Fast Epipolar Consistency without the Need for Pseudo Matrix Inverses

Alexander Preuhs, Michael Manhart and Andreas Maier

***Abstract*—Interventional C-arm systems allow flexible 2-D imaging of a 3-D scene while being capable of cone beam computed tomography. Due to the flexible structure of the C-arm, the rotation speed is limited, increasing the acquisition time compared to conventional computed tomography. Therefore, patient motion frequently occurs during data acquisition inducing inconsistencies in the projection raw data. A framework using Grangeat's theorem and epipolar consistency was successfully applied for compensating rigid motion. This algorithm was efficiently parallelized, however, before each iteration, the pseudo-inverse of each projection matrix must be calculated. We present a geometric modification of the presented algorithm which can be used without a pseudo-inverse. As such, the complete algorithm can be implemented for low-level hardware without the need of a linear algebra package that supports the calculation of matrix inverse. Both algorithms are applied for head motion compensation and the runtime of both is compared.**

I. Introduction

A fundamental assumption in computed tomography (CT) is that the scanned object remains static during the acquisition process. If this assumption cannot be fulfilled, images produced with conventional reconstruction algorithms will suffer from artifacts. Current C-arm CT acquisitions last about 20 seconds. During the acquisition time, involuntary patient motion is often inevitable without patient fixation. However, if the motion can be assumed to be rigid and smooth, a motion compensated reconstruction can be computed by finding the correct geometric correspondence between the motion affected projections and the calibration data. Four categories of compensating motion artifacts have emerged in literature and they can be grouped into approaches using external markers [1], image metrics on the reconstruction volume [2], 3-D/2-D registration of the projection data to digitally rendered radiographs from the reconstruction volume [3], [4] and projection data consistency based metrics [5]–[8].

In this work, we focus on a consistency method based on the 3-D radon transform. The method exploits epipolar geometry to find lines on two detectors corresponding to an epipolar plane. Grangeat's theorem can be used to find a mapping between each epipolar line pair and the 3-D radon value corresponding to the epipolar plane [9]. This algorithm is denoted as epipolar consistency and was presented by Aichert et al. [5]. As the algorithm directly works on the projection domain without the need of a reconstruction, the computational cost is low. It basically consists of comparing corresponding line configurations. This can be accelerated by parallelizing the algorithm using graphics processing units (GPU) [10].

To apply the algorithm for rigid motion compensation, the consistency between all possible line pairs is evaluated in an iterative optimization process in order to find the set of parameters describing the motion within the scan [11], [12]. In [10] before each iteration the pseudo-inverse of the projection matrices must be calculated on the CPU. We propose a geometric modification that allows to calculate corresponding epipolar lines without the need of a pseudo-inverse.

A. Preuhs and A. Maier are with the Pattern Recognition Lab, Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany.

M. Manhart is with Siemens Healthcare GmbH, Forchheim, Germany.

Email: alexander.preuhs@fau.de

II. Methods

A. Grangeat's Theorem

In cone-beam CT an X-ray source radially emits photons, that — after attenuation — are measured at a detector. The attenuation process for a ray can be described by an integral. However, due to the radial structure of the rays, integrating along a detector line does not result in a plane integral of the underlying object f, instead it differs by a radial weighting.

Grangeat's theorem describes the connection between this weighted integral and a plane integral — i.e. the 3-D radon value $\mathcal{R}f(\mathbf{n}, d)$ describing the integral along a plane with normal $\mathbf{n} \in \mathcal{S}^2$ at distance d. Using a derivative operation the radial weighting can be canceled out. Grangeat defined an intermediate function $S_\lambda(\mathbf{n})$ that is calculated from the projection data which can be related to the derivative of the 3-D radon transform

$$S_\lambda(\mathbf{n}) = \int_{\mathcal{S}^2} \delta'(\mathbf{x}^\top \mathbf{n}) g_\lambda(\mathbf{x}) d\mathbf{x} = \frac{\partial}{\partial d} \mathcal{R}f(\mathbf{n}, d)|_{d=\mathbf{c}_\lambda^\top \mathbf{n}} \ , \quad (1)$$

where $g_\lambda(\mathbf{x})$ describes a single value on the detector with λ describing the projection index, $\mathbf{c}_\lambda$ the source position and $\mathbf{x}$ a vector from the source to a detector pixel. The geometry for two projections $\lambda = a$ and $\lambda = b$ is visualized in Fig. 1. Here $\delta'(\cdot)$ describes the derivative of the Dirac delta distribution. A detailed evaluation of Eq. (1) can be found in [9], and some simplifications are discussed in [5].

B. Epipolar Consistency

It directly follows from Eq. (1) that two projections a, b must satisfy

$$S_a(\mathbf{n}) = S_b(\mathbf{n}) \qquad \forall \mathbf{n} \in \mathcal{S}^2 : \mathbf{c}_b^\top \mathbf{n} = \mathbf{c}_a^\top \mathbf{n} \ . \quad (2)$$

If the geometry information is wrong, e.g. due to rigid object motion, then Eq. (2) will not hold. Thus, we can use it as a measure of inconsistency. Below we summarize the framework proposed by Aichert et al. [5], [10], which is used to evaluate the consistency of two views.

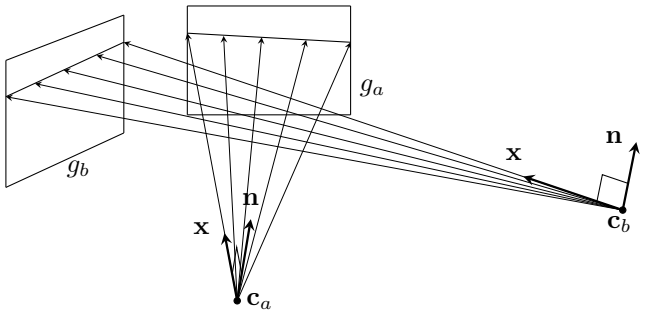

Fig. 1. Schematic drawing of a scene including two projections. The vector $\mathbf{n}$ describes the normal of a radon plane. Several realizations of the vectors $\mathbf{x}$ are drawn that are perpendicular to $\mathbf{n}$. The pixel intensity measured at the detector along ray $\mathbf{x}$ is described by $g_a(\mathbf{x})$ or $g_b(\mathbf{x})$, respectively.

The intermediate function $S_\lambda(\mathbf{n})$ can be precomputed for each projection. Then, the global indexing by the plane normal $\mathbf{n}$ can be replaced by a local projection-pair-dependent indexing using a line $\mathbf{l}_\lambda^\kappa$ defined on the detector described by g_λ. By epipolar geometry two epipolar lines $\mathbf{l}_a^\kappa$ and $\mathbf{l}_b^\kappa$ are found that belong to the same epipolar plane $\mathbf{E}^\kappa$ — i.e. the radon plane. The algorithm starts with a configuration of two projections described by their projection matrix $\mathbf{P}_a$ and $\mathbf{P}_b$, respectively. Using these two projection matrices a mapping matrix is derived that maps an angle κ to an epipolar plane $\mathbf{E}^\kappa$. Using the pseudo-inverse the respective epipolar lines $\mathbf{l}_a^\kappa$ and $\mathbf{l}_b^\kappa$ are computed. The respective values are then used to look up the values at the precomputed intermediate function S_a and S_b. This allows the indexing of Eq. (1) using an angle κ and two projection matrices

$$S_a(\kappa, \mathbf{P}_a, \mathbf{P}_b) = S_a(\mathbf{n}) \quad \forall \mathbf{n} \in \mathcal{S}^2 : \mathbf{c}_b^\top \mathbf{n} = \mathbf{c}_a^\top \mathbf{n} \ . \tag{3}$$

To evaluate the consistency of a whole scan, many different views must be compared to each other, while in each two-view comparison a multitude of line pairs are evaluated. As the operations are independent from each other, this can be evaluated in parallel allowing the efficient parallelization of the algorithm using GPUs.

C. Projective Geometry

Projective geometry can be seen as an extension to the common Euclidean geometry. In the context of image reconstructions, projective geometry is mostly used to describe the projection of a world point to a detector. Therefore, a projection matrix is created that performs a projective transformation on a world point. In this context, the world point must be converted to homogeneous coordinates first.

Homogeneous coordinates are the representation of n-dimensional points in the projective space and are written as $(n+1)$-component vectors. In $\mathbb{P}^3$ a point is described by $(x, y, z, w)^\top$, and we can obtain the euclidean representation by dividing with the last component $(x/w, y/w, z/w)^\top$. Similarly, a plane is described by $(a, b, c, d)^\top$. The vector can be understood as the parameters of a Hessian normal form, where the first three components describe the normal of the plane, and d is the scaled distance to the origin. If $a^2 + b^2 + c^2 = 1$ then d is exactly the signed distance to the origin. The concept that a four-component vector can either be interpreted as a point or a plane is called duality, where we refer to the point interpretation as primal form and the plane interpretation as dual form.

A special case is the representation of a line in $\mathbb{P}^3$. There is no direct description but we can construct the line as the connection of two points or the intersection of two planes. An intuitive derivation can be found in [13], we only state the relevant result of this derivation. The creation of a line as the incident of two planes $\mathbf{a}, \mathbf{b} \in \mathbb{P}^3$ is obtained by

$$\text{meet}(\mathbf{a}, \mathbf{b}) = \mathbf{L} = \begin{pmatrix} p \\ q \\ r \\ s \\ t \\ u \end{pmatrix} = \begin{pmatrix} a_z b_w - a_w b_z \\ a_y b_w - a_w b_y \\ a_y b_z - a_z b_y \\ a_x b_w - a_w b_x \\ a_x b_z - a_z b_x \\ a_x b_y - a_y b_x \end{pmatrix} \ , \tag{4}$$

where the six components of $\mathbf{L}$ are often referred to as Plücker coordinates. We can build an anti-symmetric matrix $\mathbf{L}_K$ from the Plücker coordinates that represents a line as the intersection of two planes — i.e. the dual representation of a line. A point $\mathbf{x}$ common to a plane $\mathbf{p}$ and the line $\mathbf{L}$ can be found by right-multiplication of $\mathbf{p}$ to $\mathbf{L}_K$

$$\mathbf{x} = \text{meet}(\mathbf{L}, \mathbf{p}) = \mathbf{L}_K \, \mathbf{p} = \begin{pmatrix} 0 & -p & -q & r \\ p & 0 & s & -t \\ q & -s & 0 & u \\ -r & t & -u & 0 \end{pmatrix} \mathbf{p} \ . \tag{5}$$

Note that there is also a primal representation of $\mathbf{L}$ which will not be discussed in this paper.

An extension in the projective geometry is the concept of geometric primitives at infinity. They are regular objects and thus can be handled as any other objects. A point at infinity is defined by a homogeneous coordinate $w = 0$. In $\mathbb{P}^3$ the plane at infinity is defined by $\boldsymbol{\pi}_\infty = (0, 0, 0, 1)^\top$. All previously introduced equations are also valid for objects at infinity. We could for example use Eq. (5) to find the incident of a line $\mathbf{L}$ with $\boldsymbol{\pi}_\infty$, which will be a point at infinity, where the first three component of that point are the direction of the line.

An advantage of using projective geometry is the representation of transformations based on matrix multiplication. A point $\mathbf{x}'$ which is the transformation of the point $\mathbf{x}$ under $\mathbf{T}$ is simply found by

$$\mathbf{x}' = \mathbf{T}\,\mathbf{x} \ . \tag{6}$$

The transformation rule for planes can be derived from the property that the distance from a point $\mathbf{x}$ incident to a plane $\mathbf{p}$ is zero. The distance between the transformed point $\mathbf{x}'$ and the plane $\mathbf{p}'$ will remain zero if they have been transformed under the same transformation $\mathbf{T}$. It therefore holds that

$$\mathbf{p}'^\top \mathbf{x}' = \mathbf{p}^\top \mathbf{x} = 0 \ . \tag{7}$$

Solving Eq. (6) for $\mathbf{x}$ and plugging that in Eq. (7) gives

$$\mathbf{p}'^\top \mathbf{x}' = \mathbf{p}^\top \mathbf{T}^{-1} \mathbf{x}' = \left((\mathbf{T}^{-1})^\top \mathbf{p} \right)^\top \mathbf{x}' \ , \tag{8}$$

it directly follows that

$$\mathbf{p}' = (\mathbf{T}^{-1})^\top \mathbf{p} \tag{9}$$

which describes the transformation of planes. The point $\mathbf{x}$ incident to a plane $\mathbf{p}$ and a line $\mathbf{L}$ can be found by right-multiplying the plane to the dual representation of $\mathbf{L}$ (cf. Eq. (5)). Further, a transformed point $\mathbf{x}'$ will be incident to the plane $\mathbf{p}'$ and line $\mathbf{L}'$ if both are transformed under a transformation $\mathbf{T}$, thus,

$$\mathbf{L}_K \, \mathbf{p} \; = \mathbf{x}, \qquad\qquad \mathbf{L}'_K \, \mathbf{p}' \; = \mathbf{x}' \;\; . \tag{10}$$

When we solve Eq. (6) and (9) for $\mathbf{x}$ and $\mathbf{p}$, respectively, we can plug the result in the left part of Eq. (10) which results in

$$\mathbf{L}_K \mathbf{T}^\top \, \mathbf{p}' = \; \mathbf{T}^{-1} \, \mathbf{x}' \; \Leftrightarrow \; \mathbf{T} \, \mathbf{L}_K \mathbf{T}^\top \, \mathbf{p}' = \; \mathbf{x}' \;\; . \tag{11}$$

Comparing the result with the right side of Eq. (10) it immediately emerges that the line $\mathbf{L}'$ which is the transformation of $\mathbf{L}$ under $\mathbf{T}$ can be calculated by

$$\mathbf{L}'_K \; = \; \mathbf{T} \, \mathbf{L}_K \, \mathbf{T}^\top \;\; . \tag{12}$$

D. Optimized Algorithm

The main purpose of the algorithm presented in Section II-B is to find the mapping between two lines $\mathbf{l}_a^\kappa$ and $\mathbf{l}_b^\kappa$ that can be used to look up the corresponding precomputed values S_a and S_b, respectively. This is achieved by first finding epipolar planes $\mathbf{E}^\kappa$ which are then mapped to the corresponding epipolar lines. The algorithm presented in [5] makes use of the pseudo-inverse to compute that mapping. However, the calculation of a pseudo-inverse is not supported on many GPUs, and must therefore be done on the CPU beforehand, whereas the rest of the framework is parallelizable. In addition a linear algebra library must be included to support the calculation of pseudo matrix inverses.

We propose a geometric modification that creates the mapping without the need of a pseudo-inverse. As shown in Section II-C, the transformation rule depends on the object that is to be transformed. It can be seen from Eq. (12) that lines are transformed using the transformation matrix and its transpose. Thus, transforming the plane to a line while preserving the relevant information will make the pseudo-inverse dispensable.

We can achieve this using the concept of infinity. The projective three-space is covered by the infinity plane $\boldsymbol{\pi}_\infty = (0, 0, 0, 1)$. Any plane intersects the infinity plane in a line incident to $\boldsymbol{\pi}_\infty$ and the plane itself, i.e. a line at infinity. The orientation of the plane is persevered by the direction of the line. In a last step, we can simply use Eq. (12) to project the line at infinity, resulting in the desired epipolar lines.

Therefore, we start with the epipolar plane $\mathbf{E}^\kappa$. Using Eq. (4) we can compute the line at infinity $\mathbf{L}^\kappa$ as the intersection of the epipolar plane with $\boldsymbol{\pi}_\infty$

$$\mathbf{L}^\kappa = \text{meet}(\mathbf{E}^\kappa, \boldsymbol{\pi}_\infty) \;\; . \tag{13}$$

Using the representation of the line at infinity now allows us to use the transformation rule as described by Eq. (12) to obtain the epipolar line $\mathbf{l}_\lambda^\kappa$

$$[\mathbf{l}_\lambda^\kappa]_\times \; = \; \mathbf{P}_\lambda \, \mathbf{L}_K^\kappa \, {\mathbf{P}_\lambda}^T \;\; . \tag{14}$$

The parameters of $\mathbf{l}_\lambda^\kappa$ are available from the 3×3 skew matrix $S = [\mathbf{l}_\lambda^\kappa]_\times$ as $\mathbf{l}_\lambda^\kappa = (S_{12}, S_{20}, S_{01})^\top$. As a result Eq. (13) and (14) replace the mapping from epipolar planes to lines presented in [5] and, therefore, makes the computation of pseudo-inverses unnecessary. The additional cost is the implementation of Eq. (4) on the GPU, however, this can be reused to simplify the calculation of source positions. As the three rows of the projection matrix can be interpreted as planes all passing the source, the incident of two of these planes will create a line. Using matrix multiplication (cf. Eq. 5) the source position is then found by the incident of that line with the third plane.

E. Optimization

If rigid motion occurs during the scan, the calibrated trajectory does not represent the true geometry of the acquired data. In order to restore the true geometry, a rigid transformation $\mathbf{T}_\lambda$ for each projection matrix $\mathbf{P}_\lambda$ must be found. The true geometry is expected to have minimal inconsistency. We therefore define the inconsistency between two projections a and b in dependence of the respective rigid transformations $\mathbf{T}_a$ and $\mathbf{T}_b$ by

$$d(\mathbf{P}_a\mathbf{T}_a, \mathbf{P}_b\mathbf{T}_b) = \frac{1}{N_\kappa} \sum_{k=0}^{K} \left[S_a(k\Delta\kappa, \mathbf{P}_a\mathbf{T}_a, \mathbf{P}_b\mathbf{T}_b) - S_b(k\Delta\kappa, \mathbf{P}_b\mathbf{T}_b, \mathbf{P}_a\mathbf{T}_a) \right]^2 \;\; , \tag{15}$$

where N_κ is the number of epipolar planes that hit both detectors and K is the total number of sampled epipolar planes. The angular step-size is denoted by $\Delta\kappa$. To be more robust for outliers we use the robust Cauchy norm and define the inconsistency of two views by

$$e_{a,b} = \frac{d(\kappa, \mathbf{P}_a\mathbf{T}_a, \mathbf{P}_b\mathbf{T}_b)}{1 + \frac{1}{c}\, d(\kappa, \mathbf{P}_a\mathbf{T}_a, \mathbf{P}_b\mathbf{T}_b)} \;\; . \tag{16}$$

The parameter c controls the penalty and should be selected according to the intensity of the projection images. We denote the vector of rigid transformations $\mathbf{T} = [T_1, ...T_N]$, with N being the number of projections of the trajectory. The corrected geometry is denoted by $\hat{\mathbf{T}}$ and found by solving

$$\hat{\mathbf{T}} = \arg\min_{\mathbf{T}} \sum_{a,b=1}^{N} e_{a,b} \;\; . \tag{17}$$

Since motion is expected to be smooth we model each rigid motion parameter in $\mathbf{T}$ by an Akima spline [14]. This also allows the reduction of the search space, as we must not find a transformation for each λ, but only for the nodes of the spline. The optimum is then found using the open source non-linear optimizer JPOP[1] in CONRAD [15].

III. Experiments

To evaluate the proposed method, we have acquired a 200° short scan (496 projections) of a head phantom using a robotic

[1] https://www5.cs.fau.de/research/software/java-parallel-optimization-package/

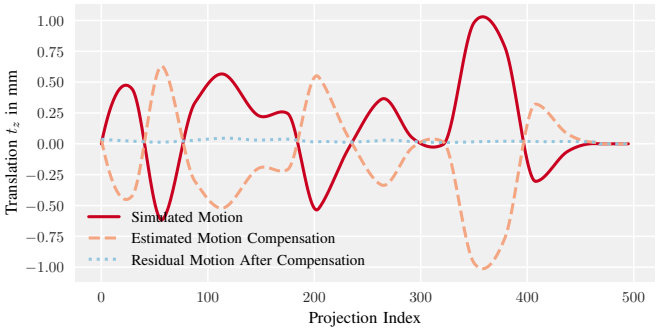

Fig. 2. Simulated, estimated and residual motion t_z for each projection.

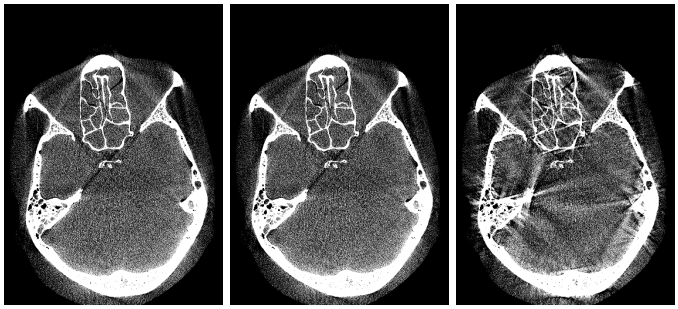

Fig. 3. Central slices of the reconstructed volume HU [-100, 100]. Left: ground truth, mid: with simulated motion after compensation, right: with simulated motion.

C-arm system (Artis zeego, Siemens Healthcare GmbH, Germany). Thereafter, we simulate rigid motion, which is directly incorporated in the projection matrices. This is done using a rigid motion creator[2].

Epipolar consistency is known to produce mostly horizontal epipolar lines in a majority of the projection pairs within a short scan. Only view pairs that are almost opposed to each other present diverging epipolar lines. Motion that is parallel to the epipolar lines is not detectable by the presented consistency measure. Thus, we only concentrate on motion orthogonal to the epipolar lines in all pairs, which is typically denoted as out-plane motion. Defining the rotation axis of the short scan as the z-axis, we only simulate translations in z-direction. The simulated motion pattern consisting of 17 spline nodes is shown in Fig. 2.

IV. Results

The reconstructions of the acquired head phantom is shown in Fig. 3 for the motion corrupted case (right), the motion compensated case (mid) and the ground truth (left). The corresponding motion is depicted in Fig. 2. Both algorithms produce the same results, only the runtime is expected to change. By skipping the sequential calculation of pseudo-inverses the runtime could be reduced by 1.29% using a standard computer with an Intel Core i7-4910MQ and a NVIDIA Quadro K2100M. The overall runtime for the motion parameter estimation was 841.7 seconds using the proposed modifications and 852.8 seconds if the inverse is pre-calculated before each optimization step.

V. Conclusion and Discussion

We presented a modification to the algorithm presented in [10] which avoids the calculation of inverse projection matrices. This is achieved by transforming the respective epipolar planes to lines at infinity. Lines are transformed — in contrast to planes — using only the transformation matrix and its transposed. Thus, only the projection matrix and its transposed must be available.

The runtime could be improved by 1.29% using a Java environment. Using more high-level programming languages, e.g. python, the runtime advantage could eventually increase, as more computations can be performed on a dedicated GPU. Furthermore, when implementing the algorithm in low-level programming languages, e.g. C++/CUDA, no linear algebra libraries need to be included.

Disclaimer: The concepts and information presented in this paper are based on research and are not commercially available.

[2]https://github.com/alPreuhs/MotionCreator

References

[1] J.-H. Kim, J. Nuyts, Z. Kuncic, and R. Fulton, "The feasibility of head motion tracking in helical CT," *Medical physics*, vol. 40, no. 4, 2013.

[2] A. Sisniega, J. W. Stayman, Q. Cao, J. Yorkston, J. H. Siewerdsen, and W. Zbijewski, "Motion estimation using a penalized image sharpness criterion for resolution recovery in extremities cone-beam CT," *Proceedings of the 3rd CT-meeting*, 2016.

[3] W. Wein and A. Ladikos, "Towards General Motion Recovery in Cone-Beam Computed Tomography," *Proceedings of the 12th Fully 3D*, 2013.

[4] M. Berger, K. Müller, M. Unberath, J. Thies, J.-H. Choi, R. Fahrig, and A. Maier, "Marker-free motion correction in weight-bearing cone-beam CT of the knee joint," *Medical Physics*, vol. 43, no. 3, 2016.

[5] A. Aichert, M. Berger, J. Wang, N. Maass, A. Doerfler, J. Hornegger, and A. K. Maier, "Epipolar Consistency in Transmission Imaging," *IEEE Transactions on Medical Imaging*, vol. 34, no. 11, 2015.

[6] C. Debbeler, N. , M. Elter, F. Dennerlein, and T. M. Buzug, "A New CT Rawdata Redundancy Measure applied to Automated Misalignment Correction," *Proceedings of the 12th Fully 3D*, 2013.

[7] N. Maass, F. Dennerlein, A. Aichert, and A. Maier, "Geometrical Jitter Correction in CT," in *Proceedings of the 3rd CT-meeting*, 2014.

[8] M. Berger, Y. Xia, W. Aichinger, K. Mentl, M. Unberath, A. Aichert, C. Riess, J. Hornegger, R. Fahrig, and A. Maier, "Motion Compensation for Cone-Beam CT Using Fourier Consistency Conditions," *Physics in Medicine and Biology*, vol. 62, no. 17, 2017.

[9] M. Defrise and R. Clack, "A Cone-Beam Reconstruction Algorithm Using Shift-Variant Filtering and Cone-Beam Backprojection," *IEEE Transactions on Medical Imaging*, vol. 13, no. 1, 1994.

[10] A. Aichert, K. Breininger, T. Köhler, and A. Maier, "Efficient epipolar consistency," *Proceedings of the 4rd CT-meeting*, 2016.

[11] R. Frysch and G. Rose, "Rigid motion compensation in C-arm CT using consistency measure on projection data," in *MICCAI*, 2015.

[12] M. Unberath, A. Aichert, and A. Maier, "Consistency-based Respiratory Motion Estimation in Rotational Angiography," *Medical Physics*, 2017.

[13] J. F. Blinn, "A Homogeneous Formulation for Lines in 3 Space," *Computer Graphics (SIGGRAPH '77 Proceedings)*, vol. 11, 1977.

[14] H. Akima, "A New Method of Interpolation and Smooth Curve Fitting Based on Local Procedures," *Journal of the ACM*, vol. 17, no. 4, 1970.

[15] A. Maier, H. Hofmann, M. Berger, P. Fischer, C. Schwemmer, H. Wu, K. Müller, J. Hornegger, J.-H. Choi, C. Riess, A. Keil, and R. Fahrig, "CONRAD - A software framework for cone-beam imaging in radiology," *Medical Physics*, vol. 40, no. 11, 2013.