Camera Pose Estimation from Line and Point Correspondences

Alexander Vakhitov\textsuperscript{1}

\textsuperscript{1}Skolkovo Institute of Science and Technology


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

1. Problem Settings
2. Introduction
3. Efficient PnP
4. Optimal PnP
5. Biblio
6. Conclusion
Some Applications of Multiple View Geometry

<table>
<thead>
<tr>
<th>Structure from Motion</th>
<th>Visual SLAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLMAP (2016)</td>
<td>ORB-SLAM (2014)</td>
</tr>
</tbody>
</table>

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

\begin{itemize}
  \item \textbf{U} - point of $\mathbb{R}^3$
  \item \textbf{u} - point of $\mathbb{R}^2$
  \item \textbf{R} - matrix
\end{itemize}
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 $\mathbf{p}_d, \mathbf{q}_d$, its reprojection onto 3D line $\mathbf{P}_d, \mathbf{Q}_d$
- Model has $n_p$ points, $n_l$ lines
- $\theta$ - camera pose parameters encoding $\mathbf{R}, \mathbf{t}$
- Normalized camera (unit focal, zero center shift)
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 $60^\circ$-wide camera, generate random $R, t$, give a rotated by $R^{-1}$ and shifted by $-R^{-1}t$ model to the methods, willing to get $R, t$. 

![Graphs showing rotation and translation errors for OPnP and DLT methods.]
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 reconstruction 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
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: \( \tilde{x} = \begin{pmatrix} x \\ 1 \end{pmatrix} \).

Perspective projection with matrix \( P \):

\[
\lambda \tilde{x} = P \begin{pmatrix} x \\ 1 \end{pmatrix}
\]

(1)

Get rid of \( \lambda \):

\[
\tilde{x} \times \left( P \begin{pmatrix} x \\ 1 \end{pmatrix} \right) = 0
\]

(2)

For \( n \geq 6 \), we find \( P \).
Euclidean reconstruction with DLT

Having \( P, R, t \) - \\

\[
\bar{X} = \frac{1}{n} \sum_i X_i \\
C = \frac{1}{n} \sum_i (X_i - \bar{X})(P \left( X_i - \bar{X} \right))^T.
\]

Orthogonal Procrustes

\[
[U, S, V] = SVD(C) \quad \implies \quad R = UV^T.
\]

\[
\sum_i \|RX_i + t - sP \left( \begin{array}{c} X_i \\ 1 \end{array} \right) \|^2 \to \min_{s,t}.
\]
General scheme of PnP method

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

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

(3)

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

Example

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

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

\[ \lambda_i \tilde{x} = \pi(\theta, X) + \lambda_i \tilde{\xi}_i \]

From eq. 3 express \( \lambda_i \) and substitute into 1,2:

\[ \pi^{(3)}(\theta, X)x = \pi^{(1,2)}(\theta, X). \]  \hspace{1cm} (4)

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

\[ E_p = \sum_i \| \pi^{(3)}(\theta, X)x - \pi^{(1,2)}(\theta, X) \|^2 \to \min. \]  \hspace{1cm} (5)
OPnP vs EPnP

\[ E_p = \sum_i \|\pi^{(3)}(\theta, X)x - \pi^{(1,2)}(\theta, X)\|^2 \to \min. \]

\[ \nabla_{\theta} E_p = 0. \]

OPnP - \( \pi(\theta, x) \) - polynomial, \( \text{dim}(\theta) = 4 \), solve polynomial equations

EPnP - \( \pi(\theta, x) \) - linear, \( \text{dim}(\theta) = 12 \), but there are quadratic constraints on the components of \( \theta \). Relinearization.
Comparison of OPnP and EPnP

![Graph showing Mean Rotation Error and Average time for different numbers of points using DLT, OPnP, and EPnP_GN methods.]

- **Mean Rotation Error**
  - DLT: Green triangles
  - OPnP: Blue diamonds
  - EPnP_GN: Magenta circles

- **Average time**
  - DLT: Green triangles
  - OPnP: Blue diamonds
  - EPnP_GN: Magenta circles
Generalization to $PnPL$

Line equation from the detected segment endpoints:

$$\hat{l}^i = \hat{p}_d^i \times \hat{q}_d^i, \quad l^i = \frac{\hat{l}^i}{|\hat{l}^i|} \in \mathbb{R}^3. \quad (6)$$

Algebraic point-to-line distance:

$$E_{pl}(\theta, P^i, l^i) = (l^i)^\top \pi(\theta, P^i), \quad (7)$$

Algebraic segment-to-line distance:

$$E_1(\theta, P^i, Q^i, l^i) = E_{pl}^2(\theta, P^i, l^i) + E_{pl}^2(\theta, Q^i, l^i). \quad (8)$$
Efficient PnP

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

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

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

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

$a_{i,j}$ do not change under rotation and translation.
Efficient PnP (2)

\[
\pi_{\text{EPnP}}(\theta, X_i) = \sum_{j=1}^{4} a_{i,j} C_j
\]  

(9)

We get 2 equations w.r.t. \( \mu = (C_{c,1}^T, C_{c,2}^T, C_{c,3}^T, C_{c,4}^T) \) for a single 3D-2D match, no using all correspondences we form a linear system:

\[
M\mu = 0, \quad M \in \mathbb{R}^{2n \times 12}
\]  

(10)

Without noise \( \xi_i \) in point detections, \( M \) 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 \( v_1, \ldots, v_N \) \( M \), corresponding to \( N = 1, 2, 3, 4 \) smallest singular values of \( M \).
Efficient PnP (3)

So, $\mu$ can be represented as

$$\mu = \sum_{i=1}^{N} \beta_i v_i \quad (11)$$

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

$$\|C_i - C_j\|^2 = r_{ij}^2, \quad i, j = 1, \ldots, 4, i \neq j. \quad (12)$$

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

$$\Gamma \gamma = w. \quad (13)$$

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

$$(\beta_{i1} \beta_{i2})(\beta_{i3} \beta_{i4}) = (\beta_{i1} \beta_{i3})(\beta_{i2} \beta_{i4}).$$
Constraints are:

\[ \lambda_i \tilde{x} = RX_i + t. \]  

(14)

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

\[ \hat{\lambda}_i \tilde{x} = \hat{R}X_i + \hat{t}. \]

(15)

Summing up equation triplets, get

\[ \hat{t}^{(3)} = n(1 - \hat{r}_3^T \bar{X}), \quad \bar{X} = \frac{1}{n} \sum X_i. \]
Algorithm OPnP (2)

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

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

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

$$
E_{\text{points}}(\hat{\mathbf{R}}(\mathbf{q}), \hat{\mathbf{t}}) = \| \mathbf{G}_p \hat{\mathbf{R}}(\mathbf{q}) + \mathbf{H}_p \hat{\mathbf{t}}^{(1,2)} + \mathbf{k}_p \|^2 \rightarrow \min, \quad (16)
$$

for known $\mathbf{G}_p, \mathbf{H}_p$ and $\mathbf{k}_p$.

$$
\nabla_{\mathbf{q}} E_{\text{points}} = 0, \\
\nabla_{\hat{\mathbf{t}}} E_{\text{points}} = 0.
$$
Algorithm OPnP (3)

From $\nabla \hat{t} E_{\text{points}} = 0$:

$$H_p^\top (G_p \hat{r} + H_p \hat{t}^{(1,2)} + k_p) = 0 \quad \implies \quad \hat{t}^{(1,2)} = P \hat{r} + u,$$  \hspace{1cm} (17)

$$P = -(H_p^\top H_p)^{-1}(H_p^\top G_p), \quad u = -(H_p^\top H_p)^{-1}H_p^\top k_p.$$  \hspace{1cm} (18)

From $\nabla q E_{\text{points}} = 0$, for the derivative w.r.t. first quaternion component $q = (a, \ldots)$:

$$\frac{\partial \hat{r}}{\partial a} G_p^\top (G_p \hat{r} + H_p \hat{t}^{(1,2)} + k_p) = 0.$$  \hspace{1cm} (19)

Remains to solve a system of 4 polynomial equations (19) of deg 3.
Camera pose from points and lines. Accuracy w.r.t. feature number.
### PnPL using NYU2 dataset

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(21.3, 100.0)</td>
<td>(9.3, 30.5)/(9.5, 28.2)</td>
<td>(61.5, 605.4)</td>
<td>(0.4, 6.0)/(0.3, 5.7)</td>
</tr>
<tr>
<td>(1.3, 33.8)</td>
<td>(0.6, 9.1)/(0.2, 2.6)</td>
<td>(11.5, 100.0)</td>
<td>(2.0, 71.8)/(1.3, 30.1)</td>
</tr>
</tbody>
</table>
Time for big $n$, PnPL problem

![Graph showing runtime vs. $n_p + n_l$ for different methods: Mirzaei, RPnL, Pluecker, EPnP_GN, OPnP, DLT, EPnPL, OPnPL.](image-url)
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


• E. Kanaeva, L. Gurevich and A. Vakhitov. Camera Pose and Focal length Estimation Using Regularized Distance Constraints, BMVC 2015
Datasets


Conclusion

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