A better visual odometry with bundle adjustment and uncertainty modeling

Ilan Shimshoni, Ehud Rivlin, Alex Kreimer

May 1, 2015

Abstract

This draft summarizes the work I did so far: implemented bundle adjustment which allows to take uncertainties into account while estimating motion parameters and also implemented a simulated environment which allows to easily inject various types of noise and verify the behaviour of the algorithms.

1 Notation

This section makes definitions that are used throughout the document.

Let the camera j be parameterized by a vector a_j, $j = 1, \ldots, n$, and 3d the point i by vector b_i, $i = 1, \ldots, m$. We assume calibrated cameras and thus parations that are used throughout the document.

Let the camera j be parameterize only the extrinsic camera parameters. The rotation is represented by three Euler angles and the translation is taken "as is". *Parameter* vector $\theta \in R^M$ is defined as: $\theta = [a_1^T, \ldots, a_n^T, b_1^T, \ldots, b_m^T]^T$.

$x_{ij} \in \mathcal{R}^k$ denotes the image coordinates 3d point i as seen in image j.

Measurement vector is defined by image observations: $X = [x_{11}^T, \ldots, x_{1m}^T, x_{21}^T, \ldots, x_{2m}^T, \ldots, x_{n1}^T, \ldots, x_{nm}^T]$, $X \in \mathcal{R}^N$ and it is accompanied by its covariance data $\Sigma_X = diag(\Sigma_{x_{11}}, \ldots, \Sigma_{x_{nm}})$, $\Sigma_{ij} \in \mathcal{R}^{k \times k}$. In case of missing measurement it just disappears from the measurement vector, so the indices may have "holes" in them.

Prediction vector is a function of the parameter vector θ: $\hat{X} = [\hat{x}_{11}^T, \ldots, \hat{x}_{1m}^T, \hat{x}_{21}^T, \ldots, \hat{x}_{2m}^T, \ldots, \hat{x}_{n1}^T, \ldots, \hat{x}_{nm}^T]$, s.t. $\hat{x}_{ij} = f_{ij}(P)$.

Bundle adjustment is problem of finding a set of parameters that make observations in different views as "consistent" as possible. Different formulations of consistency are possible with maximul likelihood one dominating the literature.

2 Problem Formulations

2.1 Maximum Likelihood

Maximum likelihood estimator seeks to maximize the likelihood of the data given the parameters:

$$\theta_{ML} = \underset{\theta}{\operatorname{argmax}}\ p(X|\theta)$$

We assume that the measurements adhere to the following statistical model:

$$x_{ij} = \hat{x}_{ij} + \epsilon_{ij}$$

where $\hat{x}_{ij}$ is the model prediction, ϵ_{ij} is a Gaussian noise with zero mean and a known covariance Σ_{ij}. Thus x_{ij} is a k-vector Gaussian random variable with mean $\hat{x}_{ij}$ and covariance matrix Σ_{ij}. Hence the probability density of x_{ij} given θ is:

$$p(x_{ij}|\theta) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_{ij}|}} \exp^{-(x_{ij}-\hat{x}_{ij}, \Sigma_{ij}^{-1}(x_{ij}-\hat{x}_{ij}))/2}$$

Thus ML estimator is:

$$\theta_{ML} = \underset{\theta}{\operatorname{argmax}}\ p(X|\theta) = \underset{\theta}{\operatorname{argmax}} \prod_{i,j} p(x_{ij}|\theta) = \underset{\theta}{\operatorname{argmin}}\ \sum_{ij} \|x_{ij} - \hat{x}_{ij}\|_{\Sigma_{ij}} \tag{1}$$

This is typical formulation of BA procedure found in the literature. It accounts for the uncertainty in the measurements while the prior distribution of the parameters is assumed to be uniform.

Note: if the covariances are all of the form $\Sigma_{ij} = aI$ then

$$\underset{\theta}{\operatorname{argmin}}\ \sum_{ij} \|x_{ij} - \hat{x}_{ij}\|_{\Sigma_{ij}} = \underset{\theta}{\operatorname{argmin}}\ \sum_{ij} \|x_{ij} - \hat{x}_{ij}\|_2^2$$

thus minimization of unweighted reprojection errors implicitly assumes equal measurement covariance.

2.2 Maximum a posteriori probability

Now we assume a joint distribution $p(X, \theta)$ and seek to maximize the posterior distribution on the parameters

$$\theta_{MAP} = \underset{\theta}{\operatorname{argmax}}\ p(\theta|X) = \underset{\theta}{\operatorname{argmax}}\ p(X|\theta)p(\theta)$$

In case of a Gaussian zero mean and a known covariance prior over the parameters

$$\theta_{MAP} = \underset{\theta}{\operatorname{argmin}}\ \sum_{ij} \|x_{ij} - \hat{x}_{ij}\|_{\Sigma_{ij}} + \sum_{i} \left\|b_i - b_i^0\right\|_{\Sigma_{b_i}} + \sum_{j} \left\|a_j - a_j^0\right\|_{\Sigma_{a_j}}$$

Note Motion priors may be modelled as a distribution over the camera extrinsic parameters while structure uncertainty may be modeled as a prior distribution over the structure parameters.

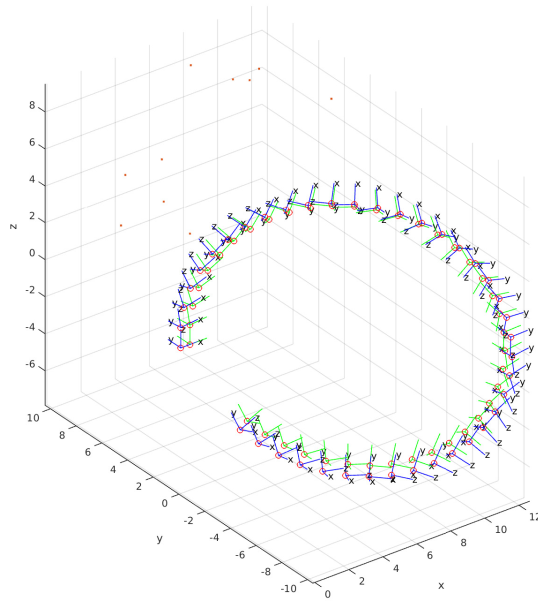


Figure 1: A possible simulation scene that depicts stereo camera motion and scene structure.

2.3 A choice of parameterization

3d points are represented by 3 vectors $b_j = X_j = [X_j, Y_j, Z_j]^T$. Camera motion is parameterized by a 6 vector $a_i = [r_x, r_y, r_z, t_x, t_y, t_z]$. Rotations are parameterized using Euler angles $R = R_z(r_z)R_y(r_y)R_x(r_x)$. Translation is represented by a 3 vector $dt = [d_x, d_y, d_z]^T$. Left (right) camera i projection of 3d point X_j:

$$\hat{x}_{ij}^{(l)} = \pi^{(l)}(a_i, b_j) = K[R^{(l)}(a_i)\ T^{(l)}(a_i)]b_j$$
$$\hat{x}_{ij}^{(r)} = \pi^{(r)}(a_i, b_j) = K[R^{(r)}(a_i)\ T^{(r)}(a_i)]b_j$$

3 Implementation

We implemented a simulator that allows to move the camera in a predefined (known) way. As they move the cameras observe a set of known 3d points. Parameter estimation procedure is used to estimate the motion and/or structure which may be compared to the true parameter value.

Figure 1 depicts a possible scene with cameras and structure. Figure 2 shows a typical image seen by a stereo rig in the simulator.

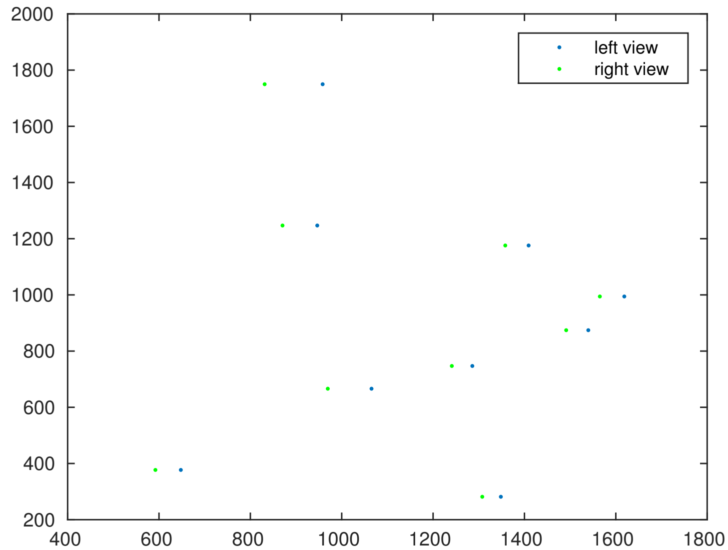


Figure 2: Scene projection as seen by one of the stereo pairs

Camera motion and structure parameters are estimated in 3 steps. Note that this procedure is not intended to be optimal but rather to get a feeling of what various parameter estimation schemes are capable of on the synthetic data:

- Camera motion between each subsequent stereo frames is estimated. We triangulate both sets of observations to obtain the 3d in both frames. Then we solve $R = \underset{\Omega}{\operatorname{argmin}} \|A\Omega - B\|_F$ s.t. $\Omega^T\Omega = I$. If $M = A^TB$ and $M = U\Sigma V^*$ is its SVD decomposition, then $R = UV^*$
- Single step re-projection minimization $\theta = \underset{\theta}{\operatorname{argmin}} \|X - \hat{X}(\theta)\|_2^2$. The optimization is solved using LM algorithm, the result of the previous section is used as an initial guess
- We refine the parameters from previous step, by solving problem 1 (e.g., bundle adjustment over all frames and structure). The parameters that are obtained as an output of this step are our best estimate. Bundle adjustment is performed twice, once with all unit co-variances another time with real co-variance information.

We perform the above parameter estimation procedure a number of times each time adding a noise to the pixel measurements. For each experiment we set a maximum value for standard deviation of noise, $\sigma_{max} \in \{1 \dots 4\}$. For each coordinate of each measurement we generate $\sigma \sim U[0, \sigma_{max}]$ and then perturb the measurement coordinate by $\epsilon \sim N(0, \sigma^2)$.

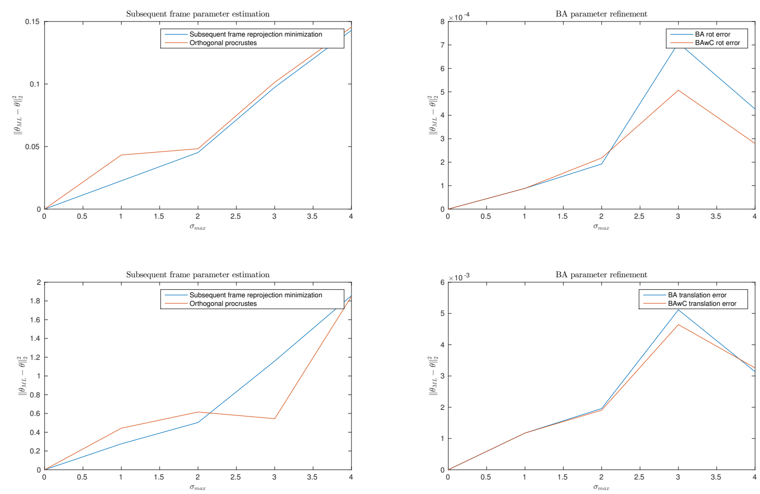


Figure 3: First column presents results for subsequent frame motion estimation, while second column shows results for BA performed over all frames seen up to this point. The results are averaged over 10 camera motions. Each camera observed 20 3d points. No outliers were introduced into the experiment. BAwC stands for bundle adjustment with co-variance, while BA stands for bundle adjustment with all unit co-variances

Figure 3 presents results.

4 Essential bundle adjustment

4.1 Estimation of motion parameters

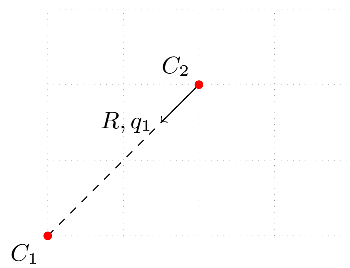
R describes world frame in terms of camera frame. q_1 is a direction from C_2 to C_1 described in C_2.

Thus for every pair of correspondences x, x'; (C_1, C_2) holds

$$x'^T K^{-1} [q]_\times R K^{-1} x = 0$$

We define the following objective (includes multiple features):

$$F(q, R) = \sum_i^N [x_i'^T K^{-1} [q]_\times R K^{-1} x_i]^2 + [\lambda(q^T q - 1)]^2$$



4.2 Gauss Newton iteration

Let $r(q(t_x, t_y, t_z), R(r_x, r_y, r_z)) \in \mathbf{R}^{N+1}$ be defined as (we further omit the parameterization of the translation and the rotation):

$$r_i = x_i'^T K^{-1} [q]_\times R K^{-1} x_i$$
$$r_{N+1} = \lambda(q^T q - 1)$$

So,

$$F(q, R) = r^T r = \sum_i^{N+1} r_i^2$$

$$(\nabla F)_j = \sum_i^{N+1} 2 r_i (\nabla r_i)_j \text{ and thus, } \nabla F = 2 J^T r \quad \text{where, } J = [\nabla r_1 \nabla r_2 \ldots \nabla r_{N+1}]^T$$

$$(\nabla^2 F)_{jk} = 2 \sum_i^{N+1} (\nabla r_i)_k (\nabla r_i)_j + r_i (\nabla^2 r_i)_{jk} \approx 2 \sum_i^{N+1} (\nabla r_i)_k (\nabla r_i)_j$$

$$\nabla^2 F \approx 2 J^T J$$

4.3 Dogleg algorithm

The algorithm sets a trust region radius Δ. Possible search direction is $p = p_{sd} + \lambda(p_N - p_{sd})$ where $\lambda \in [0,1]$ (a compromise between steepest descent and Newton directions):

$$\lambda^* = \underset{\lambda}{\operatorname{argmax}}\ \|p\| \text{ s.t. } \|p\| \leq \Delta$$

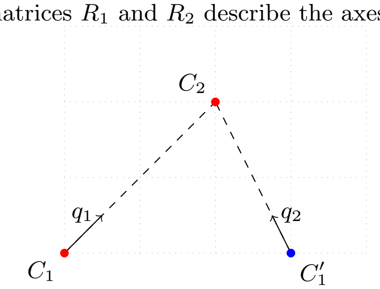
$$\begin{aligned}\|p\|^2 &= \|p_{sd} + \lambda(p_N - p_{sd})\|^2 \\ &= \|p_{sd}\|^2 + 2\lambda p_{sd}^T(p_N - p_{sd}) + \lambda^2\|p_N - p_{sd}\|^2 = \Delta^2\end{aligned}$$

This is a quadratic equation in λ and $\lambda^* = \frac{-b+-\sqrt{b^2-4ac}}{2a}$, $a = \|p_N - p_{sd}\|^2$, $b = 2p_{sd}^T(p_N - p_{sd})$, $c = \|p_{sd}\|^2 - \Delta^2$

4.4 Stereo

Let C_i and C_i' be the locations of the left/right camera at time i. The rig is moving rigidly and we are interested to recover this motion.

Epipolar constraint for C_1 and C_2 may be expressed $E_1 = [q_1]_\times R_1$ and for C_1' and C_2' as $E_2 = [q_2]_\times R_2$. Here q_1, q_2 are line of sight vectors that are described in coordinate frames C_1 and C_2 respectively. Rotation matrices R_1 and R_2 describe the axes of ? in terms of ?.



$$f_i(r_x, r_y, r_z, t_x, t_y, t_z) = x_i^T[t]_x R x_i'$$

where

$$R(r_x, r_y, r_z) = R_x(r_x)R_y(r_y)R_z(r_z) = \begin{bmatrix} c_y c_z & -c_y s_z & s_y \\ s_x s_y c_z + c_x s_z & -s_x s_y s_z + c_x c_z & -s_x c_y \\ -c_x s_y c_z + s_x s_z & c_x s_y s_z + s_x c_z & c_x c_y \end{bmatrix}$$

and $[t]_x = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$

$$\frac{\partial f}{\partial r_x} = x_i^T[t]_x \frac{\partial R}{\partial r_x} x_i' \quad = x_i^T \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ c_x s_y c_z - s_x s_z & -c_x s_y s_z - s_x c_z & -c_x c_y \\ s_x s_y c_z + c_x s_z & -s_x s_y s_z + c_x c_z & -s_x c_y \end{bmatrix} x_i'$$

$$\frac{\partial f}{\partial r_y} = x_i^T[t]_x \frac{\partial R}{\partial r_y} x_i' \quad = x_i^T \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \begin{bmatrix} -s_y c_z & s_y c_z & c_y \\ c_x c_y c_z & -c_x c_y s_z & c_x s_y \\ -s_x c_y c_z & s_x c_y s_z & -s_x s_y \end{bmatrix} x_i'$$

$$\frac{\partial f}{\partial r_z} = x_i^T [t]_x \frac{\partial R}{\partial r_z} x_i' \quad = x_i^T \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \begin{bmatrix} -c_y s_z & c_y s_z & 0 \\ -s_x s_y s_z + c_x c_z & -s_x s_y c_z - c_x s_z & 0 \\ c_x s_y s_z + c_x s_z & c_x s_y c_z - c_x s_z & 0 \end{bmatrix} x_i'$$