SOFSEM 2008, High Tatras, Slovakia, 19-25 January 2008

An Algorithm for Computation of the Scene Geometry by the Log-Polar Area Matching Around Salient Points

Bogusław Cyganek

AGH - University of Science and Technology

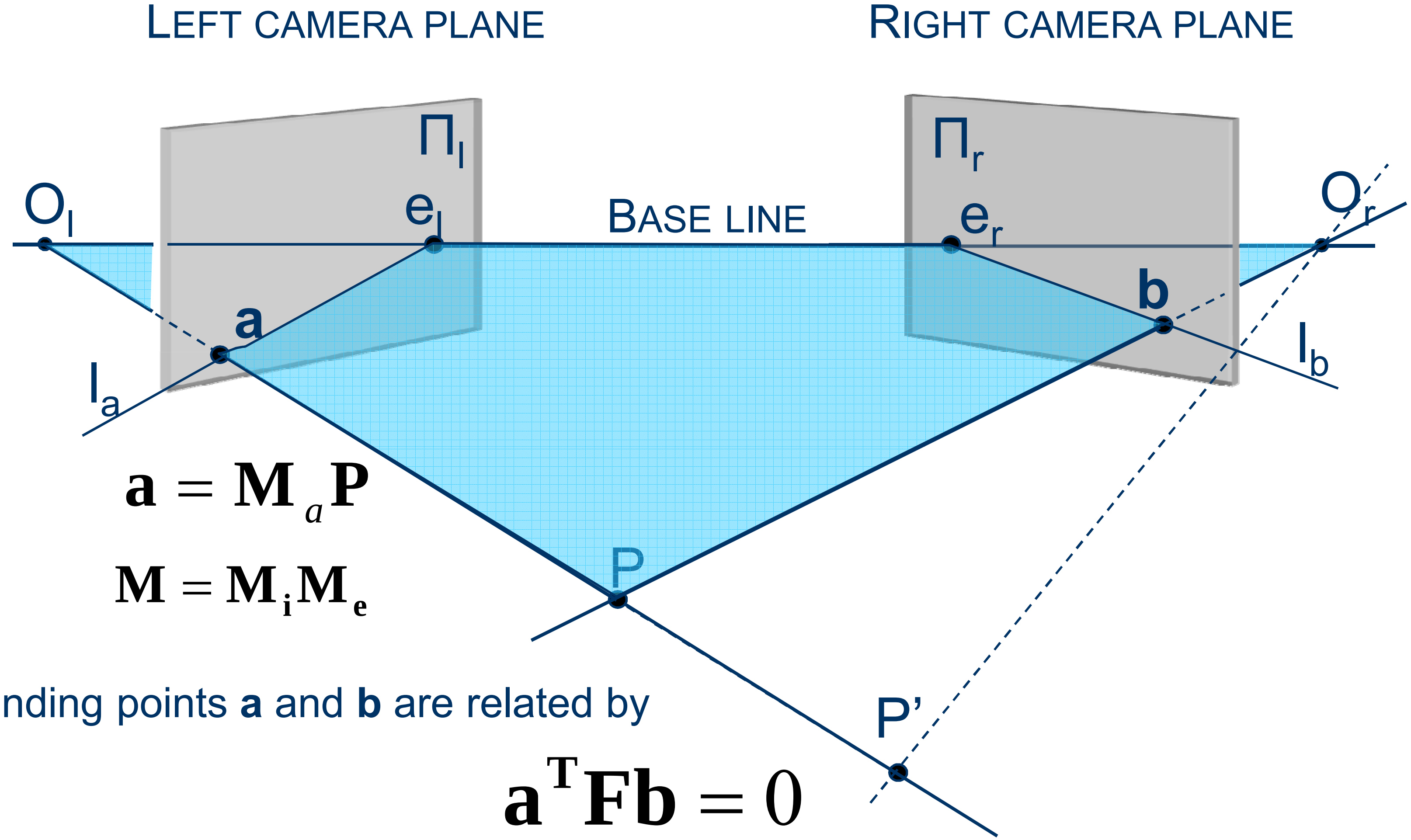
Al. Mickiewicza 30, 30-059 Kraków, Poland

cyganek@uci.agh.edu.pl

Outline

- Stereovision – mathematical formulation.
- Computation of the scene geometry – determination of the fundamental matrix by the linear optimization.
- Problems – point normalization & outliers.
- Salient points from the structural tensor.
- Matching in the extended log-polar space.
- Experimental results (other applications…).
- Conclusions.

Stereovision



$$\mathbf{a} = \mathbf{M}_a \mathbf{P}$$

$$\mathbf{M} = \mathbf{M}_i \mathbf{M}_e$$

The corresponding points **a** and **b** are related by

$$\mathbf{a}^T \mathbf{F} \mathbf{b} = 0$$

Once the matrix **F** is known, the epipolar lines, $\mathbf{l}_a$ for the left and $\mathbf{l}_b$ for the right camera, can be determined as follows

$$\mathbf{l}_a = \mathbf{F}\mathbf{b}, \qquad \mathbf{l}_b = \mathbf{F}^T\mathbf{a}.$$

$$\boxed{\mathbf{F} = ?}$$

Fundamental Matrix – Formulation

$$\mathbf{a}^{\mathrm{T}}\mathbf{F}\mathbf{b} = 0$$

$$r = \mathbf{q}^{\mathrm{T}}\mathbf{f} = \sum_{i=1}^{9} q_i f_i = 0$$

r is called a residual, vectors **q** and **f** are given as follows

$$\mathbf{q} = [a_1 b_1, a_2 b_1, b_1, a_1 b_2, a_2 b_2, b_2, a_1, a_2, 1]^{\mathrm{T}}$$

$$\mathbf{f} = [F_{11}, F_{12}, F_{13}, F_{21}, F_{22}, F_{23}, F_{31}, F_{32}, F_{33}]^{\mathrm{T}}$$

Then, a $K \geq 8$ number of the corresponding points is gathered into a compound matrix $\mathbf{Q}_{K\times 9}$. From **Q** the so called moment matrix $\mathbf{M}=\mathbf{Q}^{\mathrm{T}}\mathbf{Q}$ is created which is of size 9x9**. The matrix F** is found as an eigenvector **w** of **M** which corresponds to the lowest eigenvalue of **M**. This way found **F** minimizes the sum of squares of algebraic residuals

$$E = \sum_{k=1}^{K} \rho_k$$

Fundamental Matrix – Computation

$$E = \sum_{k=1}^{K} \rho_k = \sum_{k=1}^{K} \frac{r_k^2}{\mathbf{f}^T \mathbf{J} \mathbf{f}} = \sum_{k=1}^{K} \frac{\mathbf{a}_k^T \mathbf{F} \mathbf{b}_k}{\mathbf{f}^T \mathbf{J} \mathbf{f}} = \frac{\mathbf{f}^T \mathbf{M} \mathbf{f}}{\mathbf{f}^T \mathbf{J} \mathbf{f}}$$

$$\boxed{\min \left\{ E \right\}}$$

where $\mathbf{J}=\mathbf{J}_1=diag[1,1,\ldots,1]$ is the normalization matrix which corresponds to the optimization constraint in the form: $||f||=\sum_i f^2_i=1$.

Solution to the optimization problem is obtained as an eigenvector $\mathbf{f}_s=\mathbf{w}$ that corresponds to the **lowest eigenvalue** λ_k of the moment matrix ***M***.

Instead of $\mathbf{J}=\mathbf{J}_1$ Torr and Fitzgibbon proposed to apply a constraint which is invariant to the Euclidean transformations in the image planes. They showed that the Frobenius norm of the form $f^2_1+f^2_2+f^2_4+f^2_5$ fulfils such invariance requirement. This corresponds to $\mathbf{J}=\mathbf{J}_2=diag[1,1,0,1,1,0,0,0,0]$. Finding $\mathbf{f}_s$ in this case is more complicated since it is equivalent to solving the generalized eigenvector problem:

$$\mathbf{f}^T \mathbf{J} \mathbf{f} - \mathbf{f}^T \mathbf{M} \mathbf{f} = 0$$

Fundamental Matrix – Problems

There are at least two additional problems:
1. Wide dynamical range of the coordinates.
2. Outliers.

➔ Normalization

The normalization is done by an affine transformation **T**, consisting of translation and scaling, so that the centroid of the reference points is at the origin of the coordinate space and the root-mean-square distance of the points from the origin is √2.

$$\mathbf{a}' = \mathbf{T_a a} = \begin{bmatrix} s_a & 0 & -m_{a1}s_a \\ 0 & s_a & -m_{a2}s_a \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ 1 \end{bmatrix}$$

$\mathbf{m_a}=[m_{a1}, m_{a2}, 1]$ is a mean point,
$s_a=\sqrt{2}/d_{av}$ for d_{av} is an average point distance from the origin point *[0, 0, 1].*

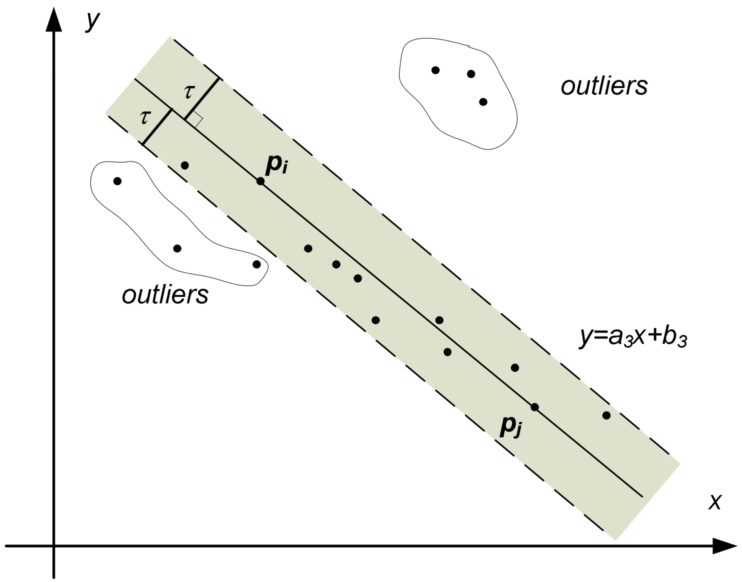
Normalization: $\mathbf{F}' = \mathbf{T_a^{-T} F T_b^{-1}}$

DeNormalization: $\mathbf{F} = \mathbf{T_a^{T} F' T_b}$

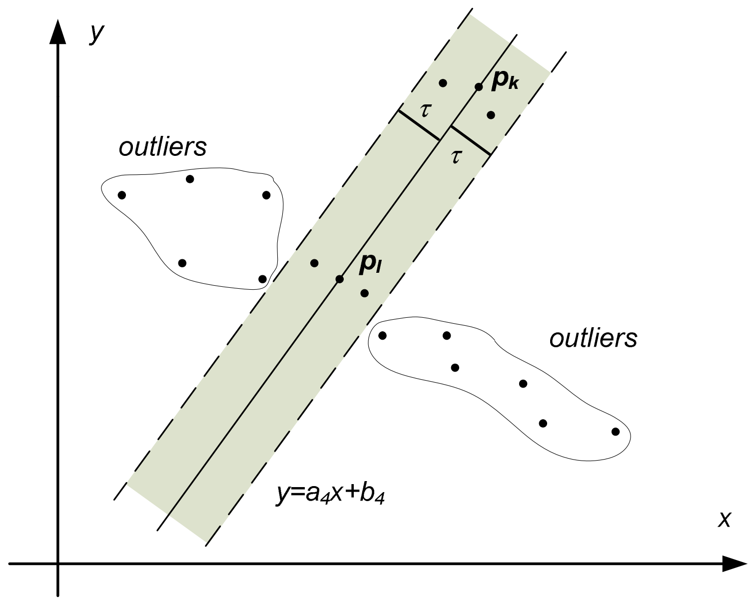
Fundamental Matrix – Problems

➔ <u>RANSAC + Log-Polar matching</u>

Randomly choose a number of samples from the set of all measurements, try to fit a model to them, and check how many other points are in consensus with this model estimate. The process is repeated and the best fit, i.e. an estimate supported by the maximal number of measurements is left as a solution. All other points are treated as outliers.

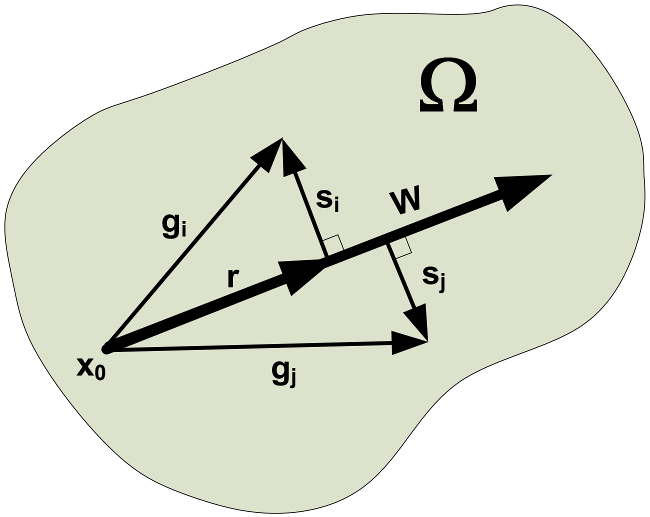


Fitting a line to points – a step of the RANSAC algorithm. Randomly selected pair of points x_i and x_j serves an initial estimate $y=a_3x+b_3$. Distances of all other points to this estimate are checked and only those within a predefined threshold (grey area) are considered as inliers



Fitting a line to points by the RANSAC algorithm. Another randomly selected pair of points x_k and x_l serves other estimate $y=a_4x+b_4$. New region of inliers denoted in grey

Points-of-interest from Structural Tensor



A local neighbourhood Ω is represented with a single orientation vector **w**, which is accurate if most of the gradients $\mathbf{g}_\mathbf{i}$ in Ω coincide with **w** . This is equivalent to minimization of a cumulative sum of the residual vectors $\mathbf{s}_\mathbf{i}$ in Ω which is also equivalent to finding eigenvectors of the structural tensor

$$\mathbf{s} = \mathbf{g} - \mathbf{r} = \mathbf{g} - \frac{\mathbf{w}}{\|\mathbf{w}\|}\|\mathbf{r}\| = \mathbf{g} - \frac{\mathbf{w}\mathbf{g}^T\mathbf{w}}{\mathbf{w}^T\mathbf{w}}$$

The error function:

$$e(\mathbf{x},\mathbf{x}_0) = \|\mathbf{s}\| = \left\|\mathbf{g}(\mathbf{x}) - \mathbf{w}(\mathbf{x}_0)\mathbf{g}^T(\mathbf{x})\mathbf{w}(\mathbf{x}_0)\right\| \qquad \mathbf{w}^T\mathbf{w} = 1$$

$$E(\mathbf{x}_0) = \int_\Omega e^2(\mathbf{x},\mathbf{x}_0) G_\sigma(\mathbf{x},\mathbf{x}_0) d\mathbf{x}$$

$$\boxed{\min_\mathbf{w} \|E\|}$$

Points-of-interest from Structural Tensor

$$\mathbf{w} = \left[T_{11} - T_{22} \quad 2T_{12} \right]^T$$

where

$$\mathbf{T} = \int_{\Omega} \mathbf{g}^T \mathbf{g} G_{\sigma}\left(\mathbf{x}, \mathbf{x}_0\right) d\mathbf{x}$$

STRUCTURAL TENSOR

Corner points (x_i, y_i) for matching fulfil the following condition

$$\lambda_1\left(x_i, y_i\right) \geq \lambda_2\left(x_i, y_i\right) \geq \mu$$

$$\lambda_{1,2} = \frac{1}{2}\left[\left(T_{11} + T_{22}\right) \pm \sqrt{\left(T_{11} - T_{22}\right)^2 + 4{T_{12}}^2}\right]$$

A priority queue for selection of image points with the strongest responses

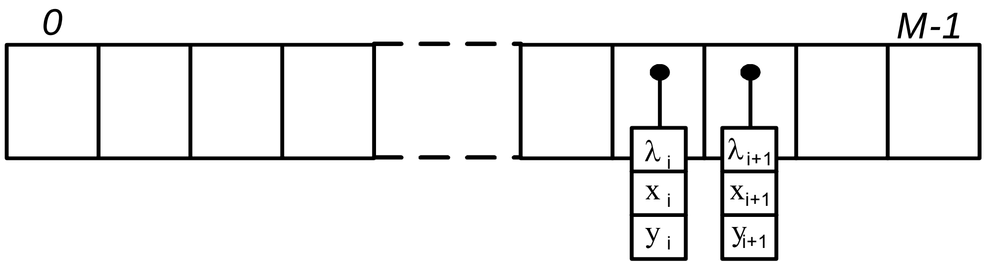
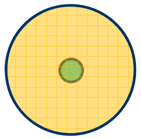


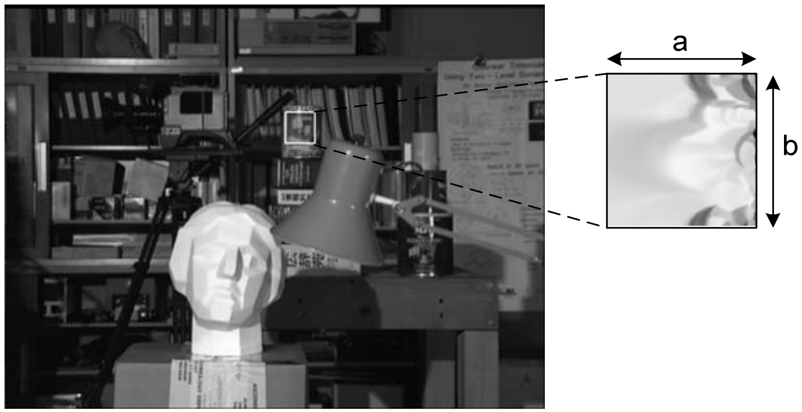
Image Matching in the Log-Polar Domain

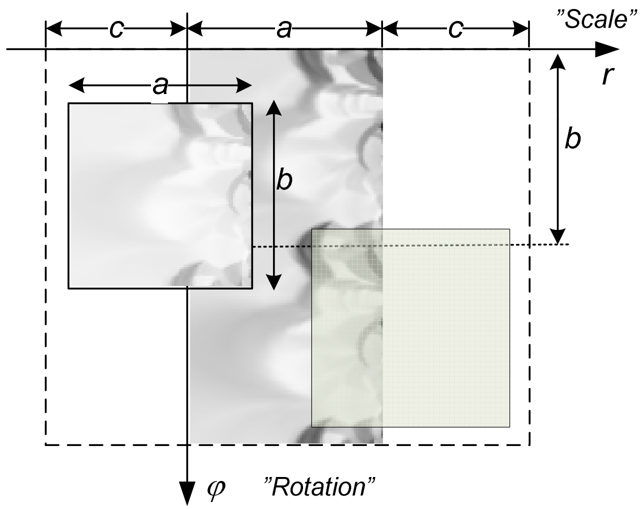
The log-polar transformation takes points *(x,y)* from the Euclidean space into the *(r,φ)* points in the polar space



$$r = \log_B\left(\sqrt{\left(x-x_0\right)^2+\left(y-y_0\right)^2}\right) \qquad \varphi = \arctan\frac{y-y_0}{x-x_0}, \quad \textit{for} \quad x \neq x_0$$

Matching a pattern in the extended log-polar space: For each position search is done in two dimensions to account for scale and rotation which results in a 4D search





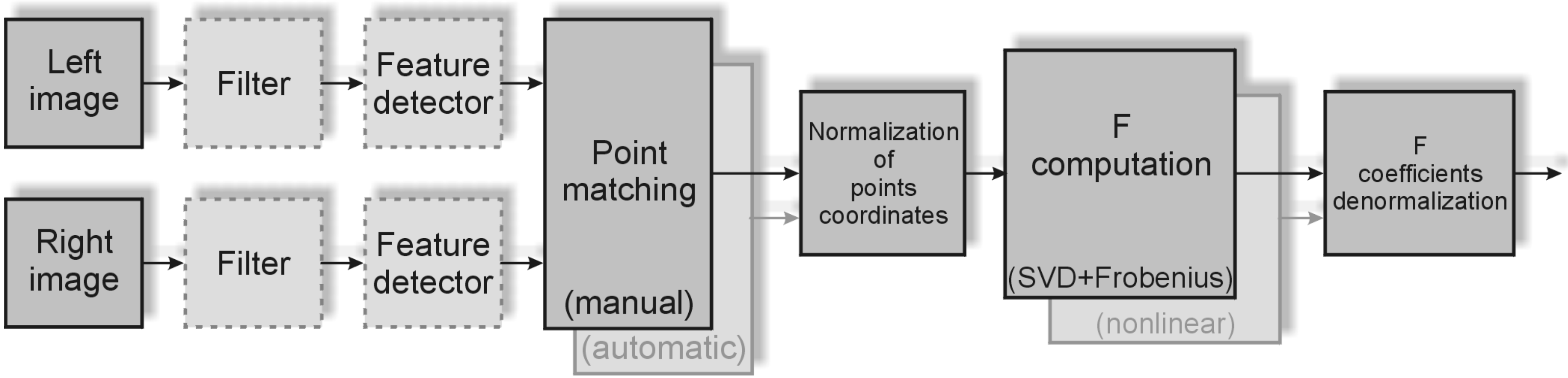
Each position in the extended space is matched pixel-by-pixel with ρ

$$\rho = \frac{\sum_{(i,j)\in U}\left(I_1(x_1+i,x_2+j)-\overline{I_1(x_1,x_2)_U}\right)\cdot\left(I_2(x_1+i,x_2+j)-\overline{I_2(x_1,x_2)_U}\right)}{\sqrt{\sum_{(i,j)\in U}\left(I_1(x_1+i,x_2+j)-\overline{I_1(x_1,x_2)_U}\right)^2\cdot\sum_{(i,j)\in U}\left(I_2(x_1+i,x_2+j)-\overline{I_2(x_1,x_2)_U}\right)^2}}$$

Experimental Setup

The ability of the LP matching to detect local rotation and scale can be used to **sieve out the outliers**. Our idea here is to reject all the matching pairs for which their local rotation or scale deviates significantly from 0.

The code was written in C++ on the Microsoft® .NET 2005 IDE. The tests were performed on the laptop computer with the Intel® Core™ Duo processor (dual-core) 2MB L2 cache (T2600 with 2.16GHz speed) as well as the 2GB of the operational memory (RAM).



Experimental Results

Matching of the exemplary video sequence. First frame (a) with salient points from the structural tensor detector. The next (5th) frame in a sequence with the matched points (b).
Outliers are the pairs of points no. 2 and 5

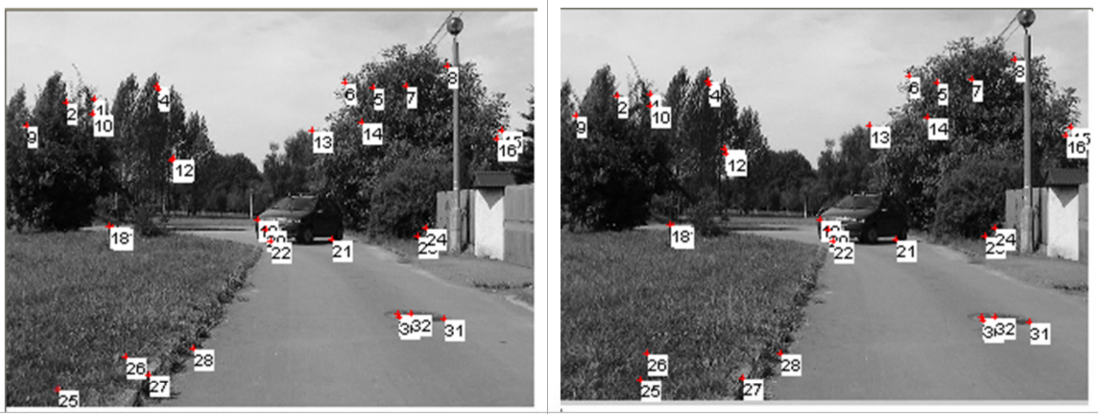
a b

Results of the log-polar matching process of the image above. Outliers in grey:

No	Left pt. (x,y)	Right pt. (x,y)	Best match val. ρ	LP (*scale*, *rotation*)
1	248, 51	(253, 52)	0.929104	(0, 0)
2	158, 66	(253, 48)	0.924074	(15, 2)
3	227, 24	(232, 25)	0.912579	(0, 0)
4	16, 105	(20, 105)	0.840755	(0, 0)
5	288, 80	(293, 144)	0.634407	(-14, 4)
6	135, 117	(135, 117)	0.746453	(0, 1)
7	170, 136	(173, 136)	0.876785	(0, 0)
8	192, 175	(199, 175)	0.846410	(0, 0)

Experimental Results

Matching of the “Car” stereo pair. Left image (a) with salient points obtained by the structural tensor detector. The right image with the matched points (b). Two outliers



a b

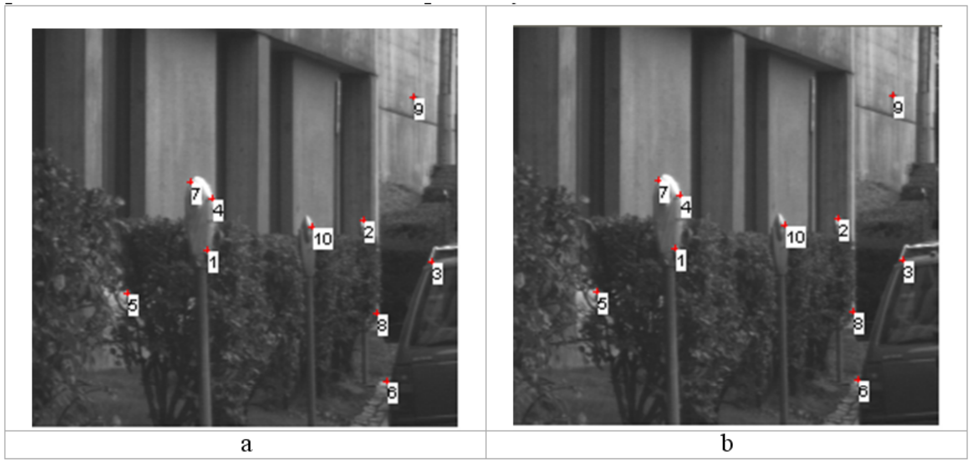
$$\mathbf{F} = \begin{bmatrix} 0.492126 & -0.343832 & 0.169008 \\ -0.269639 & -0.190649 & 0.0927313 \\ 0.167368 & 0.117041 & -0.0574843 \end{bmatrix}$$

Results of the LP matching process of the above images. Outliers in grey:

No	Left pt. (x,y)	Right pt. (x,y)	Best match val. ρ	LP (*scale, rotation*)
1	(53,53)	(54,51)	0.929504	(0,0)
2	(36,55)	(34,52)	0.826554	(0,0)
…	…	…	…	…
25	(31,229)	(48,224)	0.807608	(0,13)
26	(72,209)	(52,208)	0.795199	(0,0)
…	…	…	…	…
32	(245,183)	(263,186)	0.976108	(0,0)

Experimental Results

Matching of the “Parkmeter” images. Left image (a) with salient points obtained by the structural tensor detector. The right image with the matched points (b). No outliers



Matching “Tsukuba” pair divided into 4×4 tiles. 27×27 LP window used for matching. Left (a) with detected corners. The right with the matched points (b). No outliers

Conclusions

- Efficient point matching method: stereovision (in this presentation), but also object detection (Internet), object tracking, etc..
- Computation of the salient points (corners) from the structural tensor.
- Matching around salient points but in **the log-polar** space.
- Advantage of this approach is possible detection of **outliers** based on their excessive **local rotation** and/or **scale**.

Thank you!