Non-uniqueness in C-arm calibration with bundle adjustment

Anastasia Konik, Laurent Desbat, Yannick Grondin

Non-uniqueness in C-arm calibration with bundle adjustment

Anastasia Konik[1], Laurent Desbat[1], and Yannick Grondin[2]

[1]TIMC-GMCAO, Univ. Grenoble Alpes, Grenoble, France
[2]SurgiQual Institute, Meylan, France

Abstract In this work, we consider the problem of C-arm geometric calibration with the bundle adjustment (BA) method. This method was initially developed and used in computer vision. It's based on markers, but with unknown geometry. We don't know the 3D positions of the markers from which we have to estimate the calibration parameters. Thus, the geometric calibration is only based on the positions of the projected markers on 2D projection images. Pollefeys et al. [1] have shown that such calibration from BA can be performed up to a similarity transformation. In our work, we present a numerical solution to the C-arm geometric calibration with the BA method. We show the non-uniqueness of the geometric calibration with the BA method for cone-beam tomography. Just like in computer vision, we can't find the solution better than up to a similarity transformation.

1 Introduction

A C-arm X-ray imaging system is designed as a C-shaped arm which connects a X-ray source and a X-ray detector. We consider isocentric C-arms rotating around their isocenter. Usually projection images of a patient placed at the isocenter are collected. In the figure 1, we present some geometric parameters of a C-arm in a schematic view. This geometric model is classical in the cone-beam (CB) geometry.

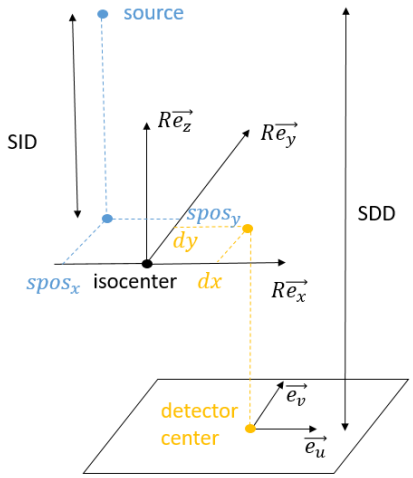

Figure 1: Some geometric parameters of a C-arm in a schematic view.

In the section 2 we recall the geometric model of a C-arm system. This projection geometric model maps 3D patient points to 2D detector points. The model contains geometric parameters. In general, these parameters need to be calibrated for each projection. The identification of these parameters is necessary for an accurate 3D reconstruction [2]. For most C-arms, in order to take into account mechanical vibrations over time, it is necessary to periodically perform the C-arm geometric calibration. In this work, we want to discuss bundle adjustment (BA) geometric calibration.

This C-arm geometric calibration process is similar to the camera geometric calibration in computer vision [3]. By analogy with computer vision, we can divide all image-based calibration methods for C-arms into two groups: calibration with markers and without markers as in [4]. For the first group, the calibration problem is solved with specific scans of a calibration object, usually based on few opaque markers. Either the 3D coordinates of marker centers are known in the world coordinate system or not. In the second case, both these 3D coordinates of marker centers and the geometric calibration parameters need to be identified. In this work, we consider this bundle adjustment problem and the method to solve this problem described in [5]. Just as for the calibration without markers, BA only uses the projection data and thus belongs to self-calibration methods.

2 Geometric calibration of a C-arm

As we know from computer vision [3], a camera can be modelled by a projection matrix P mapping Q, a 3D point in the world coordinate system, to q, its corresponding projection onto the image plane. We usually have the decomposition of the projection matrix P:

$$P \sim \begin{pmatrix} K & 0 \end{pmatrix} \begin{pmatrix} R & -Rt \\ 0^T & 1 \end{pmatrix} = KR \begin{pmatrix} I & -t \end{pmatrix}. \qquad (1)$$

To be more precise, we can connect homogeneous coordinates of $q = (u, v, 1)^T$ in the pixel coordinate system and homogeneous coordinates of $Q = (X^w, Y^w, Z^w, 1)^T$ in the world coordinate system with special matrices K, R and t:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim KR \begin{pmatrix} I & -t \end{pmatrix} \begin{pmatrix} X^w \\ Y^w \\ Z^w \\ 1 \end{pmatrix}, K = \begin{pmatrix} f_x & s & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix}. \qquad (2)$$

The matrix K consists of intrinsic calibration parameters. For the classical pinhole camera model the skew $s = 0$, $f_x = f_y = f$ is the focal length, u_0 and v_0 are the coordinates in the camera image of the orthogonal projection of the optical center. The rotation matrix R and the translation vector t are extrinsic calibration parameters; they describe the orientation and the position of the camera in the world coordinate system. Let us use here a IEC 61217 standard in order to describe the C-arm used in our simulation in the same way as cameras. In order to do this, 9 parameters are used, see the table 1.

Usually, we don't know exactly these parameters, so we need to calibrate them. We present some of these parameters in the figure 1.

Parameter	Value	Noise bounds
SDD (mm): source-detector distance	1000	± 3.5
SID (mm): source-isocenter distance	700	± 6.9
$spos_x$ (mm): x-coordinate of the position of the source in the rotated frame	0	± 6.9
$spos_y$ (mm): y-coordinate of the position of the source in the rotated frame	0	± 6.9
dx (mm): x-coordinate of the center of the detector in the rotated frame	0	± 13.9
dy (mm): y-coordinate of the center of the detector in the rotated frame	0	± 13.9
θ_x (degrees): orientation of the rotated frame relative to the world frame along the x axis	0	± 1.4
θ_z (degrees): orientation of the rotated frame relative to the world frame along the z axis	0	± 1.4
θ_y (degrees): angle of scan	$i\delta$	± 0.7

Table 1: C-arm calibration parameters: initial values and noise bounds used to simulate mechanical vibrations (we show in the third column bounds for the uniform distribution, we added a small uniform noise to initial values except the case of the SDD parameter for which we have completely different initial value, its realistic values for all projections are around 1300 mm). Here $i \in \mathbb{N}$ is the projection index, δ is the angular step.

We can describe the C-arm with approximately the same projection matrix as the basic pinhole camera. As in computer vision, we can build the intrinsic geometric calibration matrix K with zero skew ($s = K_{12} = 0$). The rotation matrix R is here the rotation around the isocenter defined by the position of the rotated frame. The translation $-t$ is the source position in the rotated frame:

$$P = K\begin{pmatrix} R & t \end{pmatrix}, t = \begin{pmatrix} -spos_x \\ -spos_y \\ -SID \end{pmatrix}, K = \begin{pmatrix} -f & 0 & u_0 \\ 0 & -f & v_0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{3}$$

$$f = \frac{SDD}{dim_{\text{pixel}}}, u_0 = \frac{spos_x - dx}{dim_{\text{pixel}}}, v_0 = \frac{spos_y - dy}{dim_{\text{pixel}}}, \tag{4}$$

$$R = \begin{pmatrix} c_z & -s_z & 0 \\ s_z & c_z & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_x & -s_x \\ 0 & s_x & c_x \end{pmatrix} \begin{pmatrix} c_y & 0 & s_y \\ 0 & 1 & 0 \\ -s_y & 0 & c_y \end{pmatrix}, \tag{5}$$

$$c_\alpha = \cos(-\theta_\alpha), s_\alpha = \sin(-\theta_\alpha), \alpha \in \{x, y, z\}. \tag{6}$$

So, during the calibration we want to identify the elements of the projection matrix P or the calibration parameters which define the projection matrix.

3 Bundle adjustment

In tomographic situations, we assume that we have collected many X-ray projections. For each X-ray projection i, we want to estimate both projection matrices $\hat{P}^i$, $i = 1, \ldots, N_{\text{projections}}$ and unknown 3D marker points $\hat{Q}_j$, $j = 1, \ldots, N_{\text{markers}}$ from known image points q^i_j. We minimize the mean of Euclidean distances between the projected points and the measured image points for all X-ray images, i.e.

$$\min_x D(x) \stackrel{\text{def}}{=} \min_{P^i, Q_j} \frac{1}{N_{\text{markers}} N_{\text{projections}}} \sum_{i,j} d(P^i Q_j, q^i_j)^2, \tag{7}$$

where $d(q_1, q_2)$ is a geometric image distance between homogeneous points q_1 and q_2, D is the cost function, x is a vector containing the parameters of P^i, $i = 1, \ldots, N_{\text{projections}}$, Q_j, $j = 1, \ldots, N_{\text{markers}}$, so in our case x contains $9N_{\text{projections}} + 3N_{\text{markers}}$ parameters to be identified. This is the general formulation of the BA problem.

In [5] authors described basic local optimization methods for differentiable functions to solve the BA problem. Let us try to minimize the cost function $D(x)$ over x with the initial estimate x_0. We want to find a displacement δx which locally minimizes $D(x)$. This cost function can be replaced by an approximate local model. The quadratic local model is based on the Taylor expansion:

$$D(x + \delta x) \approx D(x) + g^T \delta x + \frac{1}{2} \delta x^T H \delta x, \tag{8}$$

where g is a gradient vector of D at x and H is a Hessian matrix at x. In [5] authors proposed methods to optimize such as the damped Newton methods which solve the following regularized system:

$$(H + \lambda W)\delta x = -g, \tag{9}$$

where λ is a weighting factor and W is a positive definite weight matrix. This is the basis for trust region methods, for example, the popular Levenberg-Marquardt method. We use this method in our numerical experiments.

4 Numerical experiments: calibration with BA

In order to solve numerically the optimization problem (7) we used the C++ package Ceres [6]. We simulated data for our numerical experiments. Firstly, we started with 20 markers, see the 3D plot in the figure 2. We call these points true values $Q_{j,\text{true}}$. Then we fixed $dim_{\text{pixel}} = 0.5$ mm, $N_{\text{projections}} = 181$ with the angular step as 2 degrees. We computed from the table 1 the initial values for the 9 calibration parameters f^i,

u_0^i, v_0^i, θ_x^i, θ_y^i, θ_z^i, t_x^i, t_y^i, t_z^i for each projection i. These values were used as initial estimations for our optimization algorithm. In order to simulate realistic values for calibration parameters and P^i_{real} which correspond to clinical situations with mechanical vibrations of the C-arm, we added a uniform noise to each calibration parameter (see the table 1 for details).

We simulated q^i_j by the multiplication of P^i_{real} by $Q_{j,\text{true}}$. We usually don't know exactly q^i_j, because we detect these points on X-ray images using specific algorithms. In order to simulate this process, we also added a uniform noise with bounds ± 0.3 pix to the image points $P^i_{\text{real}} Q_{j,\text{true}}$. Thus, as inputs for the optimization algorithm we had noisy image points q^i_j.

In order to start the minimization algorithm, we also need initial estimations for 3D points Q_j (we described before just the initial calibration parameters). We did our first guess with the basic triangulation algorithm of Python's OpenCV from two known initial projection matrices and known projections for 0 and 90 degrees. A full description of the basic triangulation algorithm could be found in [3]. After this simulation and initialisation, the optimization algorithm was launched. We started from the initial cost 1337. With the Levenberg-Marquardt method we achieved the final cost 0.003.

We show in the figure 2 estimated 3D points obtained by this algorithm. We calculated the reprojection error as $\frac{1}{N_{\text{markers}} N_{\text{projections}}} \sum_{i,j} ||\hat{P}^i Q_{j,\text{true}} - q^i_j||_2$. It is equal to 34.41 pix. The maximum errors for the estimated calibration parameters through all projections are: 9.78 pix for f, 10.28 pix for u_0, 9.65 pix for v_0 (1 pix is 0.5 mm), 102.09 mm for t. In order to compare rotation matrices, we calculated the error rotation matrix for each projection i as $R^i_{\text{err}} = (R^i_{\text{real}})^{-1} \hat{R}^i$. According to the Euler rotation theorem, each rotation R^i_{err} in three dimensions is defined by its axis and its angle ψ_i. We found the absolute value of ψ_i from the error rotation matrix R^i_{err} for each projection i with $|\psi_i| = \arccos \frac{\text{tr}(R^i_{\text{err}})-1}{2}$. The maximum error for angles $|\psi_i|$ through all projections was 0.79 degrees. We observed the high reprojection error and high errors in calibration parameters. The same for 3D points: true and estimated 3D points are far to be exactly the same. Thus, we found the solution of the calibration problem and it differs from the true solution. But what is the reason?

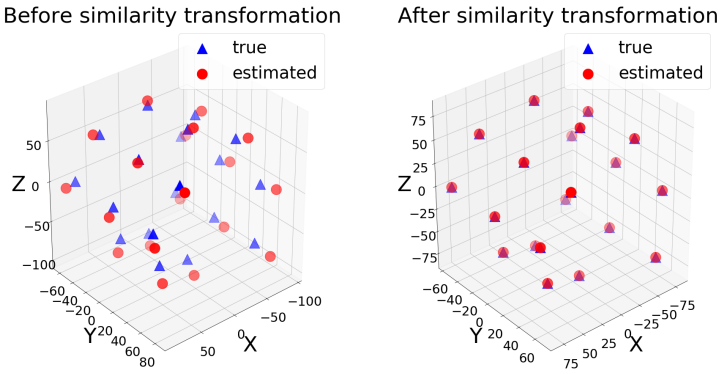

Figure 2: True and estimated 3D points for Levenberg-Marquardt method.

5 Theoretical explanation of non-uniqueness

5.1 Computer vision BA limits

Let us start with different classes of transformations of 3D space. Let us remind that in computer vision we usually use homogeneous coordinates of the point, so for the point in 3D we have four coordinates. We identify each transformation by its matrix form. Moreover, these transformations form a hierarchy. So, we start with the general one.

Definition 5.1. A projective transformation is a transformation of the form $\begin{pmatrix} A & t \\ h^T & v \end{pmatrix}$, where A is an invertible 3×3 matrix, h is a general 3-vector.

Definition 5.2. An affine transformation is a transformation of the form $\begin{pmatrix} A & t \\ 0^T & 1 \end{pmatrix}$, where A is an invertible 3×3 matrix.

Definition 5.3. A similarity transformation is a transformation of the form $\begin{pmatrix} \sigma R & t \\ 0^T & 1 \end{pmatrix}$, where R is a 3×3 rotation matrix and $\sigma \neq 0$.

Definition 5.4. An Euclidean transformation is a transformation of the form $\begin{pmatrix} R & t \\ 0^T & 1 \end{pmatrix}$, where R is a 3×3 rotation matrix.

From the literature [3] we know that we have a solution of the BA problem up to a projective transformation. We can take an invertible matrix H and have as a solution also $\hat{P}^i H^{-1}$, $H\hat{Q}_j$. Moreover, we found in [1] that for special calibration matrices with zero skews ($s = 0$) the solution could be found up to a similarity transformation. The following theorem from [1] is true:

Theorem 5.1. The class of transformations which preserves the absence of skew is the group of similarity transformations.

If we have projection matrices as solutions of our calibration problem, they differ by some projective transformation. With this theorem, if the sequence of views is general enough and if in decompositions of the projection matrices we have zero skews, this projective transformation should be a similarity transformation. We build projection matrices for the C-arm BA problem exactly such that in decompositions they have zero skews. In essentially all digital X-ray CB systems the skew is zero because the lines and columns of digital X-ray detectors are perpendicular. Unfortunately, the sequence of C-arm positions often couldn't be general enough, which complicates the application of the theorem.

5.2 Cone-beam geometric self-calibration limits

Any CB system can be also described by an integral model. We consider cone-beam data in the form

$$d(\vec{s}_\lambda, \vec{\zeta}) = Df(\vec{s}_\lambda, \vec{\zeta}) = \int_0^{+\infty} f(\vec{s}_\lambda + l\vec{\zeta})dl, \qquad (10)$$

where $\lambda \in \Lambda \subset \mathbb{R}$ is a trajectory parameter of the source, $\vec{s}_\lambda$ is the 3D position of the source at λ, $\vec{\zeta}$ is a unit vector in $\mathbb{R}^3$, the direction of the integration line.

Theorem 5.2. Let $f_{R,\vec{t}}(\vec{x}) \stackrel{\text{def}}{=} f(R\vec{x}+\vec{t})$ for any $\vec{x} \in \mathbb{R}^3$, for any rotation R and any translation vector $\vec{t} \in \mathbb{R}^3$, then

$$Df_{R,\vec{t}}(\vec{s}_\lambda, \vec{\zeta}) = Df(R\vec{s}_\lambda + \vec{t}, R\vec{\zeta}). \qquad (11)$$

Proof. We have $Df_{R,\vec{t}}(\vec{s}_\lambda, \vec{\zeta}) = \int_0^{+\infty} f_{R,\vec{t}}(\vec{s}_\lambda + l\vec{\zeta})dl = \int_0^{+\infty} f(R\vec{s}_\lambda + lR\vec{\zeta} + \vec{t})dl = Df(R\vec{s}_\lambda + \vec{t}, R\vec{\zeta})$. □

Thus, the cone-beam data $Df_{R,\vec{t}}$ of $f_{R,\vec{t}}$ from the source $\vec{s}_\lambda$ in the direction $\vec{\zeta}$ is nothing but the cone-beam data Df of f from the source position $R\vec{s}_\lambda + \vec{t}$ toward the direction $R\vec{\zeta}$. Conversely, let $\vec{v}_\lambda = R\vec{s}_\lambda + \vec{t}$ or $\vec{s}_\lambda = R^T\left(\vec{v}_\lambda - \vec{t}\right)$, let $\vec{\eta} = R\vec{\zeta}$, thus $\vec{\zeta} = R^T\vec{\eta}$. Then for all $\vec{v}_\lambda$ and all unit $\vec{\eta}$ Eq. (11) is equivalent to

$$Df_{R,\vec{t}}\left(R^T\left(\vec{v}_\lambda - \vec{t}\right), R^T\vec{\eta}\right) = Df\left(\vec{v}_\lambda, \vec{\eta}\right). \qquad (12)$$

Thus, the projection data Df of f acquired from the source position $\vec{v}_\lambda$ toward the direction $\vec{\eta}$ is equal to the projection $Df_{R,\vec{t}}$ acquired from the source position $R^T\left(\vec{v}_\lambda - \vec{t}\right)$ toward the direction $R^T\vec{\eta}$ for any rotation R and translation vector $\vec{t}$. This shows that for the cone-beam geometry the geometric self-calibration problem can not be solved better than up to an Euclidean transformation. Moreover, we have

Theorem 5.3. Let $f_{\sigma R,\vec{t}}(\vec{x}) \stackrel{\text{def}}{=} f(\sigma R\vec{x}+\vec{t})$ for any $\vec{x} \in \mathbb{R}^3$, rotation R, translation $\vec{t} \in \mathbb{R}^3$ and scaling $\sigma > 0$, then

$$D\left(\sigma f_{\sigma R,\vec{t}}\right)\left(\vec{s}_\lambda, \vec{\zeta}\right) = Df\left(\sigma R\vec{s}_\lambda + \vec{t}, R\vec{\zeta}\right). \qquad (13)$$

Proof. We have $Df_{\sigma R,\vec{t}}(\vec{s}_\lambda, \vec{\zeta}) = \int_0^{+\infty} f_{\sigma R,\vec{t}}(\vec{s}_\lambda + l\vec{\zeta})dl = \int_0^{+\infty} f(\sigma R\vec{s}_\lambda + \sigma lR\vec{\zeta} + \vec{t})dl = \int_0^{+\infty} f(\sigma R\vec{s}_\lambda + nR\vec{\zeta} + \vec{t})d\frac{n}{\sigma} = \frac{1}{\sigma}Df(\sigma R\vec{s}_\lambda + \vec{t}, R\vec{\zeta})$. □

Thus, the cone-beam data $D(\sigma f_{\sigma R,\vec{t}})$ of $\sigma f_{\sigma R,\vec{t}}$ from the source $\vec{s}_\lambda$ in the direction $\vec{\zeta}$ is nothing but the cone-beam data Df of f from the source position $\sigma R\vec{s}_\lambda + \vec{t}$ toward the direction $R\vec{\zeta}$. Conversely, let $\vec{v}_\lambda = \sigma R\vec{s}_\lambda + \vec{t}$ or $\vec{s}_\lambda = \frac{1}{\sigma}R^T\left(\vec{v}_\lambda - \vec{t}\right)$ and let $\vec{\eta} = R\vec{\zeta}$, thus $\vec{\zeta} = R^T\vec{\eta}$, then for all $\vec{v}_\lambda$ and all unit vector $\vec{\eta}$ Eq. (13) is equivalent to

$$D\left(\sigma f_{\sigma R,\vec{t}}\right)\left(\frac{1}{\sigma}R^T\left(\vec{v}_\lambda - \vec{t}\right), R^T\vec{\eta}\right) = Df(\vec{v}_\lambda, \vec{\eta}). \qquad (14)$$

Thus, for the cone-beam geometry the geometric self-calibration problem can not be solved better than up to a similarity transformation.

6 Numerical experiments: similarity error identification

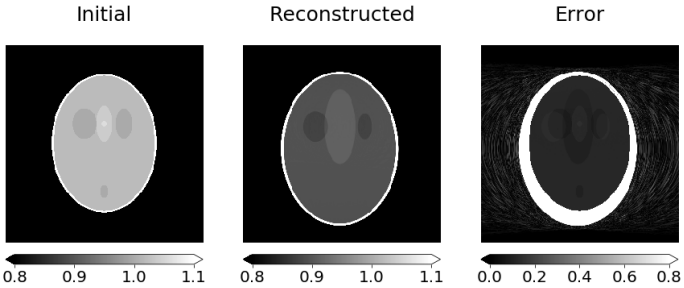

Figure 3: Slices $z = 6.5$ mm of the initial 3D Shepp–Logan phantom $f(\vec{x})$ (left), the reconstruction $g(\vec{x})$ from the estimated acquisition geometry (center) and $|f(\vec{x}) - g(\vec{x})|$ (right).

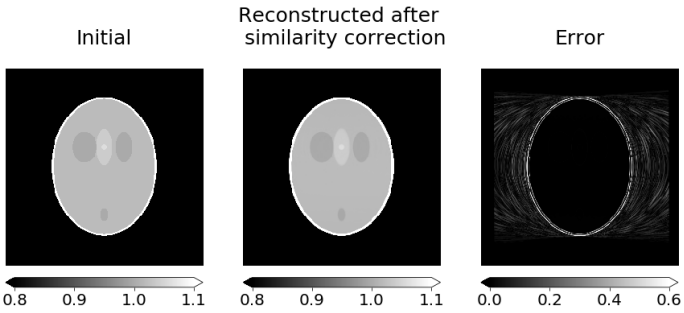

Figure 4: Slices $z = 6.5$ mm of the initial 3D Shepp–Logan phantom $f(\vec{x})$ (left), the reconstruction from the estimated acquisition geometry after the similarity correction $\frac{1}{\sigma}g\left(\frac{1}{\sigma}R^T\left(\vec{x} - \vec{t}\right)\right)$ (center) and $\left|f(\vec{x}) - \frac{1}{\sigma}g\left(\frac{1}{\sigma}R^T\left(\vec{x} - \vec{t}\right)\right)\right|$ (right).

The numerical experiments from the section 4 provided a scaling, a rotation and a translation, i.e. a similarity transformation, for computing the set of the true 3D marker coordinates from the set of the estimated 3D marker coordinates. Firstly, we computed the scaling factor. We computed the barycenters b_{true} and b_{est} of the true and the estimated 3D points. The mean of $\frac{||Q_{j,\text{true}} - b_{\text{true}}||_2}{||\hat{Q}_j - b_{\text{est}}||_2}$ is a simple (and sufficient) estimation of the scaling. It is equal to 0.88. Then, after the scaling correction, we numerically found the rotation and the translation with the algorithm described in [7]. We can then apply to the set of scaled estimated points the rotation approximately equal to the identity matrix and the translation $(0.96, -3.14, 5.18)^T$ in mm. We show the result of such transformation of the estimated 3D points to the true 3D points in the figure 2.

The estimated calibration parameters from the section 4 can be used to perform a reconstruction. We started with $f(\vec{x})$ being the 3D Shepp–Logan phantom. We computed projections with the true acquisition geometry. From these data we performed a FDK reconstruction denoted $g(\vec{x})$ with the estimated acquisition geometry using the Python package RTK [8]. According to Eq. (13), the reconstructed image corresponds to the function $\sigma f_{\sigma R,\vec{t}}(\vec{x})$. In the figure 3, we show the same slice of both f and g, and of $|f - g|$.

We then computed the similarity correction applied to g. The reconstruction $g(\vec{x})$ obtained from the estimated geometric parameters should be equal to $\sigma f_{\sigma R,\vec{t}}(\vec{x}) = \sigma f(\sigma R\vec{x}+\vec{t})$, thus $f(\vec{x})$ should be equal to $\frac{1}{\sigma} g\left(\frac{1}{\sigma} R^T\left(\vec{x}-\vec{t}\right)\right)$. We used $\sigma, R, \vec{t}$ estimated at the beginning of this section from the BA results obtained in the section 4 in order to compute the similarity correction, thus an estimation of the original image f. Note that an interpolation is needed for the image grid computation: we used the linear interpolation method `interpolate.RegularGridInterpolator()` from SciPy. After such similarity correction applied to the image g we obtained an estimation $\frac{1}{\sigma} g\left(\frac{1}{\sigma} R^T\left(\vec{x}-\vec{t}\right)\right)$ of the initial image $f(\vec{x})$ (see the figure 4).
The widely used root-mean-square error (RMSE) between the initial 3D image and the reconstructed 3D image after the similarity correction was 0.08. For example, RMSE between the initial 3D image and the reconstructed with the true acquisition geometry 3D image was 0.09, which is normal for the numerical reconstruction implemented in RTK. Thus, we showed that the reconstruction with the estimated acquisition geometry can be performed and we verified that then the reconstructed image has the form $\sigma f_{\sigma R,\vec{t}}(\vec{x})$.

7 Conclusion

We have presented a numerical solution to the C-arm geometric calibration problem with the BA method. In simulations, we have observed the following phenomena: there are high errors in the estimated 3D marker positions and calibration parameters, but true and estimated 3D marker points differ almost by a similarity transformation. We analyzed the existing computer vision theory and translated it to the X-ray cone-beam geometry. Cone-beam geometric self-calibration problems can not be solved better than up to a similarity transformation. In the previous section we also presented the numerical verification of our theoretical result.

Acknowledgement

This work is supported by the Univ. Grenoble Alpes IDEX grant CQFD, the "Fonds unique interministériel", the European Union FEDER in Auvergne-Rhône-Alpes (3D4Carm), the ANR ROIdoré (ANR-17-CE19-0006-01) and the ANR CAMI LABEX (ANR-11-LABX-0004-01).

References

[1] M. Pollefeys, R. Koch, and L. Van Gool. "Self-calibration and metric reconstruction in spite of varying and unknown intrinsic camera parameters". *International Journal of Computer Vision* (1999), pp. 7–25. DOI: 10.1023/A:1008109111715.

[2] M. J. Daly, J. H. Siewerdsen, Y. B. Cho, et al. "Geometric calibration of a mobile C-arm for intraoperative cone-beam CT". *Medical Physics* 35 (May 2008), pp. 2124–2136. DOI: 10.1118/1.2907563.

[3] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2004. DOI: 10.1017/CBO9780511811685.

[4] J. Lesaint. "Data consistency conditions in X-ray transmission imaging and their application to the self-calibration problem". PhD thesis. 2018.

[5] B. Triggs, P. McLauchlan, R. Hartley, et al. "Bundle adjustment - a modern synthesis". *International Workshop on Vision Algorithms* (2000), pp. 298–372. DOI: 10.1007/3-540-44480-7_21.

[6] S. Agarwal, K. Mierle, and Others. *Ceres Solver*. http://ceres-solver.org.

[7] K. S. Arun, T. S. Huang, and S. D. Blostein. "Least-squares fitting of two 3-D point sets". *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1987), pp. 698–700. DOI: 10.1109/TPAMI.1987.4767965.

[8] S. Rit, M. Vila Oliva, S. Brousmiche, et al. "The Reconstruction Toolkit (RTK), an open-source cone-beam CT reconstruction toolkit based on the Insight Toolkit (ITK)". *Journal of Physics: Conference Series* 489 (Feb. 2014). DOI: 10.1088/1742-6596/489/1/012079.