Geometry and Algebra in Computer Vision & Robotics

T o m a s P a j d l a

with contributions from

Z. Kukelova, M. Bujnak, T. Schilling, J. Heller, C. Albl

Czech Technical University Prague

Center for Machine Perception

Geometry of Vision & Robotics Group

Tomas Pajdla
Assistant professor
Center for Machine Perception, Dpt. of Cybernetics, FEE, CTU in Prague
pajdla@cmp.felk.cvut.cz

Research & applications

- 3D reconstruction
- Photogrammetry
- Robotics & manipulation
- Algebra, geometry, optimization

Teaching (PhD, MSc, BSc)

Geometry of Computer Vision & Robotics

Inria, ETH Zürich, Penn University of Pennsylvania

Research funding

Seventh Framework Programme, Ministerstvo školství, mládeže a tělovýchovy, Technologická agentura České republiky

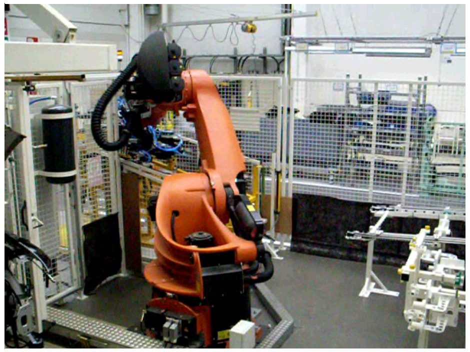
Industry collaboration

Daimler, Astrium (an EADS company), Neovision Industrial Vision Systems, NASA JPL Jet Propulsion Laboratory California Institute of Technology, SciSys

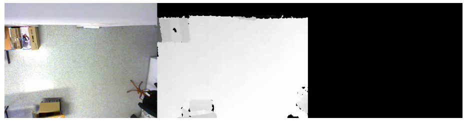
NEOVISION Industrial Vision Systems | News | Solutions | Products | Support | About

www.neovision.cz

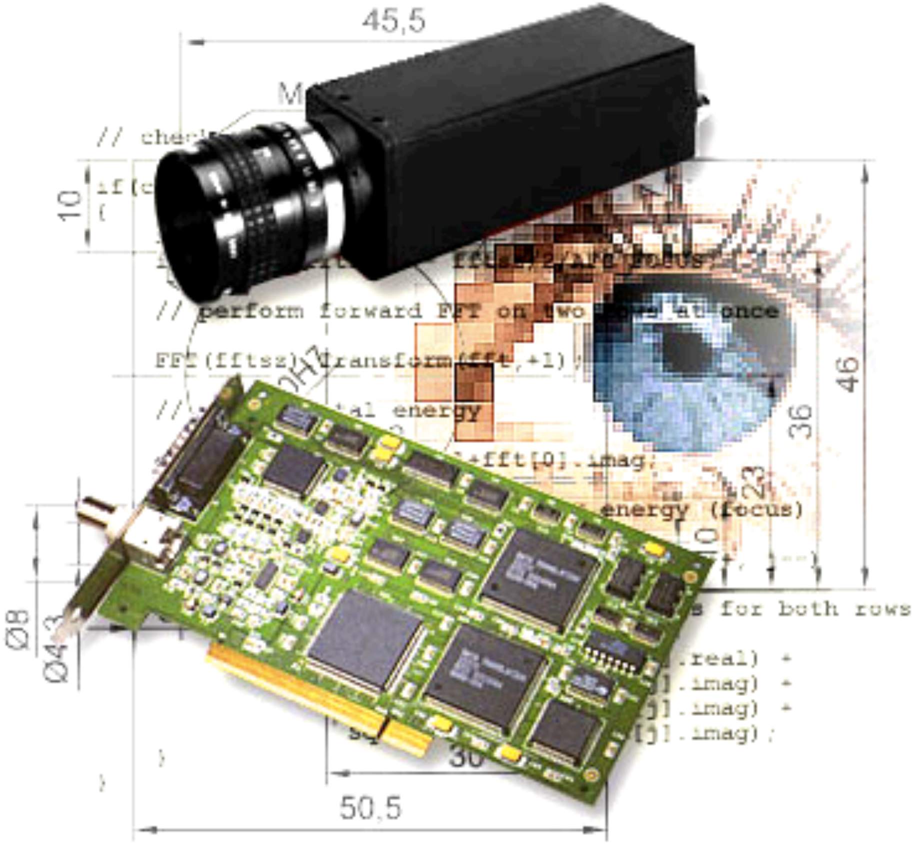
SERVIS robotics



AUTOMOTIVE robotics



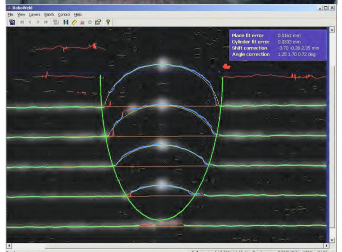
RETAIL monitoring

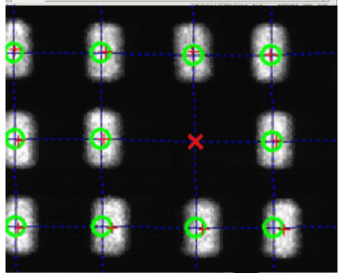


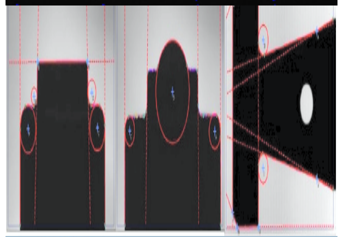
NEOVISION

Industrial Vision Systems

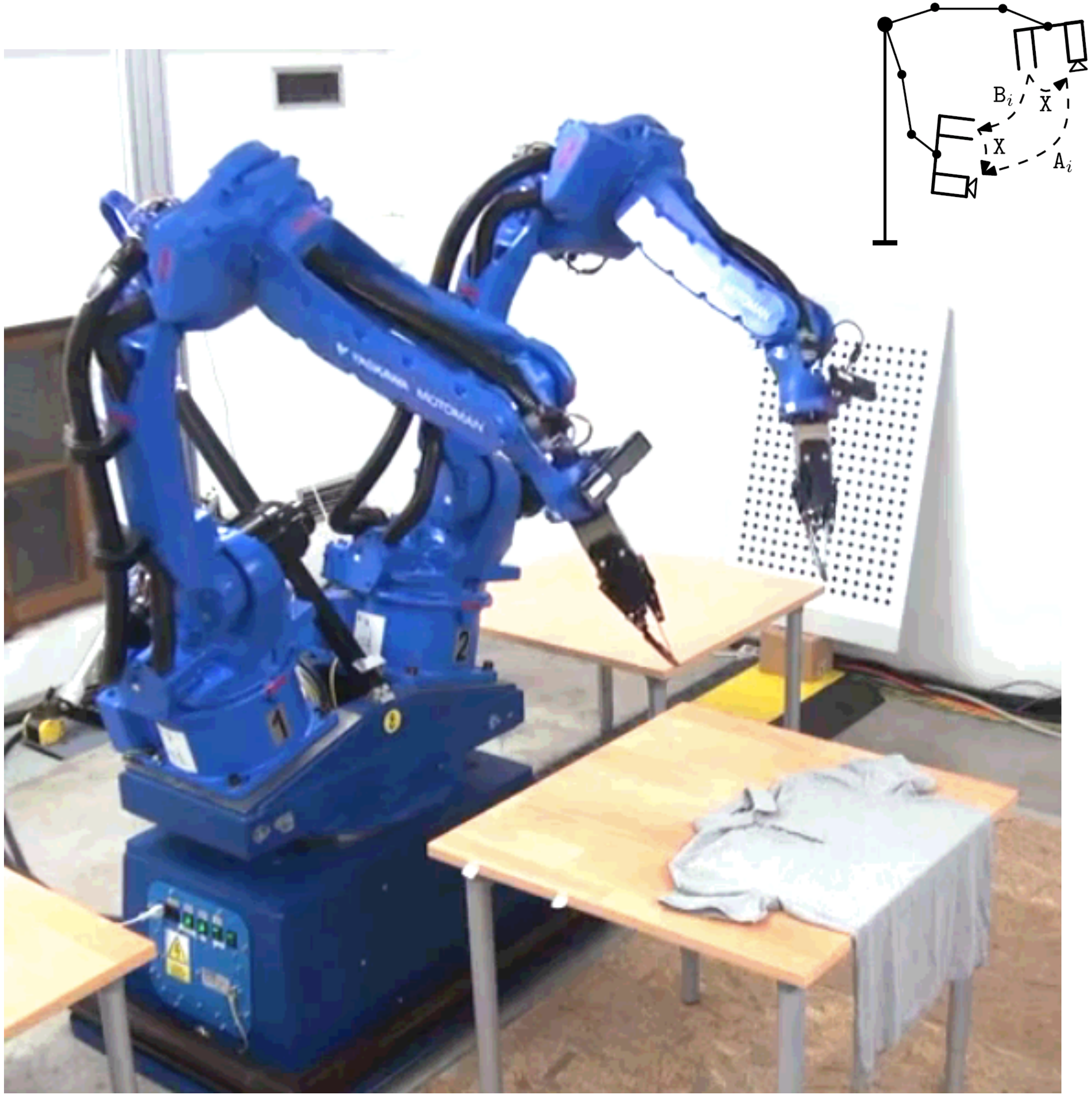
Machine vision and image processing
technology for industry and medicine

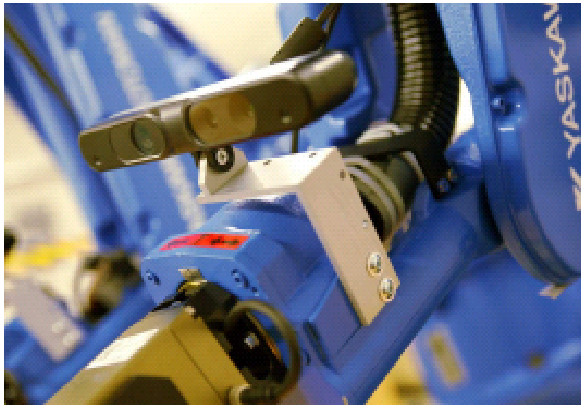




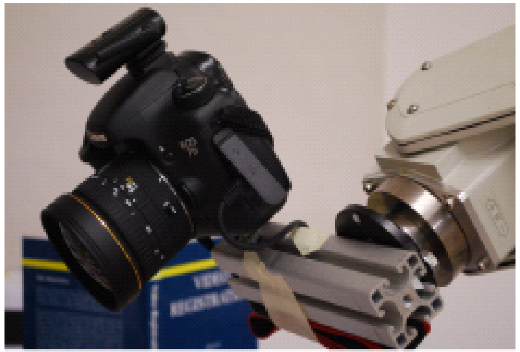


3D robotics





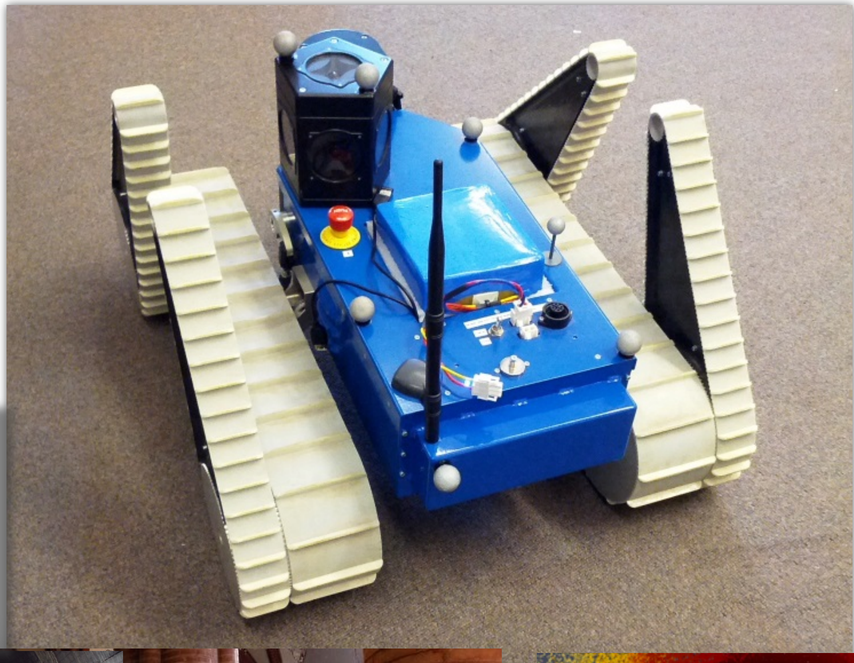
(a)

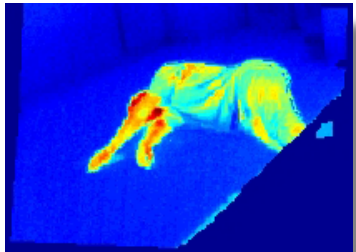


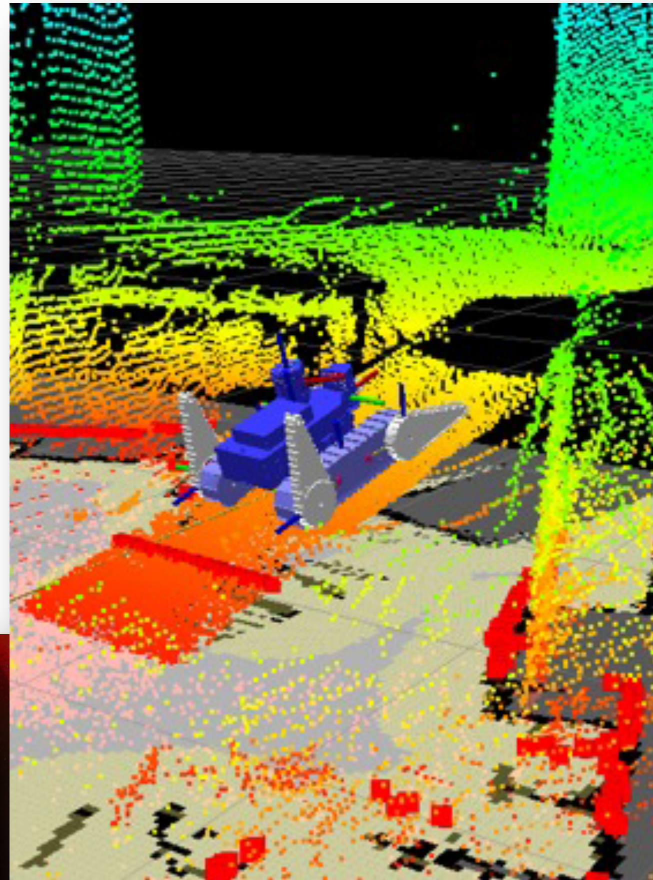
pajdla@cmp.felk.cvut.cz

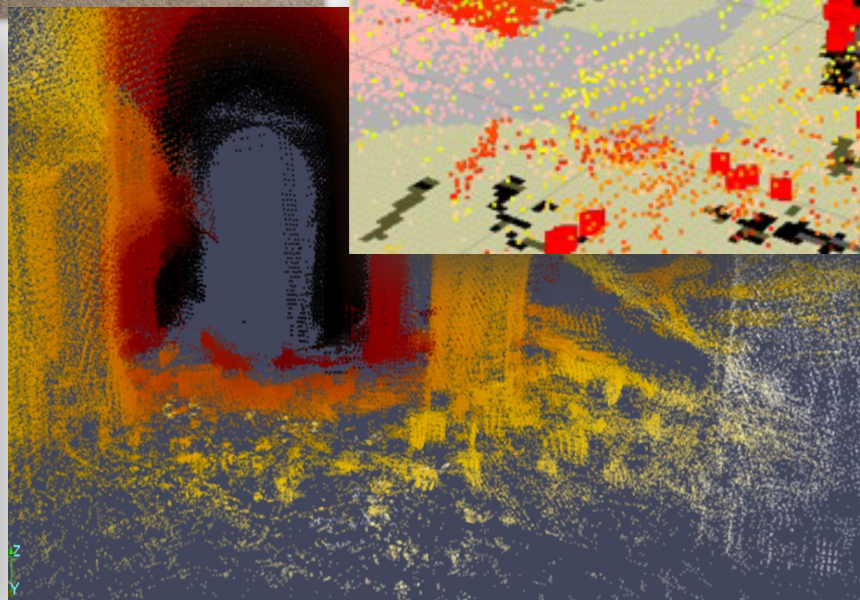
Intelligent Robotics

NIF-Ti
Natural Human-Robot Cooperation in Dynamic Environments

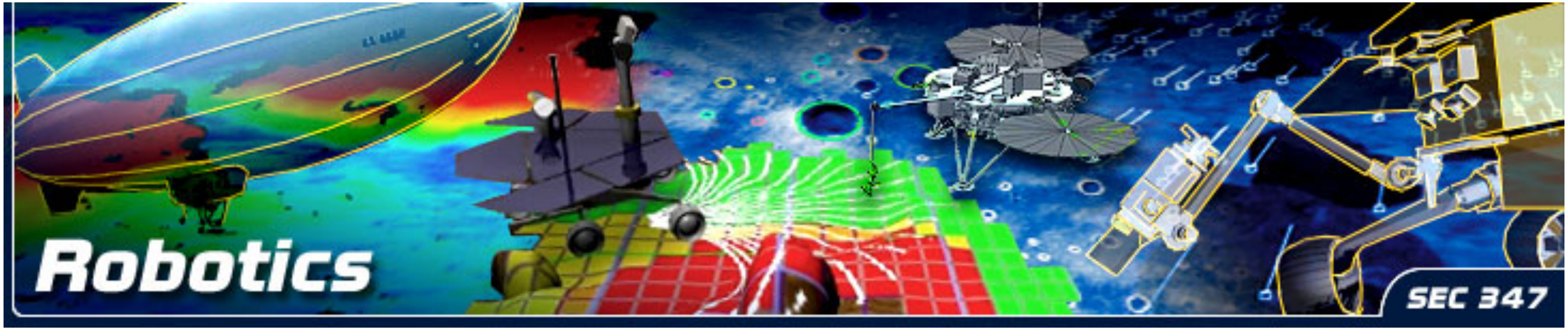


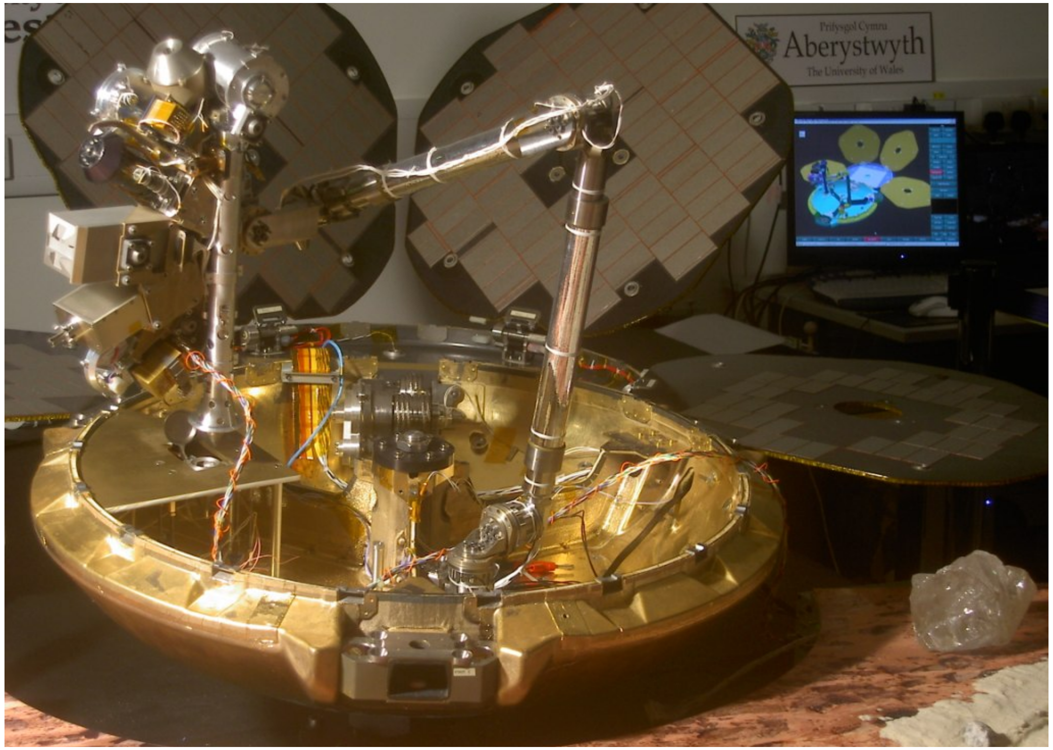


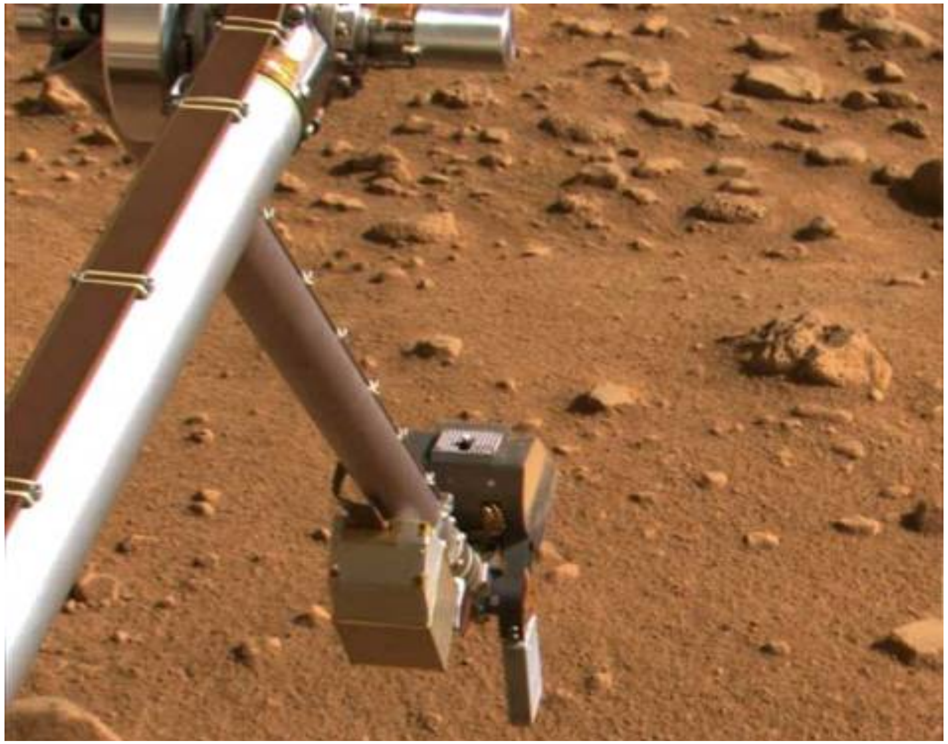




pajdla@cmp.felk.cvut.cz

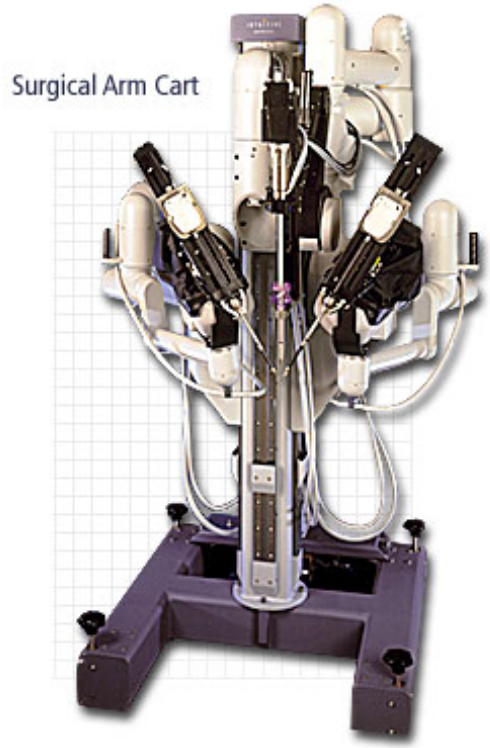


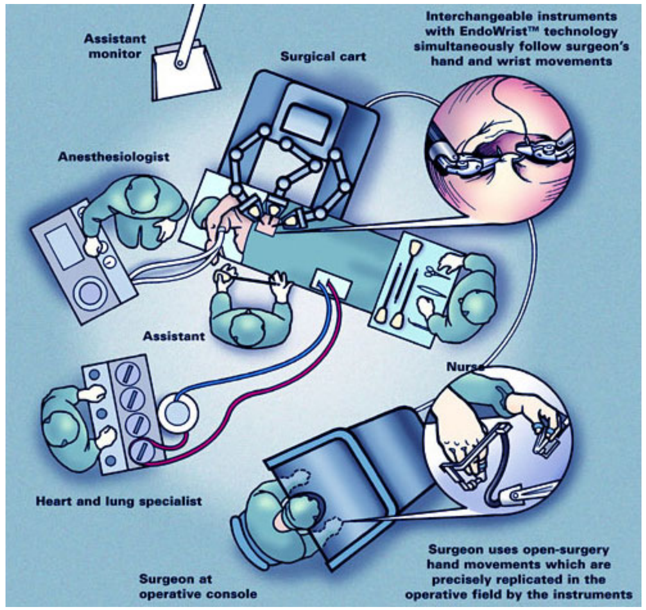




pajdla@cmp.felk.cvut.cz

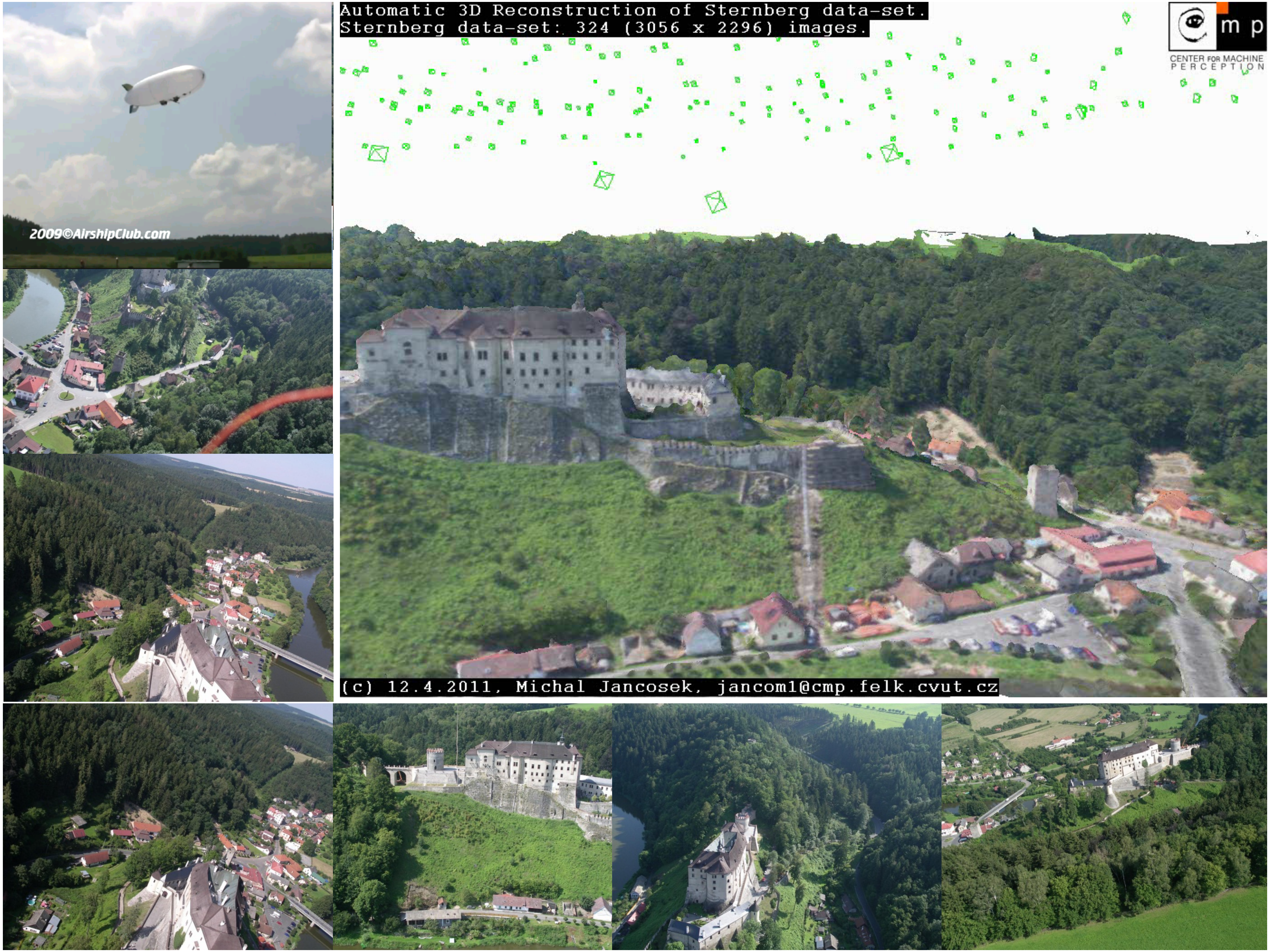
Precision for robotic surgery

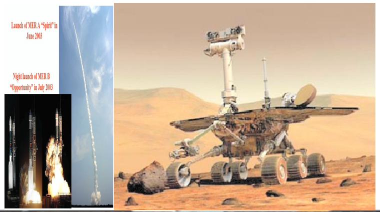




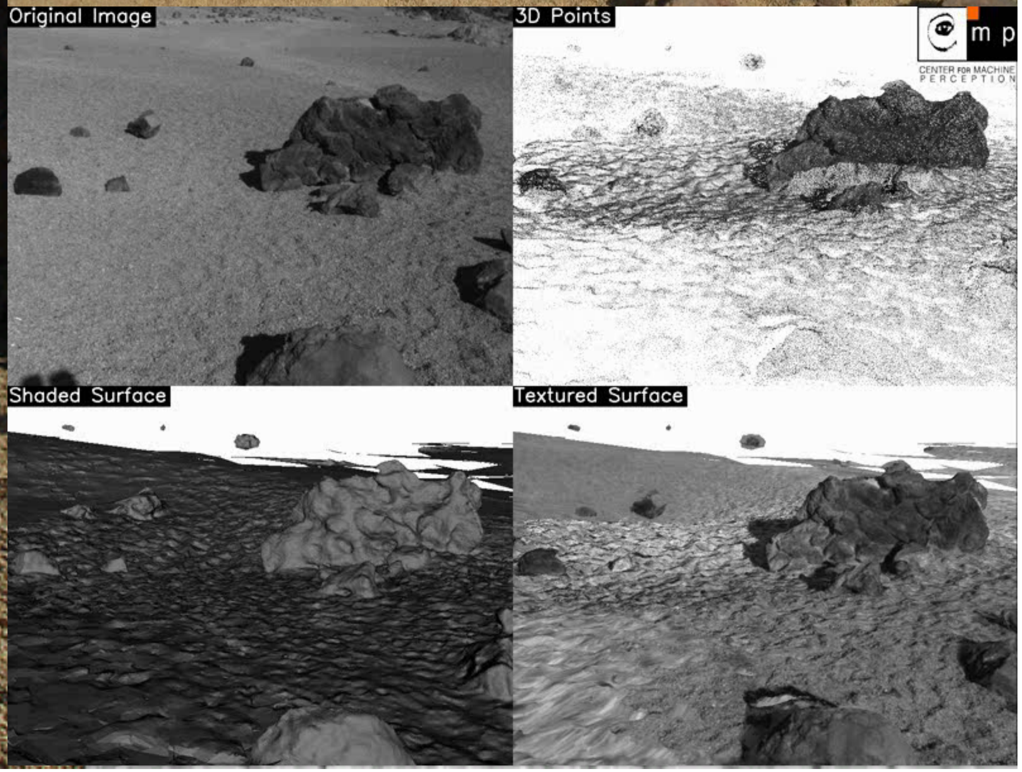
http://www.cts.usc.edu/rsi-article-robotputsuscatforefront.html


Original Image
3D Points
Shaded Surface
Textured Surface
cmp
CENTER FOR MACHINE PERCEPTION


2009©AirshipClub.com
Automatic 3D Reconstruction of Sternberg data-set.
Sternberg data-set: 324 (3056 x 2296) images.
CENTER FOR MACHINE PERCEPTION
(c) 12.4.2011, Michal Jancosek, jancom1@cmp.felk.cvut.cz



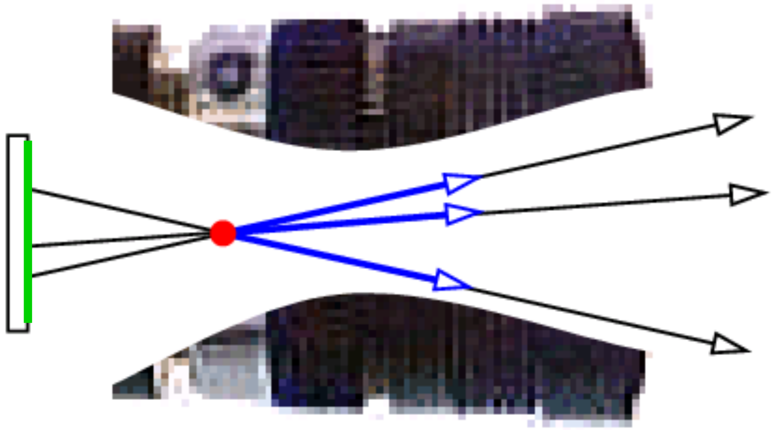

PRoVisG Field Trials Tenerife


Original Image
3D Points
CENTER FOR MACHINE PERCEPTION
Shaded Surface
Textured Surface

Perspective camera model

T. Pajdla. Elements of Geometry for Computer Vision
http://cmp.felk.cvut.cz/~pajdla/gvg/GVG-2014-Lecture.pdf

Camera



Digital cameras

Image projection model

1. Light extends along straight rays
2. Projection center
3. Projection plane

Image

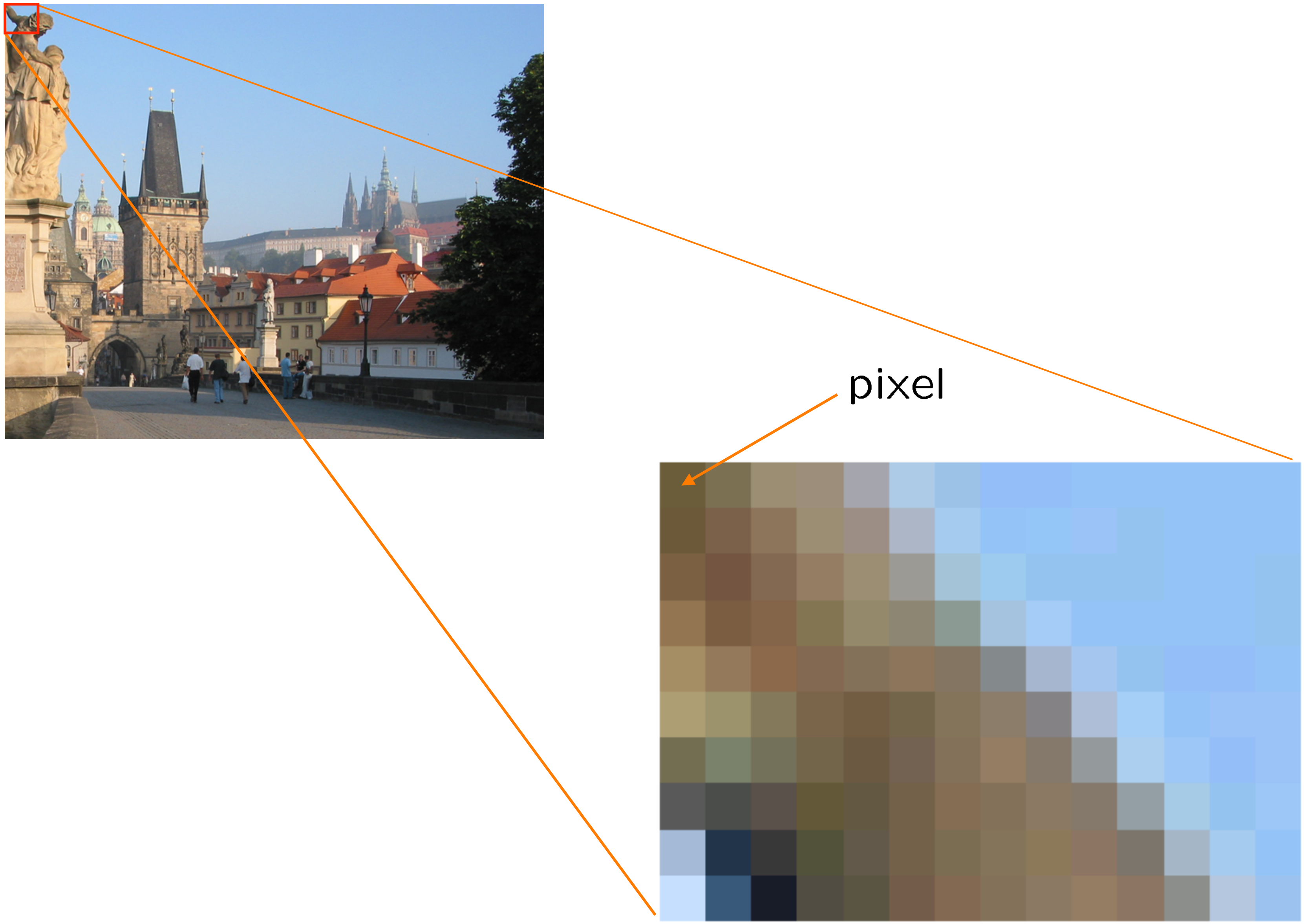
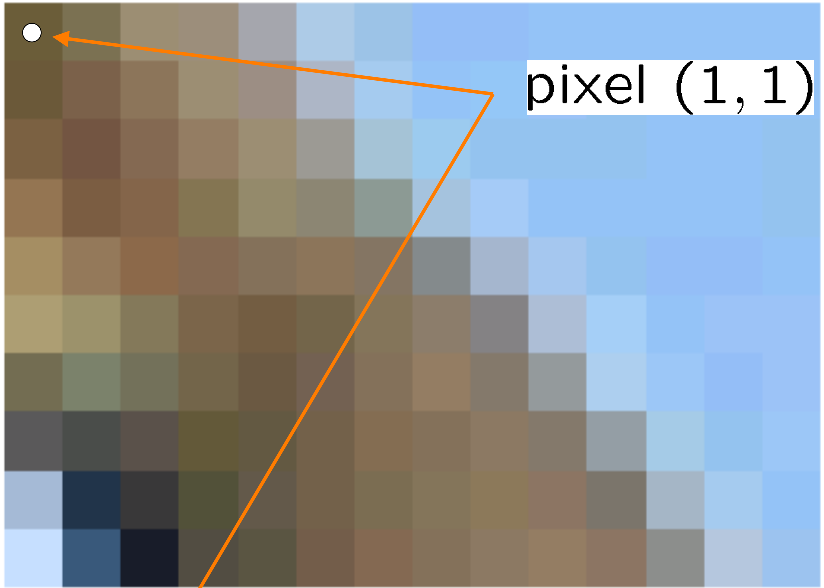


Image is a *m x n x 3* matrix in Matlab

$(0,0)^\top$

j

i




```
>>im = imread('karluv-most.jpg');
>>imagesc(im(1:10,1:14,:));
>>axis image
```

Indexing

i j R ← RGB

```
>>im(3,4,1)
ans = 144
```

1 … R

2 … G

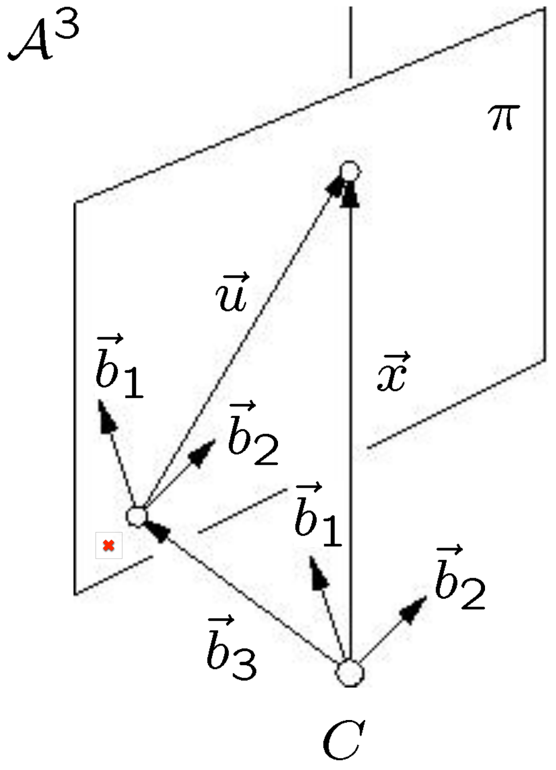
3 … B

j

i

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	107	126	155	154	161	172	155	145	149	151	150	150	150	149
2	107	121	139	156	154	174	166	146	149	152	150	148	149	151
3	125	116	130	144	159	157	167	159	149	150	148	150	151	151
4	148	125	128	132	147	140	140	167	160	146	148	151	150	150
5	164	149	136	128	131	137	130	133	160	164	150	151	149	151
6	168	154	134	122	114	118	134	137	131	173	165	148	152	153
7	112	127	116	112	106	116	130	144	129	145	170	152	147	152
8	89	76	91	98	102	115	128	131	139	130	148	167	151	152
9	164	37	62	85	98	115	127	131	139	139	125	165	164	149
10	194	62	27	84	95	112	128	131	137	150	139	142	176	158

Direction vector of projection ray



Miracle: The coordinates od the direction vector of a projection ray can be constructed by a adding "1" to image coordinates:

We measure in image

$$\vec{u} = u\,\vec{b}_1 + v\,\vec{b}_2 \sim \mathbf{u}_{(\vec{b}_1,\vec{b}_2)} = \begin{pmatrix} u \\ v \end{pmatrix}$$

Coordinate system with origin C

$$\begin{aligned} \beta &= (\vec{b}_1, \vec{b}_2, \vec{b}_3) \\ S &= (C, \beta) \\ \vec{b}_3 &= \varphi(C, o) \end{aligned}$$

Triangle equality

$$\vec{x} = \vec{u} + \vec{b}_3$$

$$\vec{x} = u\,\vec{b}_1 + v\,\vec{b}_2 + 1\,\vec{b}_3 \sim \mathbf{x}_\beta = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$$

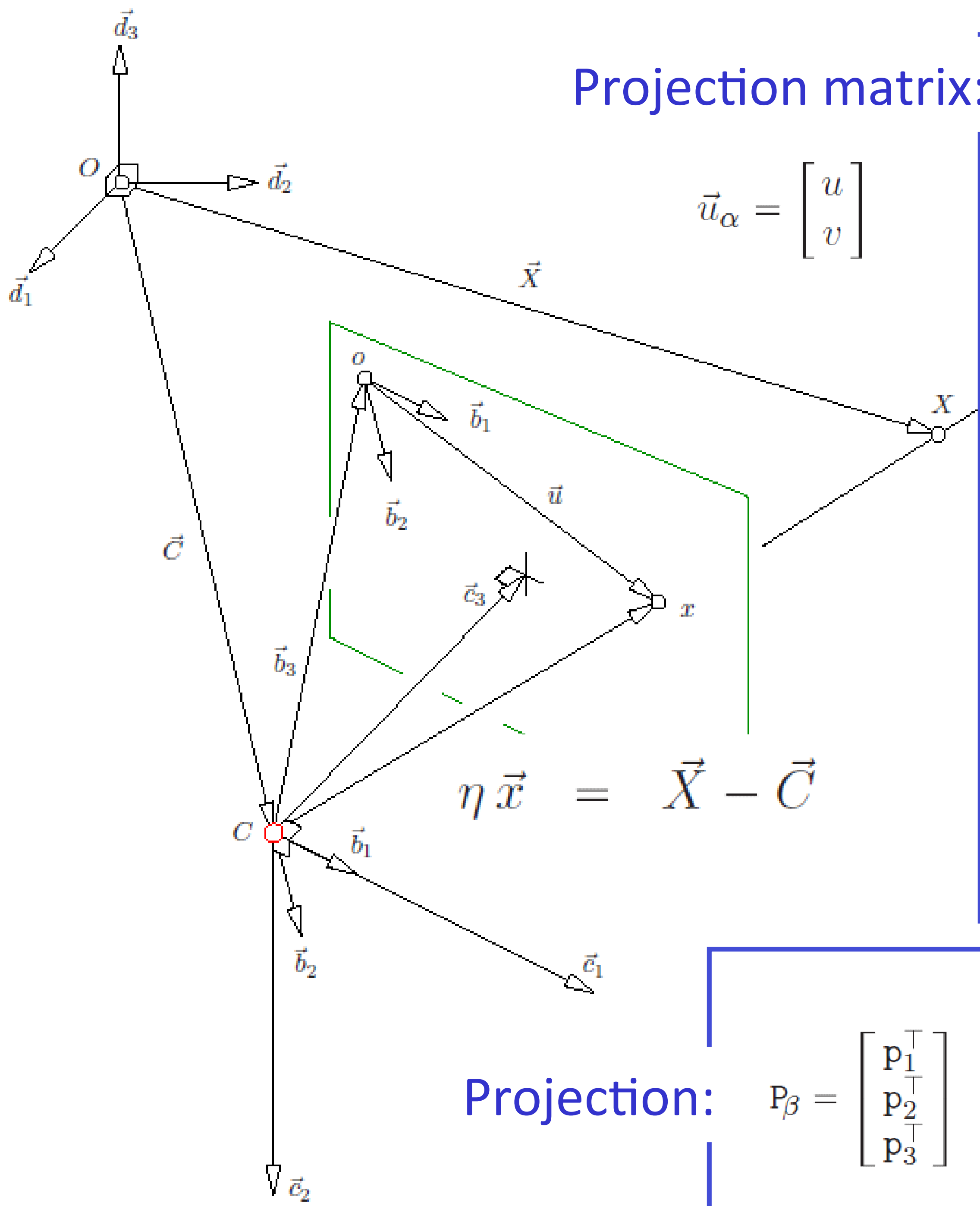


Figure 6.2: Coordinate systems of perspective camera.

$$\vec{u}_\alpha = \begin{bmatrix} u \\ v \end{bmatrix}$$

$$\eta\, \vec{x} = \vec{X} - \vec{C}$$

Projection matrix:

$$\eta\, \vec{x}_\beta = \vec{X}_\beta - \vec{C}_\beta$$

$$\eta \begin{bmatrix} \vec{u}_\alpha \\ 1 \end{bmatrix} = \vec{X}_\beta - \vec{C}_\beta$$

$$\vec{y}_\beta = \mathtt{A}\, \vec{y}_\delta$$

$$\eta \begin{bmatrix} \vec{u}_\alpha \\ 1 \end{bmatrix} = \mathtt{A}\, (\vec{X}_\delta - \vec{C}_\delta)$$

$$\eta \begin{bmatrix} \vec{u}_\alpha \\ 1 \end{bmatrix} = \mathtt{A} \left[\mathtt{I} \mid -\vec{C}_\delta \right] \begin{bmatrix} \vec{X}_\delta \\ 1 \end{bmatrix}$$

$$\eta \begin{bmatrix} \vec{u}_\alpha \\ 1 \end{bmatrix} = \mathtt{P}_\beta \begin{bmatrix} \vec{X}_\delta \\ 1 \end{bmatrix}$$

$$\eta\, \vec{x}_\beta = \mathtt{P}_\beta \begin{bmatrix} \vec{X}_\delta \\ 1 \end{bmatrix}$$

Projection:

$$\mathtt{P}_\beta = \begin{bmatrix} \mathtt{p}_1^\top \\ \mathtt{p}_2^\top \\ \mathtt{p}_3^\top \end{bmatrix} \quad \text{and} \quad \mathtt{X} = \begin{bmatrix} \vec{X}_\delta \\ 1 \end{bmatrix} \qquad \vec{u}_\alpha = \begin{bmatrix} \dfrac{\mathtt{p}_1^\top \mathtt{X}}{\mathtt{p}_3^\top \mathtt{X}} \\ \dfrac{\mathtt{p}_2^\top \mathtt{X}}{\mathtt{p}_3^\top \mathtt{X}} \end{bmatrix}$$

pajdla@cmp.felk.cvut.cz

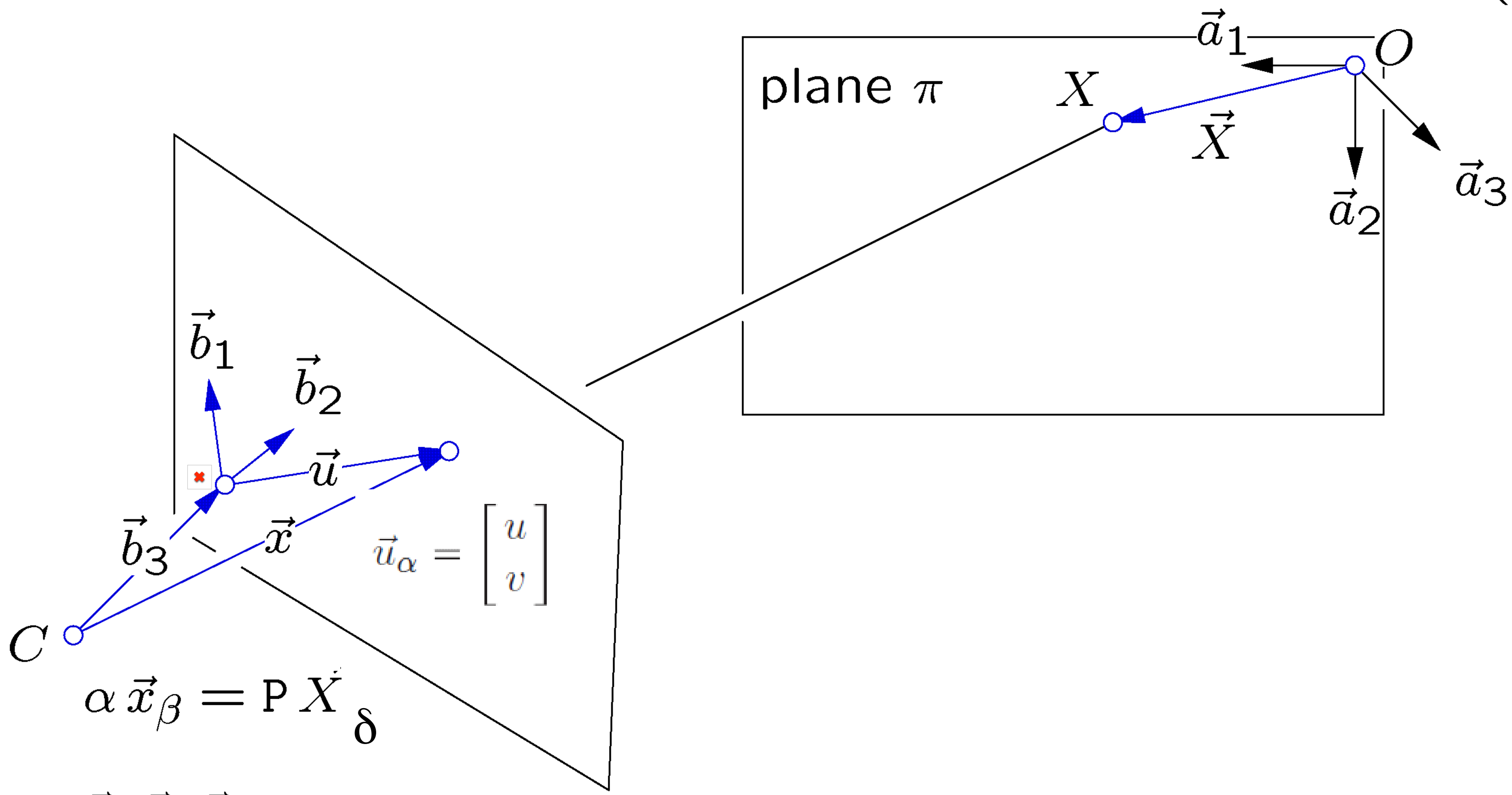
Homography

pajdla@cmp.felk.cvut.cz

Camera observes a plane

World coordinate system

$$\delta = (\vec{a}_1, \vec{a}_2, \vec{a}_3)$$
$$W = (O, \delta)$$



$\beta = (\vec{b}_1, \vec{b}_2, \vec{b}_3)$
$S = (C, \beta)$
Camera coordinate system

$$\zeta \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \zeta\, \vec{x}_\beta = \mathrm{P} \begin{bmatrix} \vec{X}_\delta \\ 1 \end{bmatrix} = \begin{bmatrix} \mathrm{p}_1 & \mathrm{p}_2 & \mathrm{p}_3 & \mathrm{p}_4 \end{bmatrix} \begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \mathrm{p}_1 & \mathrm{p}_2 & \mathrm{p}_4 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathrm{H}\, \vec{y}_\tau$$

COMPUTING THE HOMOGRAPHY
from 4 correspondences

pajdla@cmp.felk.cvut.cz

Computing the homography

$\exists\, \mathtt{H} \in \mathbb{R}^{3\times 3}$, rank $\mathtt{H} = 3$, so that $\forall\, (u,v) \overset{corr}{\leftrightarrow} (u',v')\ \exists \alpha \in \mathbb{R}$:

$$\alpha \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = \mathtt{H} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$$

Introduce symbols for rows of homography $\mathtt{H}$

$$\mathtt{H} = \begin{pmatrix} \mathbf{h}_1^\top \\ \mathbf{h}_2^\top \\ \mathbf{h}_3^\top \end{pmatrix}$$

and rewrite the above matrix equation as

$$\begin{aligned} \alpha\, u' &= \mathbf{h}_1^\top \mathbf{x} \\ \alpha\, v' &= \mathbf{h}_2^\top \mathbf{x} \\ \alpha &= \mathbf{h}_3^\top \mathbf{x} \end{aligned}$$

Computing the homography

Eliminate α from the first two equations using the third one

$$\begin{aligned}(\mathbf{h}_3^\top \mathbf{x})\, u' &= \mathbf{h}_1^\top \mathbf{x} \\ (\mathbf{h}_3^\top \mathbf{x})\, v' &= \mathbf{h}_2^\top \mathbf{x}\end{aligned}$$

move all to the left hand side and reshape it using $\mathbf{x}^\top \mathbf{y} = \mathbf{y}^\top \mathbf{x}$

$$\begin{aligned}\mathbf{x}^\top \mathbf{h}_1 - (u' \mathbf{x}^\top)\, \mathbf{h}_3 &= 0 \\ \mathbf{x}^\top \mathbf{h}_2 - (v' \mathbf{x}^\top)\, \mathbf{h}_3 &= 0\end{aligned}$$

Introduce notation

$$\mathbf{h} = \begin{pmatrix} \mathbf{h}_1^\top & \mathbf{h}_2^\top & \mathbf{h}_3^\top \end{pmatrix}^\top$$

and express the above two equations in a matrix form

$$\begin{pmatrix} u & v & 1 & 0 & 0 & 0 & -u'u & -u'v & -u' \\ 0 & 0 & 0 & u & v & 1 & -v'u & -v'v & -v' \end{pmatrix} \mathbf{h} = 0$$

pajdla@cmp.felk.cvut.cz

Computing the homography

Notice that $\mathtt{A}$ can be written in the form

$$\mathtt{A} = \begin{pmatrix} u_1 & v_1 & 1 & 0 & 0 & 0 & -u'_1 u_1 & -u'_1 v_1 & -u'_1 \\ u_2 & v_2 & 1 & 0 & 0 & 0 & -u'_2 u_2 & -u'_2 v_2 & -u'_2 \\ & & & & \vdots & & & & \\ 0 & 0 & 0 & u_1 & v_1 & 1 & -v'_1 u_1 & -v'_1 v_1 & -v'_1 \\ 0 & 0 & 0 & u_2 & v_2 & 1 & -v'_2 u_2 & -v'_2 v_2 & -v'_2 \\ & & & & \vdots & & & & \end{pmatrix}$$

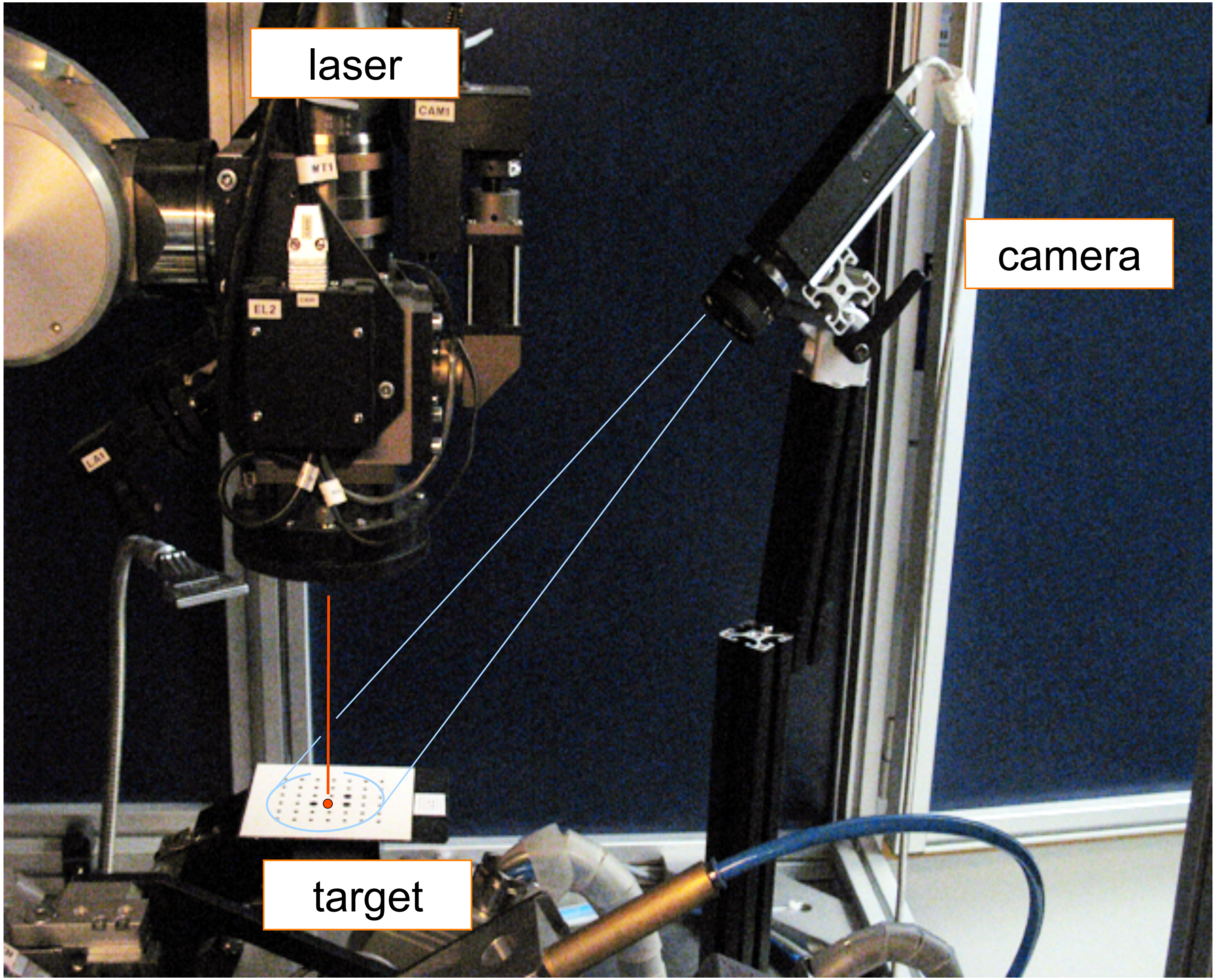
which can be rewritten more concisely as

$$\mathtt{A} = \begin{pmatrix} \mathbf{x}_1^\top & \mathbf{0} & -u'_1\,\mathbf{x}_1^\top \\ \mathbf{x}_2^\top & \mathbf{0} & -u'_2\,\mathbf{x}_2^\top \\ & \vdots & \\ \mathbf{0} & \mathbf{x}_1^\top & -v'_1\,\mathbf{x}_1^\top \\ \mathbf{0} & \mathbf{x}_2^\top & -v'_2\,\mathbf{x}_2^\top \\ & \vdots & \end{pmatrix}$$

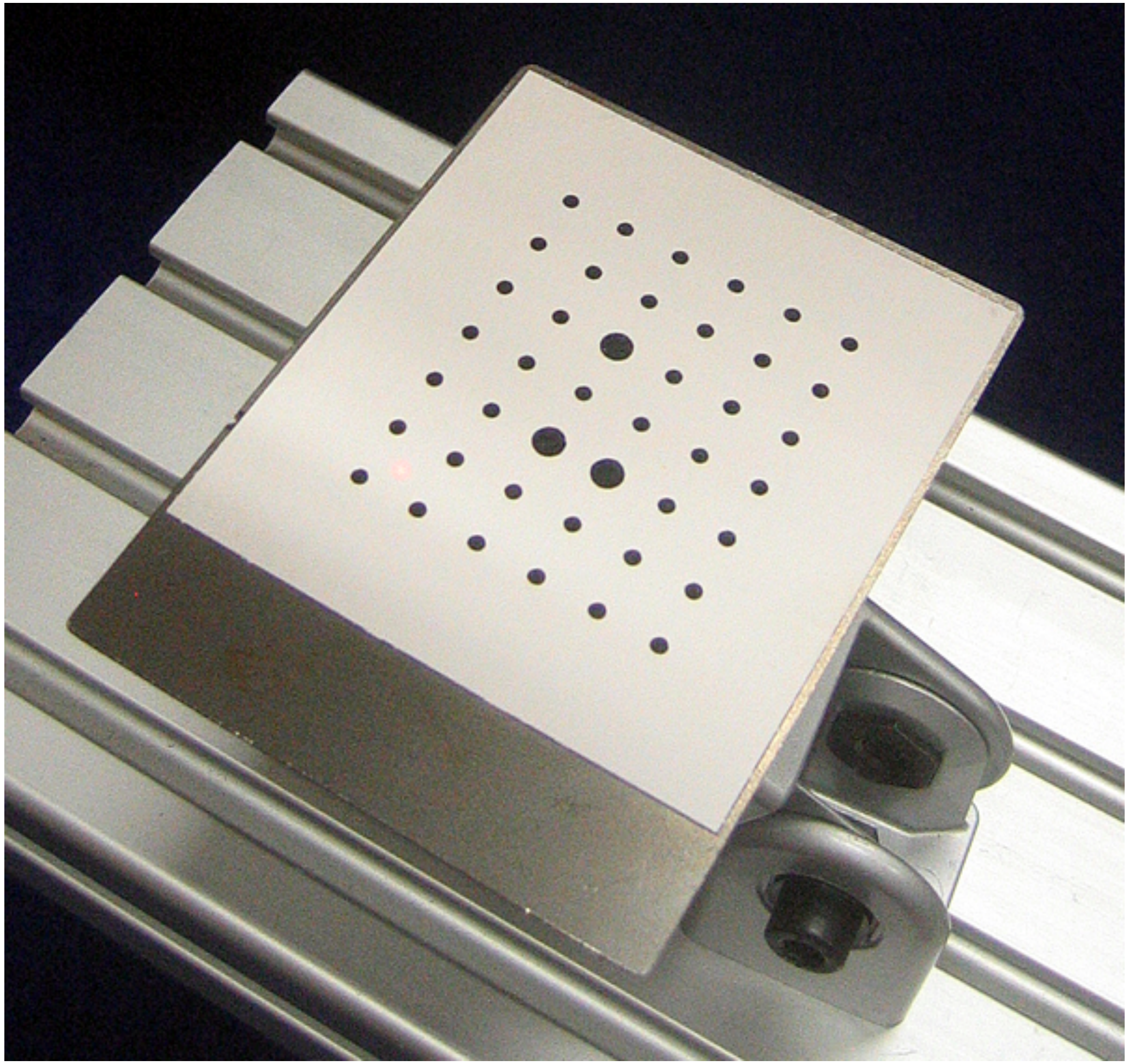
Computing the homography from 4 points on 2 lines in Matlab

```
% 4 points

>>x = [0 0 1;1 0 1;0 1 1;1 1 1]';
>>y = [1 1 1;1 0 1;0 1 1;0 0 1]';

% the 2-line algorithm

>>A       = [[x' zeros(size(x')) [-y(1,:)'*ones(1,3)].*(x')]
             [zeros(size(x')) x' [-y(2,:)'*ones(1,3)].*(x')]];
>>H       = reshape(null(A),3,3)';

% verification

>> e = y - (H*x)./[[1;1;1]*(H(3,:)*x)]

e =

  1.0e-015 *

         0    0.0481   -0.2220   -0.4441
         0    0.2220    0.0961   -0.2220
         0         0         0         0
```

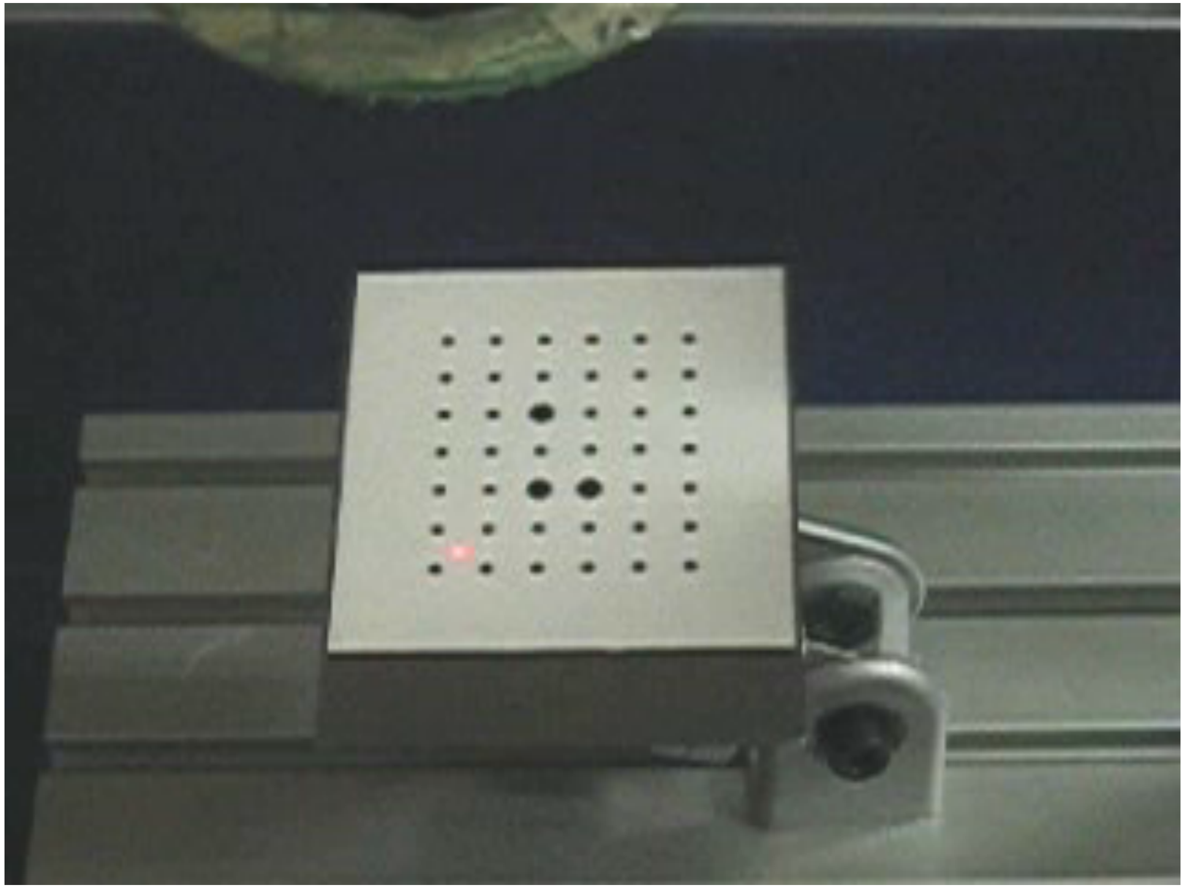
Setup



Calibration tatget

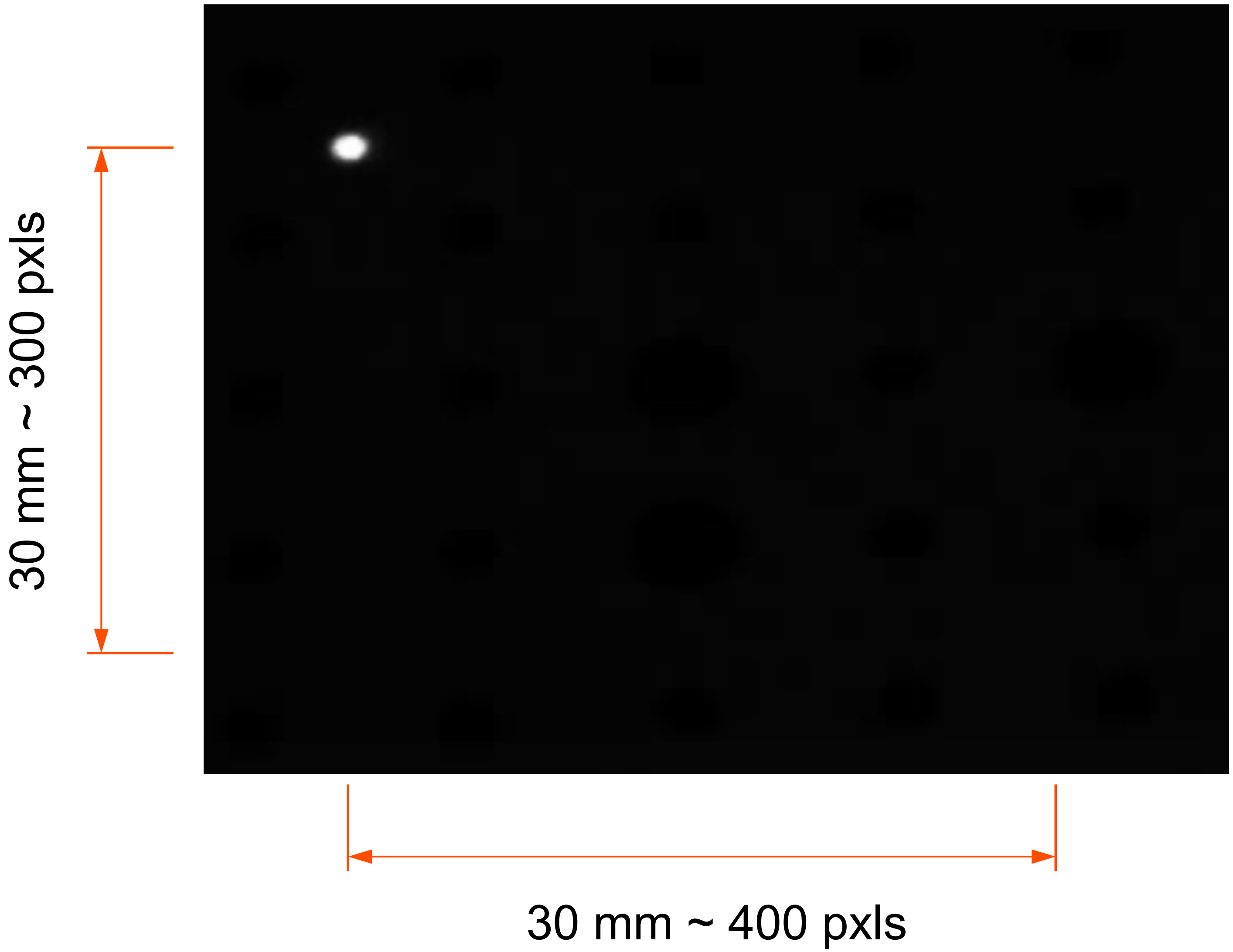


Laser spot



Laser spot in the camera view

1 pxl ~ 0.1 mm, i.e. resolution 0.1 pxl ~ 0.01 mm

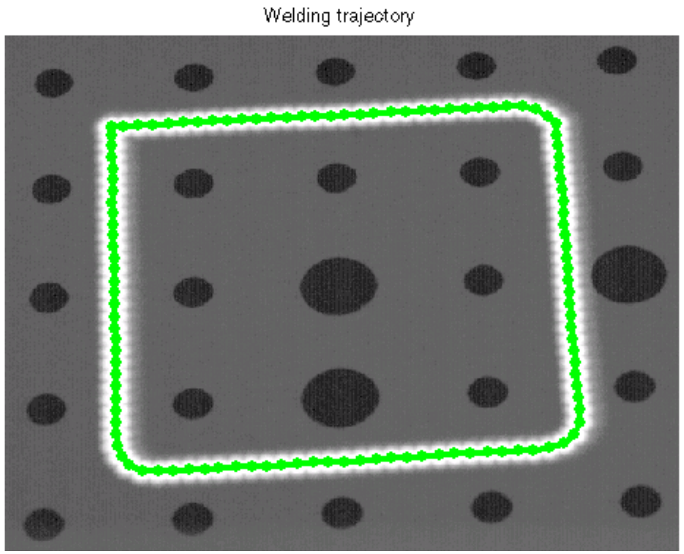


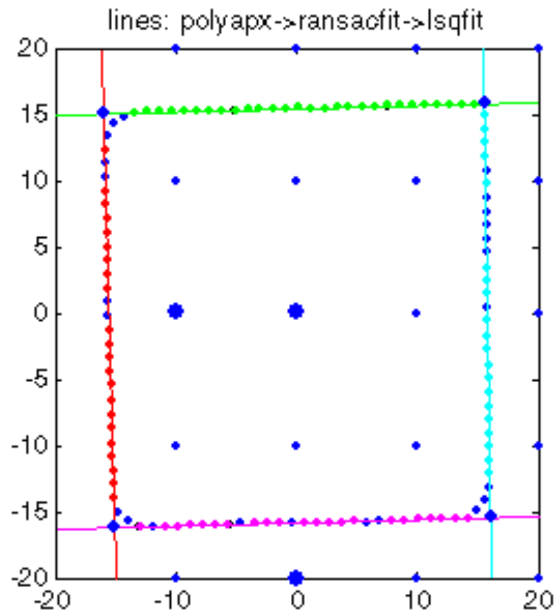
pajdla@cmp.felk.cvut.cz

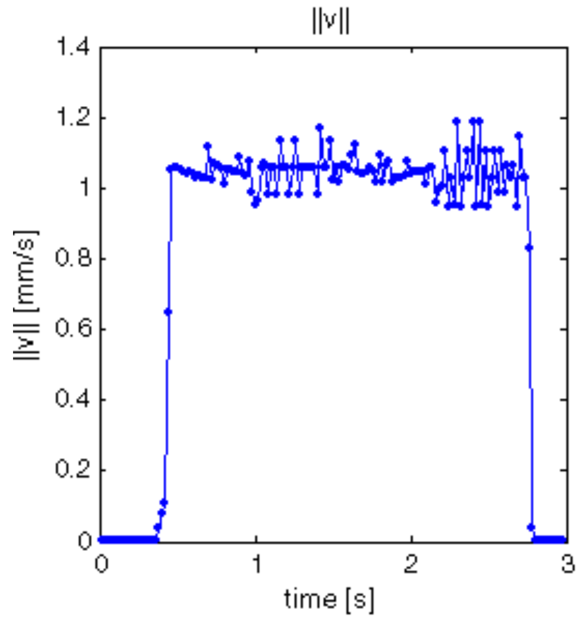
abb-sq-02:

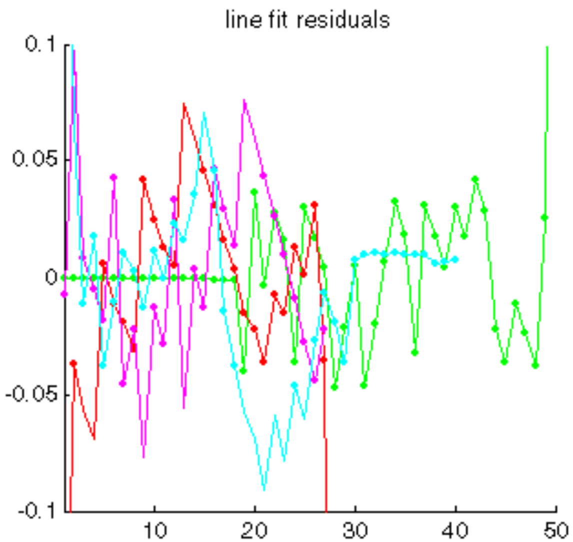
VAR speeddata speed:= [15,15,0,0];

CONST zonedata zn := [FALSE, 3.0, 3.0, 3.0, 0.30, 3.0, 0.30];



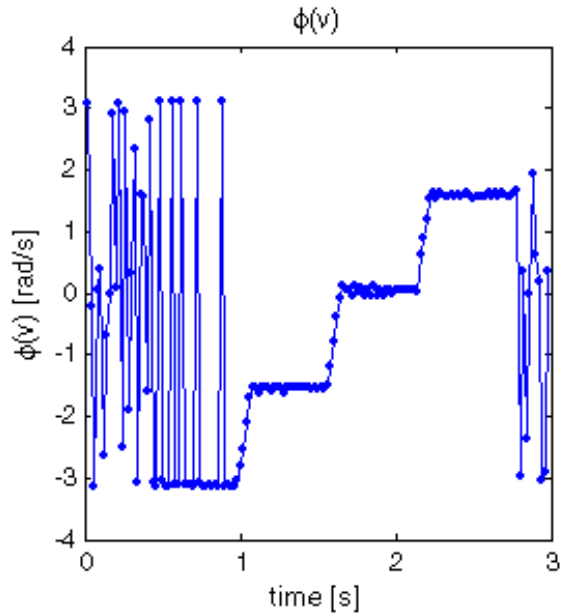






4 - corners

segment 1		2	3	4	5	6
D [mm]	31.31	31.16	31.26	31.49	44.22	44.32
dD [mm]	0.01	-0.15	-0.05	0.18	-0.05	0.05
dphi [deg]	-0.12	-0.49	0.38	0.23	-0.07	NaN

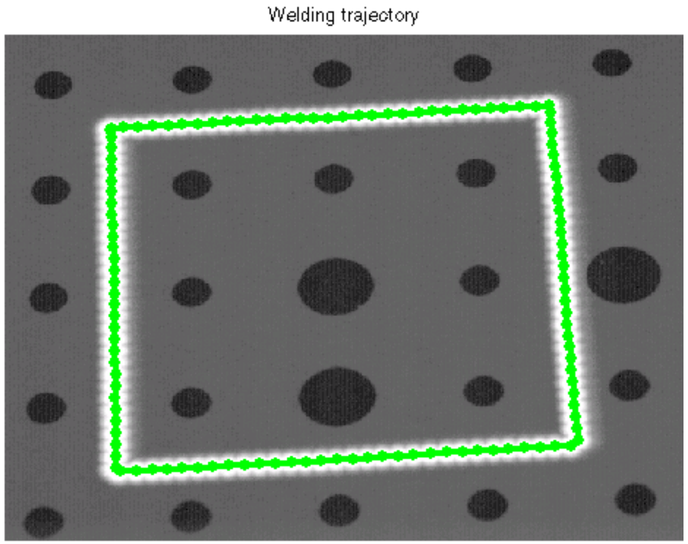


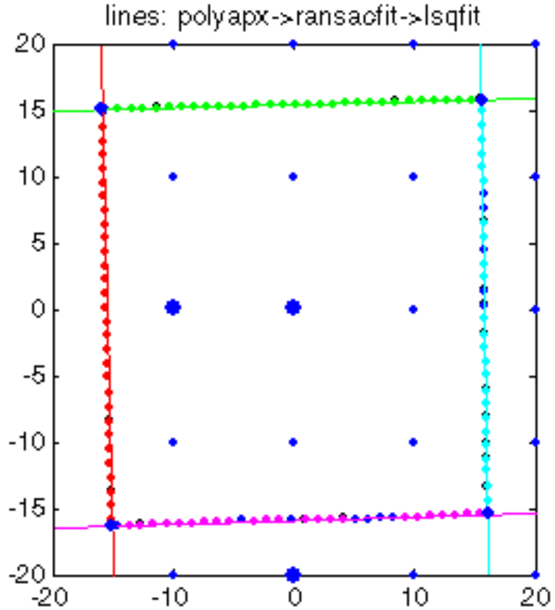
pajdla@cmp.felk.cvut.cz

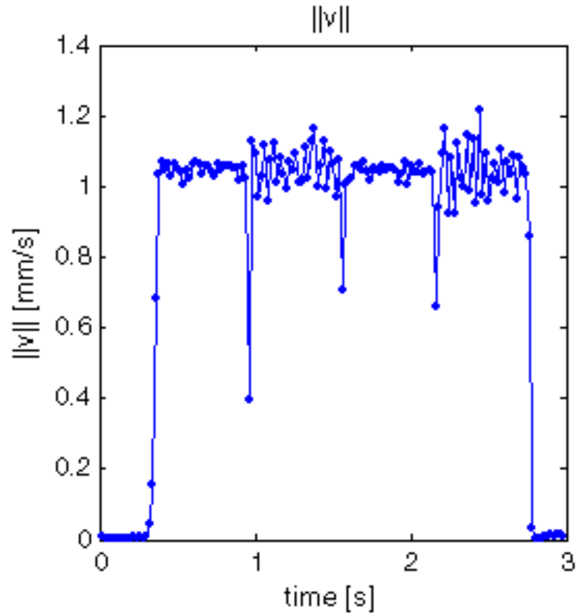
abb-sq-03:

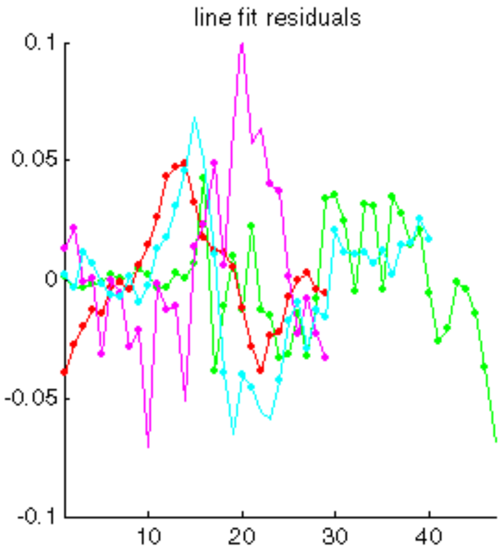
VAR speeddata speed:= [15,15,0,0];

CONST zonedata zn := [FALSE, 0.1, 0.1, 0.1, 0.01, 0.1, 0.01];



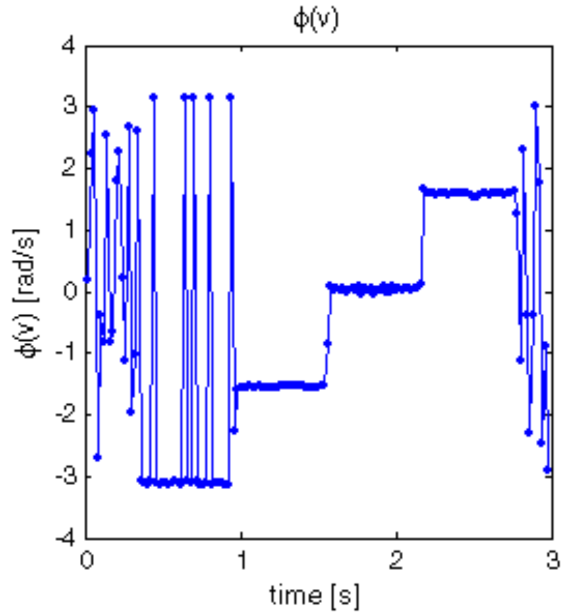






4 - corners

	segment 1	2	3	4	5	6
D [mm]	31.36	31.19	31.20	31.48	44.18	44.36
dD [mm]	0.06	-0.12	-0.11	0.17	-0.09	0.09
dphi [deg]	0.12	-0.65	0.35	0.19	-0.10	NaN

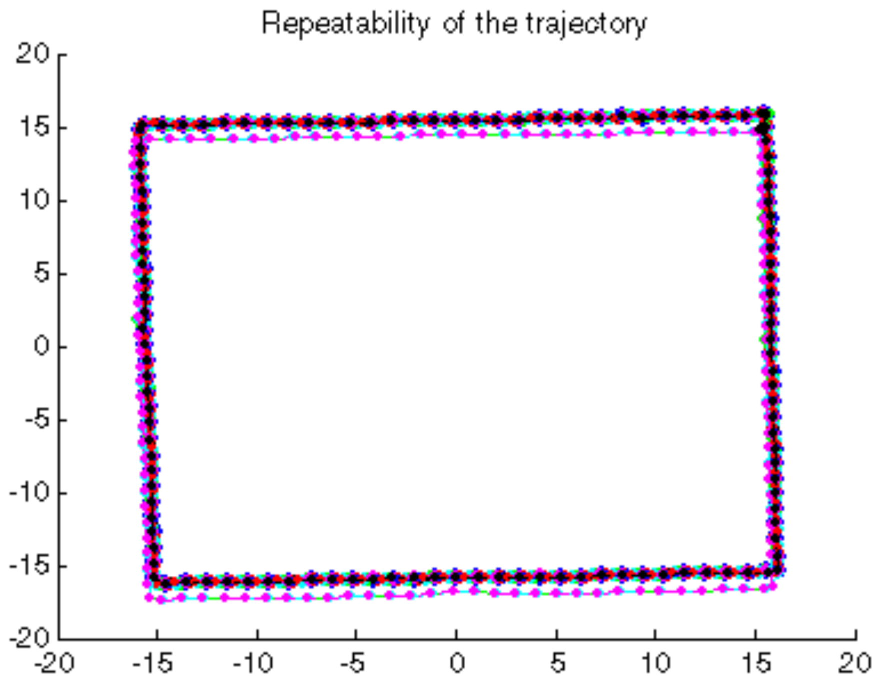


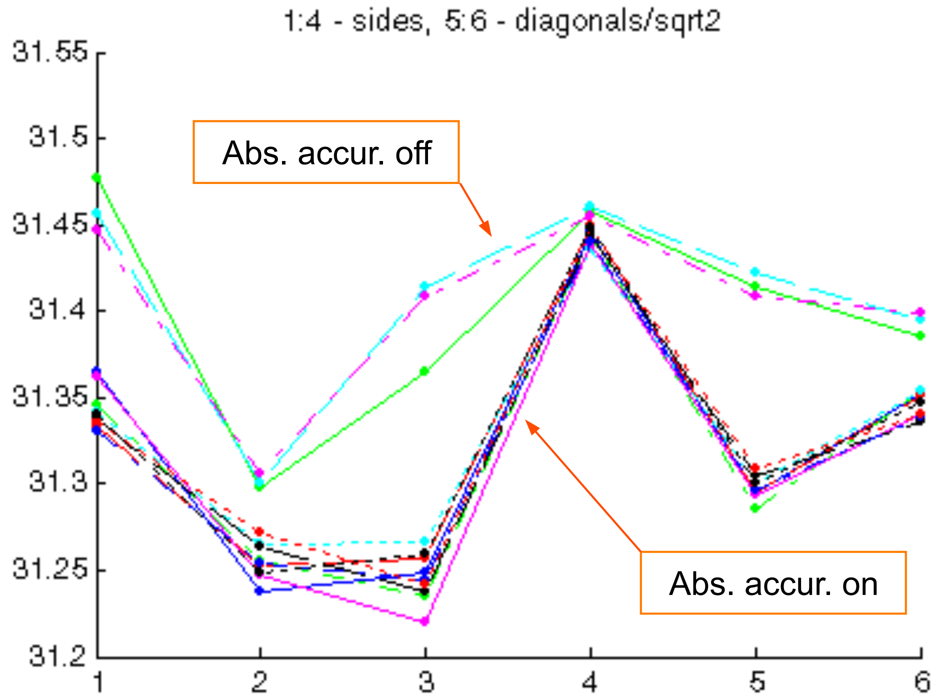
pajdla@cmp.felk.cvut.cz

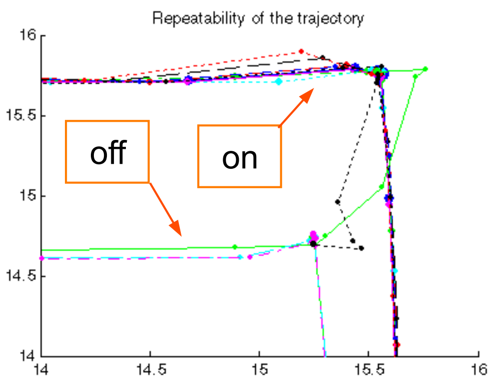
abb-sq-03:

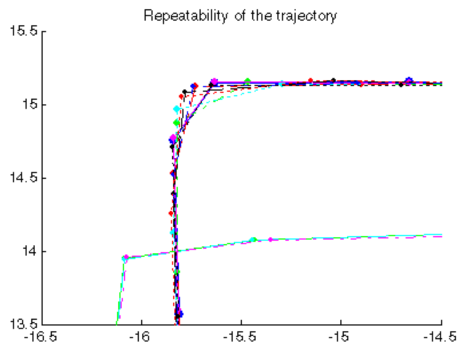
VAR speeddata speed:= [15,15,0,0];

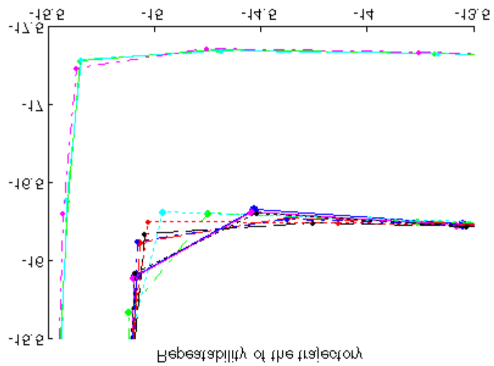
CONST zonedata zn := [FALSE, 0.1, 0.1, 0.1, 0.01, 0.1, 0.01];









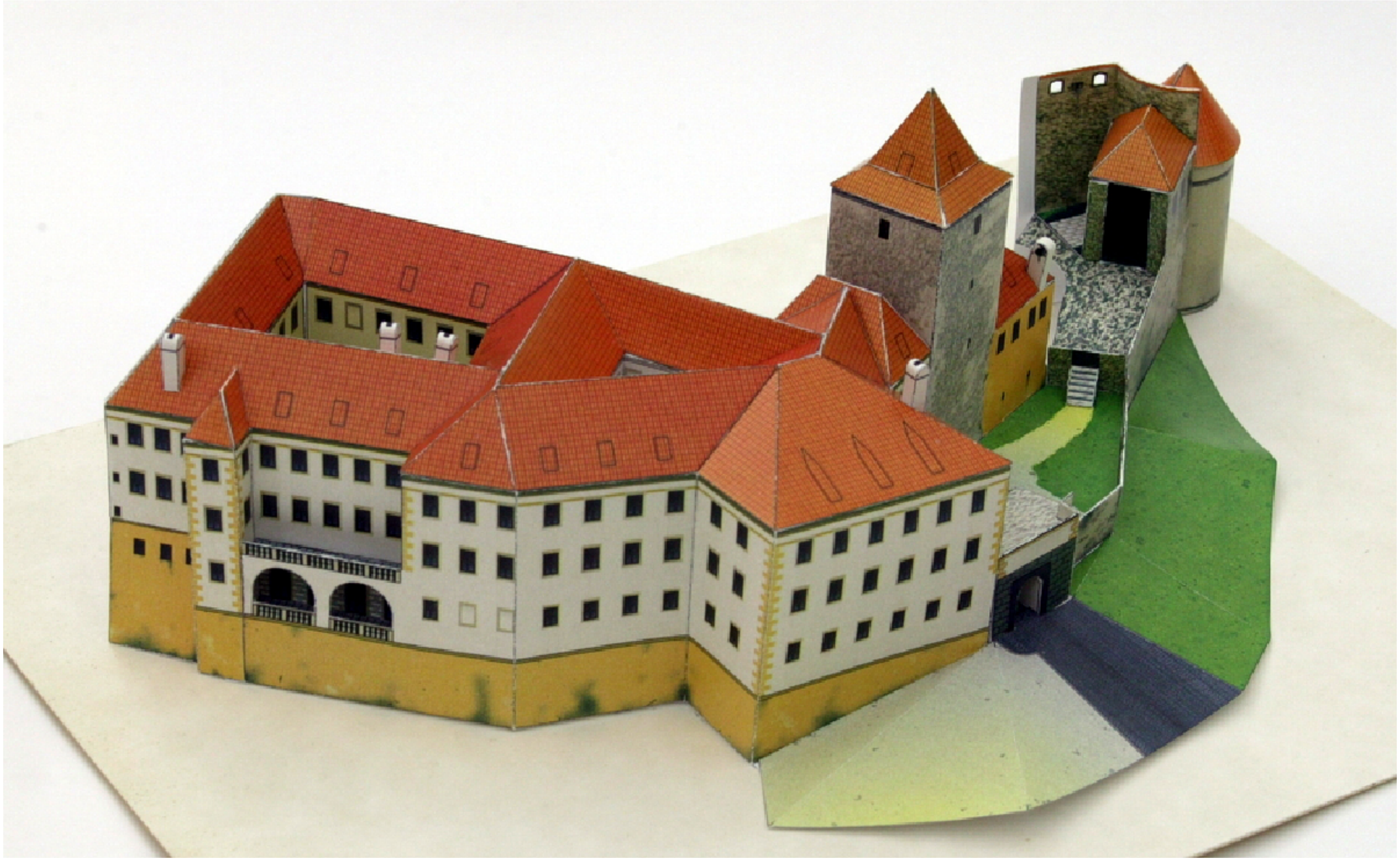


pajdla@cmp.felk.cvut.cz

pajdla@cmp.felk.cvut.cz

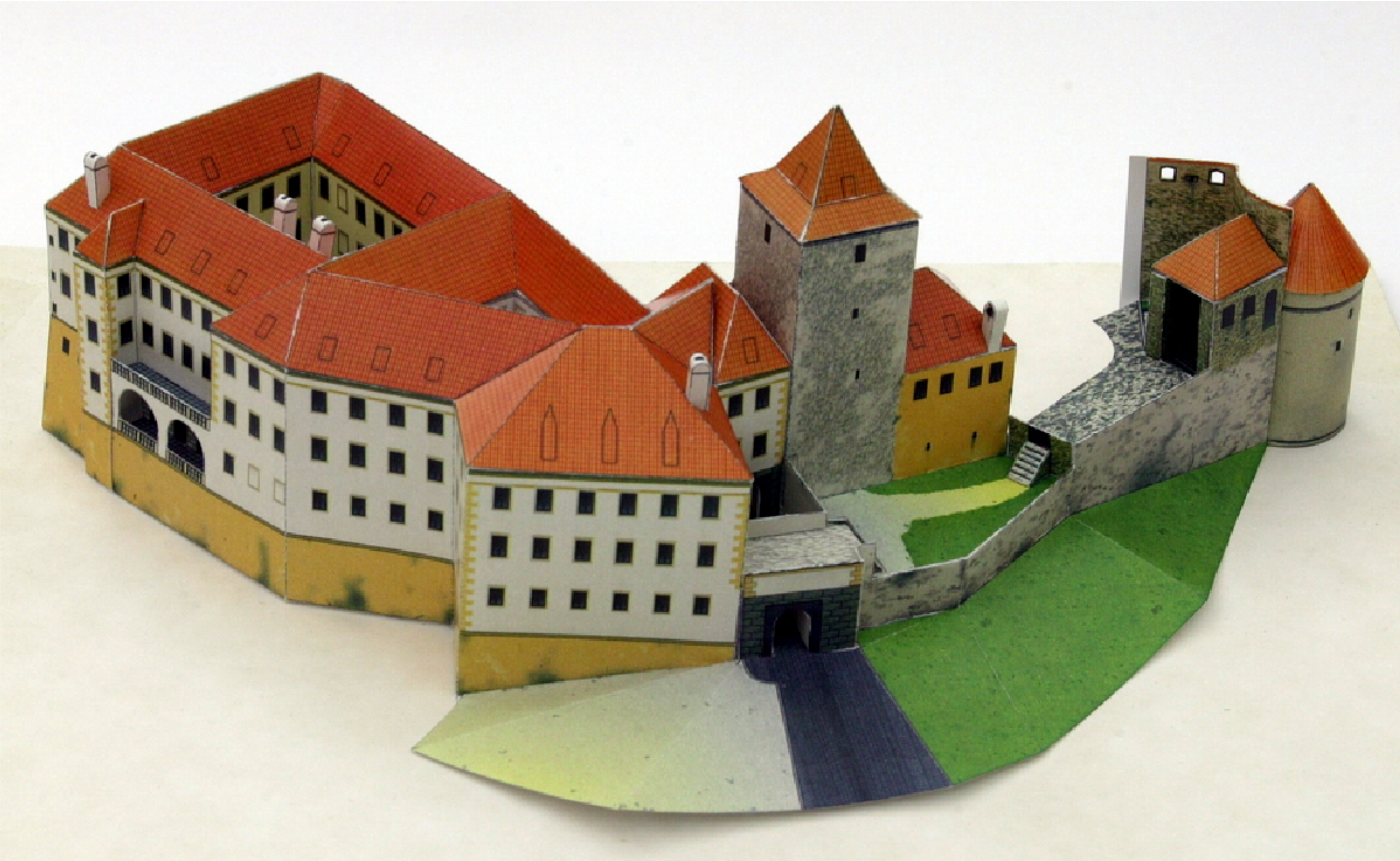

Original Image
3D Points
CENTER FOR MACHINE PERCEPTION
Shaded Surface
Textured Surface

IMAGE No. 1



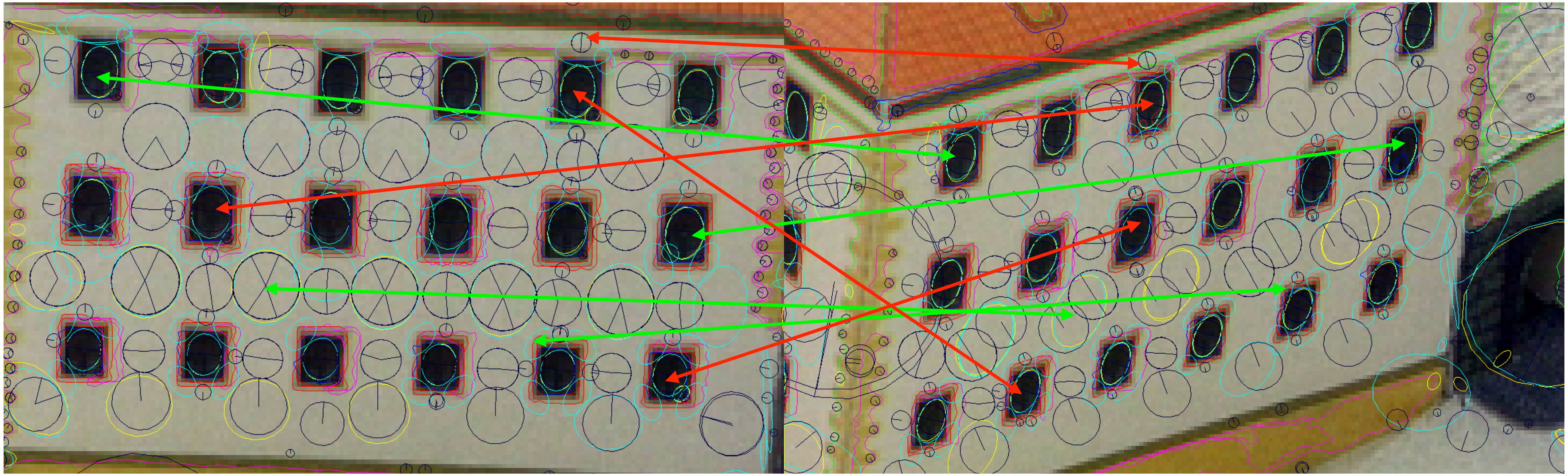
pajdla@cmp.felk.cvut.cz

IMAGE No. 2



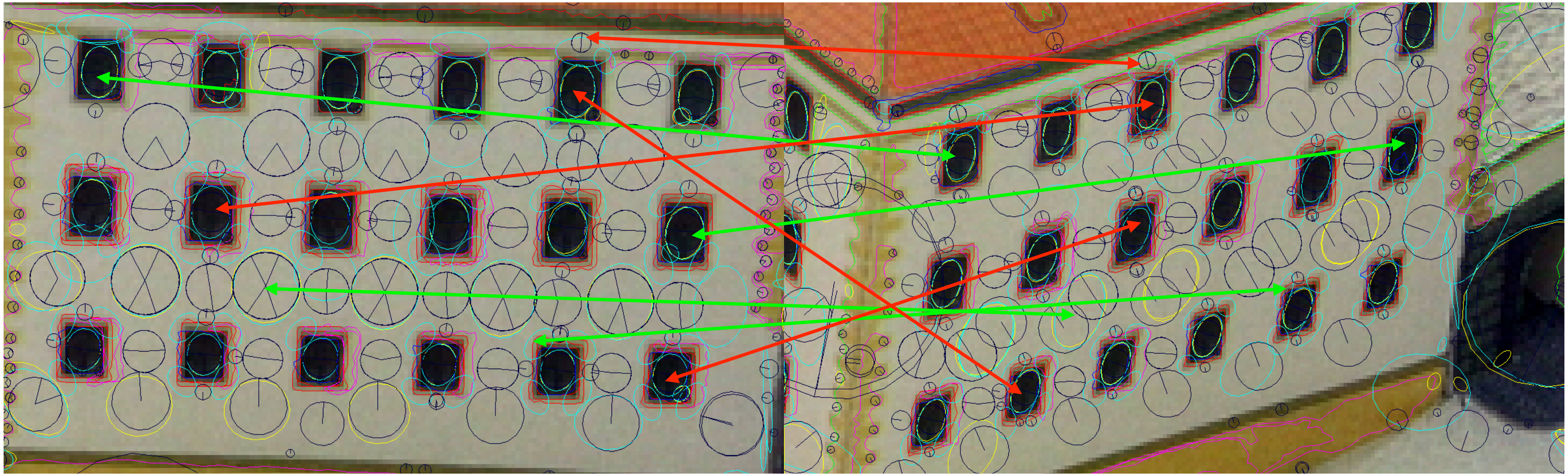
pajdla@cmp.felk.cvut.cz

SIMILAR FEATURES FORM MATCHES

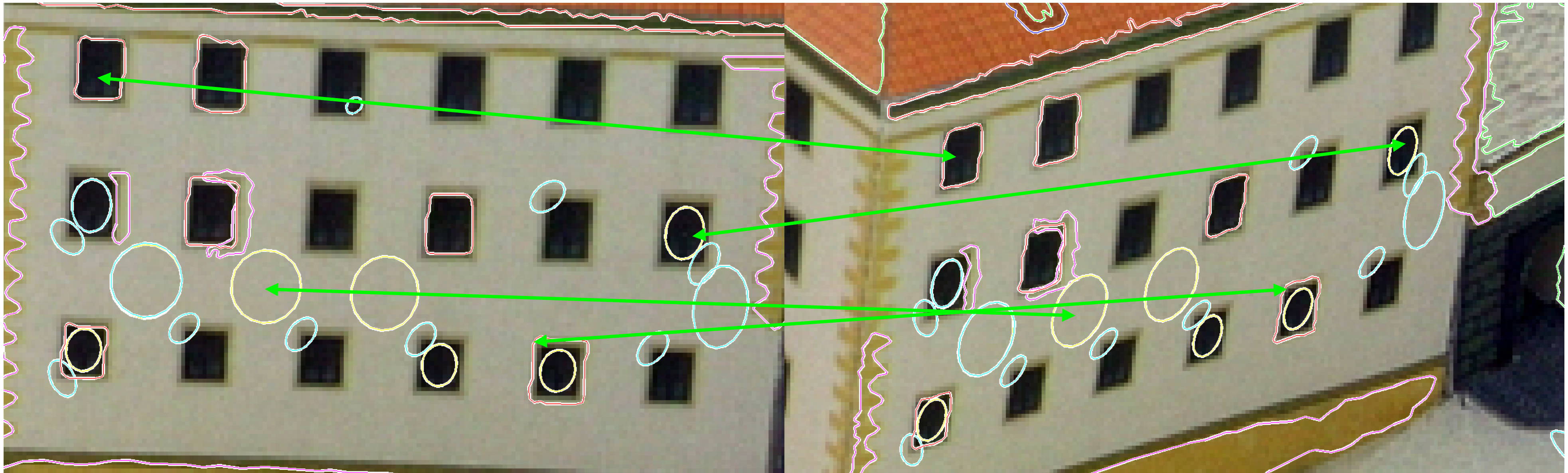


Some feature are similar but do not really match

NOT ALL MATCHES ARE CORRECT

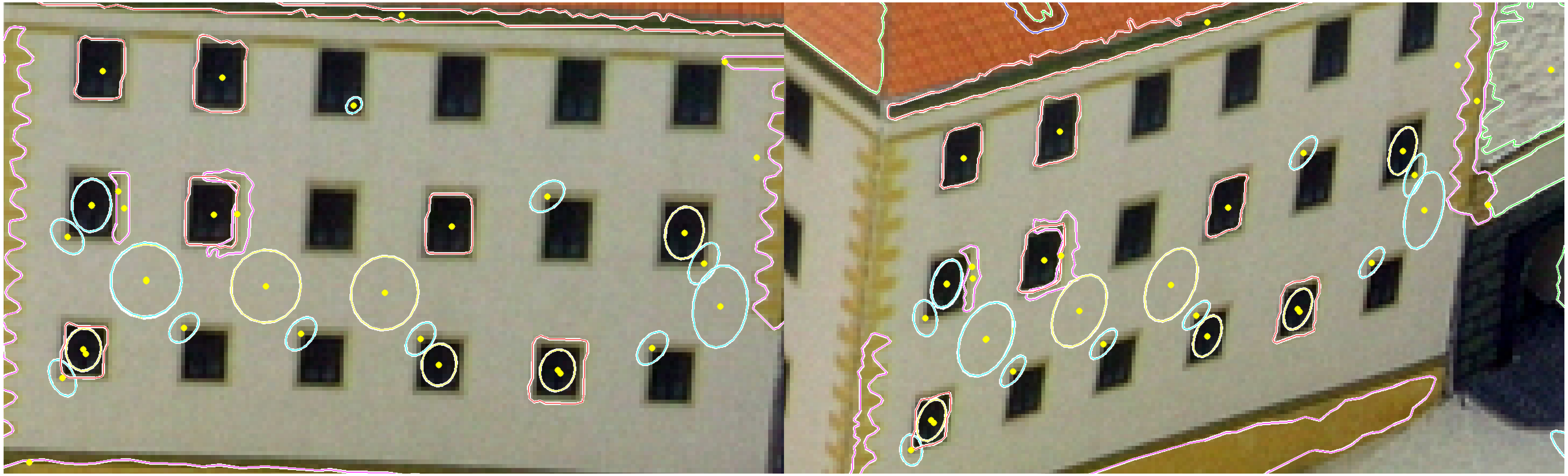


BUT SOME ARE

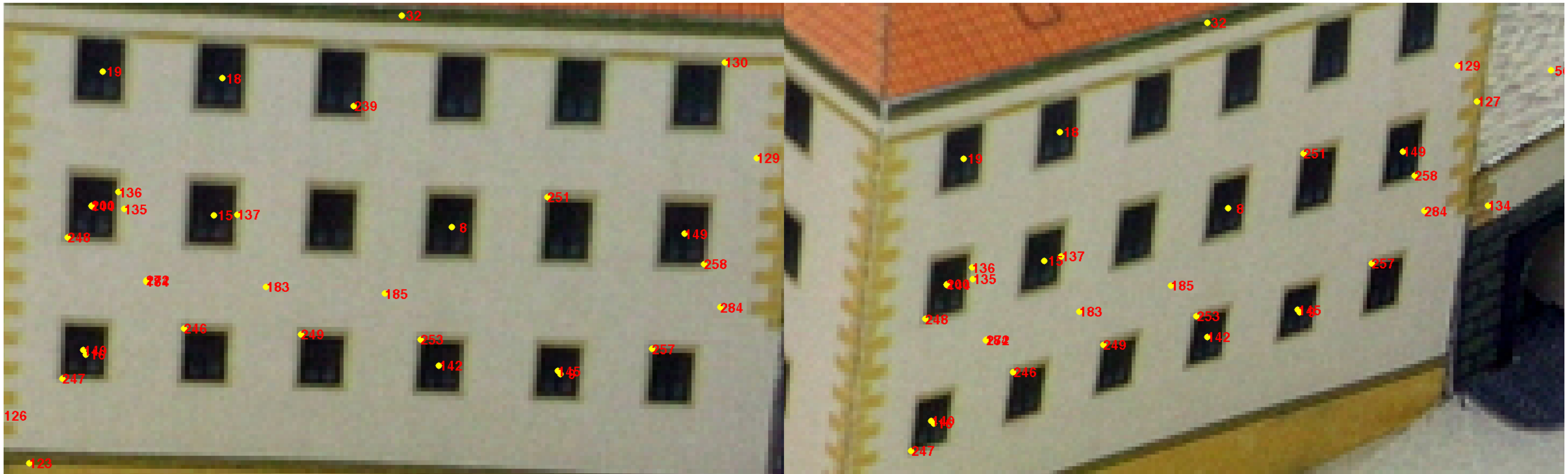


pajdla@cmp.felk.cvut.cz

FEATURE CENTERS – POINTS

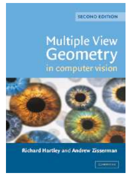


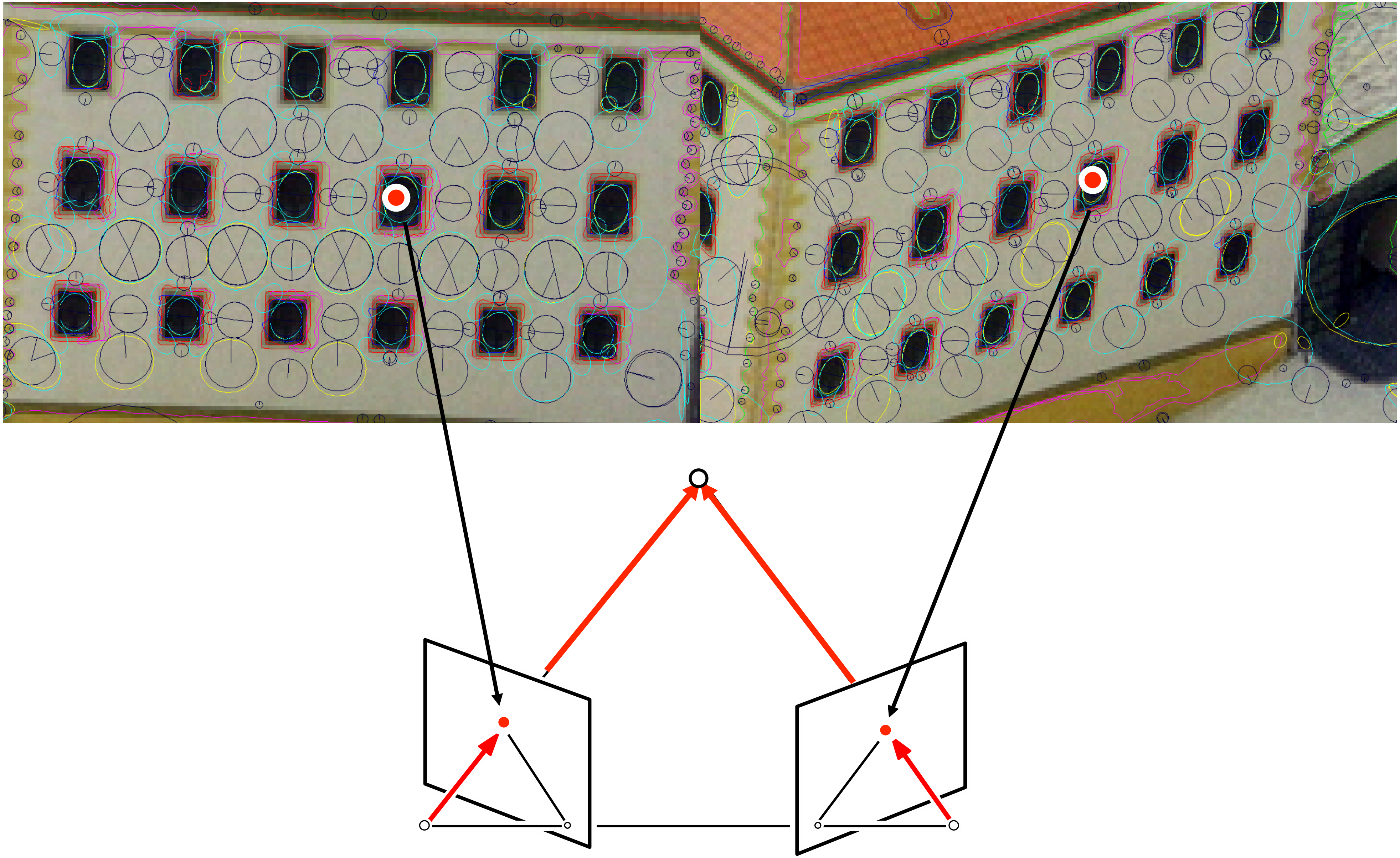
CHECK THE NUMBERS ☺



pajdla@cmp.felk.cvut.cz

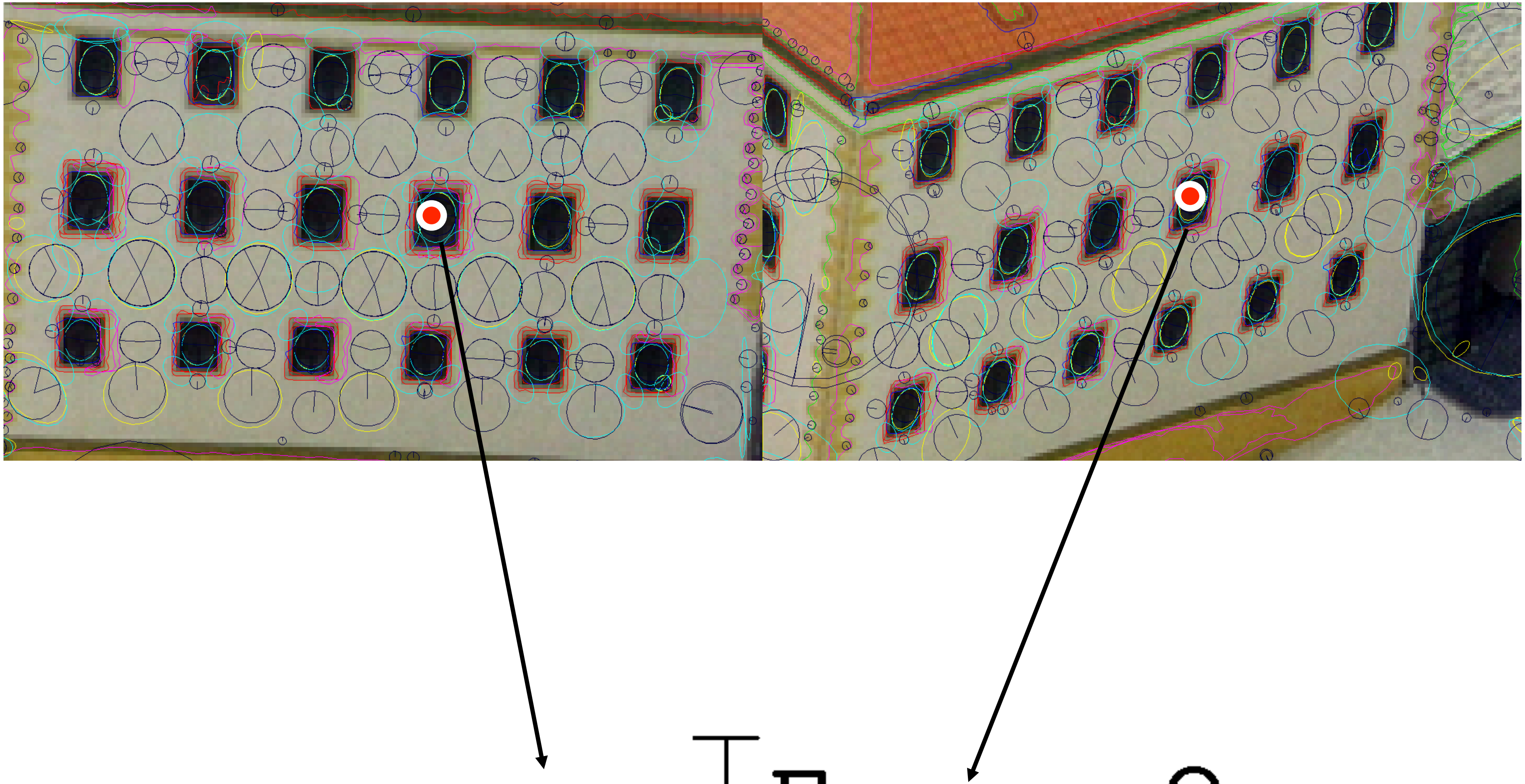
EPIPOLAR CONSTRAINT





pajdla@cmp.felk.cvut.cz

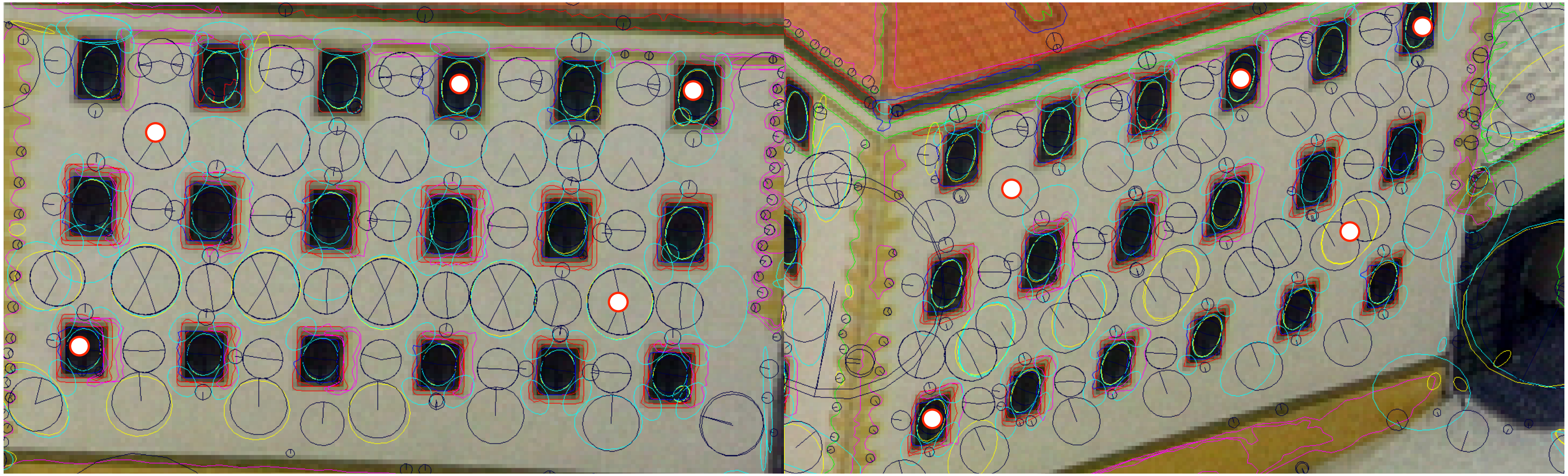
E P I P O L A R C O N S T R → algebraic equation



$$\mathbf{x}_2^\top F \mathbf{x}_1 = 0$$

pajdla@cmp.felk.cvut.cz

HOW TO GET ONE F?

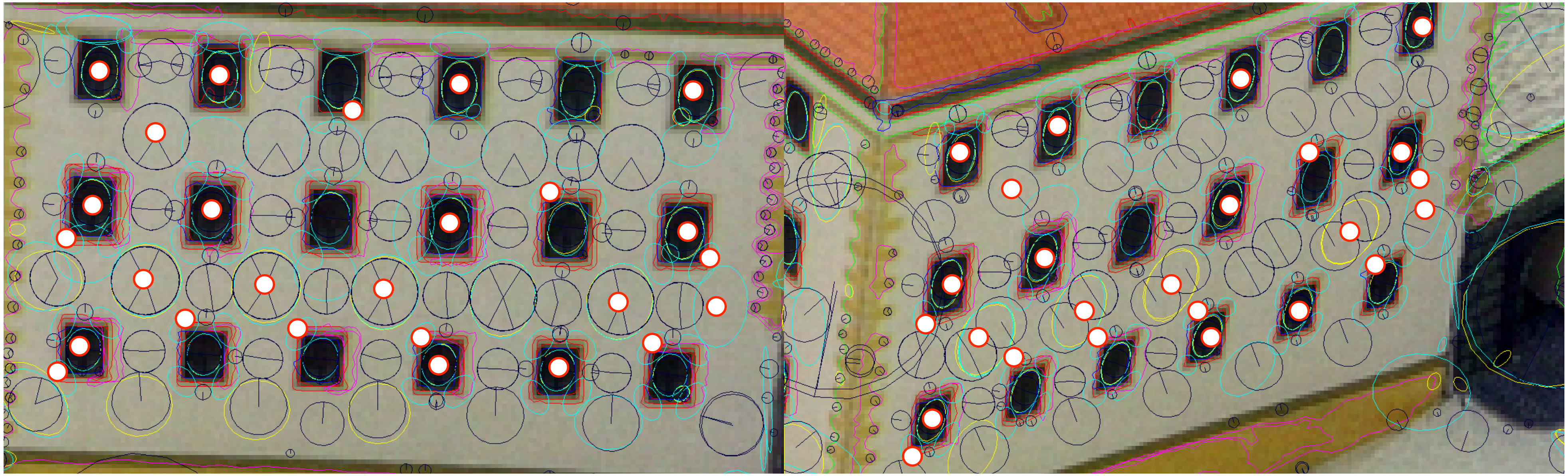


F can be computed from 5 good matches

$$\mathbf{x}_2^\top \mathbf{F} \mathbf{x}_1 = 0$$

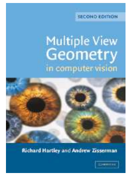
pajdla@cmp.felk.cvut.cz

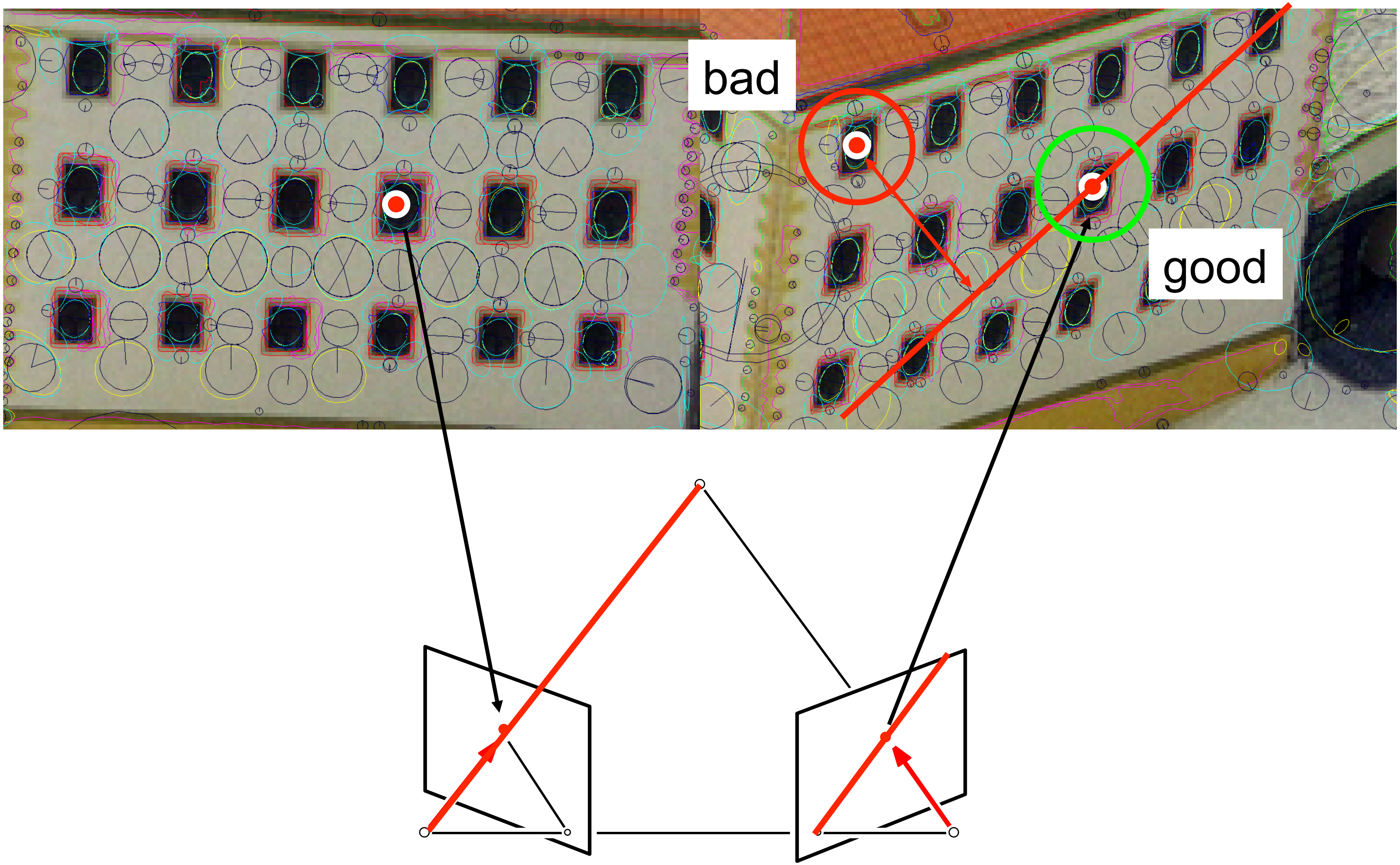
HOW TO GET THE BEST F?



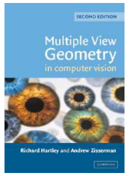
The best F is consistent with the highest number of matches

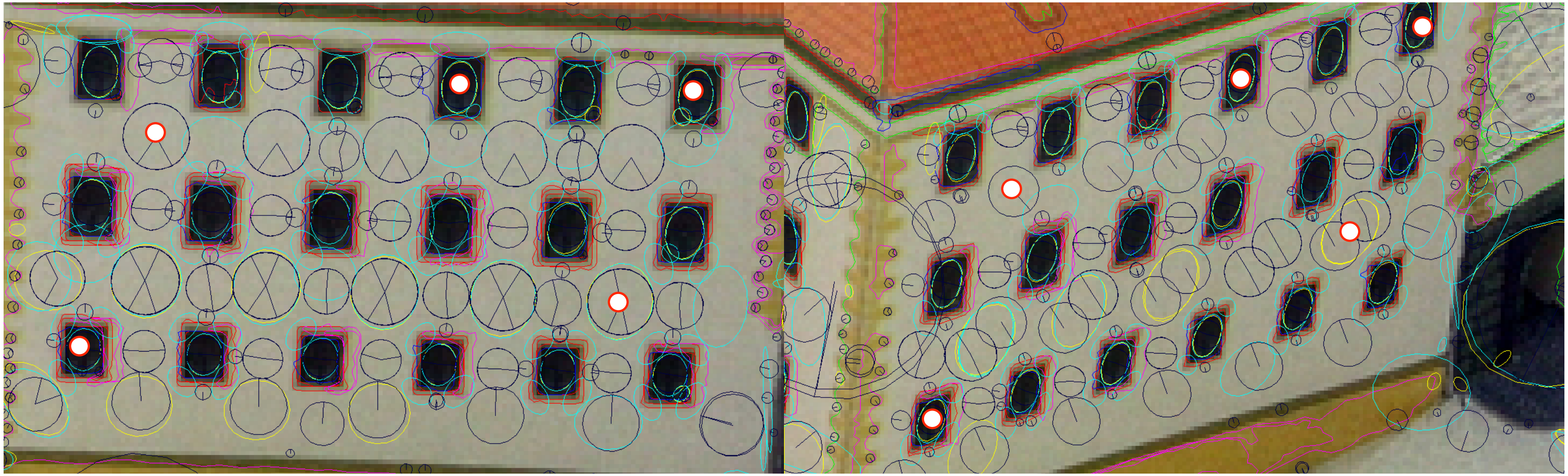
CONSISTENCY WITH F





FINDING THE BEST F



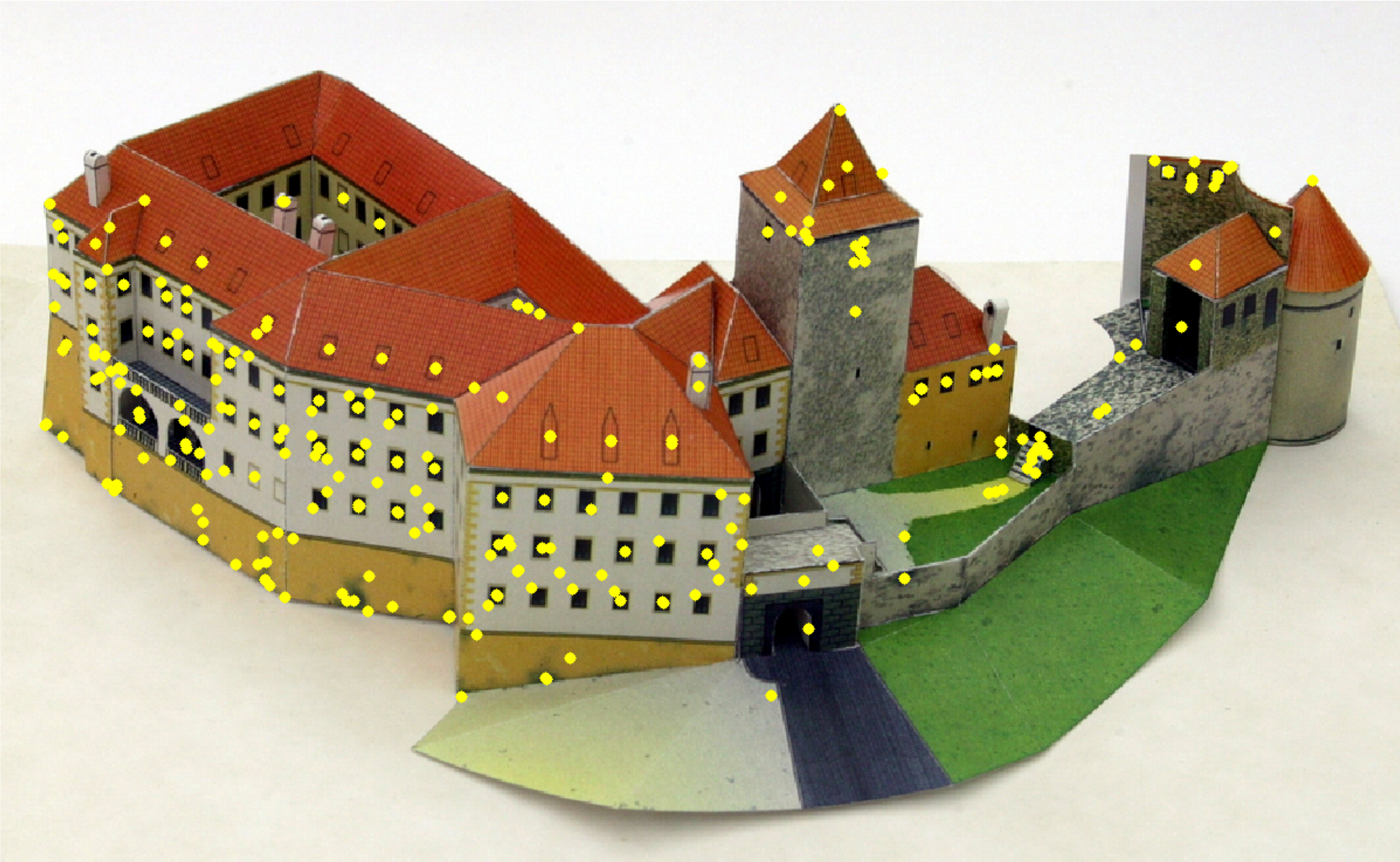


RANSAC (RANdom SAmpling Consensus)

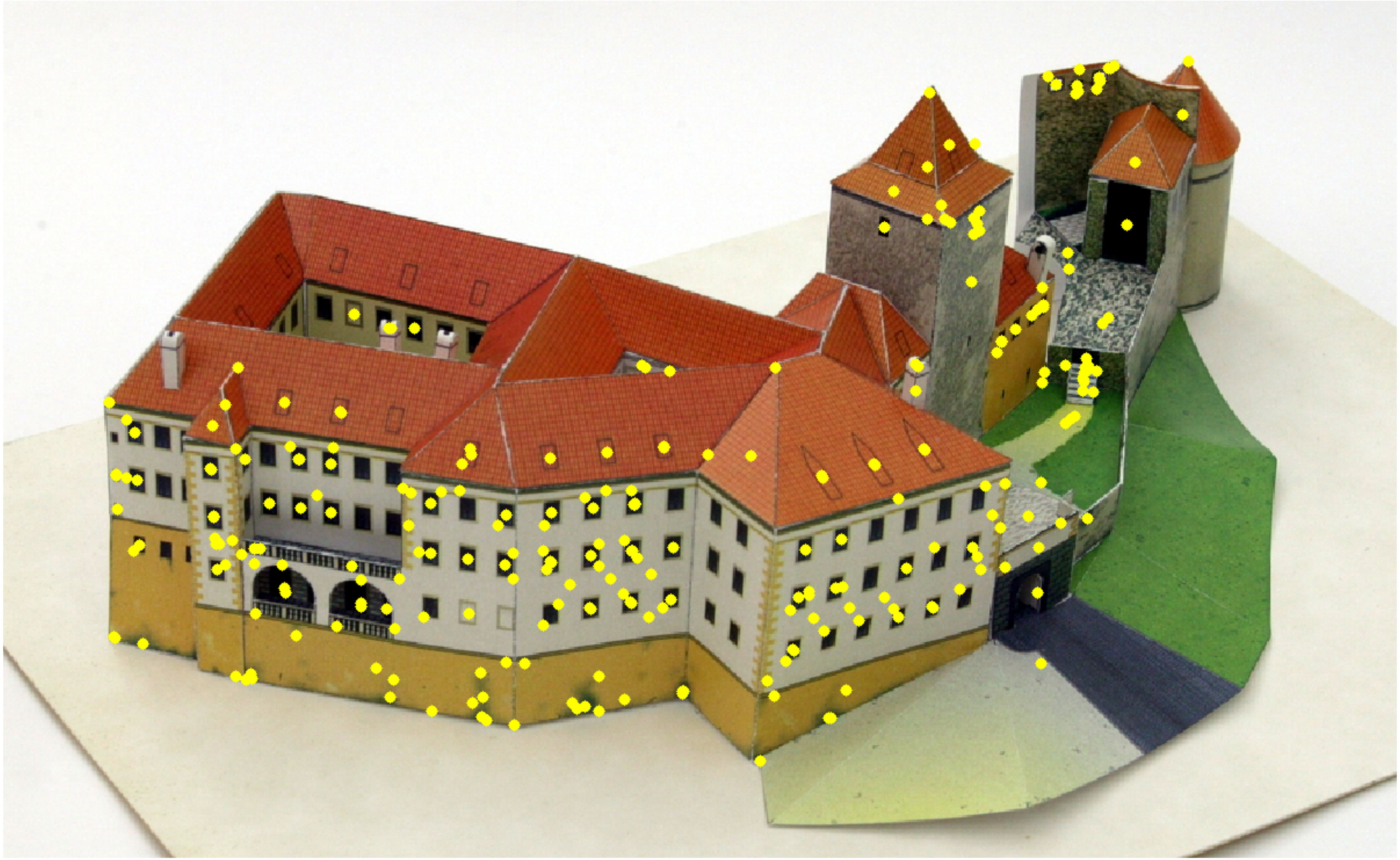
1. Generate random 5-tuples of matches
2. Compute F by solving $\mathbf{x}_2^\top \mathrm{F}\, \mathbf{x}_1 = 0$ (not so trivial)
3. Count the number of good matches

Return the largest set of good matches

pajdla@cmp.felk.cvut.cz



pajdla@cmp.felk.cvut.cz



pajdla@cmp.felk.cvut.cz

INTEGRATED FROM MANY CAMERAS

pajdla@cmp.felk.cvut.cz

pajdla@cmp.felk.cvut.cz

RESULT

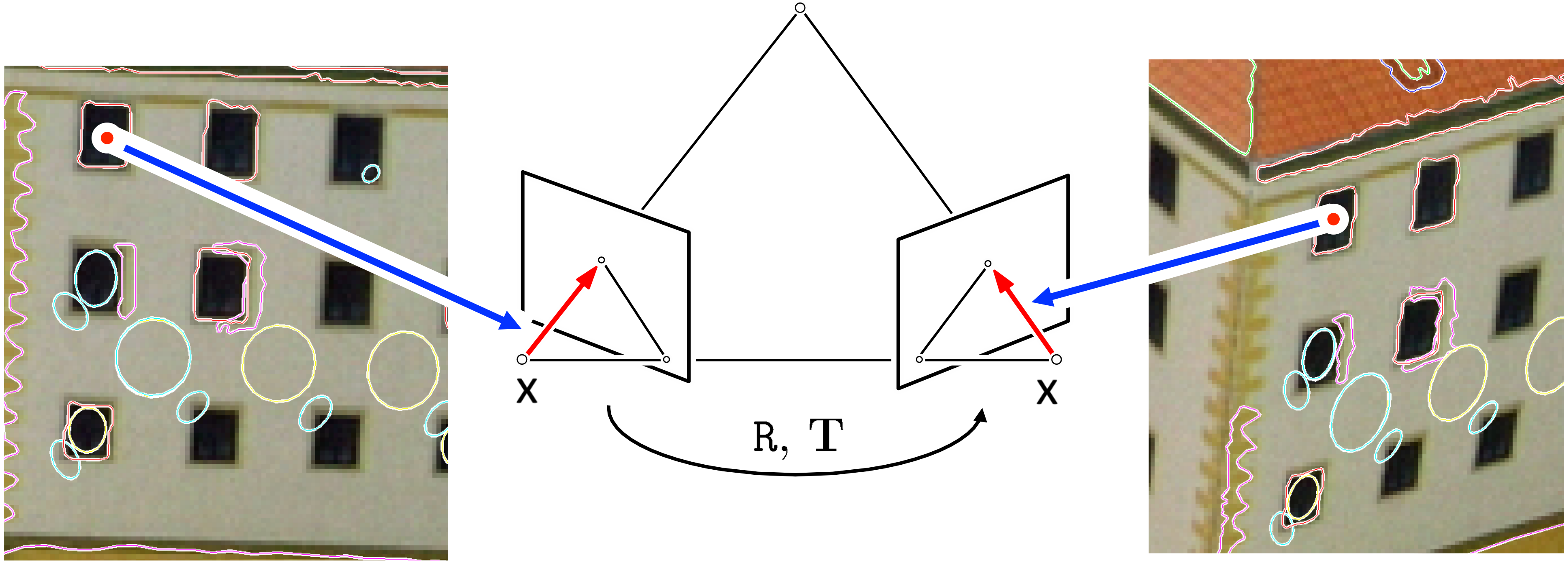
Dense 3D by fish scales of R. Sara et al.

pajdla@cmp.felk.cvut.cz

Dense 3D by fish scales of R. Sara et al.

pajdla@cmp.felk.cvut.cz

RELATIVE CAMERA POSE PROBLEM



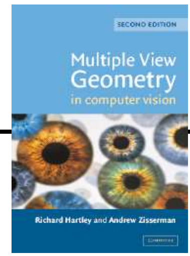
$$x_2^\top F\, x_1 = 0$$

$$\det F = 0$$

$$2\, F F^\top F - \mathrm{trace}(F F^\top) F = 0$$

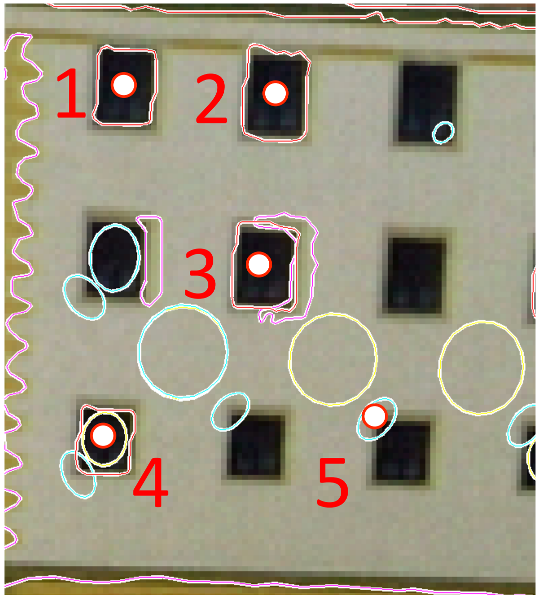
Algebraic equations

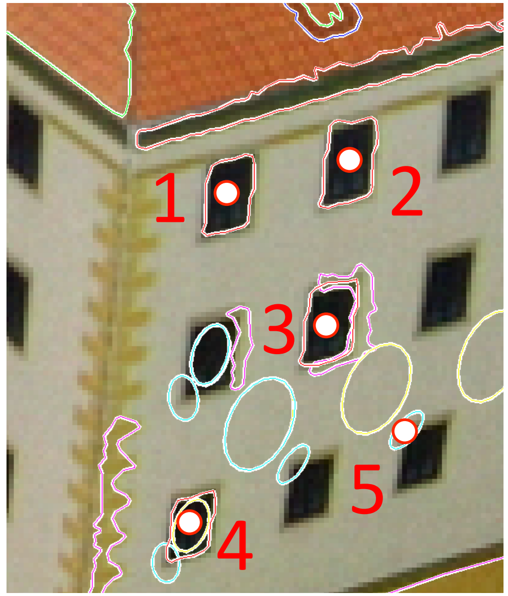
solve → F →



→ $R, \frac{T}{\|T\|}$

MINIMAL PROBLEMS & RANSAC



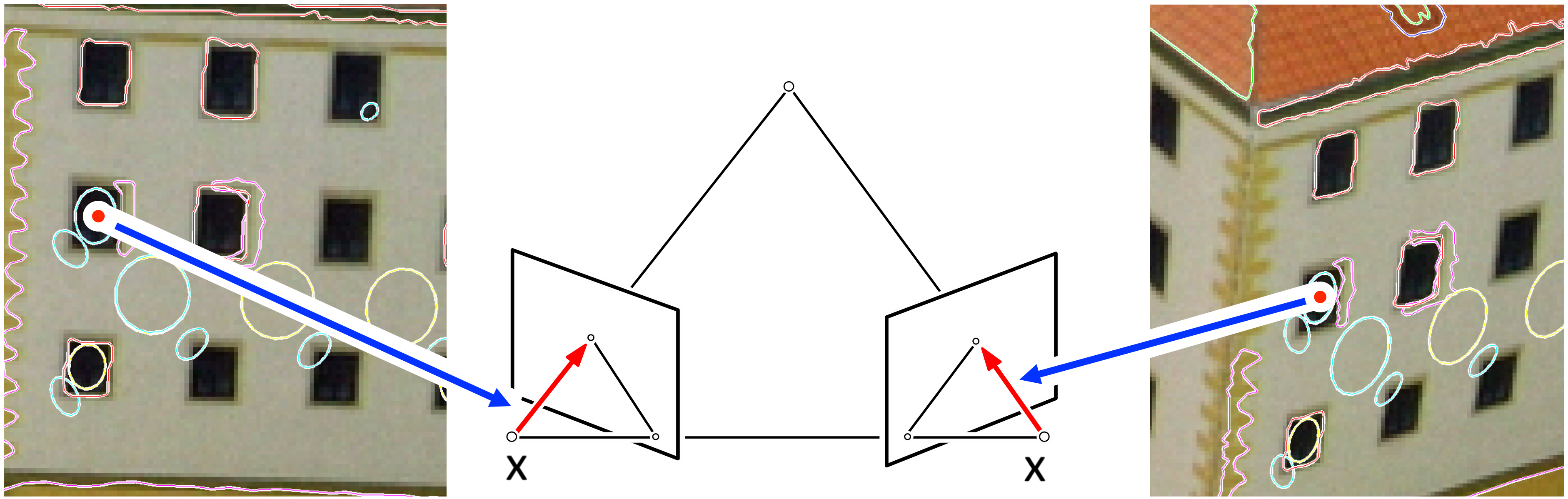


5 matches are necessary and sufficient

RANSAC: find F to maximize the # of good matches

select 5 matches → compute F → record the support

FORMULATION



$$\mathbf{x}_2^\top \mathtt{F}\,\mathbf{x}_1 = 0$$

$$\det \mathtt{F} = 0$$

$$2\,\mathtt{F}\mathtt{F}^\top\mathtt{F} - \mathrm{trace}(\mathtt{F}\mathtt{F}^\top)\mathtt{F} = 0$$

Algebraic equations

pajdla@cmp.felk.cvut.cz

UNKNOWNS

$$\mathtt{F} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}$$

$$\underbrace{\hphantom{f_{11} \quad f_{12} \quad f_{13}}}$$

9 unknowns but only 8 have to be found (F up to scale)

$\rightarrow$ we need at least 8 independent equations

pajdla@cmp.felk.cvut.cz

EQUATIONS

$$\det F = 0$$ 1 equation, degree 3

$$2FF^\top F - \mathrm{trace}(FF^\top)F = 0$$ 9 equations, degree 3

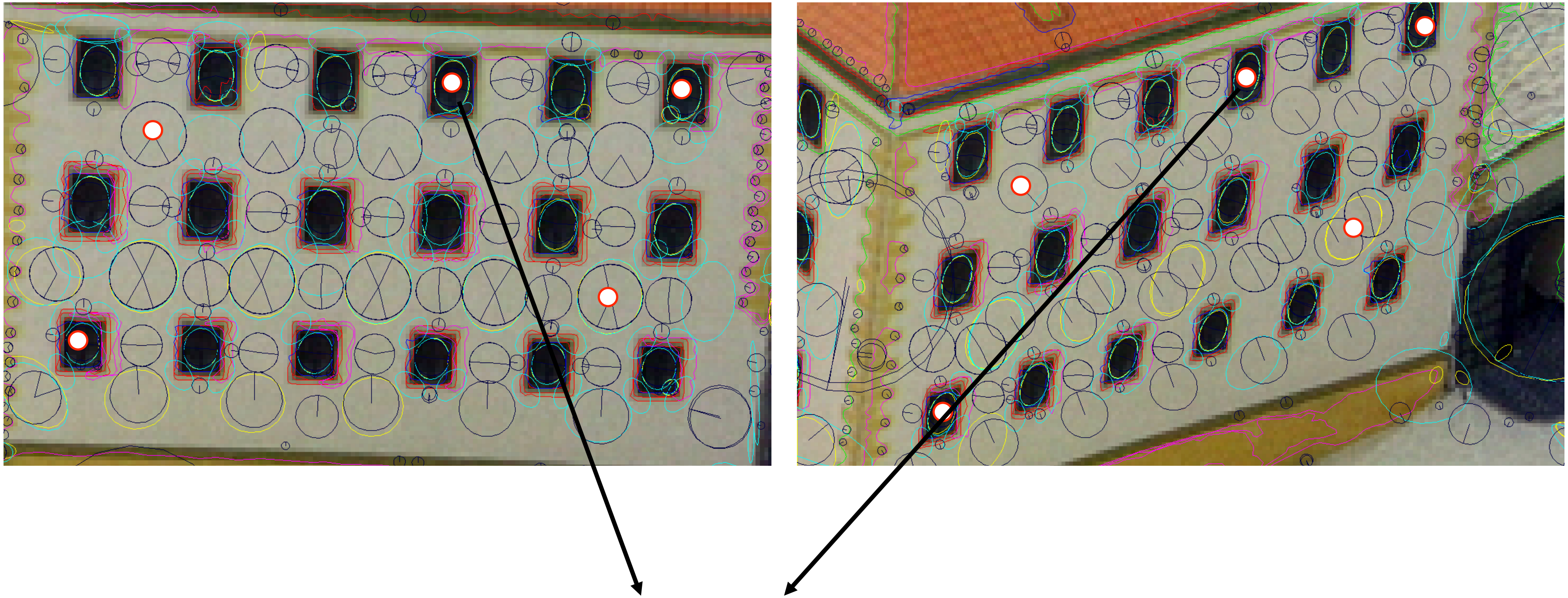
10 equations but only 3 "independent"

8 = 3 + 5

$\rightarrow$ 5 more equations needed

pajdla@cmp.felk.cvut.cz

5 EQUATIONS from image points



$$\mathbf{x}_2^\top \mathrm{F}\, \mathbf{x}_1 = 0$$

5 linear equations:

$$\bullet f_{11} + \bullet f_{12} + \bullet f_{13} + \bullet f_{21} + \bullet f_{22} + \bullet f_{23} + \bullet f_{13} + \bullet f_{23} + \bullet f_{31} + \bullet f_{32} + \bullet f_{33} \quad = \quad 0$$

pajdla@cmp.felk.cvut.cz

ELIMINATING UNKNOWNS

5 linear equations:

$$\bullet f_{11} + \bullet f_{12} + \bullet f_{13} + \bullet f_{21} + \bullet f_{22} + \bullet f_{23} + \bullet f_{13} + \bullet f_{23} + \bullet f_{31} + \bullet f_{32} + \bullet f_{33} \quad = \quad 0$$

can be written in a matrix form

$$\underbrace{\begin{bmatrix} \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \end{bmatrix}}_{\mathtt{A}} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ \vdots \\ f_{33} \end{bmatrix} = 0$$

pajdla@cmp.felk.cvut.cz

ELIMINATING UNKNOWNS

$$\underbrace{\begin{bmatrix} \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \end{bmatrix}}_{\mathtt{A}} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ \vdots \\ f_{33} \end{bmatrix} = 0$$

$\mathtt{A}$ 5×9 matrix $\longrightarrow$ it has a 4 dimensional nullspace

$$\mathtt{A}\, N = 0$$

known

$$N = \boxed{x} N_1 + \boxed{y} N_2 + \boxed{z} N_3 + \boxed{w} N_4$$

new unknowns

pajdla@cmp.felk.cvut.cz

ELIMINATING UNKNOWNS

$F \sim x\,N_1 + y\,N_2 + z\,N_3 + w\,N_4$ ... 4 unknowns

F is up to scale → choose a representative by setting w = 1

$$\underbrace{F := x\,N_1 + y\,N_2 + z\,N_3 + N_4}$$

3 unknowns x, y, z

↓ substitute

$$\left.\begin{aligned} \det F &= 0 \\ 2\,F\,F^\top F - \mathrm{trace}(F\,F^\top)F &= 0 \end{aligned}\right\}$$

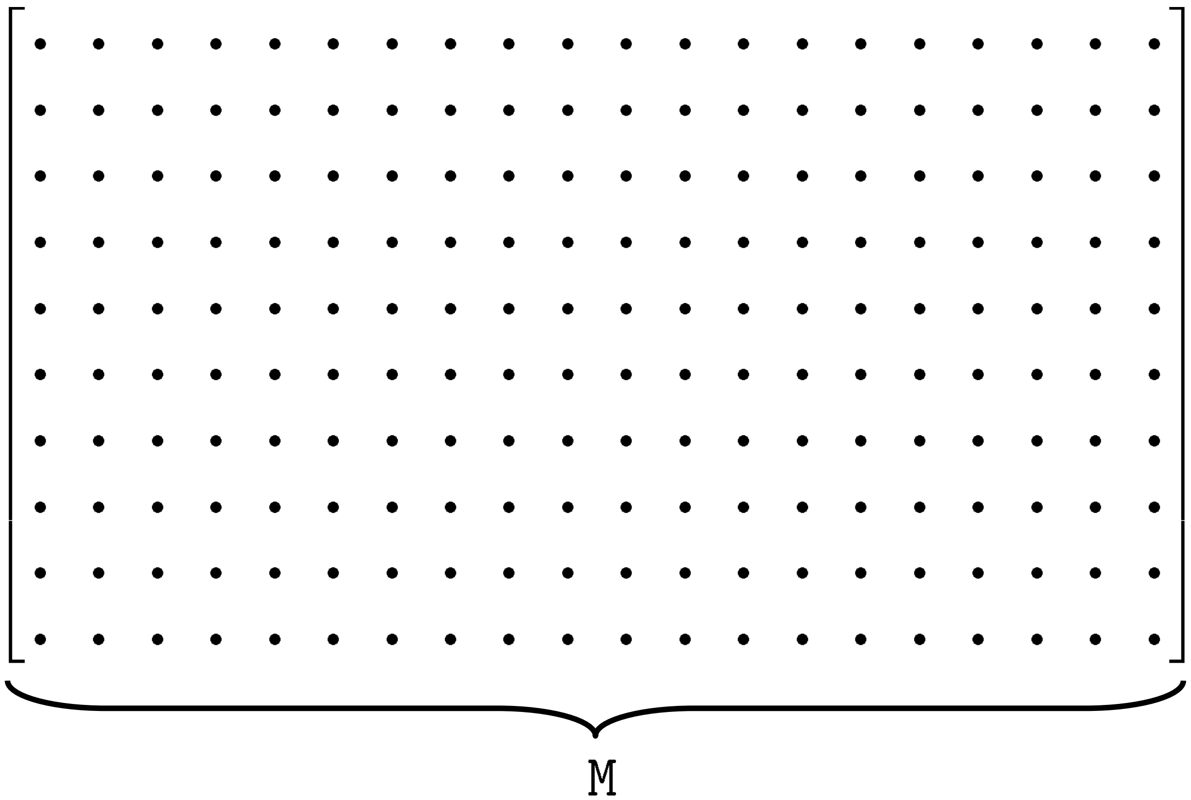
10 3rd order equations in 3 unknowns

pajdla@cmp.felk.cvut.cz

SOLVING IT

$$\det \mathtt{F} = 0$$
$$2\,\mathtt{F}\,\mathtt{F}^\top \mathtt{F} - \mathrm{trace}(\mathtt{F}\,\mathtt{F}^\top)\mathtt{F} = 0$$

10 3rd order equations in 3 unknowns

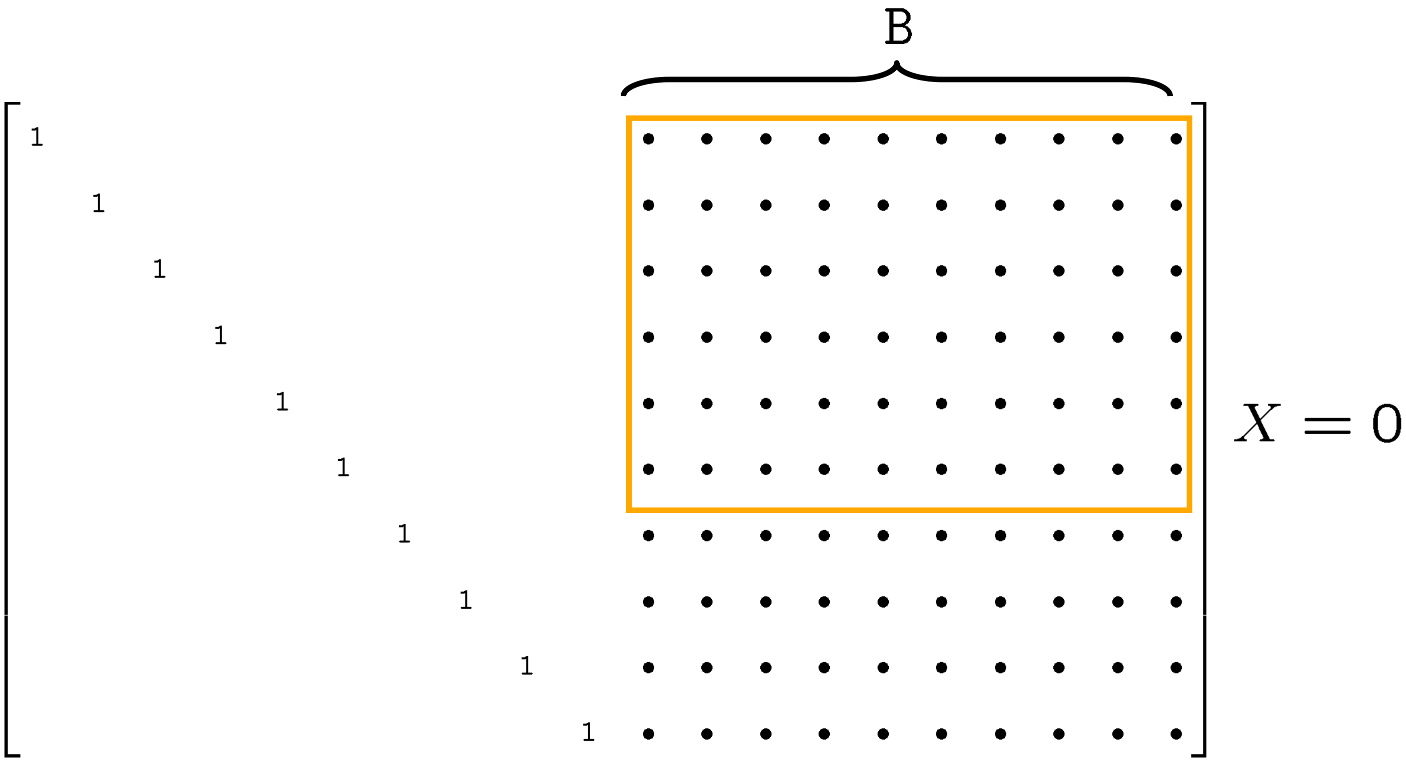


← 10 x 20 matrix

$$X = 0$$

$$X = [x^3, x^2 y, x^2 z, x\,y^2, x\,y\,z, x\,z^2, y^3, y\,z^2, z^3, x^2, x\,y, x\,z, y^2, y\,z, z^2, x, y, z, 1]^\top$$

Gauss-Jordan elimination



pajdla@cmp.felk.cvut.cz

SOLVING IT

$$A_t = \begin{bmatrix} \boxed{\quad -\mathrm{B} \quad} \\ 1 \\ \quad 1 \\ \qquad 1 \\ \qquad\qquad\qquad 1 \end{bmatrix}$$

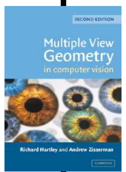
pajdla@cmp.felk.cvut.cz

SOLUTIONS

$[x, y, z] = \mathrm{eig}(A_t)$ ... up to 10 solutions

back-substitution

$F := x\,N_1 + y\,N_2 + z\,N_3 + N_4$



R, T ... camera relative motion

A L G O R I T M

1. Construct matrix M

2. Build matrix At

```
B = inv( M(:,1:10) )*M(:,11:20);
At = zeros(10);
At(1:6,:) = -B([1:6], :);
At(7,1)  = 1;
At(8,2)  = 1;
At(9,3)  = 1;
At(10,7) = 1;
```

3. Compute eigenvalues

$$[x, y, z] = \mathrm{eig}(A_t)$$

4. Recover camera relative motion R, T

pajdla@cmp.felk.cvut.cz

A L G O R I T M

1. Not trivial to find
2. Simple & fast

SOLVING ALGEBRAIC EQUATIONS

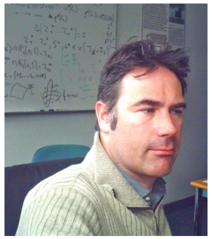
The previous procedure was a particular case of a general technique for solving systems of algebraic equations.

H I S T O R Y

1888 David Hilbert: Finitness theorem
Every ideal has a finite generating set

1965 Bruno Burchberger: Groebner bases
Computational procedure for solving systems of polynomial equations
(Extremely simple: 20 lines of Maple code!)

1998 Hans Stetter: Multiplication matrix
A stable numerical procedure via eigenvectors



1999 Jean-Charles Faugere: F4 algorithm
An efficient computational tool for cryptography

SOLVING ALGEBRAIC EQUATIONS

1 equation, 1 variable $\rightarrow$ companion matrix $\rightarrow$ eigenvalues

$$f(x) = x^3 + 4\,x^2 + x - 6 \;=\; -6 + 1\,x + 4\,x^2 + 1\,x^3$$

$$\mathtt{M}_x = \begin{bmatrix} 0 & 0 & 6 \\ 1 & 0 & -1 \\ 0 & 1 & -4 \end{bmatrix}$$

... a simple rule

```
>> e=eig(Mx)
```

$$\mathtt{e} \;=\; \begin{bmatrix} 1 \\ -2 \\ -3 \end{bmatrix} \qquad x_1 = 1,\; x_2 = -2,\; x_3 = -3$$

It works when eig works, i.e. order 100 in Matlab is often OK.

pajdla@cmp.felk.cvut.cz

SOLVING ALGEBRAIC EQUATIONS

m equations, n variables

$$\begin{aligned} f_1(x,y) &= 25\,x\,y - 15\,x - 20\,y + 12 \\ f_2(x,y) &= x^2 + y^2 - 1 \end{aligned}$$

$\rightarrow$ Groebner basis

(a set of polynomials with the same solutions but easier to solve)

$$\begin{aligned} g_1(x,y) &= 125\,y^3 - 75\,y^2 + 27 - 45\,y \\ g_2(x,y) &= 25\,x\,y - 15\,x - 20\,y + 12 \\ g_3(x,y) &= x^2 + y^2 - 1 \end{aligned}$$

pajdla@cmp.felk.cvut.cz

SOLVING ALGEBRAIC EQUATIONS

m equations, n variables

$$\begin{aligned} f_1(x,y) &= 25\,x\,y - 15\,x - 20\,y + 12 \\ f_2(x,y) &= x^2 + y^2 - 1 \end{aligned}$$

$\rightarrow$ Groebner basis $\rightarrow$ generalized companion matrix

$$\mathtt{M}_{x+y} = \begin{bmatrix} 0 & 125 & 0 & 125 \\ -60 & 100 & 125 & 75 \\ -63 & 45 & 175 & 45 \\ 65 & 100 & -125 & 75 \end{bmatrix}$$

pajdla@cmp.felk.cvut.cz

SOLVING ALGEBRAIC EQUATIONS

m equations, n variables

$$
\begin{aligned}
f_1(x,y) &= 25\,x\,y - 15\,x - 20\,y + 12 \\
f_2(x,y) &= x^2 + y^2 - 1
\end{aligned}
$$

→ Groebner basis → generalized companion matrix → eigenvectors

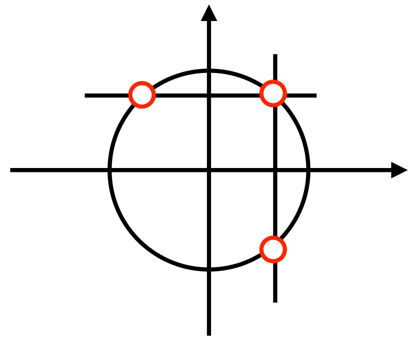
$$
\mathtt{v} \sim \begin{bmatrix} 1 & 1 & 1 & 1 \\ -\frac{4}{5} & \frac{4}{5} & \frac{4}{5} & \frac{4}{5} \\ \frac{3}{5} & -\frac{3}{5} & \frac{3}{5} & \frac{3}{5} \\ \frac{16}{25} & \frac{16}{25} & \frac{16}{25} & \frac{16}{25} \end{bmatrix}
$$

$$
x_1 = -\frac{4}{5} \quad , \quad y_1 = \frac{3}{5}
$$

pajdla@cmp.felk.cvut.cz

SOLVING ALGEBRAIC EQUATIONS

m equations, n variables



$$f_1(x,y) = 25\,x\,y - 15\,x - 20\,y + 12$$
$$f_2(x,y) = x^2 + y^2 - 1$$

$\rightarrow$ Groebner basis $\rightarrow$ generalized companion matrix $\rightarrow$ eigenvectors

$$\mathtt{v} \sim \begin{bmatrix} 1 & 1 & 1 & 1 \\ -\frac{4}{5} & \frac{4}{5} & \frac{4}{5} & \frac{4}{5} \\ \frac{3}{5} & -\frac{3}{5} & \frac{3}{5} & \frac{3}{5} \\ \frac{16}{25} & \frac{16}{25} & \frac{16}{25} & \frac{16}{25} \end{bmatrix}$$

$$x_1 = -\frac{4}{5}\ , \quad y_1 = \frac{3}{5}$$
$$x_2 = \frac{4}{5}\ , \quad y_3 = -\frac{3}{5}$$
$$x_3 = \frac{4}{5}\ , \quad y_3 = \frac{3}{5}$$

pajdla@cmp.felk.cvut.cz

THE DIFFICULT PART

Equations → Groebner basis

↑

no simple rule:

1. NP-complete in m, n (~ 3-coloring of graphs)
   (takes very long time to compute)

2. EXPSPACE-complete problem
   (needs huge space to remember intermediate results)

pajdla@cmp.felk.cvut.cz

GB COMPUTATION ALGORITHMS

Manipulation ... polynomial multiplication & pseudo-division

$$f_1 = 25\,x\,y - 15\,x - 20\,y + 12$$
$$f_2 = x^2 + y^2 - 1$$

? $g_1 = (-5\,y - 3)\,f_1 + (125\,x - 10)\,f_2$

$$g_1 = 125\,y^3 - 75\,y^2 + 27 - 45\,y$$
$$g_2 = 25\,x\,y - 15\,x - 20\,y + 12$$
$$g_3 = x^2 + y^2 - 1$$

How to find coefficients?

GB COMPUTATION ALGORITHMS

1. “Standard” (Hironaka 1964) and
   Groebner (Burchberger 1965) bases

2. 1965: Buchberger’s algorithm

   – a generalization of the Gauss-Jordan elimination
   – extremely simple: 7 (+ 13 for the rem division) lines of code
   – not efficient
   – not good for numerical approximations

3. 1999 (2005): F4 (F5) algorithm (J.-C. Faugere)

   – more efficient, more robust, more complex

pajdla@cmp.felk.cvut.cz

COMPUTING GB MAY BE VERY HARD

Example:

4 polynomials, 3 variables, degree ≤ 6, small integer coeffs

$$
\begin{aligned}
f_1 &= 8\,x^2\,y^2 + 5\,x\,y^3 + 3\,x^3\,z + x^2\,y\,z \\
f_2 &= x^5 + 2\,y^3\,z^2 + 13\,y^2\,z^3 + 5\,y\,z^4 \\
f_3 &= 8\,x^3 + 12\,y^3 + x\,z^2 + 3 \\
f_4 &= 7\,x^2\,y^4 + 18\,x\,y^3\,z^2 + y^3\,z^3
\end{aligned}
$$

have extremely simple Groebner basis

$$
\begin{aligned}
g_1 &= x \\
g_2 &= y^3 + \frac{1}{4} \\
g_3 &= z^2
\end{aligned}
$$

HOWEVER

when computed by the Buchberger's algorithm over the rational numbers w.r.t. the grevlex ordering x > y > z, the following polynomial appears during the computation:

$$y^3 - \underbrace{17359065042904512907647 47182\ldots}_{\sim 80{,}000 \text{ digits}}$$

COMPUTATION

Macaulay2 program over the rational field $\mathbb{Q}$

```
R = QQ[x,y,z, MonomialOrder=>GRevLex];
I = ideal(8*x^2*y^2 +  5*x*y^3      +  3*x^3*z    +   x^2*y*z,
              x^5     +  2*y^3*z^2   + 13*y^2*z^3 + 5*y*z^4,
            8*x^3     + 12*y^3       +     x*z^2    + 3,
            7*x^2*y^4 + 18*x*y^3*z^2 +     y^3*z^3)
G = gens gb I
```

will run very long.

The problem is in remembering very long coefficients.

COMPUTATION

Macaulay2 program over the finite field $\mathbb{Z}/13$

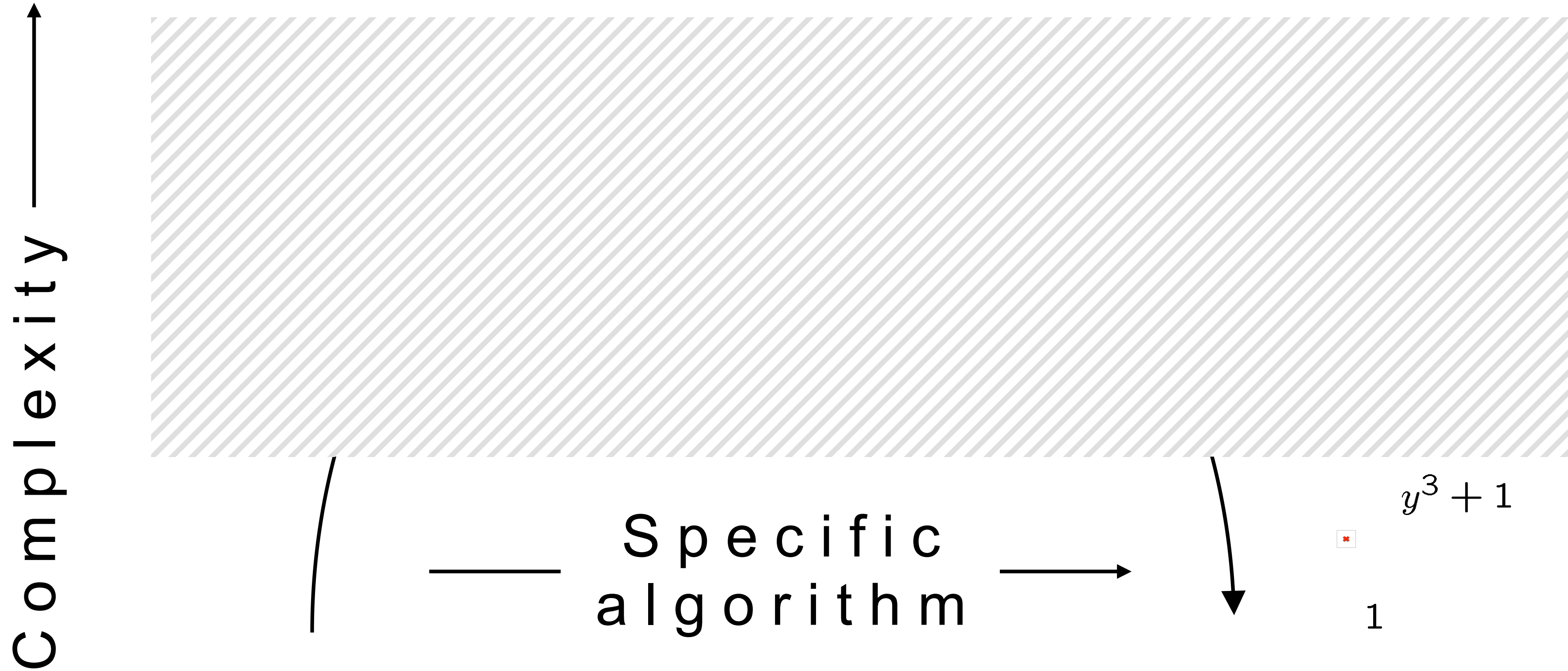
```
R = ZZ/13[x,y,z, MonomialOrder=>GRevLex];
I = ideal(8*x^2*y^2 +  5*x*y^3     +  3*x^3*z   +   x^2*y*z,
                x^5     +  2*y^3*z^2   + 13*y^2*z^3 + 5*y*z^4,
              8*x^3     + 12*y^3       +    x*z^2   + 3,
              7*x^2*y^4 + 18*x*y^3*z^2 +    y^3*z^3)
G = gens gb I
```

returns a very similar basis in the fraction of the second

$$\begin{aligned} g_1 &= x \\ g_2 &= y^3 + 10 \\ g_3 &= z^2 \end{aligned} \qquad \left(\begin{aligned} g_1 &= x \\ g_2 &= y^3 + \tfrac{1}{4} \\ g_3 &= z^2 \end{aligned} \right)$$

pajdla@cmp.felk.cvut.cz

WHY? General algorithms can construct all GBs but often generate many complicated polynomials



$$
\begin{aligned}
f_1 &= 8\,x^2\,y^2 + 5\,x\,y^3 + 3\,x^3\,z + x^2\,y\,z \\
f_2 &= x^5 + 2\,y^3\,z^2 + 13\,y^2\,z^3 + 5\,y\,z^4 \\
f_3 &= 8\,x^3 + 12\,y^3 + x\,z^2 + 3 \\
f_4 &= 7\,x^2\,y^4 + 18\,x\,y^3\,z^2 + y^3\,z^3
\end{aligned}
$$

$$
\begin{aligned}
g_1 &= x \\
g_2 &= y^3 + 10 \\
g_3 &= z^2
\end{aligned}
$$

Input

Groebner basis

pajdla@cmp.felk.cvut.cz

SPECIFIC GB CONSTRUCTION ALG'S

1. Find a short path towards the GB which is independent from the actual coefficients, implement it efficiently.

2. Use floating-point arithmetics to do the manipulations to avoid huge coefficients.

SHORT PATH TO GB

1. Restructure & reformulate the problem to reduce the number of variables and the degree of monomials.

2. Use a computer algebra system (Macaulay2) to compute the Groebner basis in a finite field (fast!) for random coefficients and remember the path:

```
R = ZZ/P[x,y,z, MonomialOrder=>GRevLex];
```

↑ "lucky" prime number (always exists)

try P = 1, 2, 3, 5, 7, ..., 30011, 30013, 30029, ...

until the result stabilizes (always does)

pajdla@cmp.felk.cvut.cz

FLOATING POINT ARITHMETICS

Strictly speaking, polynomial manipulations must be done in exact arithmetics

$$
\begin{aligned}
f_1 &= 25\,x\,y + 25\,x + 12 \\
f_2 &= x^2\,y + x^2 + 3
\end{aligned}
$$

$$
\begin{aligned}
x\,f_1 - 25\,f_2 &= x\,(\cancel{25\,x\,y} + \cancel{25\,x} + 12) - 25(\cancel{x^2\,y} + \cancel{x^2} + 3) \\
&= 12\,x - 75
\end{aligned}
$$

Double canceling may fail when rounding occurs

in floating point arithmetics

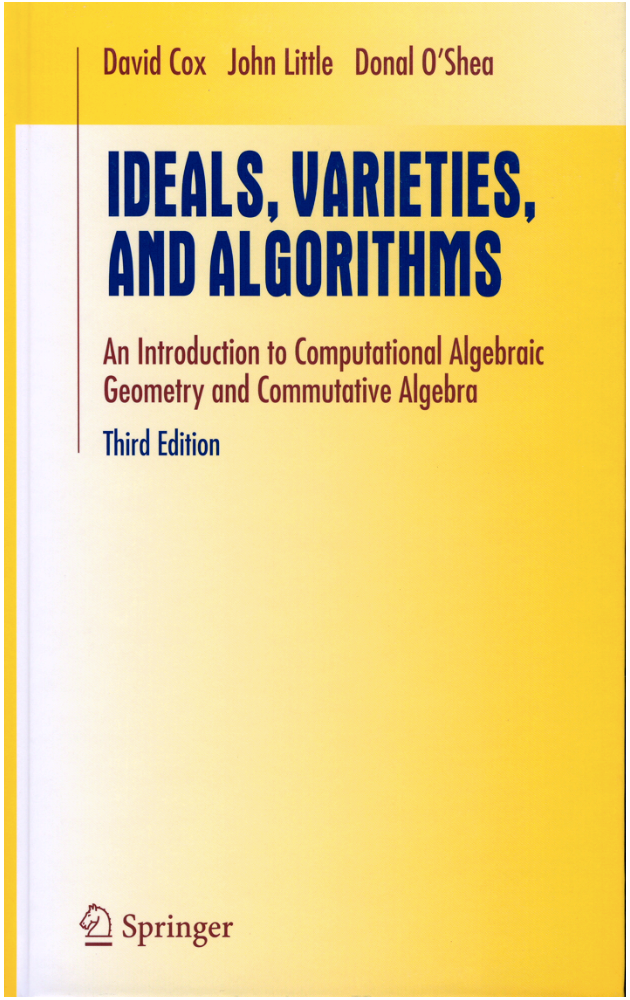
pajdla@cmp.felk.cvut.cz

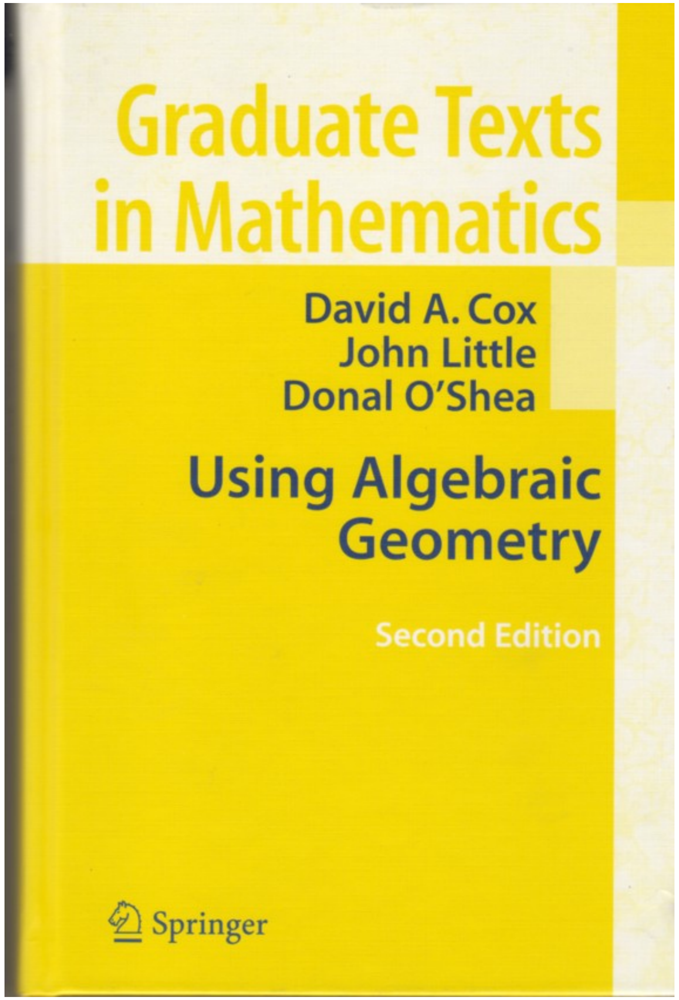
FLOATING POINT ARITHMETICS

For some computer vision problems rounding does not destroy the result.

All manipulations have to be done with care and checked on typical data in randomized experiments.

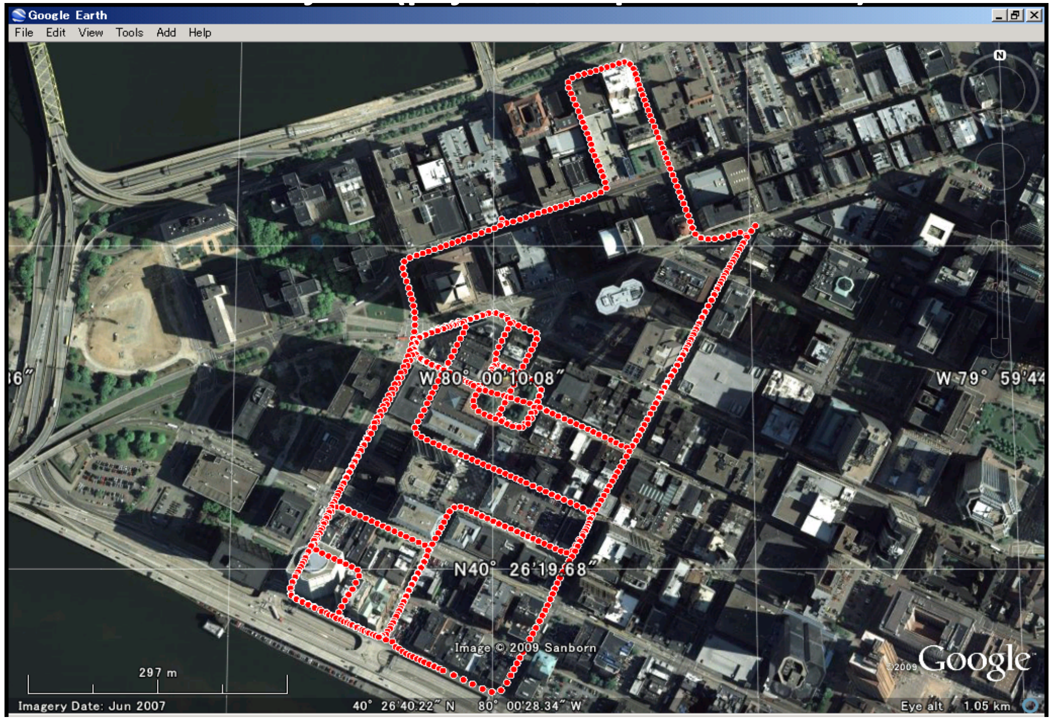
pajdla@cmp.felk.cvut.cz

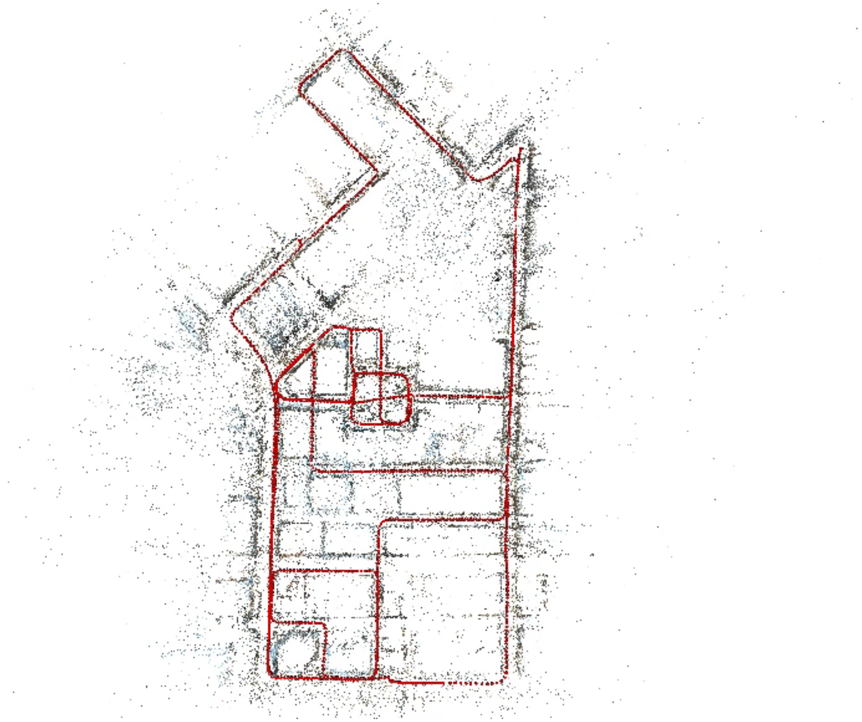




pajdla@cmp.felk.cvut.cz

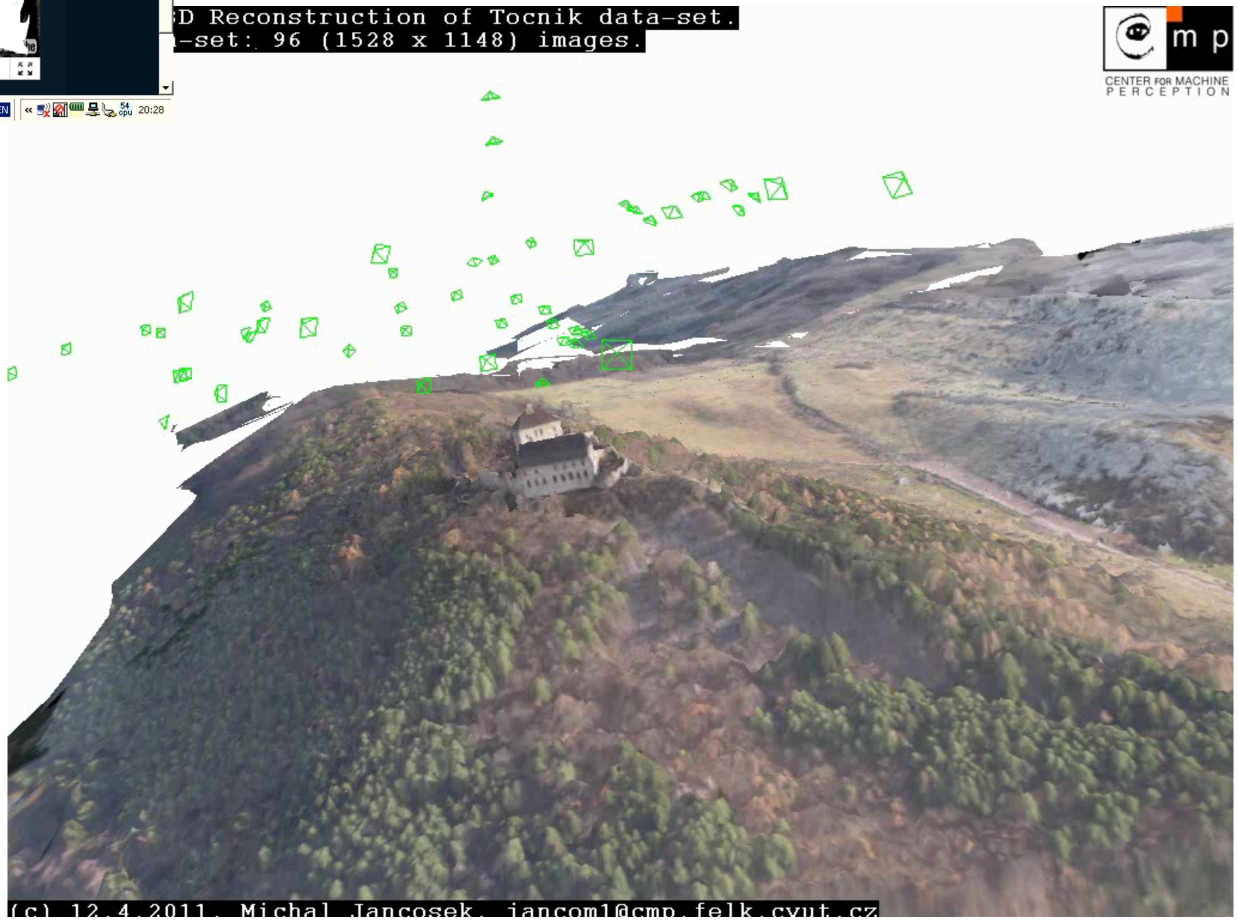
From Google Street View to 3D City Models





Welcome to CMP SfM Web ...
ptak.felk.cvut.cz/sfmservice/
CMP SfM WebService
CMP SfM Web Service
User login
Username:
Password:
Menu
SfM Web Service
Authors
Gallery
What?
CMP SfM Web Service provides a remote access to the 3D reconstruction systems developed in Center for Machine Perception. The access to the system is granted on request by email to Tomas Pajdla <pajdla@cmp.felk.cvut.cz>.
Why?
We provide the access to the service to our partners and to people in the Computer Vision community to make it easier to use our codes. There is no need to install any code on a client's computer and all the computations are performed on our dedicated computing cluster. Further, it makes it easier to compare the results of different methods to ours based on the same data.
News
17/03/11 - You can now use ▶ icon in 'Datasets table' to run SfM (No XML file needed).
06/01/11 - New web interface (version Tokyo).
Input Images
Output Model
Multi-View Stereo : Automatic 3

PRoVisG Mars 3D Challenge
PRoVisG MARS 3D Challenge
User login
Username:
Password:
Log in
(Register)
Menu
Introduction
Registration
Forum
Challenges
Stage1 - Deadline: 1 May '11
Stage2 - Deadline: 30 Jul '11
Stage3 - Deadline: 20 Sep '11
Calibration
Submit Results
Evaluation
Organizers
Participants
Welcome to the PRoVisG Mars 3D Challenge!
The Mars 3D Challenge aims at testing and improving the state of the art in visual odometry and 3D terrain reconstruction in planetary exploration.
The task of the challenge is to reconstruct depth, camera trajectory and 3D map of Mars landscape observed by the Mars Exploration Rovers (MER). Results of the challenge will help to develop technology for future planetary mission such as ESA ExoMars.
The challenge consists of three stages. The results of the challenge will be evaluated by a the PRoVisG Project consortium including the JPL NASA operating the MERs. Results of the challenge will be presented at a ICCV 2011 workshop "CVVT:E2M - Computer Vision in Vehicle Technology: From Earth to Mars" and published in a follow-up journal paper. The winners of the challenge will be awarded a travel to and a visit of a JPL/ESA space laboratory.
News
15/6/2011 - Stage 2 Challenge Released
9/3/2011- Additional info at M3DC Forum
1/3/2011 - Stage 1 Challenge Released
Register to the challenge and contribute to understanding the Universe!
All results welcome: it's possible to compete in stage 2 even if you didn't submit stage 1 results!



D Reconstruction of Tocnik data-set.
-set: 96 (1528 x 1148) images.
CENTER FOR MACHINE PERCEPTION
(c) 12.4.2011, Michal Jancosek, jancom1@cmp.felk.cvut.cz