Globally Optimal DLS Method for PnP Problem with Cayley parameterization – Appendix –

Gaku Nakano
g-nakano@cq.jp.nec.com

Information and Media Processing Labs
NEC Corporation
Kawasaki, Japan

A Building Matrix M in Eq.(3)

This derivation is essentially the same as given by [2], although the appearance is different. The cost function of Equation (2) can be rewritten as

$$\begin{aligned}\sum_{i=1}^{n}\left\|[\mathbf{m}_i]_\times(\mathbf{R}\mathbf{p}_i+\mathbf{t})\right\|^2 &= \sum_{i=1}^{n}\|\mathbf{X}_i\mathbf{Y}_i\mathbf{r}+\mathbf{X}_i\mathbf{t}\|^2 \\ &= \|\mathbf{A}\mathbf{r}+\mathbf{B}\mathbf{t}\|^2,\end{aligned} \tag{A-1}$$

where

$$\mathbf{X}_i=\begin{bmatrix}0 & -1 & v_i\\ 1 & 0 & -u_i\end{bmatrix}, \mathbf{Y}_i=\begin{bmatrix}\mathbf{p}_i^T & & \\ & \mathbf{p}_i^T & \\ & & \mathbf{p}_i^T\end{bmatrix}, \mathbf{A}=\begin{bmatrix}\mathbf{X}_1\mathbf{Y}_1\\ \vdots\\ \mathbf{X}_n\mathbf{Y}_n\end{bmatrix}, \mathbf{B}=\begin{bmatrix}\mathbf{X}_1\\ \vdots\\ \mathbf{X}_n\end{bmatrix}. \tag{A-2}$$

Note that $\mathbf{X}_i$ is a submatrix of $[\mathbf{m}_i]_\times$ extracting the first and the second rows since the rank of $[\mathbf{m}_i]_\times$ is two. We can see that Equation (A-1) is a linear least squares problem of $\mathbf{t}$. Hence, the optimal translation is given by

$$\mathbf{t}=-(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{A}\mathbf{r}. \tag{A-3}$$

Substituting this into Equation (A-1), we have

$$\begin{aligned}\|\mathbf{A}\mathbf{r}+\mathbf{B}\mathbf{t}\|^2 &= \left\|\left(\mathbf{I}-\mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\right)\mathbf{A}\mathbf{r}\right\|^2\\ &= \mathbf{r}^T\mathbf{A}^T\left(\mathbf{I}-\mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\right)\mathbf{A}\mathbf{r}\\ &= \mathbf{r}^T\left(\mathbf{A}^T\mathbf{A}-\mathbf{A}^T\mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{A}\right)\mathbf{r}.\end{aligned} \tag{A-4}$$

As a result, $\mathbf{M}$ is expressed by

$$\mathbf{M}=\mathbf{A}^T\mathbf{A}-\mathbf{A}^T\mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{A}. \tag{A-5}$$

Here, it is possible to compute $\mathbf{A}^T\mathbf{A}$, $\mathbf{A}^T\mathbf{B}$, and $\mathbf{B}^T\mathbf{B}$ explicitly without matrix multiplication:

$$\mathbf{A}^T\mathbf{A} = \begin{bmatrix} \mathbf{Z} & & \\ & \mathbf{Z} & \\ & & \mathbf{Z} \end{bmatrix}, \mathbf{B}^T\mathbf{B} = \begin{bmatrix} n & 0 & -\sum u_i \\ 0 & n & -\sum v_i \\ -\sum u_i & -\sum v_i & \sum r_i \end{bmatrix},$$

$$\mathbf{B}^T\mathbf{A} = \begin{bmatrix} \sum \mathbf{p}_i^T & & -\sum u_i \mathbf{p}_i^T \\ & \sum \mathbf{p}_i^T & -\sum v_i \mathbf{p}_i^T \\ -\sum u_i \mathbf{p}_i^T & -\sum v_i \mathbf{p}_i^T & \sum r_i \mathbf{p}_i^T \end{bmatrix}, \tag{A-6}$$

where

$$\mathbf{Z} = \begin{bmatrix} \sum x_i^2 & \sum x_i y_i & \sum x_i z_i \\ \sum x_i y_i & \sum y_i^2 & \sum y_i z_i \\ \sum x_i z_i & \sum y_i z_i & \sum z_i^2 \end{bmatrix}, r_i = u_i^2 + v_i^2. \tag{A-7}$$

B MATLAB Code for Kukelova's automatic generator [1]

Rotation Matrix

```
% register the generator
setpaths;

% coefficient matrix
M = gbs_Matrix('M%d%d', 9, 9); % 9x9
M = M - tril(M) + triu(M).';    % symmetrization

% rotation matrix
syms a b c d e f g h k
R = [a b c; d e f; g h k];
r = reshape(R.', 9, 1);  % vector representation of R

% build polynomial equations
matMr = reshape(M * r, 3, 3).';     % Eq.(8)
P      = R.' * matMr - matMr.' * R; % Eq.(11)
Q      = matMr * R.' - R * matMr.'; % Eq.(12)
g1     = R.'*R - eye(3);            % Eq.(6)
g2     = R*R.' - eye(3);
g3     = R(:,1) - cross(R(:,2),R(:,3)); % equal to det(R)=1 under Eq.(6)
g4     = R(:,2) - cross(R(:,3),R(:,1));
g5     = R(:,3) - cross(R(:,1),R(:,2));

eqs    = [g1(:,1); g1(2:3,2); g1(3,3);
          g2(:,1); g2(2:3,2); g2(3,3);
          g3; g4; g5;
          P(1,2); P(1,3); P(2,3); Q(1,2); Q(1,3); Q(2,3)]; % Eq.(13)

% collect known & unknown variables
unknown = {'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'k'};
vars = M(:).';
known = {};
for var = vars, known = [known {char(var)}]; end

% gbsolver
[res export] = gbs_CreateCode('solver_optDLS_rotmat', eqs, known, unknown);
```

Quaternion

```
% register the generator
setpaths;

% coefficient matrix
M = gbs_Matrix('M%d%d', 9, 9); % 9x9
M = M - tril(M) + triu(M).';   % symmetrization

% unit quaternion
syms a b c d
R = quat2mat([a b c d]); % Eq.(14)
r = reshape(R.', 9, 1);  % vector representation of R

% build polynomial equations
matMr = reshape(M * r, 3, 3).';     % Eq.(8)
P     = R.' * matMr - matMr.' * R; % Eq.(11)
Q     = matMr * R.' - R * matMr.'; % Eq.(12)
g1    = a^2 + b^2 + c^2 + d^2 - 1; % Eq.(15)
eqs   = [g1; P(1,2); P(1,3); P(2,3); Q(1,2); Q(1,3); Q(2,3)]; % Eqs.(13)

% collect known & unknown variables
unknown = {'a', 'b', 'c', 'd'};
vars = M(:).';
known = {};
for var = vars, known = [known {char(var)}]; end

% gbsolver
[res export] = gbs_CreateCode('solver_optDLS_quat', eqs, known, unknown);
```

Cayley parameterization

```
% register the generator
setpaths;

% coefficient matrix
M = gbs_Matrix('M%d%d', 9, 9); % 9x9
M = M - tril(M) + triu(M).';   % symmetrization

% Cayley parameterization
a = 1;                   % constant
syms b c d               % unknown variables
R = quat2mat([a b c d]); % Eq.(16) without 1/s
r = reshape(R.', 9, 1);  % vector representation of R

% build polynomial equations
matMr = reshape(M * r, 3, 3).';                                 % Eq.(8)
P     = R.' * matMr - matMr.' * R;                              % Eq.(11)
Q     = matMr * R.' - R * matMr.';                              % Eq.(12)
eqs   = [P(1,2); P(1,3); P(2,3); Q(1,2); Q(1,3); Q(2,3)];   % Eq.(13)

% collect known & unknown variables
unknown = {'b', 'c', 'd'};
vars    = M(:).';
known   = {};
for var = vars, known = [known {char(var)}]; end

% call code generator
[res export] = gbs_CreateCode('solver_optDLS_cayley', eqs, known, unknown);
```

References

[1] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Automatic generator of minimal problem solvers. In *Computer Vision–ECCV 2008*, pages 302–315. Springer, 2008.

[2] Yinqiang Zheng, Shigeki Sugimoto, and Masatoshi Okutomi. Aspnp: An accurate and scalable solution to the perspective-n-point problem. *IEICE TRANSACTIONS on Information and Systems*, 96(7):1525–1535, 2013.