Camera Pose Estimation from Line and Point Correspondences

Alexander Vakhitov[1]

[1]Skolkovo Institute of Science and Technology

Second Christmas Colloqiuim on Computer Vision, 28.12.2016

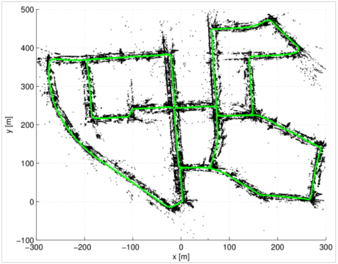
The report is supported by the Ministry of Education and Science grant RFMEFI61516X0003

Talk Outline

Some Applications of Multiple View Geometry

Structure from Motion	Visual SLAM
Bundler (2006)	MonoSLAM (2007)
Visual SFM (2013)	PTAM (2007)
Theia (2015)	LSD-SLAM (2014)
COLMAP (2016)	ORB-SLAM (2014)

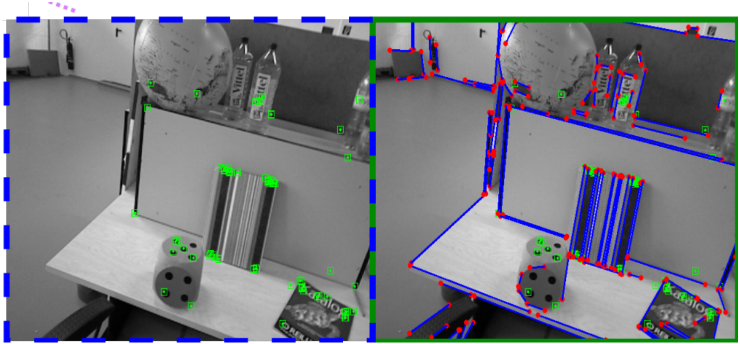
Bundler (N. Snavely et al., 2006)



ORB-SLAM (Mur et al., 2015)

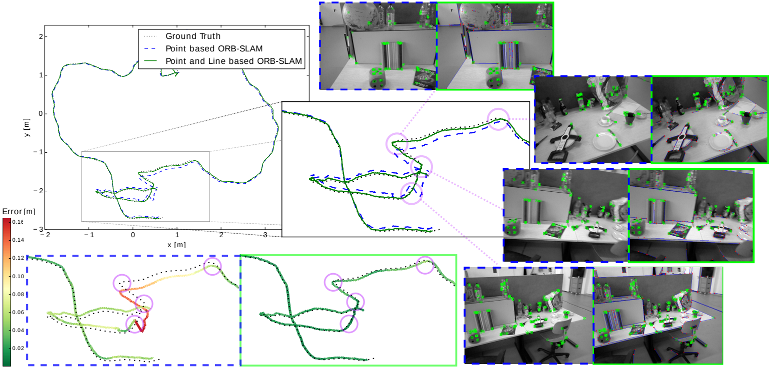
Features in Multiple View Geometry

- Points the **only** widely used in Visual SLAM and SfM features
- But we need more...



(TUM-RGBD dataset, image kindly provided by A. Pumarola, IRI-UPC)

Lines Meet ORB-SLAM



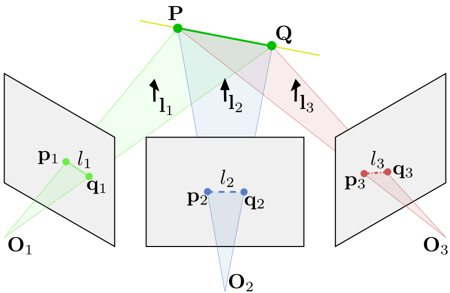
(TUM-RGBD dataset, image kindly provided by A. Pumarola, IRI-UPC)

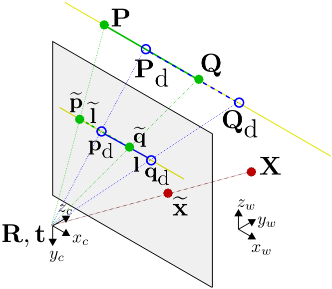
Notation

$\mathbf{U}$ - point of $\mathbb{R}^3$
$\mathbf{u}$ - point of $\mathbb{R}^2$
$\mathtt{R}$ - matrix

Some Geometric Tasks in Incremental SfM

- *Relative pose.* Having two or three 2D images, find their relative location (position+orientation) in space
- *Absolute pose.* Having 3D model and 2D image, estimate camera location w.r.t. the model

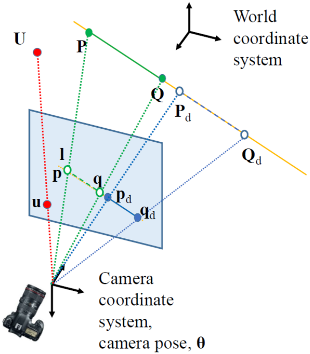




Line Matching Difficulties

- Blue - detected
- Green - reprojected
- White - manually marked model contours

Perspective-n-Point+Lines

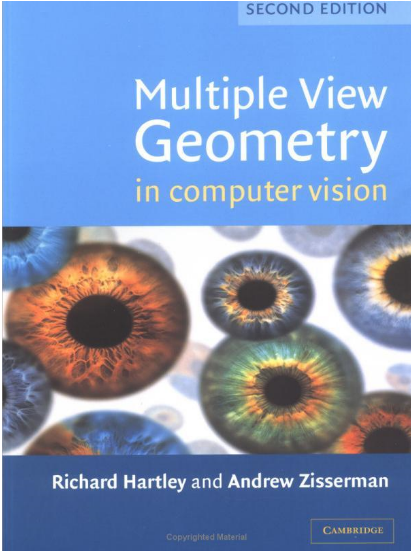


- Points: 3D $\mathbf{U}$ and 2D $\mathbf{u}$
- Line segments: 3D $\mathbf{P}, \mathbf{Q}$ and 2D $\mathbf{p}, \mathbf{q}$
- Detected line segment $\mathbf{p}_\mathrm{d}, \mathbf{q}_\mathrm{d}$, its reprojection onto 3D line $\mathbf{P}_\mathrm{d}, \mathbf{Q}_\mathrm{d}$
- model has n_p points, n_l lines
- $\boldsymbol{\theta}$ - camera pose parameters encoding $\mathtt{R}, \mathbf{t}$
- normalized camera (unit focal, zero center shift)

Perspective-n-Point+Lines

We can construct 3D model using 3 frames with SIFT point descriptors and SMLSD line descriptors (we need observation redundancy to filter outliers)
Dataset: NYU2.

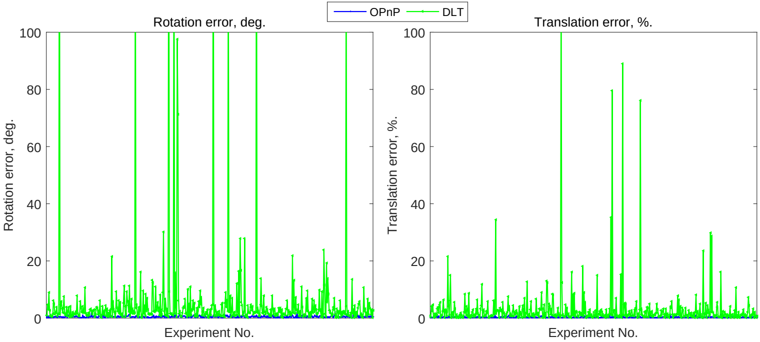
Motivating Example



If we wish to solve PnP, we open a book...
It offers Direct Linear Transform algorithm. Let's try?

Motivating Example (2). DLT vs OPnP

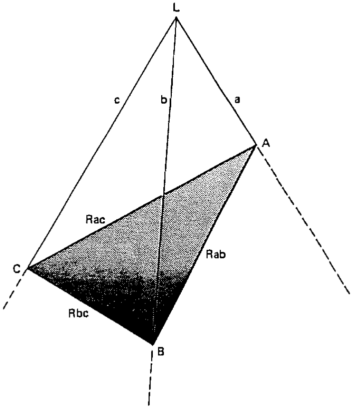
Numerical experiment: we generate 6 points in a box $[-2,2] \times [-2,2] \times [4,8]$ in front of the camera, project them with additive gaussian noise with std.dev. 1 pixel onto usual 60°-wide camera, generate random $\mathtt{R}$, $\mathbf{t}$, give a rotated by $\mathtt{R}^{-1}$ and shifted by $-\mathtt{R}^{-1}\mathbf{t}$ model to the methods, willing to get $\mathtt{R}, \mathbf{t}$.



Problem History

- XIX- beginning of XX cent. - first projective geometry results, numerical algorithms for photogrammetry
- 1960s polynomial system solving developed (Buchberger, under Groebner's supervision)
- 1970s-1980s - first PC algorithms, projective resonstruction methods (DLT)
- 1990s-2000s - minimal problem methods (P3P, P2P1L, P4Pf, etc) using Groebner bases
- 1990s - locally converging iterative algorithms for PnP
- 2000s - efficient robust algorithms for hundreds of points
- 2010s - PnP using Groebner bases

P3P Algorithm



(Fischler, Bolles, 1981)
3Cosine theorem for triangles with vertex **L**, we get 3 quadratic equations w.r.t. a, b, c, reducable to 4th order one variable equation.
At most, 4 solutions.

Direct Linear Transformation (Abdel-Aziz, Y. et al., 1971)

Homogeneous camera coordinates: $\tilde{\mathbf{x}} = \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}$.

Perspective projection with matrix $\mathtt{P}$:

$$\lambda \tilde{\mathbf{x}} = \mathtt{P} \begin{pmatrix} \mathbf{X} \\ 1 \end{pmatrix} \tag{1}$$

Get rid of λ:

$$\tilde{\mathbf{x}} \times \left(\mathtt{P} \begin{pmatrix} \mathbf{X} \\ 1 \end{pmatrix} \right) = 0 \tag{2}$$

For $n \geq 6$, we find $\mathtt{P}$.

Euclidean reconstruction with DLT

Having $\mathtt{P}$, $\mathtt{R}, \mathtt{t}$ - ?

$$\bar{\mathbf{X}} = \frac{1}{n}\sum_i \mathbf{X}_i$$

$$\mathtt{C} = \frac{1}{n}\sum_i (\mathbf{X}_i - \bar{\mathbf{X}})(\mathtt{P}\begin{pmatrix} \mathbf{X}_i - \bar{\mathbf{X}} \\ 1 \end{pmatrix})^T.$$

Orthogonal Procrustes

$$[\mathtt{U}, \mathtt{S}, \mathtt{V}] = SVD(\mathtt{C}) \quad \Longrightarrow \quad \mathtt{R} = \mathtt{U}\mathtt{V}^T.$$

$$\sum_i \|\mathtt{R}\mathbf{X}_i + \mathbf{t} - s\mathtt{P}\begin{pmatrix} \mathbf{X}_i \\ 1 \end{pmatrix}\|^2 \to \min_{s,\mathbf{t}}.$$

General scheme of PnP method

Having $\pi(\boldsymbol{\theta}, \mathbf{X})$ - projection function acting on 3D model point $\mathbf{X}$ outputting homogeneous camera projection coordinates $\tilde{\mathbf{x}}$:

$$\lambda_i \tilde{\mathbf{x}} = \pi(\boldsymbol{\theta}, \mathbf{X}) + \lambda_i \tilde{\xi}_i, \quad \pi(\boldsymbol{\theta}, \mathbf{X}) = \begin{pmatrix} \pi^{(1)}(\boldsymbol{\theta}, \mathbf{X}) \\ \pi^{(2)}(\boldsymbol{\theta}, \mathbf{X}) \\ \pi^{(3)}(\boldsymbol{\theta}, \mathbf{X}) \end{pmatrix}, \tag{3}$$

where $\tilde{\xi}_i = \begin{pmatrix} \xi_i \\ 1 \end{pmatrix}$, ξ_i - detection noise.

Example

$$\pi(\boldsymbol{\theta}, \mathbf{X}) = \mathtt{R}(\mathbf{q})\mathbf{X} + \mathbf{t}, \quad q = (a, b, c, d), \quad \|q\| = 1$$

$$\mathtt{R}(\mathbf{q}) = \begin{pmatrix} a^2 + b^2 - c^2 - d^2 & 2bc - 2ad & 2bd + 2ac \\ 2bc + 2ad & a^2 - b^2 + c^2 - d^2 & 2cd - 2ab \\ 2bd - 2ac & 2cd + 2ab & a^2 - b^2 - c^2 + d^2 \end{pmatrix}.$$

General scheme of PnP method (2)

$$\lambda_i \tilde{\mathbf{x}} = \pi(\boldsymbol{\theta}, \mathbf{X}) + \lambda_i \tilde{\xi}_i$$

From eq. 3 express λ_i and substitute into 1,2:

$$\pi^{(3)}(\boldsymbol{\theta}, \mathbf{X})\mathbf{x} = \pi^{(1,2)}(\boldsymbol{\theta}, \mathbf{X}). \tag{4}$$

Get a system of $2n$ eqs. (4) and solve it in least squares sense:

$$E_p = \sum_i \|\pi^{(3)}(\boldsymbol{\theta}, \mathbf{X})\mathbf{x} - \pi^{(1,2)}(\boldsymbol{\theta}, \mathbf{X})\|^2 \to \min. \tag{5}$$

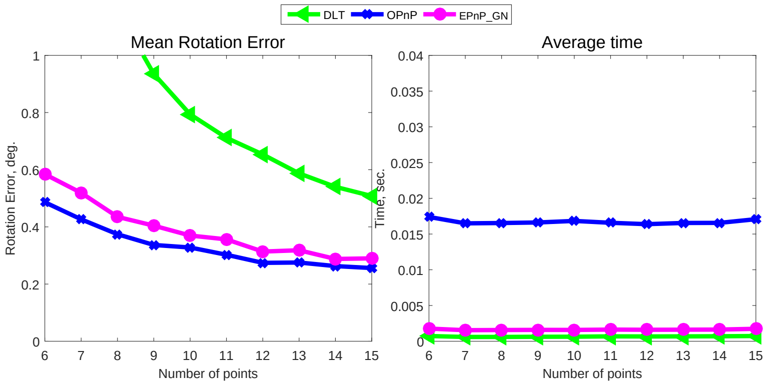
OPnP vs EPnP

$$E_p = \sum_i \|\pi^{(3)}(\boldsymbol{\theta}, \mathbf{X})\mathsf{x} - \pi^{(1,2)}(\boldsymbol{\theta}, \mathbf{X})\|^2 \to \min .$$

$$\nabla_\theta E_p = 0.$$

OPnP - $\pi(\boldsymbol{\theta}, \mathsf{x})$ - polynomial, $dim(\boldsymbol{\theta}) = 4$, solve polynomial equations
EPnP - $\pi(\boldsymbol{\theta}, \mathsf{x})$ - linear, $dim(\boldsymbol{\theta}) = 12$, but there are quadratic constraints on the components of $\boldsymbol{\theta}$. Relinearization.

Comparison of OPnP and EPnP



Generalization to PnPL

Line equation from the detected segment endpoints:

$$\hat{\mathbf{l}}^i = \tilde{\mathbf{p}}^i_{\mathrm{d}} \times \tilde{\mathbf{q}}^i_{\mathrm{d}}, \quad \mathbf{l}^i = \frac{\hat{\mathbf{l}}^i}{|\hat{\mathbf{l}}^i|} \quad \in \mathbb{R}^3. \tag{6}$$

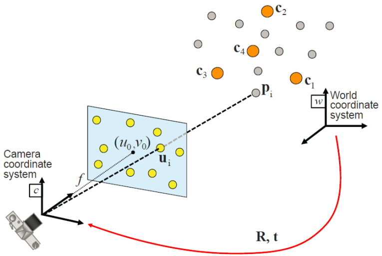
Algebraic point-to-line distance:

$$\mathrm{E}_{\mathrm{pl}}(\boldsymbol{\theta}, \mathbf{P}^i, \mathbf{l}^i) = (\mathbf{l}^i)^\top \pi(\boldsymbol{\theta}, \mathbf{P}^i), \tag{7}$$

Algebraic segment-to-line distance:

$$\mathrm{E}_{\mathrm{l}}(\boldsymbol{\theta}, \mathbf{P}^i, \mathbf{Q}^i, \mathbf{l}^i) = \mathrm{E}^2_{\mathrm{pl}}(\boldsymbol{\theta}, \mathbf{P}^i, \mathbf{l}^i) + \mathrm{E}^2_{\mathrm{pl}}(\boldsymbol{\theta}, \mathbf{Q}^i, \mathbf{l}^i). \tag{8}$$

Efficient PnP



(Lepetit, Moreno-Noguer, Fua, 2007; 2009)
First $O(n)$ algotithm for PnP.

Choose 4 control points $\mathbf{C}_i$, not in one plane.

$$\mathbf{P}_i = a_{i,1}\mathbf{C}_1 + a_{i,2}\mathbf{C}_2 + a_{i,3}\mathbf{C}_3 + a_{i,4}\mathbf{C}_4$$

$$\forall \mathtt{R}, \mathbf{t}: \quad \mathtt{R}\mathbf{P}_i + \mathbf{t} = \sum_j a_{i,j}(\mathtt{R}\mathbf{C}_j + \mathbf{t})$$

$a_{i,j}$ do not change under rotation and translation.

Efficient PnP (2)

$$\pi_{\mathrm{EPnP}}(\boldsymbol{\theta}, \mathbf{X}_i) = \sum_{j=1}^{4} a_{i,j}\mathbf{C}_j \tag{9}$$

We get 2 equations w.r.t. $\boldsymbol{\mu} = (\mathbf{C}_{c,1}^T, \mathbf{C}_{c,2}^T, \mathbf{C}_{c,3}^T, \mathbf{C}_{c,4}^T)$ for a single 3D-2D match, пo using all correspondences we form a linear system:

$$\mathbf{M}\boldsymbol{\mu} = \mathbf{0}, \quad \mathbf{M} \in \mathbb{R}^{2n\times 12} \tag{10}$$

Without noise ξ_i in point detections, $\mathbf{M}$ has a null space of dimension 1. But, in real life, we need to seek for a solution in a linear subspace of the singular vectors $\mathbf{v}_1, \ldots, \mathbf{v}_N$ $\mathbf{M}$, corresponding to $N = 1, 2, 3, 4$ smallest singular values of $\mathbf{M}$.

Efficient PnP (3)

So, $\boldsymbol{\mu}$ can be represented as

$$\boldsymbol{\mu} = \sum_{i=1}^{N} \beta_i \mathbf{v}_i \tag{11}$$

To find a unique solution we use invariance of distance between points under rotation and translation:

$$\|\mathbf{C}_i - \mathbf{C}_j\|^2 = r_{ij}^2, \quad i, j = 1, \ldots, 4, i \neq j. \tag{12}$$

Substitute (11) in (12), get quadratic system of 6 equations w.r.t. $\beta_1, \ldots, \beta_N$.
We solve it using relinearization, defining $\beta_i \beta_j$ as new unknowns.

Efficient PnP (4): Relinearization

Denote $\gamma_k = \beta_i \beta_j$, compose $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_{N+N(N-1)/2})^T$, $\mathbf{w} = (r_{12}, \ldots, r_{34})^T$ and get a system:

$$\Gamma \boldsymbol{\gamma} = \mathbf{w}. \tag{13}$$

Problem: when $N > 2$ usually system has multiple solutions. EPnP uses relinearization second time, introducing unknowns $\delta_s = \gamma_i \gamma_j$ and equations $\gamma_i \gamma_j = \gamma_k \gamma_l$, which is $(\beta_{i1}\beta_{i2})(\beta_{i3}\beta_{i4}) = (\beta_{i1}\beta_{i3})(\beta_{i2}\beta_{i4})$.

OPnP Algorithm

Constraints are:

$$\lambda_i \tilde{\mathbf{x}} = \mathtt{R}\mathbf{X}_i + \mathbf{t}. \tag{14}$$

Divide by avg depth $\bar{\lambda} = \frac{1}{n}\sum_i \lambda_i$ and denote $\hat{\mathtt{R}} = (\bar{\lambda})^{-1}\mathtt{R}$, $\hat{\mathbf{t}} = (\bar{\lambda})^{-1}\mathbf{t}$, $\hat{\lambda}_i = \frac{\lambda_i}{\bar{\lambda}}$:

$$\hat{\lambda}_i \tilde{\mathbf{x}} = \hat{\mathtt{R}}\mathbf{X}_i + \hat{\mathbf{t}}. \tag{15}$$

Summing up equation triplets, get

$$\hat{t}^{(3)} = n(1 - \hat{\mathbf{r}}_3^T \bar{\mathbf{X}}), \quad \bar{\mathbf{X}} = \frac{1}{n}\sum \mathbf{X}_i.$$

Algorithm OPnP (2)

Parameterize $\hat{\mathtt{R}}$ using non-unit quaternion $\mathbf{q} = (a,\ b,\ c,\ d)^T$:

$$\hat{\mathtt{R}}(\mathbf{q}) = \begin{pmatrix} a^2+b^2-c^2-d^2 & 2bc-2ad & 2bd+2ac \\ 2bc+2ad & a^2-b^2+c^2-d^2 & 2cd-2ab \\ 2bd-2ac & 2cd+2ab & a^2-b^2-c^2+d^2 \end{pmatrix}.$$

Write equations w.r.t. vectorized rotation matrix $\hat{\mathbf{r}}(\mathbf{q})$ и $\hat{\mathbf{t}}^{(1,2)}$:

$$\mathrm{E}_{\mathrm{points}}(\hat{\mathbf{r}}(\mathbf{q}), \hat{\mathbf{t}}) = \|\mathtt{G}_{\mathrm{p}}\hat{\mathbf{r}}(\mathbf{q}) + \mathtt{H}_{\mathrm{p}}\hat{\mathbf{t}}^{(1,2)} + \mathbf{k}_{\mathbf{p}}\|^2 \ \to \min, \tag{16}$$

for known $\mathtt{G}_{\mathrm{p}}, \mathtt{H}_{\mathrm{p}}$ and $\mathbf{k}_{\mathbf{p}}$.

$$\nabla_{\hat{\mathbf{q}}}\mathrm{E}_{\mathrm{points}} = 0,$$

$$\nabla_{\hat{\mathbf{t}}}\mathrm{E}_{\mathrm{points}} = 0.$$

Algorithm OPnP (3)

From $\nabla_{\hat{\mathbf{t}}} \mathrm{E}_{\mathrm{points}} = 0$:

$$\mathtt{H}_{\mathrm{p}}^{\top}(\mathtt{G}_{\mathrm{p}}\hat{\mathbf{r}} + \mathtt{H}_{\mathrm{p}}\hat{\mathbf{t}}^{(1,2)} + \mathbf{k}_{\mathrm{p}}) = 0 \quad \Longrightarrow \quad \hat{\mathbf{t}}^{(1,2)} = \mathtt{P}\hat{\mathbf{r}} + \mathbf{u}, \tag{17}$$

$$\mathtt{P} = -(\mathtt{H}_{\mathrm{p}}^{\top}\mathtt{H}_{\mathrm{p}})^{-1}(\mathtt{H}_{\mathrm{p}}^{\top}\mathtt{G}_{\mathrm{p}}), \;\; \mathbf{u} = -(\mathtt{H}_{\mathrm{p}}^{\top}\mathtt{H}_{\mathrm{p}})^{-1}\mathtt{H}_{\mathrm{p}}^{\top}\mathbf{k}_{\mathrm{p}}. \tag{18}$$

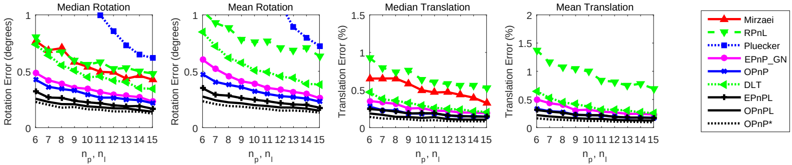
From $\nabla_{\mathbf{q}} \mathrm{E}_{\mathrm{points}} = 0$, for the derivative w.r.t. first quaternion component $\mathbf{q} = (a, \ldots)$:

$$\frac{\partial \hat{\mathbf{r}}}{\partial a}\mathtt{G}_{\mathrm{p}}^{\top}(\mathtt{G}_{\mathrm{p}}\hat{\mathbf{r}} + \mathtt{H}_{\mathrm{p}}\hat{\mathbf{t}}^{(1,2)} + \mathbf{k}_{\mathrm{p}}) = 0. \tag{19}$$

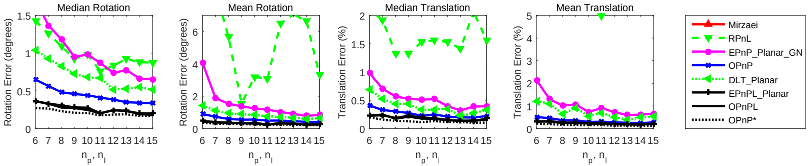
Remains to solve a system of 4 polyomial equations (19) of deg 3.

OPnPL, EPnPL

(a) General case: PnP (n_p), PnL(n_l), PnPL (n_p, n_l)

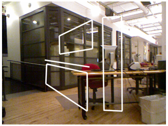
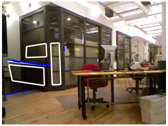
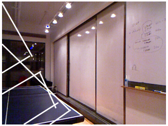
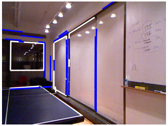
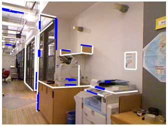
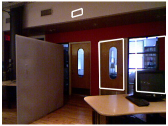
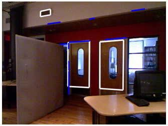


(b) COplanar points and lines: PnP (n_p), PnL(n_l), PnPL (n_p, n_l)

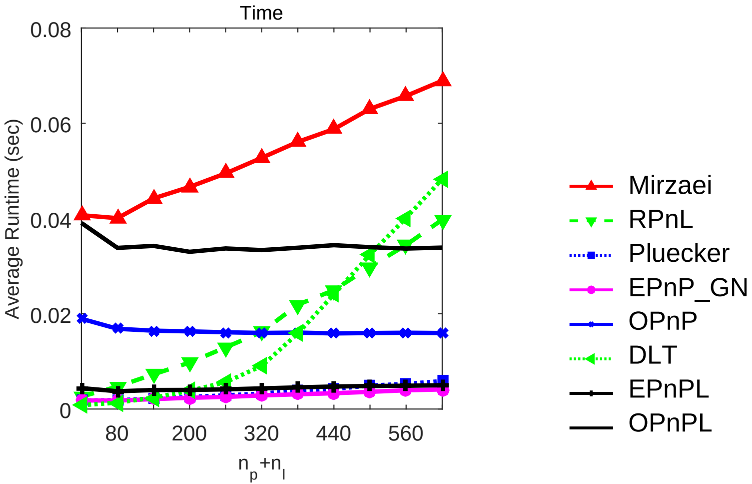


Camera pose from points and lines. Accuracy w.r.t. feature number.

PnPL using NYU2 dataset

OPnP (pt)	OPnPL (lin)/(pt+lin)	OPnP (pt)	OPnPL (lin)/(pt+lin)
$(21.3, 100.0)$	$(9.3, 30.5)/(9.5, 28.2)$	$(61.5, 605.4)$	$(0.4, 6.0)/(0.3, 5.7)$
			
$(1.3, 33.8)$	$(0.6, 9.1)/(0.2, 2.6)$	$(11.5, 100.0)$	$(2.0, 71.8)/(1.3, 30.1)$
			

Time for big n, PnPL problem



References

Descriptors

- D. Lowe, SIFT
- Bart Verhagen, Radu Timofte, and Luc Van Gool. Scale-invariant line descriptors for wide baseline matching. In IEEE Conf. on Applications of Computer Vision (WACV), 2014

Алгоритмы PnP

- Y. Zheng, Y. Kuang, S. Sugimoto, K. Astrom, and M. Okutomi. Revisiting the pnp problem: A fast, general and optimal solution. In IEEE Conf. on Computer Vision (ICCV)
- V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnp: An accurate O(n) solution to the PnP problem. International Journal of Computer Vision, 81(2):155-166, 2009.
- E. Kanaeva, L. Gurevich and A. Vakhitov. Camera Pose and Focal length Estimation Using Regularized Distance Constraints, BMVC 2015

References (2)

Datasets

- Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In Computer Vision-ECCV 2012, pages 746-760.
- Christoph Strecha, Wolfgang von Hansen, L Van Gool, Pascal Fua, and Ulrich Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In IEEE Conf. on Computer Vision and Pattern Recognition, 2008, pages 1-8.

Conclusion

Thank you for coming!
Thanks for this wonderful opportunity, and Happy New Year!