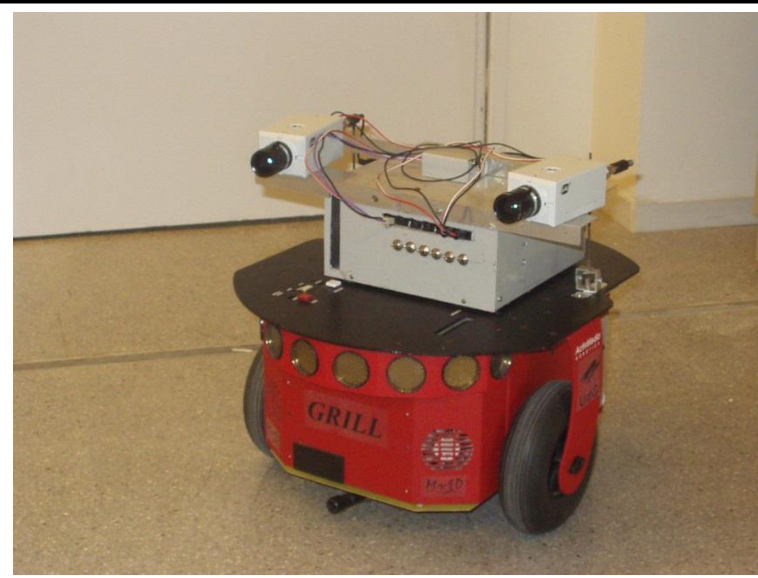


Inside stereo vision system



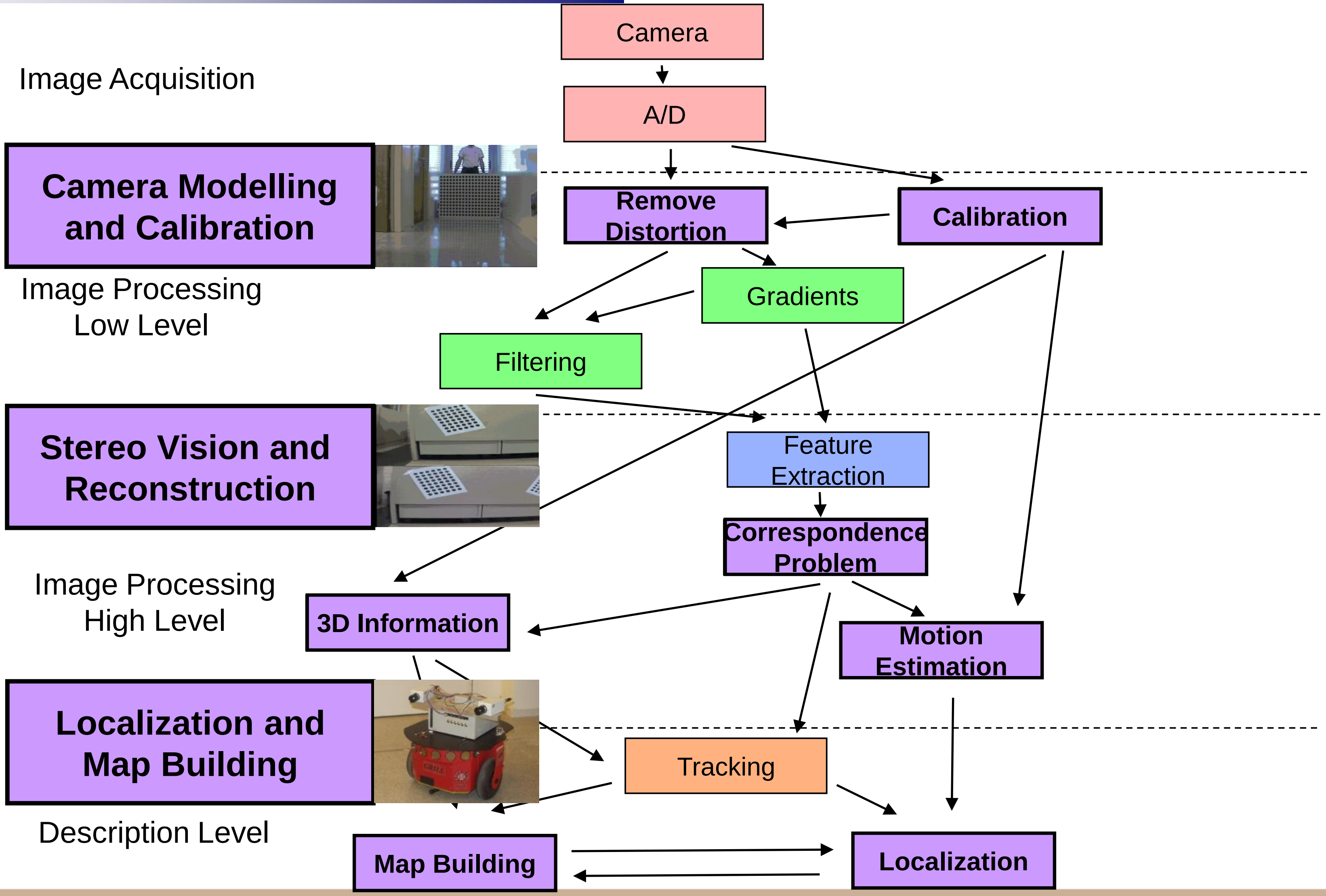
Outside stereo vision system

Pioneer 2

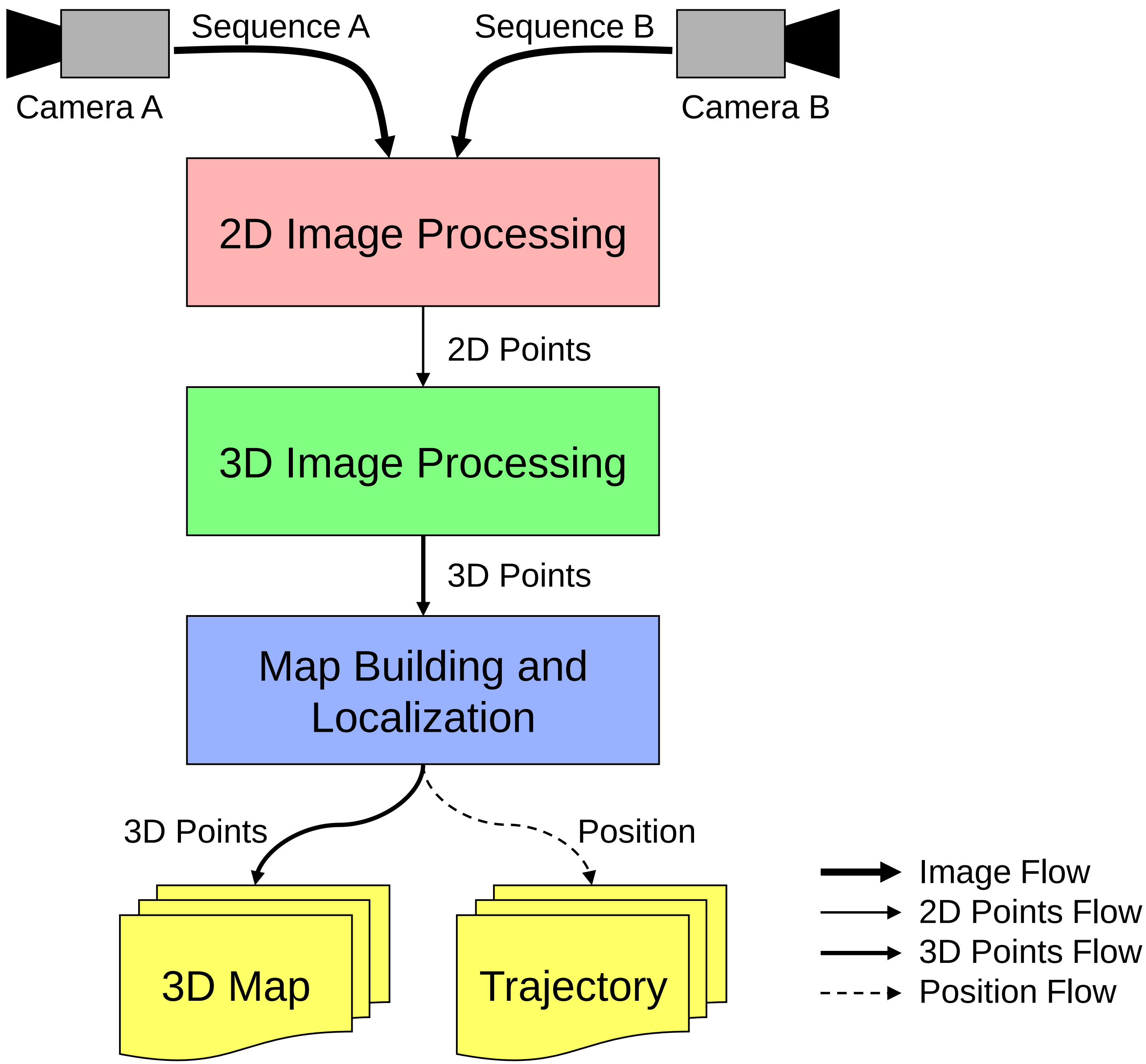


GRILL Mobile Robot

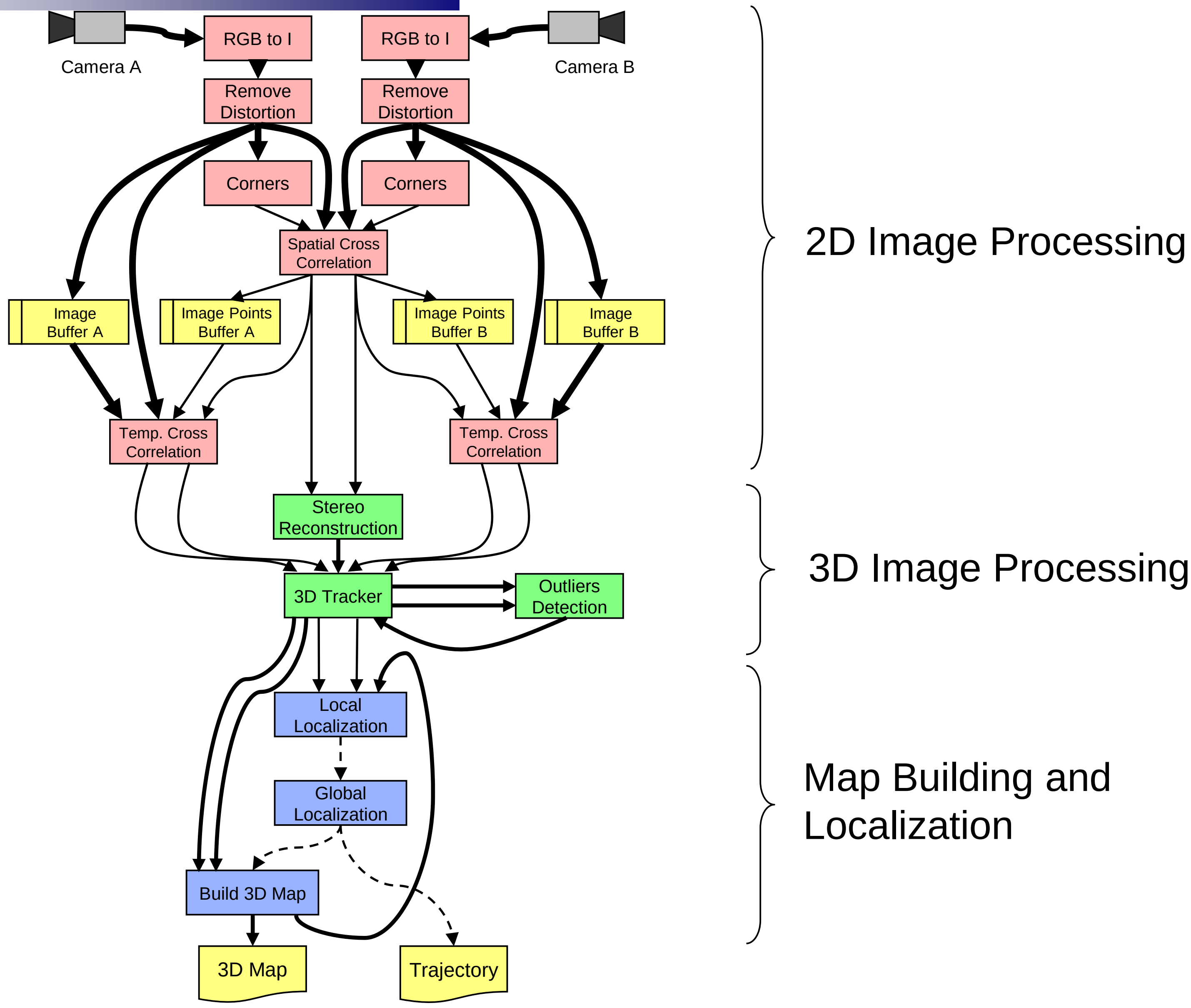
4.7 3D mapping – Data flow diagram



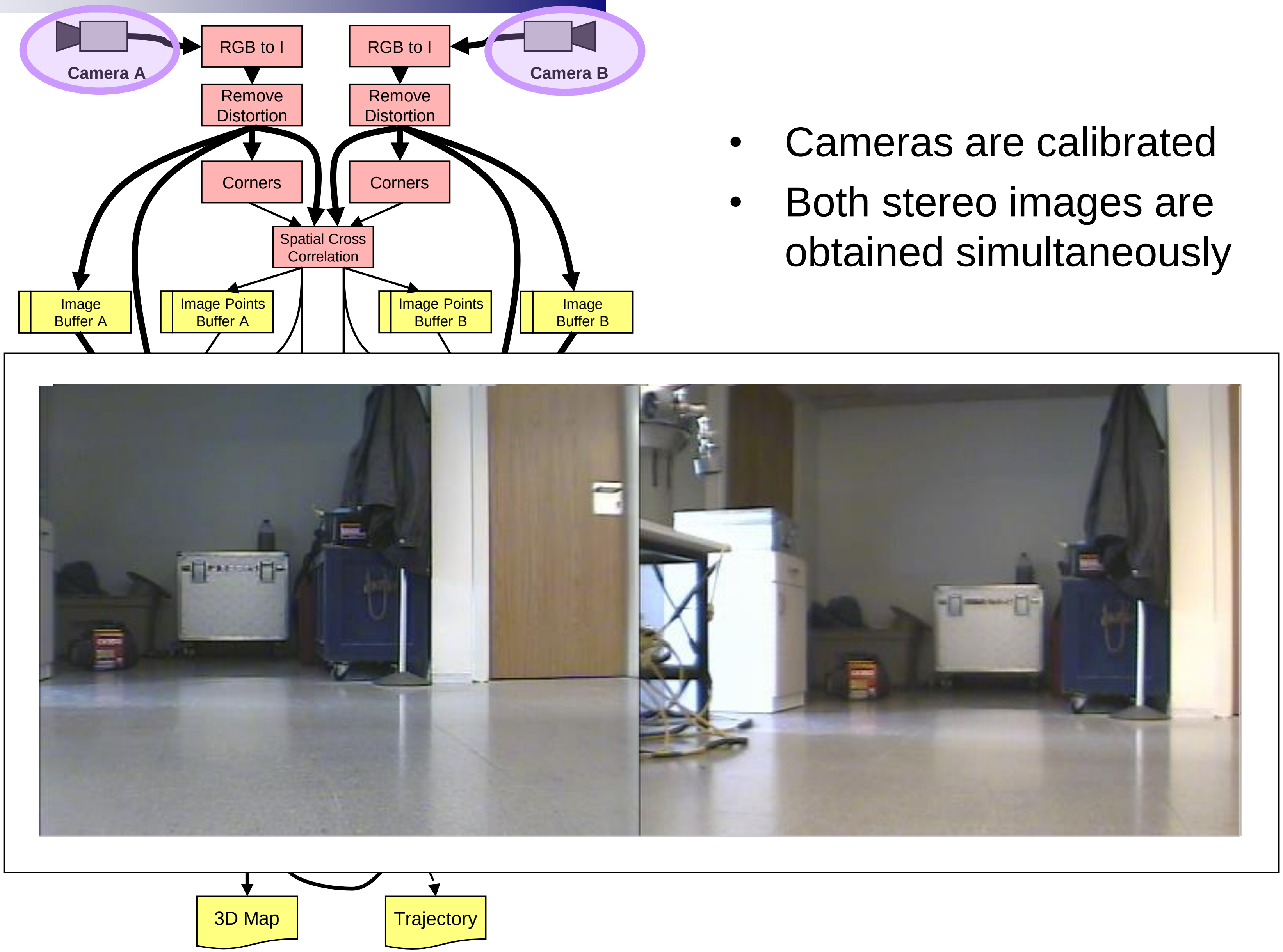
4.7 3D mapping – Data flow diagram



4.7 3D mapping – Data flow diagram

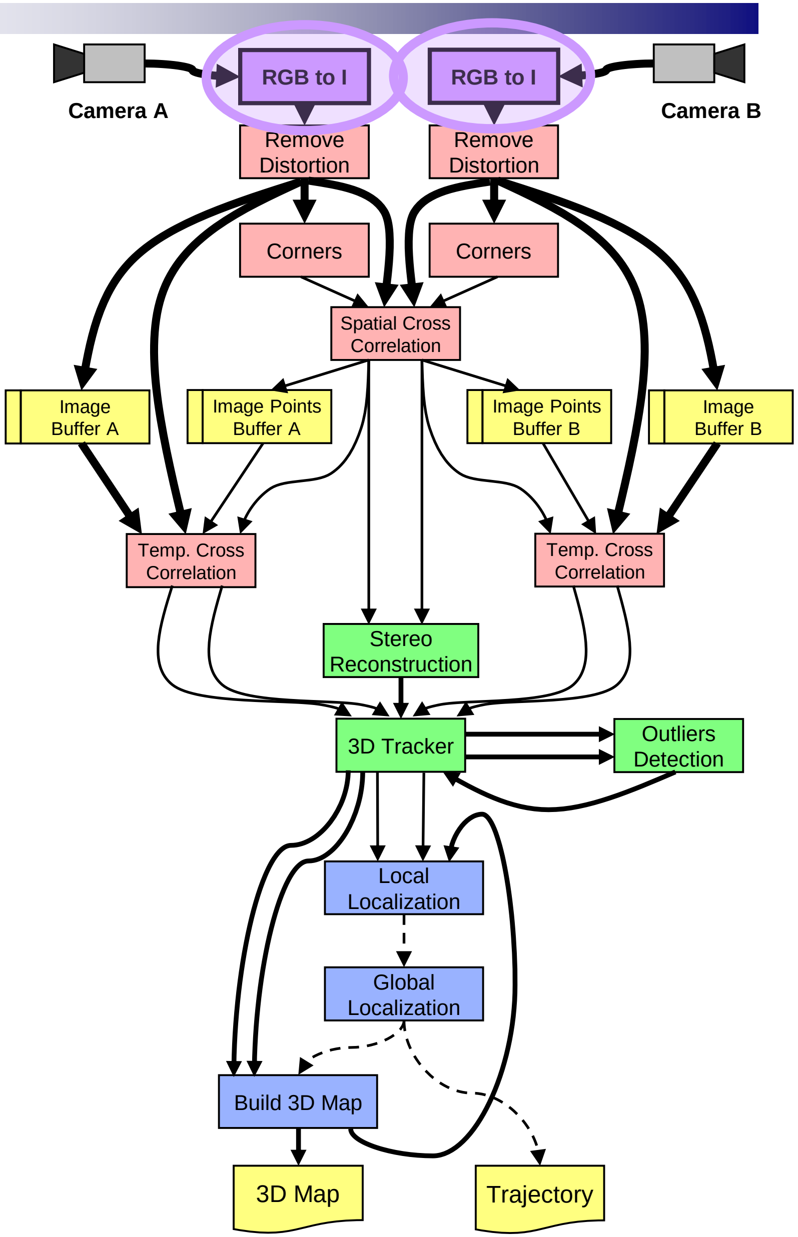


4.7 3D mapping – Input sequence



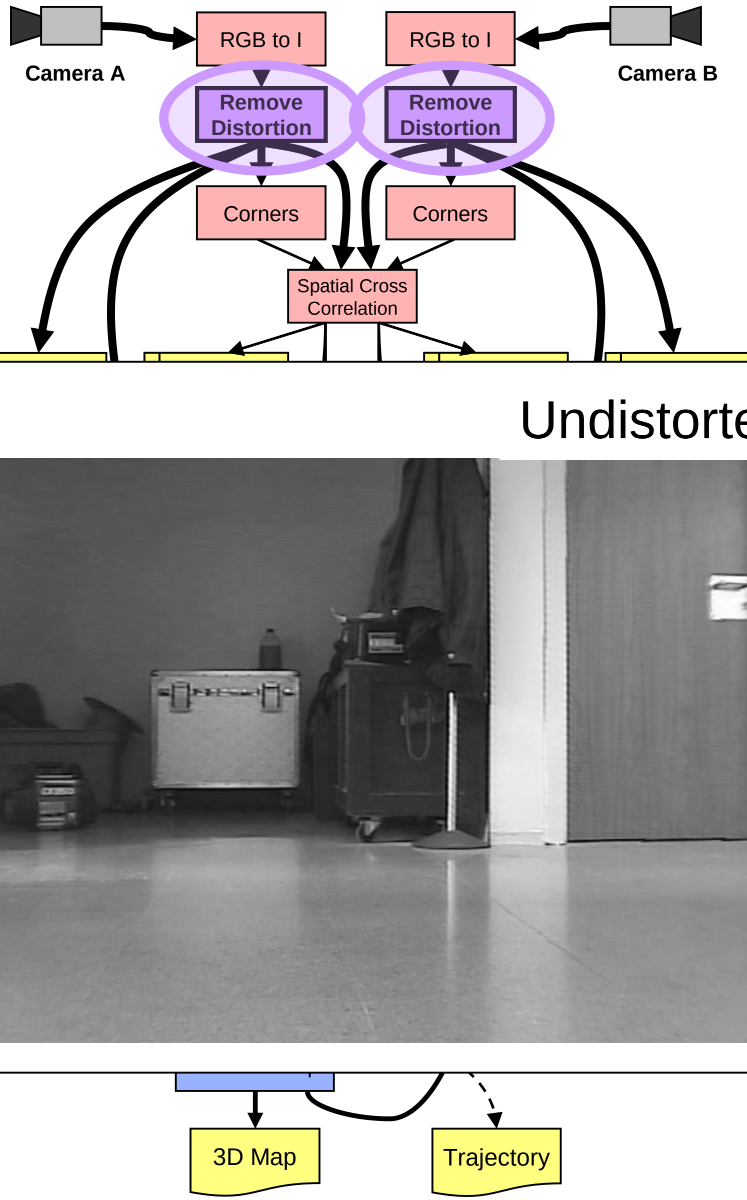
- Cameras are calibrated
- Both stereo images are obtained simultaneously

4.7 3D mapping – RGB to I



- Description
  - Converting a color image to an intensity image
- Input
  - Color image (RGB)
- Output
  - Intensity image

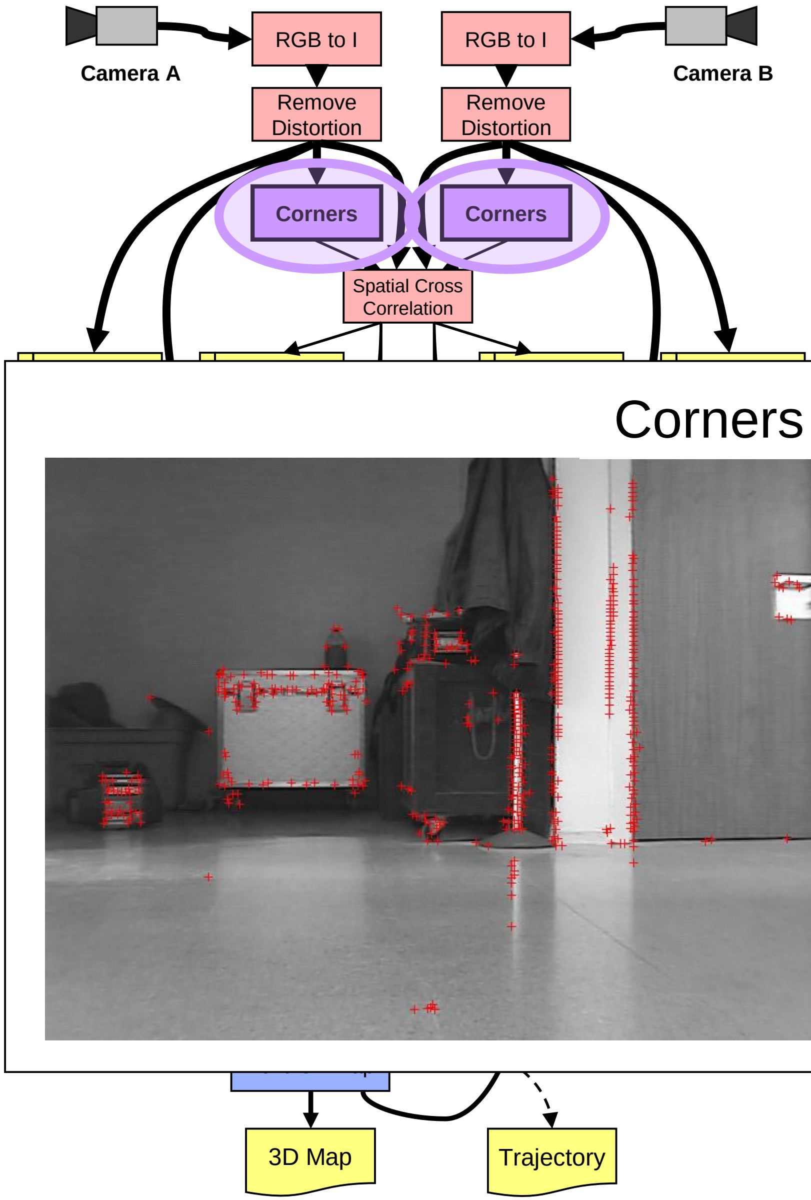
4.7 3D mapping – Remove Distortion



- Description
  - Removing distortion of an image using camera calibration parameters
- Input

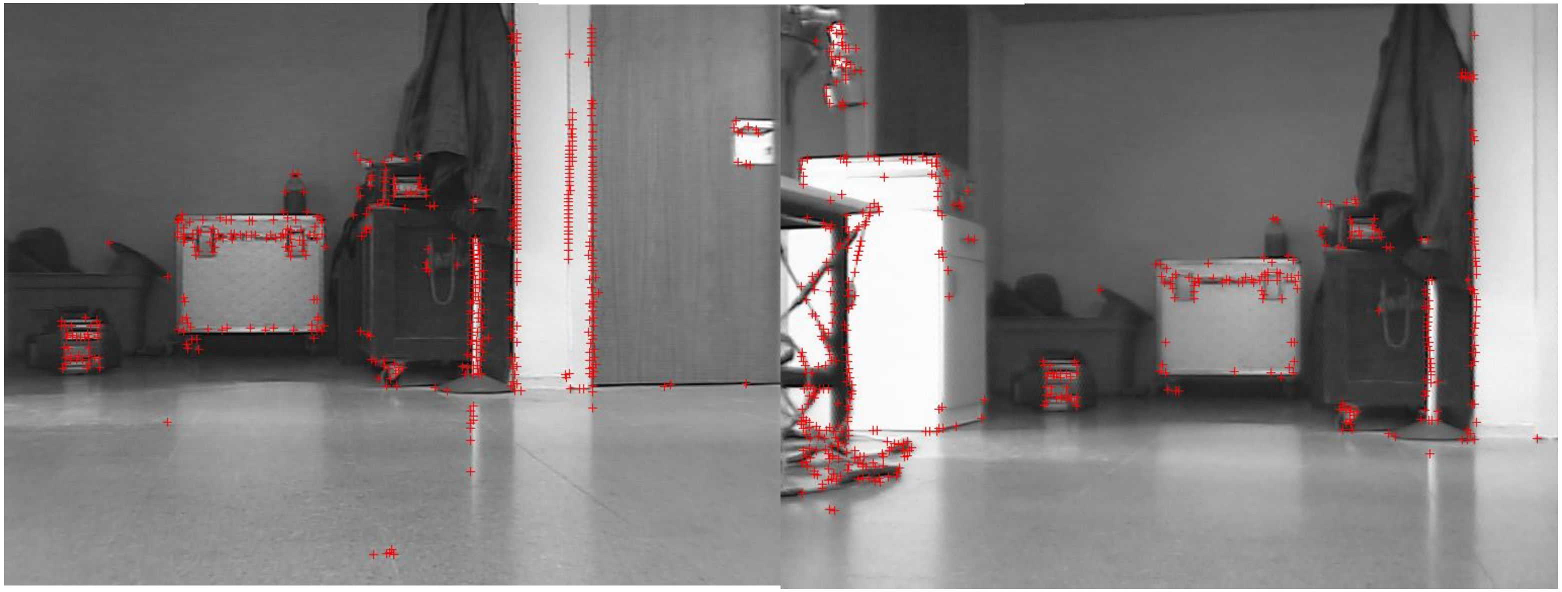
Undistorted Images

4.7 3D mapping – Corners

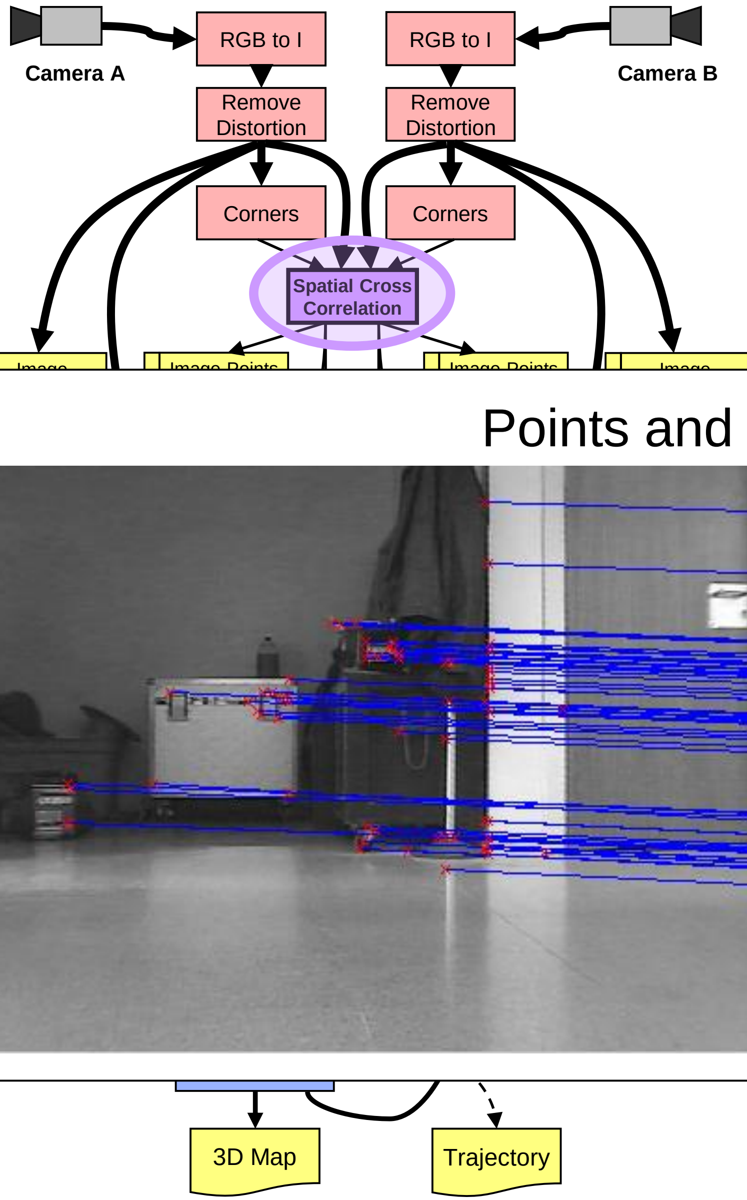


- Description
  - Detection of corners using a variant of Harris corners detector
- Input

Corners Detected

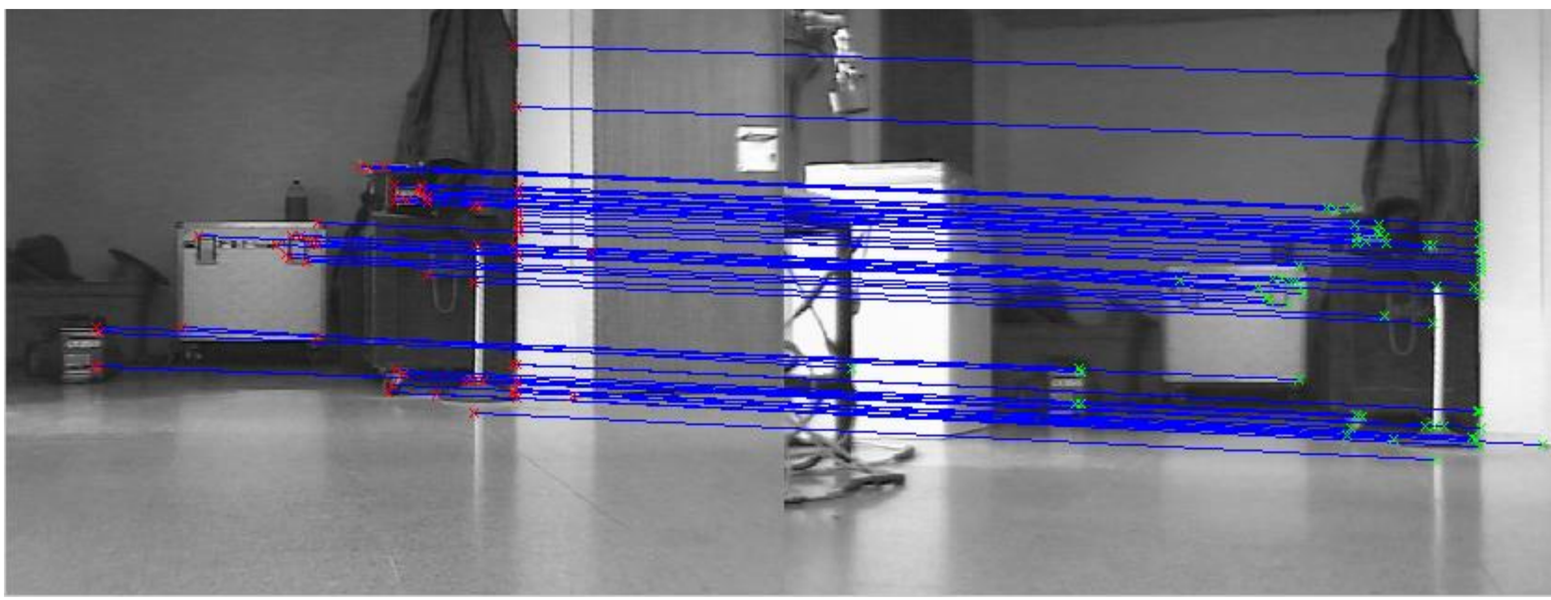


4.7 3D mapping – Spatial Cross Correlation



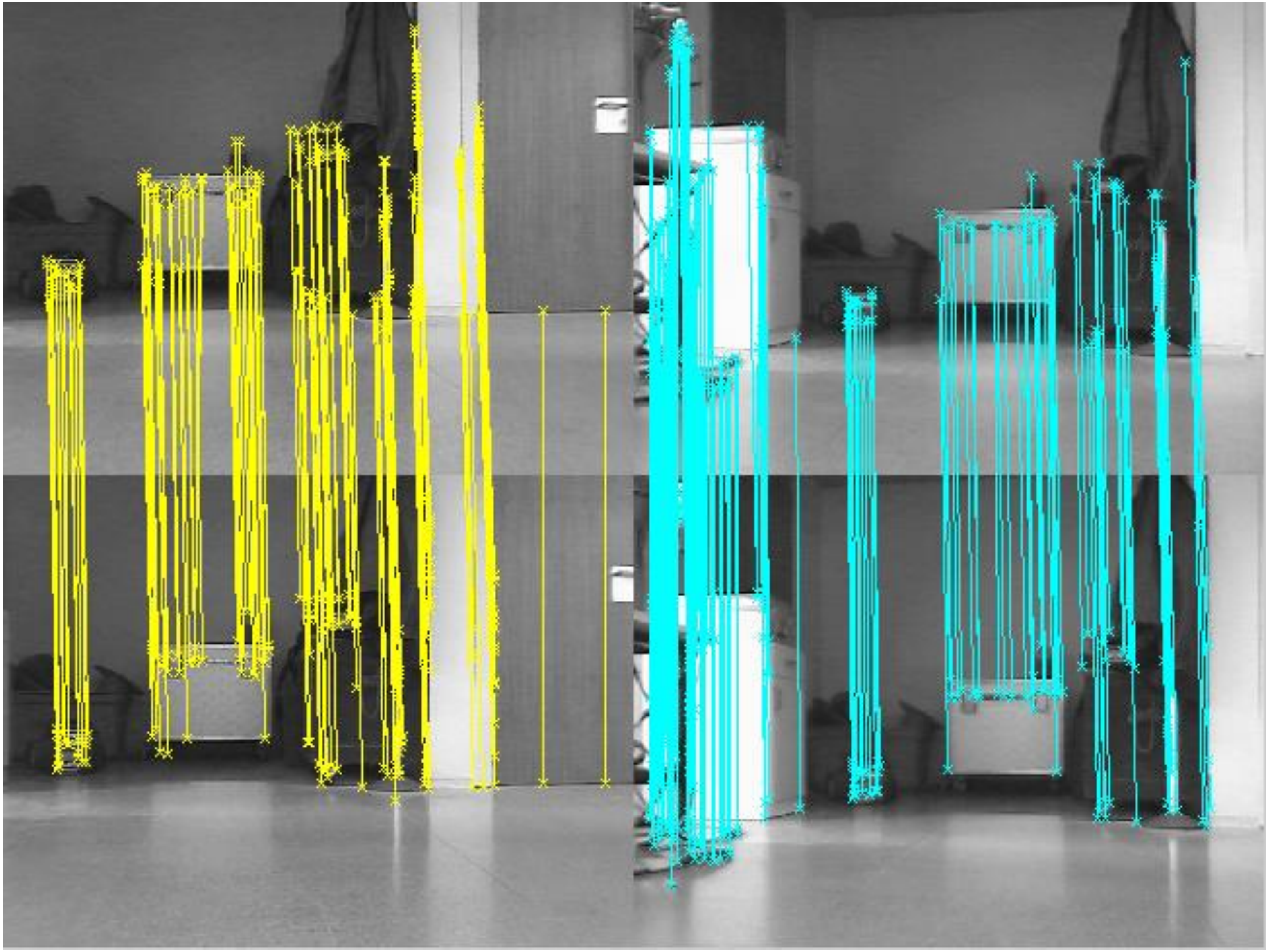
- Description
  - Spatial cross correlation using fundamental matrix obtained from camera calibration parameters

Points and matches list



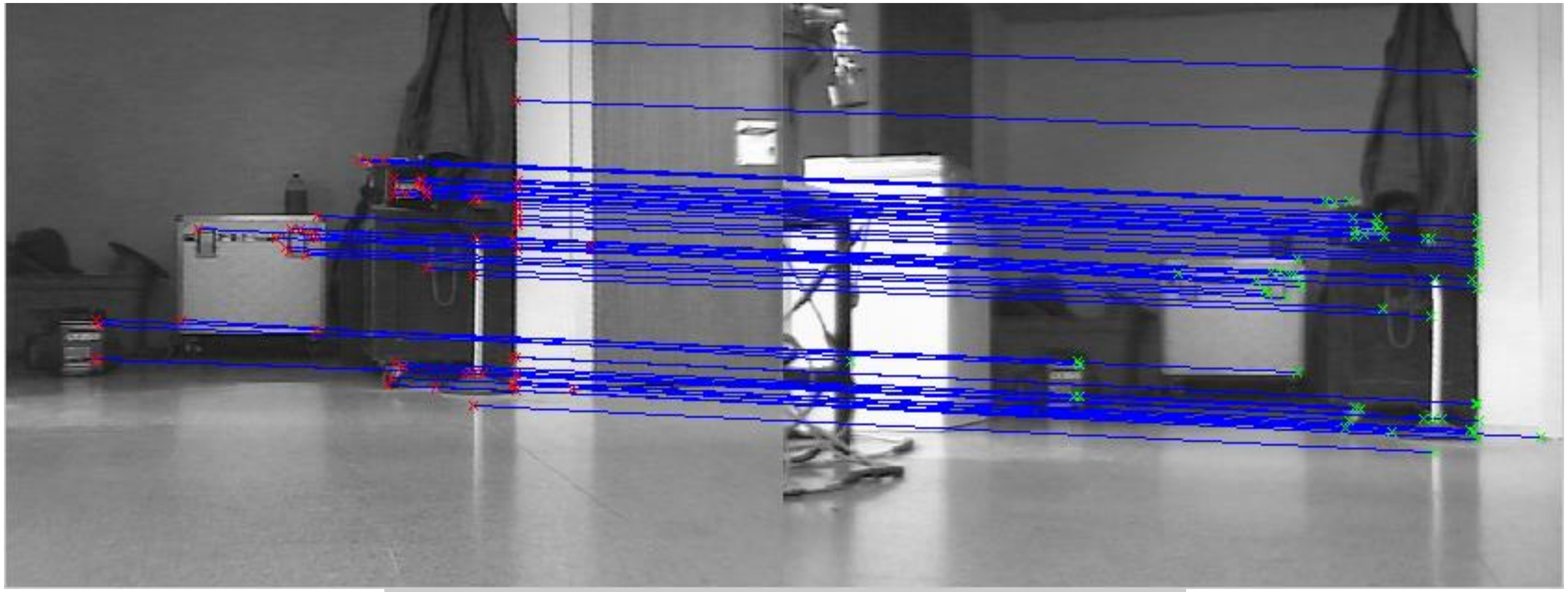
4.7 3D mapping – Temporal Cross Correlation

Points and matches list

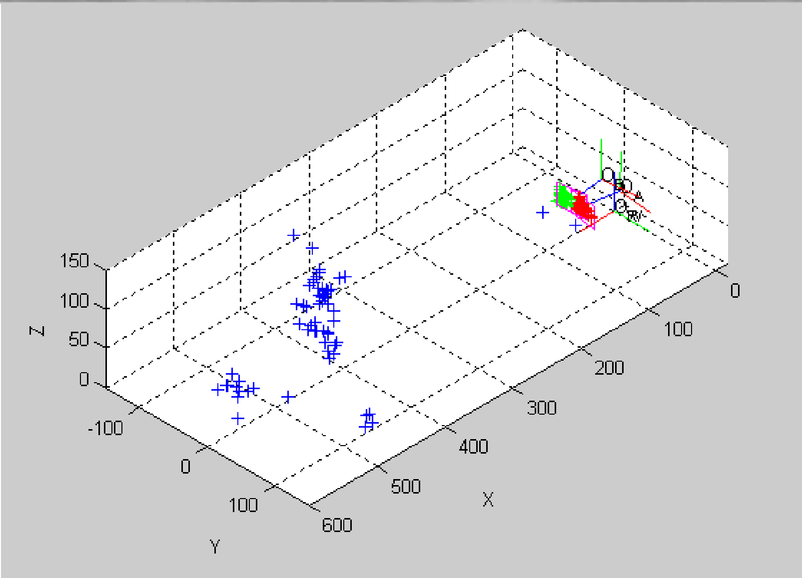


4.7 3D mapping – Stereo Reconstruction

Points and matches list

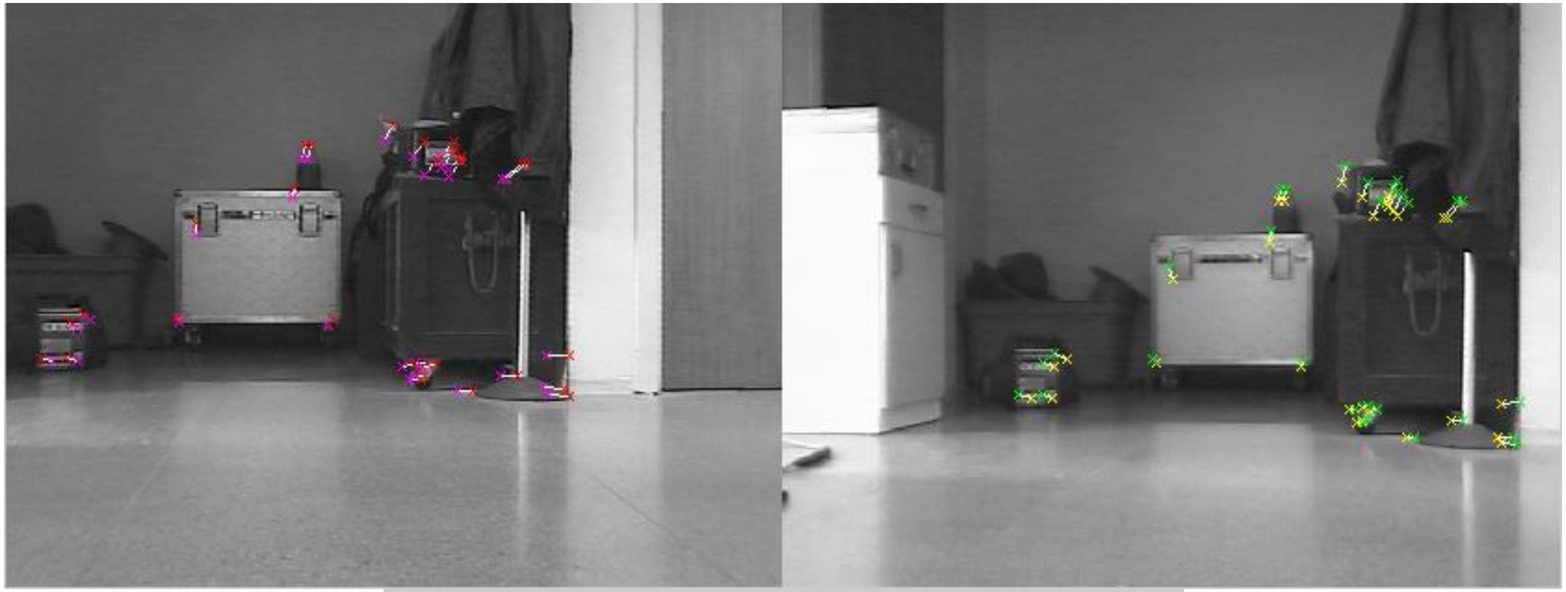


3D points list

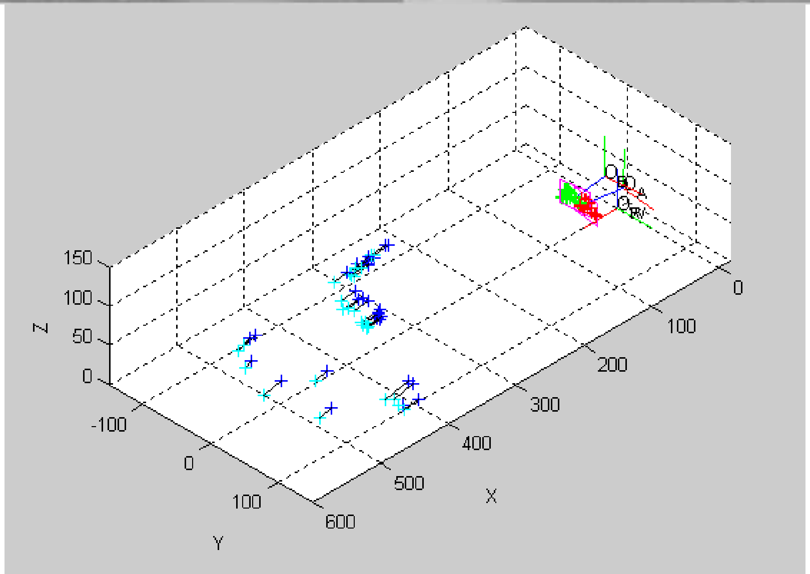


4.7 3D mapping – 3D Tracker

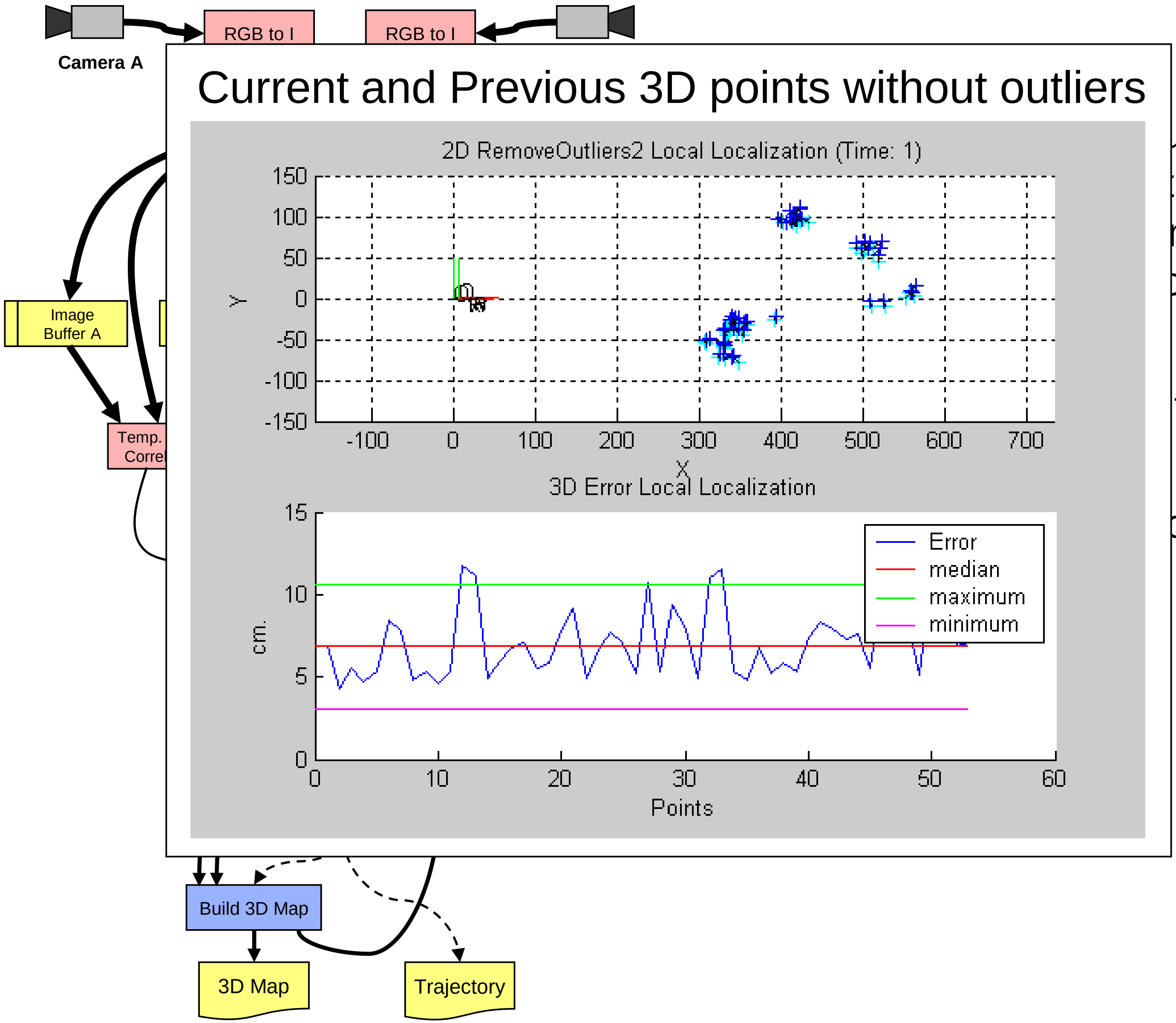
Points and matches tracker



3D points tracker



4.7 3D mapping – Outliers Detection

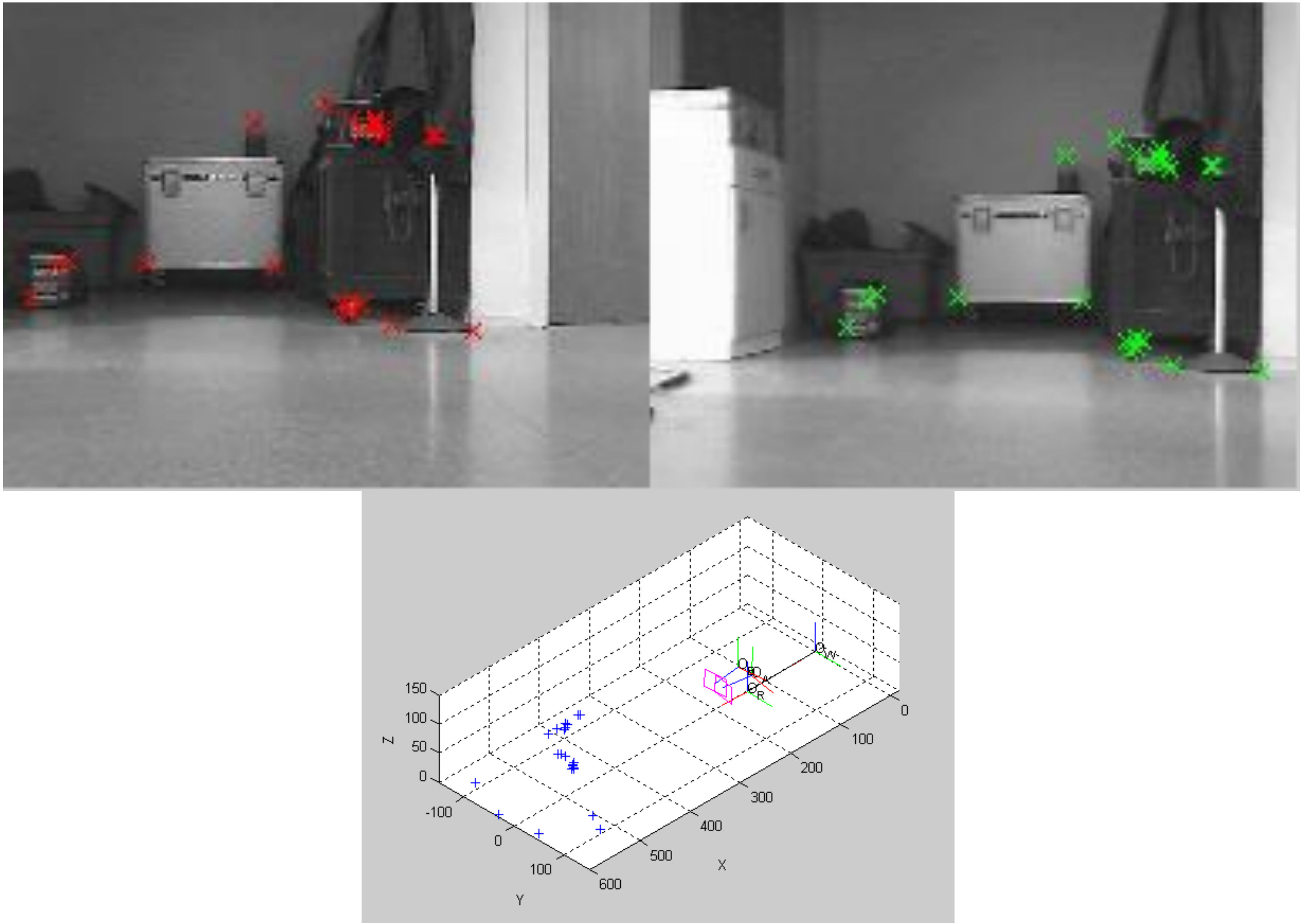


utliers
tance
nt and
oints list

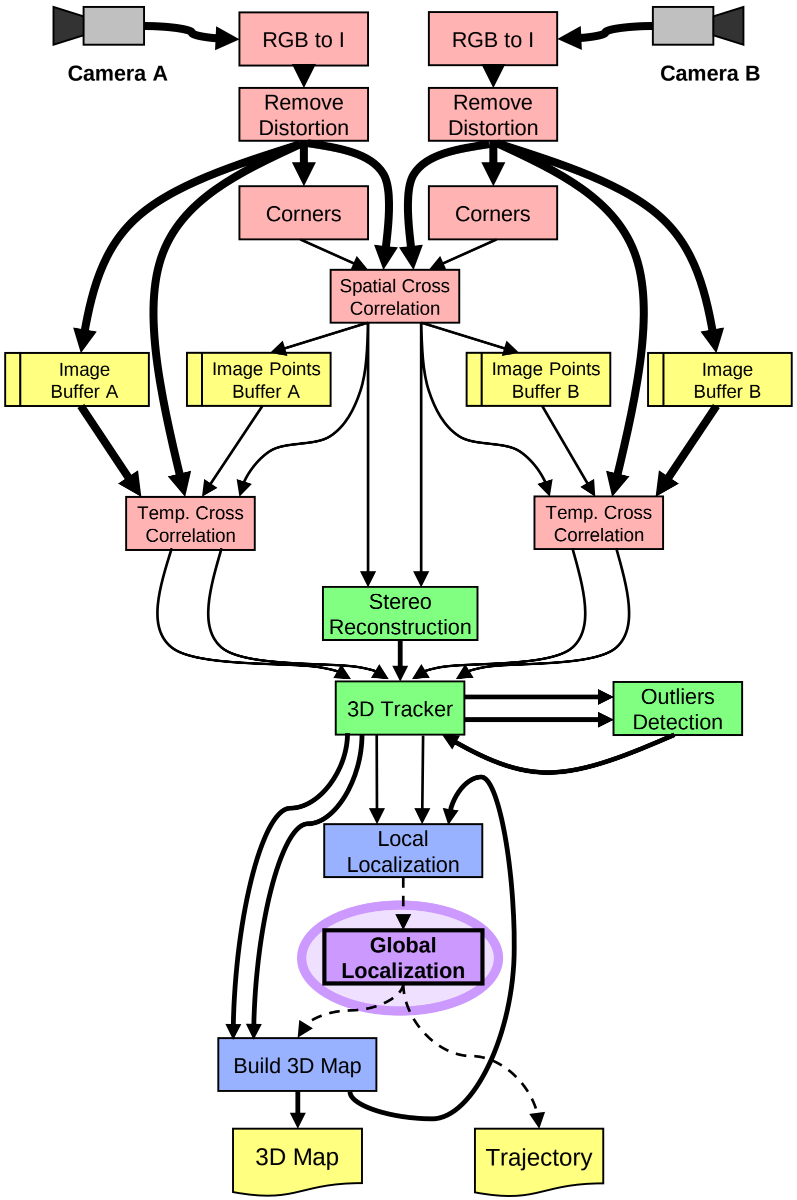
tion
nts list
oints list

4.7 3D mapping – Local Localization

Map absolute position

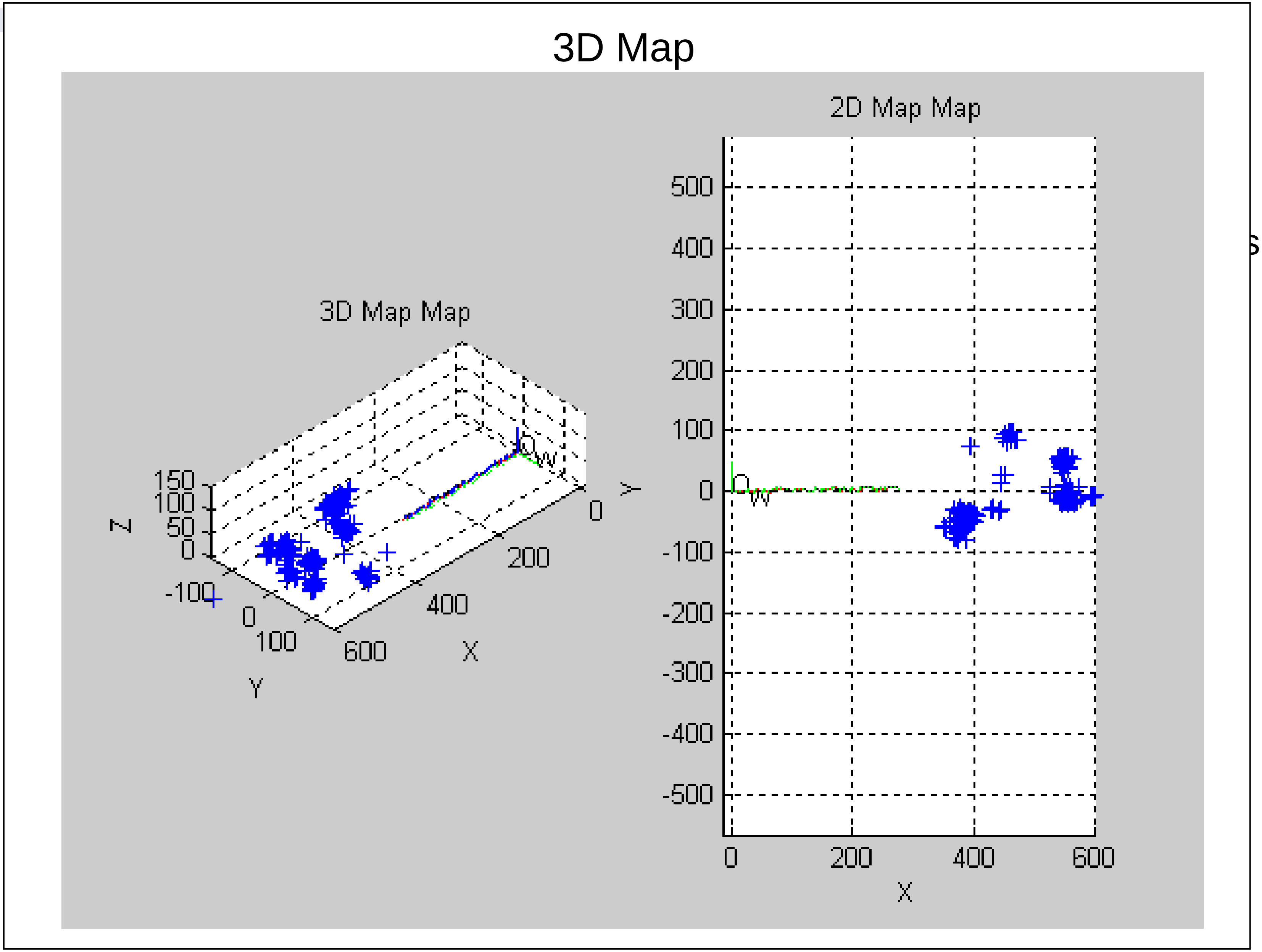


4.7 3D mapping – Global Localization



- Description
  - Computing the trajectory effect by the robot
- Input
  - Local position
- Output
  - Global position

4.7 3D mapping – Building 3D Map



4.7 3D mapping – Video