Hybrid geometric self-calibration of radiological systems

Anastasia Konik

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

École doctorale : MSTII - Mathématiques, Sciences et technologies de l'information, Informatique
Spécialité : Mathématiques Appliquées
Unité de recherche : Translational Innovation in Medicine and Complexity

Auto-étalonnage géométrique hybride des systèmes radiologiques

Hybrid geometric self-calibration of radiological systems

Présentée par :

Anastasia KONIK

Direction de thèse :

Laurent DESBAT Professeur des Universités, UNIVERSITE GRENOBLE ALPES	Directeur de thèse
Yannick GRONDIN Ingénieur Docteur, SURGIQUAL INSTITUTE	Co-encadrant de thèse

Rapporteurs :

Voichita MAXIM
Maître de conférences HDR, INSA LYON
Thomas RODET
Professeur des Universités, ENS PARIS SACLAY

Thèse soutenue publiquement le **21 février 2023**, devant le jury composé de :

Laurent DESBAT Professeur des Universités, UNIVERSITE GRENOBLE ALPES	Directeur de thèse
Voichita MAXIM Maître de conférences HDR, INSA LYON	Rapporteure
Thomas RODET Professeur des Universités, ENS PARIS SACLAY	Rapporteur
Valérie PERRIER Professeur des Universités, GRENOBLE INP	Présidente
Charles SOUSSEN Professeur des Universités, CENTRALESUPELEC	Examinateur

Invités :

Yannick GRONDIN Ingénieur Docteur, SURGIQUAL INSTITUTE	Co-encadrant de thèse

Hybrid geometric self-calibration of radiological systems

Anastasia Konik

Supervisors: Laurent Desbat, Yannick Grondin

TIMC-GMCAO, UGA

Acknowledgements

This work is supported by the French National Research Agency in the framework of the "Investissement d'avenir" program (ANR-15-IDEX-02), the "Fonds unique interministériel", the European Union FEDER in Auvergne-Rhône-Alpes (3D4Carm) and the ANR CAMI LabEx (ANR-11-LABX-0004-01).

I was very fortunate to have many opportunities to work in Academia in Russia and in France and to learn more about science. I am grateful to my first supervisors Vladimir Alexandrov and Nelly Shulenina that they showed me how researchers work. I would like to thank my current advisors Laurent Desbat and Yannick Grondin for their support, especially during the hard COVID time. Thank you for introducing me medical imaging and computer vision, for your guidance and your expertise. You taught me how to present my results and how to do independent research. I would like to thank also Eugene Chizhonkov and Zarina Lepshokova with whom I was also lucky to work in applied mathematics and even in social sciences. I would like to express a special gratitude to Sergei Grudinin for his help in difficult times after moving to France.

I would like to thank my committee members - Valérie Perrier, Voichita Maxim, Thomas Rodet, Charles Soussen - for their time invested in my defense.

I want to express my special gratitude to my family, especially my parents Galina and Aleksei, who continue to support me in my decisions day after day. But also to my sleepless relatives in Seattle, my sister Olesya, Nikita, Anna, Tom and Mix, for the fact that all these years they are a big part of my life even at a distance. You are the best family I could ever dream of. Thank you for everything you have done for me! I also want to thank my other relatives in Russia. Thank you all for the immeasurable support I received from you in my early years!

I would like to thank my friends in Grenoble. I was not able to visit home frequently because of different reasons, so I appreciated a company of many friends I have in the team GMCAO of TIMC (Nicolas, Hung, Mathurin, Matthieu), in the team PRETA (Mariel, Rémi) and many others from TIMC. Special thanks to Katia for Russian minutes in the lab, to Vera for her constant support and movie outings and to my best neighbour Parivash for long conversations and walks. The months in quarantine would have been intolerable without being in contact with my Russian friends. Thank you for remembering me!

Finally, thanks to all my teachers in Russia and in France and also to my colleagues from Grenoble INP and UGA who taught me to be a teacher. One of the greatest things was the opportunity to learn from and to work with some of the smartest people in the world. You will stay in my mind forever.

Abstract

Dans ce manuscrit nous nous concentrons sur l'auto-étalonnage des systèmes à rayons X. Par auto-étalonnage nous considérons la situation où nous devons définir des paramètres pour les modèles de projection de rayons X avec des marqueurs dans une mire à géométrie inconnue ou sans marqueurs. Nous considérons quelques modèles géométriques classiques de systèmes radiologiques. Tout d'abord, le modèle 3D en faisceau conique avec les faisceaux divergents où la trajectoire générale de la source est calibrée avec la méthode d'ajustement de faisceaux. On montre théoriquement dans le cas de la géométrie conique 3D que tout système avec un tel modèle intégral ne peut être calibré qu'à une similitude près. Deuxièmement, pour la géométrie parallèle 2D et la géométrie en éventail 2D avec des sources alignées, nous proposons la calibration basée sur les conditions de cohérence des données (DCC) sur les distributions. Dans ce cas, nous étendons les DCC connues des fonctions aux distributions, nous modélisons des marqueurs avec des distributions de Dirac et construisons les nouvelles procédures analytiques pour calibrer à l'aide de mires de calibration spéciales. Enfin, par analogie avec le cas 2D, nous construisons les procédures de calibration similaires pour les cas en faisceau conique avec des sources alignées et les cas en faisceau conique avec des sources dans le plan parallèle au plan du détecteur. Nous présentons des simulations numériques dans chaque cas.

In this manuscript, we concentrate on self-calibration of X-ray systems. By self-calibration, we consider the situation when we need to define parameters of X-ray projection models with markers in a calibration system of unknown geometry or without markers. We consider few classical X-ray models. Firstly, the 3D cone-beam model with divergent beams and with the general source trajectory is calibrated with the bundle adjustment method. It's shown theoretically in the 3D cone-beam geometry that any system with such an integral model cannot be geometrically calibrated better then up to a similarity transformation. Secondly, for the 2D parallel geometry with parallel beams and the 2D fan-beam geometry with divergent beams and sources on a line we propose a geometric calibration based on data consistency conditions (DCC) on distributions. In this case, we extend the known DCC from functions to distributions. We model markers with Dirac distributions and construct new analytical procedures to calibrate using special calibration cages. Lastly, by the analogy with the 2D case, we construct similar calibration procedures for the cases of the cone-beam with sources on a line and the cone-beam with sources in the plane parallel to the detector plane. We present numerical simulations in each case.

Contents

List of Figures

List of Tables

1 Introduction

1.1 French summary of the chapter

Le premier chapitre est consacré à l'introduction à la vision par ordinateur et à l'imagerie médicale. Dans la section 1.2 nous commençons par la brève histoire de la tomographie, la description des systèmes à rayons X et les principales géométries utilisées pour la modélisation des systèmes à rayons X. Les systèmes à rayons X sont généralement constitués d'une source et d'un détecteur. Les rayons X traversant un objet sont atténués par l'objet. Pour un objet hétérogène, cette atténuation des rayons X est différente selon les parties plus ou moins denses d'objet, créant un contraste dans l'image de projection. Un patient est ainsi modélisé par sa fonction d'atténuation f à reconstruire à partir des images de projection. Différentes configurations des sources sont introduites : les trajectoires circulaires et spirales dans les systèmes CT rotatifs, les trajectoires linéaires comme en tomosynthèse [DG03] et le cas des sources multiples comme par exemple dans les systèmes CT rotatifs à deux sources [Flo+06]. Plus de détails historiques peuvent être trouvés dans [Kal06].

La première génération de systèmes à rayons X a été construite suivant la géométrie en faisceau parallèle, plus tard avec des géométries en faisceau divergent. Dans notre travail nous nous concentrons sur la géométrie parallèle 2D décrite par la transformée de Radon. Nous rappelons également dans la section 1.2 la transformée en faisceau divergent 2D (dite transformée en éventail) et 3D (dite transformée en faisceau conique). Pour toutes les géométries nous considérons des détecteurs plans. Nous présentons également les trois familles de systèmes tomographiques aux géométries divergentes : les scanners CT, les systèmes C-arm et les systèmes de contrôle non destructif.

Dans la section 1.3 nous décrivons le modèle de caméra et la matrice de projection qui permet de projeter un point 3D de l'espace vers un point 2D dans le plan d'image. Les paramètres géométriques du modèle de projection doivent être définis. S'ils ne sont pas connus, la caméra doit être calibrée géométriquement. L'étalonnage (ou la calibration) peut être effectué avec des marqueurs dont la géométrie et la position sont connues [Stu12]. En tomographie, les marqueurs sont généralement de petits objets ronds denses (semblables à des billes de roulement) ajoutés à la scène. Dans certains cas, ces billes sont positionnées sur ou à l'intérieur d'un fantôme de sorte à connaître les coordonnées 3D : on connaît ainsi les coordonnées 3D relatives des centres des marqueurs entre eux. Ainsi, l'étalonnage va consister à estimer les éléments de la matrice de projection, connaissant les coordonnées des points 3D et les coordonnées des points projetés 2D respectifs, détectés dans l'image. Dans certaines situations, nous ne connaissons pas non plus les coordonnées des points 3D, alors nous avons deux ensembles d'inconnues : les matrices de projection P^i pour différentes vues de caméra et les coordonnées 3D des

marqueurs Q_j. Dans ce cas, nous pouvons résoudre le problème d'ajustement de faisceaux ("bundle adjustment" en anglais) en utilisant différentes techniques d'optimisation [Tri+00].

En vision par ordinateur, les méthodes d'étalonnage basées sur l'image peuvent être divisées en deux groupes : avec marqueurs et sans marqueurs. Le dernier groupe est généralement appelé un groupe d'algorithmes d'auto-étalonnage. En l'absence de marqueurs, il est classique de rechercher des points d'intérêt dans les images. Pour cela, nous pouvons utiliser des algorithmes spéciaux pour la détection de coin ou la détection de singularités. Ensuite, il est possible de définir un ensemble de correspondances des points d'intérêt dans les vues différentes en fonction de la proximité et de la similitude de leur intensité dans les voisinages [HZ04]. Certaines techniques d'auto-calibration décrites dans [HZ04] sont également présentées dans notre introduction dans la section 1.3. Nous y définissions la transformation projective et la similitude. D'après [HZ04], nous savons que nous avons une solution du problème d'ajustement de faisceaux à une transformation projective près. La classe de telles solutions nous donne une reconstruction projective de la scène. Parfois, nous pouvons avoir ce qu'on appelle la reconstruction métrique de la scène. Pour la reconstruction métrique, nous avons une solution à une similitude près. En général, il est facile d'obtenir une reconstruction projective (par exemple, à partir d'ajustement de faisceaux) à partir de l'ensemble des points d'intérêt. Ensuite pour l'auto-calibration, il faut estimer l'homographie rectifiant H, la transformation pour laquelle on peut obtenir une reconstruction métrique P^iH, $H^{-1}Q_j$ à partir de la reconstruction projective P^i, Q_j.

Le processus de calibration est une étape essentielle, entre autres, pour la reconstruction d'images 3D. Par analogie avec la vision par ordinateur, les chercheurs en tomographie envisagent des méthodes de calibration basées sur l'image avec marqueurs et sans marqueurs [Les18]. Pour le premier groupe, les coordonnées 3D des centres de marqueurs peuvent être connues ou inconnues. Dans nos travaux d'auto-étalonnage en tomographie, nous utilisons à la fois les approches des algorithmes avec marqueurs de géométrie inconnue et les algorithmes sans aucun marqueur.

Dans la section 1.4 nous passons en revue deux types de méthodes existantes pour calibrer différents systèmes à rayons X. Pour le premier groupe, les propriétés de projection aident à dériver des procédures de calibration [WT04], [RGG94], [LW12]. Très souvent, les procédures d'auto-calibration aboutissent dans ce cas à des optimisations numériques non linéaires comme dans le cas d'ajustement de faisceaux. Dans de nombreux travaux, des mires d'étalonnage avec des propriétés spéciales ont été conçues en plus [Cho+05], [Noo+00], [MCN09]. Certaines méthodes analytiques de ce groupe ont été récemment proposées dans [TSH19] pour la géométrie en faisceau conique avec la trajectoire de source circulaire et dans [Jon18] pour les géométries en éventail et en faisceau conique avec des trajectoires générales de source.

Le deuxième groupe de méthodes est basé sur les conditions de cohérence des données (DCC pour "Data Consistency Conditions" en anglais). Les DCC sont des équations qui doivent être satisfaites par les données de projection. Nous mentionnons les DCC dans la géométrie parallèle 2D en présentant les conditions de cohérence de Helgason-Ludwig (HLCC) [Hel65; Lud66] et quelques méthodes de

calibration développées à partir de celles-ci dans [BB00b; BB00a], [Pan+08] et [DS14]. Nous mentionnons également les DCC de [Cla13] pour la géométrie en éventail avec des sources alignées avec la procédure de calibration de [NDC20] que nous souhaitons généraliser dans notre travail. Pour le cas 3D, il est important de parler de DCC pour le cas du faisceau conique avec des sources alignées de [NDC20], et dans le plan parallèle au plan du détecteur de [CD13]. Ce sont des géométries que nous aborderons dans les prochains chapitres. Le principal avantage des méthodes basées sur les DCC est qu'elles peuvent aider à construire une procédure analytique basée uniquement sur des images projetées. L'inconvénient est qu'elles ne peuvent généralement pas gérer la troncature des données.

Puisque notre travail est principalement dédié aux algorithmes hybrides, c'est-à-dire aux méthodes qui combinent l'utilisation de marqueurs et de DCC, nous mentionnons également dans la section 1.4 l'algorithme hybride et l'idée de calibrer avec des données tronquées de faisceaux coniques présentées par Unberath et ses collègues dans [Unb+17], l'article dont nous nous sommes inspirés. Nous illustrons également le problème de la troncature des projections et pourquoi il est crucial pour les DCC.

1.2 Tomography and some acquisition geometries

X-ray computed tomography (CT) was widely developed in the second half of the 20th century [Kal06]. Usually, in X-ray systems there are an X-ray source and a detector. X-rays passing through a patient modelled by its attenuation map f produce projected images in the detector. The principal task of tomography is the reconstruction of f from the projected images.

The first generation of X-ray systems was built with pencil beam, single source, single detector, later the production moved to geometries with divergent beams. First detectors in 1970s were flat with one or few rows, but later X-ray systems were equipped with a ring-like detector. In 2000s more rows were added to the detector to have simultaneous acquisition of few slices leading to well-known CT scanners (see fig. 1.1 for different detector configurations). Different source configurations were introduced during the history of tomography: circular and spiral trajectories in rotating CT systems, linear trajectories as in tomosynthesis [DG03], multiple sources in one system as in [Flo+06]. Nowadays, with the relative low cost of X-ray micro-sources, new configurations are studied, for example, the stationary CT with the rectangular imaged region and multiple sources [Gon+13], the case of multi-source CT with the cube structure [Gon+18]. More historical details can be found in [Kal06]. In our work, we consider different 2D and 3D geometries of X-ray systems with different source trajectories with flat detectors.

We usually suppose that $f \in \mathscr{D}(\mathbb{R}^3)$ or $f \in \mathscr{D}(\mathbb{R}^2)$ (in other words, f belongs to the space of compactly supported smooth functions in $\mathbb{R}^3$ or $\mathbb{R}^2$, where the smoothness is classically defined by the infinite number of continuous derivatives), but very often the mathematical theory can be done in a much larger mathematical framework such as $f \in \mathscr{S}(\mathbb{R}^3)$ or $f \in \mathscr{S}(\mathbb{R}^2)$ (in other words, f belongs to the Schwartz space or to the space of rapidly decreasing smooth functions). It's important to know here

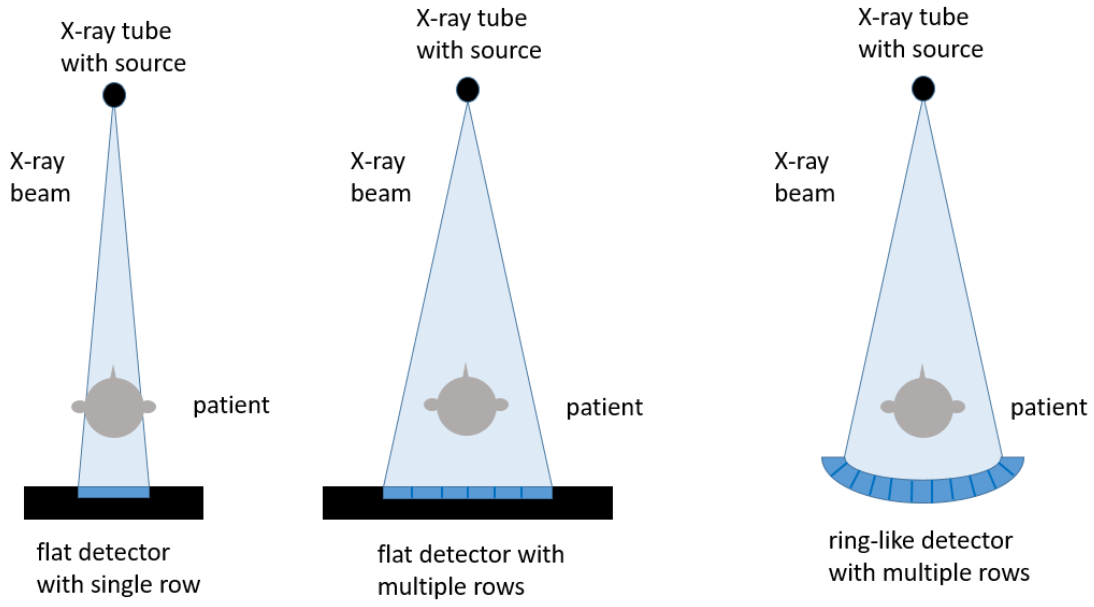

Figure 1.1: Left: flat detector system with single row, center: flat detector system with multiple rows, right: ring-like detector system with multiple rows.

that $\mathscr{D}(\mathbb{R}^N) \subset \mathscr{S}(\mathbb{R}^N)$ for any dimension N. In fact, the integral projection, see, for example, Eq. (1.1), is naturally well defined for the Lebesgue integrable function f.

One class of X-ray systems with parallel acquisition is described by the Radon transform. We can define the Radon transform of the function from the Schwartz space in any dimension (see, for example, [Nat01]):

Definition 1.1. *For $f \in \mathscr{S}(\mathbb{R}^N)$ the Radon transform is*

$$\mathfrak{R}f(\vec{\theta},s) = \int_{\vec{\theta}\cdot\vec{x}=s} f(\vec{x})d\vec{x} = \int_{\vec{y}\in\vec{\theta}^{\perp}} f(s\vec{\theta}+\vec{y})d\vec{y}, \tag{1.1}$$

where $s \in \mathbb{R}, \vec{\theta} \in S^{N-1}$, S^{N-1} is the unit sphere in the N-dimensional Euclidean space; $\vec{\theta}\cdot\vec{x}$ is the usual scalar product of two vectors.

When $N = 2$, the integral over a hyperplane in the Definition 1.1 can be rewritten in the simple way:

Definition 1.2. *For $f \in \mathscr{S}(\mathbb{R}^2)$ the Radon transform is*

$$\mathfrak{R}f(\alpha,s) = \int_{-\infty}^{\infty} f(s\vec{\theta}_\alpha + l\vec{\eta}_\alpha)dl = \int_{-\infty}^{\infty} f(s\cos\alpha - l\sin\alpha, s\sin\alpha + l\cos\alpha)dl, \tag{1.2}$$

where $\alpha \in [0,2\pi)$, $s \in \mathbb{R}$, $\vec{\theta}_\alpha = (\cos\alpha, \sin\alpha)$, $\vec{\eta}_\alpha = (-\sin\alpha, \cos\alpha)$.

The geometric parameters of this 2D model are shown in the fig. 1.2.

The Radon transform is well-defined for Lebesgue integrable functions for which we can calculate the integral from Eq. (1.2). For example, when the function is an indicator function $f = f_\Omega$, Ω is a

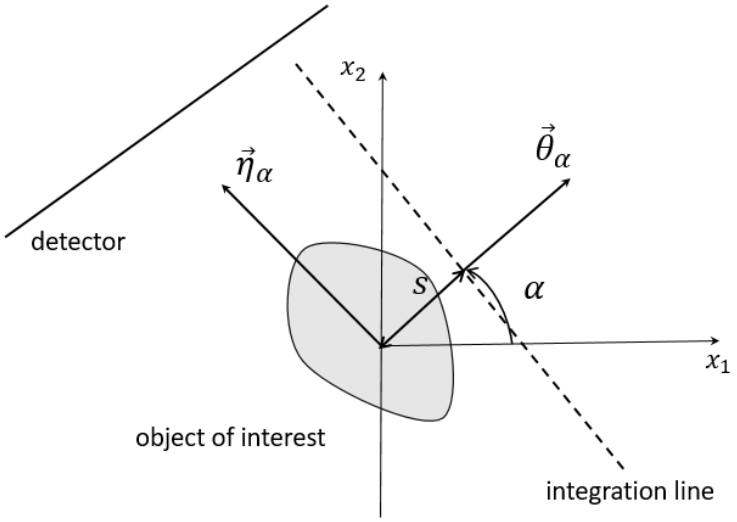

Figure 1.2: The 2D parallel geometry.

compact, its Radon transform is easy to calculate for particular cases. A simple case to calculate is the case of Ω to be the closed circular disc of radius $R > 0$ centered at the origin. Then the indicator function of Ω is given by

$$f_R(x,y) = \begin{cases} 1, & \text{if } x^2+y^2 \le R^2, \\ 0, & \text{otherwise.} \end{cases} \tag{1.3}$$

Then the Radon transform in Eq. (1.2) of this function is computed in many lecture notes in tomography:

$$\Re f_R(\alpha,s) = \begin{cases} 2\sqrt{R^2-s^2}, & \text{if } |s| \le R, \\ 0, & \text{otherwise.} \end{cases} \tag{1.4}$$

Now suppose Ω is a disc of radius $R > 0$ centered at $\vec{c} = (a,b)$ with the indicator function $f_{R,\vec{c}}$. Suppose that $f_{R,\vec{c}} := f_R(\vec{x}-\vec{c})$ is the translation of the function f_R, then

$$\begin{aligned} \Re f_{R,\vec{c}}(\alpha,s) = \int_{-\infty}^{\infty} f_R(s\vec{\theta}_\alpha + l\vec{\eta}_\alpha - \vec{c})dl &= \int_{-\infty}^{\infty} f_R((s-\vec{c}\cdot\vec{\theta}_\alpha)\vec{\theta}_\alpha + (l-\vec{c}\cdot\vec{\eta}_\alpha)\vec{\eta}_\alpha)dl \\ &= \int_{-\infty}^{\infty} f_R((s-\vec{c}\cdot\vec{\theta}_\alpha)\vec{\theta}_\alpha + t\vec{\eta}_\alpha)dt = \Re f_R(\alpha, s-\vec{c}\cdot\vec{\theta}_\alpha). \end{aligned} \tag{1.5}$$

It follows from Eq. (1.5) that for the translation of the indicator of the ball with radius R $f_{R,\vec{c}}$:

$$\Re f_{R,\vec{c}}(\alpha,s) = \begin{cases} 2\sqrt{R^2-(s-s_0(\alpha,a,b))^2}, & \text{if } |s-s_0| \le R, \\ 0, & \text{otherwise,} \end{cases} \tag{1.6}$$

where $s_0(\alpha,a,b) = a\cos\alpha + b\sin\alpha = \vec{c}\cdot\vec{\theta}_\alpha$.

Here we described known properties, other properties of the 2D Radon transform and details can be found in [Fee10].

Also, it's clear that $\mathfrak{R}$ is a linear operator: $\mathfrak{R}(\alpha g + \beta h) = \alpha \mathfrak{R} g + \beta \mathfrak{R} h$, where α and β are scalars. Thus, we can easily calculate the Radon transform of the linear combination of disk indicators on the plane.

Note that the parallel acquisition is a simplified model of geometry which is difficult to meet in real systems. The closest example might be the synchrotron as a parallel beam [Hel+05] where the source-detector distance is huge. We can also define a divergent-beam transform in any dimension. For example, for 2D we have the so-called fan-beam transform:

Definition 1.3. *Fan-beam data (or divergent projection) of the object $f \in \mathscr{S}(\mathbb{R}^2)$*

$$\mathfrak{D} f(\lambda, \vec{\zeta}) = \int_0^{+\infty} f(\vec{s}_\lambda + l\vec{\zeta}) dl, \tag{1.7}$$

where $\lambda \in \mathbb{R}$ is the trajectory parameter of the source $\vec{s}_\lambda \in \mathbb{R}^2$, $\vec{\zeta}$ defines the direction of the beam line.

In the standard theory we have that $\vec{\zeta} \in S^1$ and we have so-called non-weighted projections, where S^1 is the unit sphere in $\mathbb{R}^2$, but in our work we also consider the situation when $\vec{\zeta}$ has an arbitrary length. The last situation corresponds to so-called weighted projections in the definition of the integral transform.

For the 3D case of divergent X-rays researchers define:

Definition 1.4. *Cone-beam data (or divergent projection) of the object $f \in \mathscr{S}(\mathbb{R}^3)$*

$$\mathfrak{D} f(\vec{s}_\lambda, \vec{\zeta}) = \int_0^{+\infty} f(\vec{s}_\lambda + l\vec{\zeta}) dl, \tag{1.8}$$

where $\lambda \in \mathbb{R}$ is the trajectory parameter of the source $\vec{s}_\lambda \in \mathbb{R}^3$, $\vec{\zeta}$ - the direction of the integration line.

We should note here the same: in the standard theory we have that $\vec{\zeta} \in S^2$, where S^2 is the unit sphere in $\mathbb{R}^3$, but in our work we also consider the situation with non-unit $\vec{\zeta}$.

For the last two definitions we can parameterize differently $\vec{s}_\lambda$ and obtain different geometries. For example, let us consider the 2D fan-beam geometry from [Cla13] with sources on a line that we plan to use later. In this case, the source trajectory $\vec{s}_\lambda = (D, \lambda)^T$ is on the line $x_1 = D$, $D > 0$ (fig. 1.3). The detector is in the x_2-axis, the non-unit direction of the integration line $\vec{\zeta} = (0, y)^T - (D, \lambda)^T$, y is a parameter of the detector, then $\vec{s}_\lambda + l\vec{\zeta} = (D, \lambda)^T + l(-D, y - \lambda)^T$. Then we can rewrite that the fan-beam data of f are

$$\mathfrak{D} f(\lambda, y) = \int_0^{+\infty} f(D - lD, \lambda + ly - l\lambda) dl. \tag{1.9}$$

In this case, we usually consider f as a smooth function of compact support with support in $(D_1, D_2) \times \mathbb{R}$, $0 < D_1 < D_2 < D$, because our object should be between the detector and the source

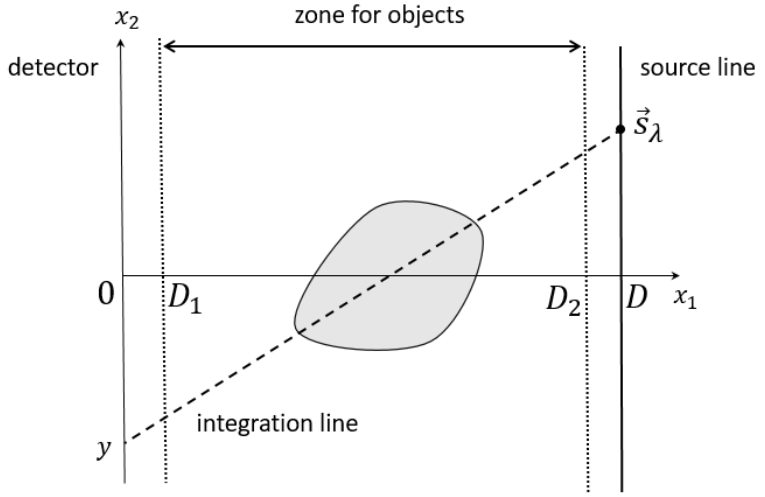

Figure 1.3: The fan-beam geometry with sources on a line.

line. But it's also possible to define the fan-beam transform of another functions, for example, for indicator functions as we did for the 2D Radon transform.

In the same way, we can construct different 3D cone-beam geometries. In the next chapters we will describe in details different modeling approaches for the 3D divergent geometry. But let us present here in details the situation from [CD13] when we have sources moving in the plane parallel to the detector plane. In the case of the fig. 1.4, the source trajectory is $\vec{s}_\lambda = (\lambda_1, \lambda_2, 0)^T$, the detector is in $x_3 = D$, the non-unit direction of the integration line $\vec{\zeta} = (u, v, D)^T - (\lambda_1, \lambda_2, 0)^T$, u and v are parameters of the detector. Then we can rewrite that the cone-beam data of f are

$$\mathfrak{D}f(\lambda_1, \lambda_2, u, v) = \int_0^{+\infty} f(\lambda_1 + l(u-\lambda_1), \lambda_2 + l(v-\lambda_2), lD)dl. \tag{1.10}$$

We are considering three families of tomographic systems with divergent geometries: CT scanners, C-arm systems and non-destructive testing systems. CT scanners have a ring shape and a translation table for patients (fig. 1.5), they are mainly intended for diagnosis. The most popular CT scanners are produced by Siemens (SOMATOM family), Phillips (for example, Incisive CT), GE (for example, Revolution CT). A C-arm connects an X-ray source and an X-ray detector by a C-shaped arm and rotates around its isocenter (fig. 1.6). A patient is usually placed at the isocenter. Projected images after C-arm rotations are collected. C-arms are intended for interventional operations. C-arms are produced by Phillips (Zenition mobile C-arm platform), Siemens (Cios family), Ziehm (Ziehm Vision RFD 3D). With non-destructive X-ray systems we want to present the case of non-medical X-ray systems with the line source trajectory, see fig. 1.7 for the example of a baggage screening system produced by Safeagle. Other systems for non-destructive testing are produced for detecting defects or the inspection of materials (see, for example, Shimadzu company, https://www.shimadzu.com/an/products/non-destructive-testing/index.html). C-arm systems are described by the 3D cone-beam model, while CT scanners and baggage screening systems can be sometimes reduced to the 2D geometry.

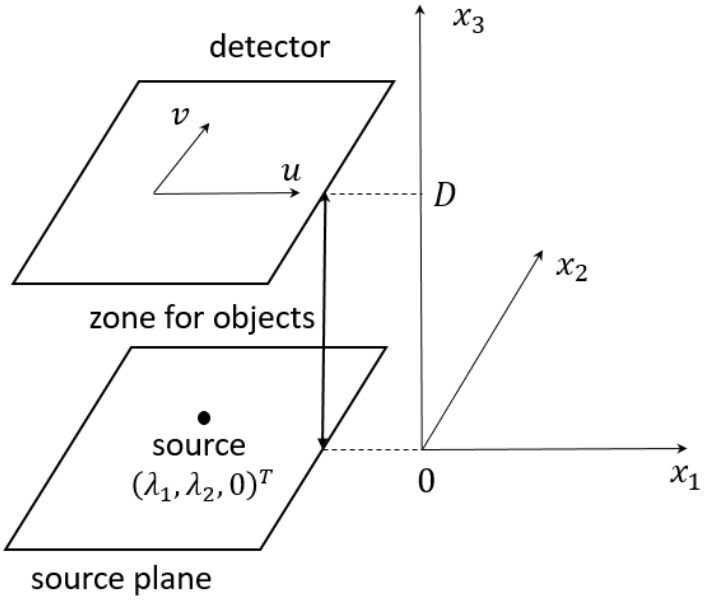

Figure 1.4: The cone-beam geometry with sources on a plane parallel to the detector.

Figure 1.5: A CT scanner SOMATOM Edge Plus by Siemens, picture from https://www.siemens-healthineers.com/fr/computed-tomography.

Note that in order to perform the reconstruction, we need to know the geometric parameters of the model of an X-ray system. Thus, a calibration of X-ray systems should be performed to estimate the geometric parameters. The reconstruction can be performed analytically or with some iterative procedures. For the 2D parallel geometry, filtered backprojection (FBP) is the most famous analytical

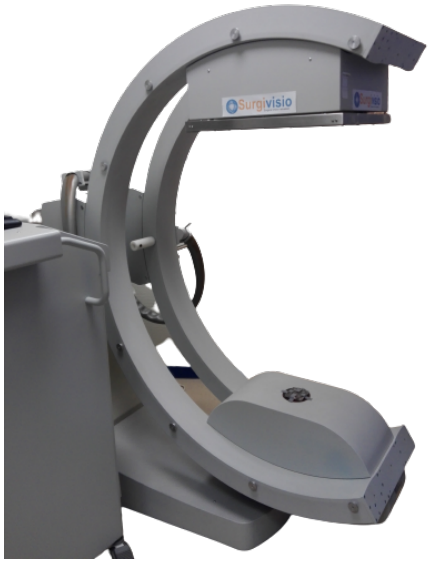

Figure 1.6: A prototype of C-arm in the TIMC lab by Surgivisio.

Figure 1.7: A baggage screening system HP-SE6040C by Safeagle, picture from https://www.safeagle.com/product-1233-x-ray-baggage-scanner-hp-se6040.html.

reconstruction method [Nat01]. The iterative or algebraic reconstruction technique (ART) for Radon's integral equations based on Kaczmarz's method can be also found in [Nat01]. The FPB inversion formula was derived also for the fan-beam geometry with a circular trajectory of the source (see, for example, [KS01]). The 2D FBP algorithm was extended to the 3D case to the Feldkamp-Davis-Kress (FDK) algorithm [FDK84]. In the case of linear source trajectories, iterative techniques are more preferred [DG03].

In this section we introduced different geometries of X-ray systems, provided mathematical definitions of the corresponding models. We also briefly discussed the reconstruction procedure and several reconstruction methods to solve this specific inverse problem to estimate a specific object from the set of its projections. The calibration should be performed before the reconstruction. We will discuss different calibration techniques in tomography and in computer vision world in the next sections.

1.3 Geometric calibration of a camera in computer vision

1.3.1 Pinhole camera model and projection matrix

In the section 1.3, we want to discuss different calibration techniques in computer vision. Later we will show the connection between the calibration process in computer vision and in tomography.

The description of a popular pinhole camera model can be found in [Stu12]. This model represents a perspective projection and helps to project the 3D world to a 2D image. An optical center and an image plane describe the pinhole model. The 3D point Q is projected along a line to the point q in the image plane (fig. 1.8).

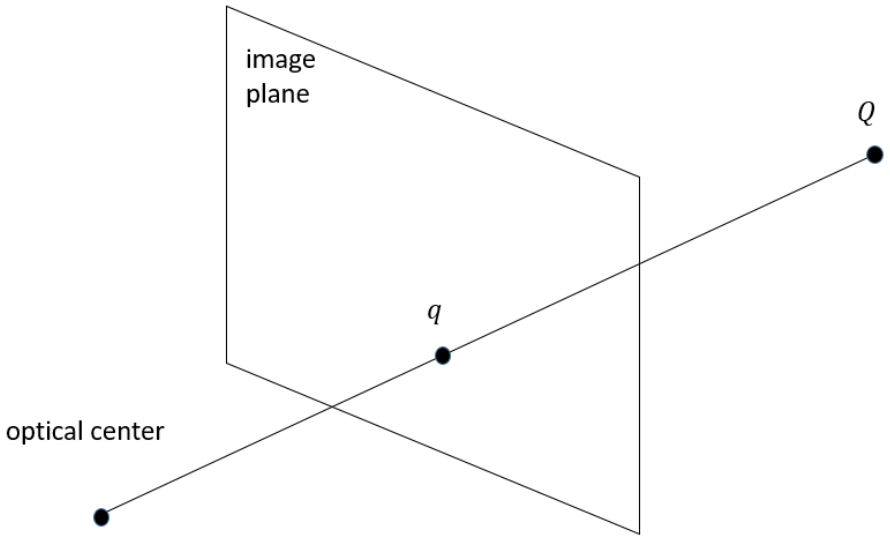

Figure 1.8: The pinhole camera model.

In order to algebraically describe this geometric camera model, we need to introduce coordinate systems and coordinates. Homogeneous coordinates with 4 coordinate systems are usually used. These coordinate systems are:

1. the fixed coordinate system in the 3D scene or the world system;

2. the 3D coordinate system of the camera with the origin in the optical center and Z-axis as the optical axis (the optical axis is the line through the optical center and perpendicular to the image plane), X and Y axes are chosen as being parallel to the image plane;

3. the 2D coordinate system for the image plane with the origin in the intersection of the optical axis and the image plane, x and y are chosen as being parallel to the X and Y axes of the camera system;

4. the 2D pixel coordinate system in pixel units.

Let us start with main equations that should connect coordinates of the 2D point q in the pixel system and the 3D point Q in the world system. Let us denote homogeneous coordinates of the optical center in the world system as $\begin{pmatrix} t \\ 1 \end{pmatrix}$, the orientation of the camera as R. We usually say that $\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$ are homogeneous coordinates for the Cartesian point $\begin{pmatrix} x \\ y \end{pmatrix}$ (for 2D, for example, but we can also define in 3D). The same Cartesian point can be represented in homogeneous coordinates as $\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$ or $\begin{pmatrix} kx \\ ky \\ k \end{pmatrix}$ for any non-zero k. Thus, a point can be represented by many homogeneous coordinates. This representation is commonly used in computer vision in order to simplify computations.

The connection between coordinates $\begin{pmatrix} X^c \\ Y^c \\ Z^c \end{pmatrix}$ of Q in the camera system and $\begin{pmatrix} X^w \\ Y^w \\ Z^w \end{pmatrix}$ in the world system can be written with a rotation matrix of the camera R and a translation of the camera t as:

$$\begin{pmatrix} X^c \\ Y^c \\ Z^c \end{pmatrix} = R \begin{pmatrix} X^w \\ Y^w \\ Z^w \end{pmatrix} - Rt. \tag{1.11}$$

Or in homogeneous coordinates:

$$\begin{pmatrix} X^c \\ Y^c \\ Z^c \\ 1 \end{pmatrix} = \begin{pmatrix} R & -Rt \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} X^w \\ Y^w \\ Z^w \\ 1 \end{pmatrix}. \tag{1.12}$$

The model provides the connection between Q in the camera system and q in the image system by knowing some parameter. Let us denote the distance between the principal point (the intersection of the optical axis and the image plane) and the optical center as the focal length f (fig. 1.9). We have

$\frac{x}{X^c} = \frac{f}{Z^c}$ (the same for y and Y^c), thus with a scalar k, where $k = Z^c$, $kx = fX^c$ and $ky = fY^c$. Thus, there is a connection:

$$q = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X^c \\ Y^c \\ Z^c \\ 1 \end{pmatrix}. \tag{1.13}$$

Here we use $\sim$, it means that the equality between vectors from the left and right sides is up to a scale.

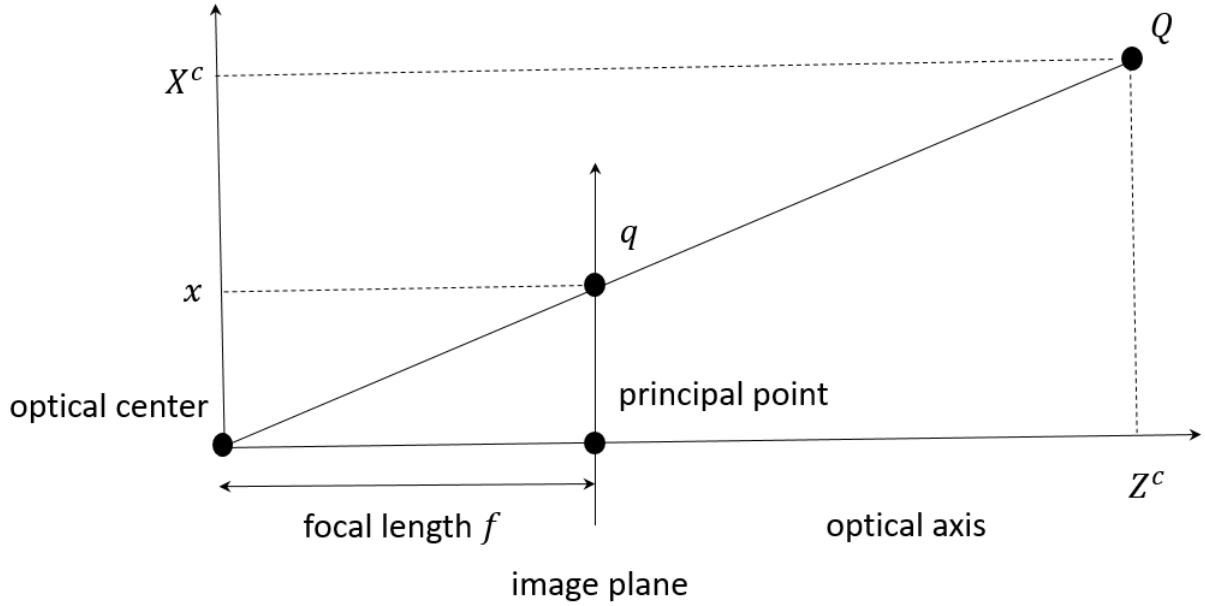

Figure 1.9: The perspective projection.

Now we can connect coordinates of q in the image system and in the pixel system. Pixels also have coordinates. The origin of the pixel coordinate system is usually the lower left corner of the image, the first coordinate axis is horizontal, the second is vertical. We can count pixels in horizontal and vertical directions starting from the origin. Thus, the transformation between image and pixel systems requires a translation and a change of unit. Let us denote the density of pixels in the u (v respectively) direction as k_u (k_v respectively). Using $\begin{pmatrix} x_0 \\ y_0 \\ 1 \end{pmatrix}$ as the translation, we can obtain coordinates of q in the pixel system:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} k_u & 0 & 0 \\ 0 & k_v & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}. \tag{1.14}$$

Thus, we can combine Eq. (1.11), (1.12), (1.13) and simplify results:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \begin{pmatrix} k_u f & 0 & k_u x_0 & 0 \\ 0 & k_v f & k_v y_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} R & -Rt \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} X^w \\ Y^w \\ Z^w \\ 1 \end{pmatrix}. \tag{1.15}$$

If we define $K = \begin{pmatrix} k_u f & 0 & k_u x_0 \\ 0 & k_v f & k_v y_0 \\ 0 & 0 & 1 \end{pmatrix}$, then

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \begin{pmatrix} K & 0 \end{pmatrix} \begin{pmatrix} R & -Rt \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} X^w \\ Y^w \\ Z^w \\ 1 \end{pmatrix} \sim P \begin{pmatrix} X^w \\ Y^w \\ Z^w \\ 1 \end{pmatrix}. \tag{1.16}$$

Definition 1.5. *A projection matrix P of the pinhole camera model is defined as* $P \sim KR \begin{pmatrix} I & -t \end{pmatrix}$. *The matrix K is called a calibration matrix. We can write this matrix as:*

$$K = \begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{1.17}$$

Four parameters α_u, α_v, u_0, v_0 *are called intrinsic parameters of the camera. Extrinsic parameters are elements of R and t.*

Usually we don't know the parameters of the model and we have to define them with a good accuracy. The calibration is a task to find the parameters of the model. It differs in the literature, but sometimes calibration means just finding the intrinsic parameters, sometimes both intrinsic and extrinsic parameters. There are two main methods to calibrate in computer vision: with markers and without markers. Markers usually mean points or small round objects added to the scene modeling points. They form the so-called calibration cage. We give few examples in the next subsections. Let us note that any object that leads to the detection of marker points can serve as a calibration cage. The example in the computer vision is the calibration with a chessboard after the detection of the chessboard corners [Hof+17].

1.3.2 Camera calibration with known markers

Theory. The simple calibration process with markers consists of taking few images of a set of points with known 3D coordinates. Let us describe the simple algorithm to solve this calibration task as

presented in [Stu12]. We want to use the correspondence between each marker in the world system and its projection. Let us consider N points (markers). For each point with index j from Eq. (1.16) we have

$$q_j = \begin{pmatrix} u_j \\ v_j \\ 1 \end{pmatrix} \sim PQ_j. \tag{1.18}$$

Here P is unknown. Firstly, we need to compute 12 elements of the projection matrix P. After intrinsic and extrinsic parameters can be found from P.

We can go from homogeneous equations to actual:

$$u_j = \frac{(PQ_j)_1}{(PQ_j)_3},\ v_j = \frac{(PQ_j)_2}{(PQ_j)_3}, \tag{1.19}$$

or

$$u_j(P_{31}Q_{j,1} + P_{32}Q_{j,2} + P_{33}Q_{j,3} + P_{34}Q_{j,4}) = P_{11}Q_{j,1} + P_{12}Q_{j,2} + P_{13}Q_{j,3} + P_{14}Q_{j,4}$$

$$v_j(P_{31}Q_{j,1} + P_{32}Q_{j,2} + P_{33}Q_{j,3} + P_{34}Q_{j,4}) = P_{21}Q_{j,1} + P_{22}Q_{j,2} + P_{23}Q_{j,3} + P_{24}Q_{j,4}.$$

In the matrix form it will be:

$$Ax = 0, \tag{1.20}$$

where

$$x = \begin{pmatrix} P_{11} \\ \vdots \\ P_{14} \\ P_{21} \\ \vdots \\ P_{24} \\ P_{31} \\ \vdots \\ P_{34} \end{pmatrix}$$

and

$$A = \begin{pmatrix} Q_{1,1} & \dots & Q_{1,4} & 0 & \dots & 0 & -u_1Q_{1,1} & \dots & -u_1Q_{1,4} \\ 0 & \dots & 0 & Q_{1,1} & \dots & Q_{1,4} & -v_1Q_{1,1} & \dots & -v_1Q_{1,4} \\ & \vdots & & & \vdots & & & \vdots & \\ & \vdots & & & \vdots & & & \vdots & \\ Q_{N,1} & \dots & Q_{N,4} & 0 & \dots & 0 & -u_NQ_{N,1} & \dots & -u_NQ_{N,4} \\ 0 & \dots & 0 & Q_{N,1} & \dots & Q_{N,4} & -v_NQ_{N,1} & \dots & -v_NQ_{N,4} \end{pmatrix}.$$

The vector x contains 12 elements of P, from each point correspondence we have 2 equations. Thus, 6 correspondences should be sufficient to find P. The calibration cage with the markers can be

degenerate or singular, meaning that the identification problem has a non-unique solution. Moreover, data could contain a noise. In this case, there is no exact solution of the system Eq. (1.20). Thus, researchers propose to compute a solution in the least squares sense. We also need to avoid the trivial solution $x = 0$ of the system Eq. (1.20). So, we need to add a constraint on x, for example $||x||_2 = 1$. Thus, we need to solve the following problem:

$$\min_x \sum_{i=1}^{2N} ((Ax)_i)^2 \text{ s.t. } ||x||_2 = 1. \tag{1.21}$$

It's possible to solve it with a singular value decomposition (SVD) of the matrix A. Let us recall SVD.

Definition 1.6. *The singular value decomposition (SVD) of an $m \times k$ real matrix A is a factorization of the form $U\Sigma V^T$, where U is an $m \times m$ orthogonal matrix, Σ is an $m \times k$ rectangular diagonal matrix with singular values $\Sigma_{ii} \geq 0$ on the diagonal, V is an $k \times k$ orthogonal matrix.*

We can rewrite Eq. (1.21) as

$$\min_x \sum_{i=1}^{2N} ((Ax)_i)^2 = \min_x x^T A^T Ax = \min_x x^T V\Sigma^T U^T U\Sigma V^T x = \min_x x^T V\Sigma^T \Sigma V^T x.$$

If we denote $y = V^T x$, $S = \Sigma^T \Sigma$, then $y^T y = x^T VV^T x = x^T x$ and we need to solve

$$\min_y y^T Sy \text{ s.t. } y^T y = 1. \tag{1.22}$$

Usually, the number of zero singular values (and zero diagonal values of S) is equal to 1 (when the calibration cage isn't degenerate). We can solve the optimization task in the following way: since the diagonal matrix S consists of the squares of singular values from the biggest to smallest, we need to find and take the smallest non-zero singular value. Then in the usual case with non-degenerate calibration cage we take $\hat{y} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \end{pmatrix}$. After we can calculate the corresponding $\hat{x} = V\hat{y}$.

The problematic situations appear, for example, when all markers are in a plane, then we have more than one non-zero vector in the null-space of A. In this case, we need to change initial points (markers) in order to have a unique solution of Eq. (1.21).

Once the projection matrix P is found, we can compute intrinsic and extrinsic parameters or elements of K, R and t. Suppose that $\bar{P}$ is the 3×3 sub-matrix of P of its first 3 columns. Then

$$\bar{P} \sim KR. \tag{1.23}$$

From $\bar{P}\bar{P}^T \sim KRR^TK^T$ we have

$$\bar{P}\bar{P}^T \sim KK^T. \tag{1.24}$$

We can use the Cholesky decomposition $\bar{P}\bar{P}^T \sim BB^T$, where B is an upper triangular matrix. Let recall that with the Cholesky decomposition each symmetric and positive definite matrix of size $m \times m$ can be decomposed as BB^T, where B is upper triangular of the size $m \times m$. K is also upper triangular by construction. Since the element K_{33} equals 1, thus all elements of B should be divided by B_{33}. Once this is done, we can extract K.

For the computation of the extrinsic parameters from $P \sim \begin{pmatrix} KR & -KRt \end{pmatrix}$ we have

$$K^{-1}P \sim \begin{pmatrix} R & -Rt \end{pmatrix}. \tag{1.25}$$

Denote $C \sim R$ be the 3×3 sub-matrix of the first three columns of $K^{-1}P$, then

$$\lambda C = R. \tag{1.26}$$

Then $\det(\lambda C) = \lambda^3 \det C = \det R = 1$. So, $\lambda = \sqrt[3]{\frac{1}{\det C}}$. Thus, we can simply compute R from C and then t from

$$\lambda K^{-1}P = \begin{pmatrix} R & -Rt \end{pmatrix}. \tag{1.27}$$

Numerical experiments. Let us illustrate the first step of the algorithm with our simulation in Python. We didn't use noise, but we took 7 markers (as in the figure 1.10), thus we can just search a solution in the least squares sense. We took true values:

- $K_{\text{true}} = \begin{pmatrix} 1 & 0 & 3 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{pmatrix}$,
- $t_{\text{true}} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$,
- R_{true} is a rotation matrix of 45 degrees around Z-axis,
- then $P_{\text{true}} = \begin{pmatrix} 0.707 & -0.707 & 3 & -0.707 \\ 0.707 & 0.707 & 3 & -0.707 \\ 0 & 0 & 1 & 0 \end{pmatrix}$.

We simulated projections of the markers, thus as an input of our algorithm we have the coordinates of the markers in the world system and coordinates of the projections of markers in the pixel system.

We obtained:

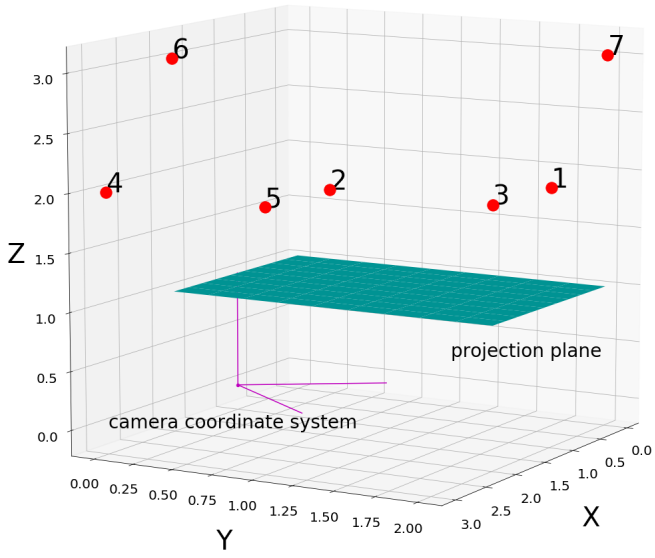

Figure 1.10: The image plane, camera coordinate system and non-coplanar markers used in our numerical simulation.

- $\hat{P} = \begin{pmatrix} -1.51 \times 10^{-1} & 1.51 \times 10^{-1} & -6.40 \times 10^{-1} & 1.51 \times 10^{-1} \\ -1.51 \times 10^{-1} & -1.51 \times 10^{-1} & -6.40 \times 10^{-1} & 1.51 \times 10^{-1} \\ -1.11 \times 10^{-16} & 6.11 \times 10^{-16} & -2.13 \times 10^{-1} & 5.76 \times 10^{-16} \end{pmatrix}$ as the solution to the optimization problem Eq. (1.21);
- since $\hat{P}/P_{33} = \begin{pmatrix} 7.07 \times 10^{-1} & -7.07 \times 10^{-1} & 3.00 & -7.07 \times 10^{-1} \\ 7.07 \times 10^{-1} & 7.07 \times 10^{-1} & 3.00 & -7.07 \times 10^{-1} \\ 2.86 \times 10^{-15} & -6.01 \times 10^{-17} & 1.00 & -1.61 \times 10^{-14} \end{pmatrix}$, then $\hat{P}$ equals to P_{true} up to a scale.

During our numerical simulation for the second step we calculated very precisely the calibration parameters from $\hat{P}$ obtained after the first step of the algorithm, see the table 1.1.

Elements of K	**Elements of R**	**Elements of t**
5.88×10^{-15}	2.86×10^{-15}	9.33×10^{-15}

Table 1.1: Maximum absolute errors (maximum absolute difference between estimated values and true values) for calibration parameters.

In the realistic setting, we usually know projections just approximately, for example, after the detection of singularities of the image. In this situation, we will have more significant errors in the calibration algorithm. Nevertheless, if we know coordinates of the markers in the world system, we can use this

algorithm, since it's easy to implement. Disadvantages are that we need to design a calibration cage carefully and we need to disturb scene by introducing this calibration cage of additional objects.

1.3.3 Calibration with markers with unknown geometry: bundle adjustment problem

Although in applications it's important to know only parameters of the system, sometimes we may face the mathematical problem to find two sets of unknowns: unknown 3D points $\hat{Q}_j$, $j = 1,\ldots,N_{\text{markers}}$, and projection matrices for different cameras $\hat{P}^i$, $i = 1,\ldots,N_{\text{projections}}$, which project exactly to known detected image points q^i_j. In order to do that, we minimize the distance between the projected point and the detected image point for every view, i.e.

$$\min_x D(x) := \min_{P^i,Q_j} \sum_{i,j} d(P^i Q_j, q^i_j)^2, \tag{1.28}$$

where $d(x,y)$ is any geometric image distance between the homogeneous points x and y. This is the general formulation of the so-called bundle adjustment (BA) problem.

In [Tri+00] the authors summed up basic optimization methods to solve the BA problem. We consider a differentiable cost function $D(x)$ over parameters x. Usually, from the initial estimate x_0 we want to find a displacement δx which locally minimizes $D(x)$. We can replace $D(x)$ by the following quadratic local model based on the Taylor expansion:

$$D(x+\delta x) \approx D(x) + g^T \delta x + \frac{1}{2}\delta x^T H \delta x, \tag{1.29}$$

where g is a gradient vector of D at x and H is a Hessian matrix at x.

According to [Tri+00], it's possible to use different methods for the optimization of the local model:

- the Newton's method with the estimation of δx as $-H^{-1}g$ for each iteration. The inversion leads to the high cost per iteration. Also, we need to calculate the matrix H with second derivatives. Moreover, it could be hard to achieve a convergence starting from the far initial point.

- damped Newton's methods which solve the regularized system

$$(H + \lambda W)\delta x = -g, \tag{1.30}$$

  where λ is a weighting factor and W is a positive definite weight matrix. This is a basis for trust region methods where we limit δx and introduce some constrains by this. Levenberg-Marquardt method is a famous algorithm from this family.

- if we consider the weighted sum of squared error with the particular cost function $D(x) = \frac{1}{2}\Delta q(x)^T W \Delta q(x)$, then we can apply the Gauss-Newton method. Here W is a symmetric positive

definite matrix, $\Delta q(x)$ is the vector of differences between projected by the model and detected image points. Then we can approximately find $H \approx J^T W J$, where $J = \frac{d\Delta q}{dx}$. In this case, we need to solve normal equations

$$J^T W J \delta x = -J^T W \Delta q. \tag{1.31}$$

- first order methods as conjugate gradient method, BFGS (which builds an explicit approximation of H) or limited-memory BFGS (which maintains only a low-rank approximation of H).

The bundle adjustment is still widely used in the literature: to identify the trajectory of the camera in the real-time tracking in [ESN06], to calibrate cameras described by models different from the pinhole camera model in [Urb+17], to take into account other sensors than cameras in the joint calibration by the bundle adjustment in robotics in [PKB14], to calibrate the Microsoft Kinect system in [CL13], to move from the bundle adjustment to photometric bundle adjustment with the minimization of some photometric energy functional and the estimation of the shape, camera parameters and the scene texture in [DP14].

1.3.4 Camera calibration without markers: some approaches

In computer vision, there is a division of image-based calibration methods into two groups: with markers and without markers. The last group is usually called a group of self-calibration algorithms. Usually researchers replace markers with interest points. In order to compute interest points in each image we can use special algorithms for corner detection or singularity detection. After this, it's possible to define a set of matches in different views of the interest points based on proximity and similarity of their intensity in neighbourhoods [HZ04]. Let us describe few methods to calibrate without markers.

Homography based algorithm. Let us start with the simple calibration without markers from [Stu12]. Suppose that we don't have a calibration object and have two images from the same viewpoint, but with different camera orientations R_1 and R_2, where R_1 and R_2 are different rotations of the camera coordinate system with respect to the world coordinate system. We can assume that $t = 0$. With the calibration matrix K we can compute projection matrices as $P_1 \sim \begin{pmatrix} KR_1 & 0 \end{pmatrix}$ and $P_2 \sim \begin{pmatrix} KR_2 & 0 \end{pmatrix}$. If we take a 3D point $Q = \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix}$, then projections $q_1 \sim KR_1 \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$ and $q_2 \sim KR_2 \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$ or $R_1^T K^{-1} q_1 \sim \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$ and $R_2^T K^{-1} q_2 \sim \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$. From that we have

$$R_2^T K^{-1} q_2 \sim R_1^T K^{-1} q_1, \tag{1.32}$$

$$q_2 \sim KR_2R_1^TK^{-1}q_1. \tag{1.33}$$

Definition 1.7. *The transformation H which links two projections of the same 3D point is called a homography:*

$$H \sim KR_2R_1^TK^{-1}. \tag{1.34}$$

If we know at least two homographies, then we can find intrinsic calibration parameters from this knowledge. We don't need to introduce calibration objects into the scene, but we need to know homographies. Note that the simplest way to estimate homographies is through point correspondences in different views, but for that we don't need to use real calibration objects, we can only use interest points to replace real marker points. For intrinsic calibration parameters:

1. after few manipulations we have $HKK^TH^T \sim KK^T$;
2. let us denote $A = KK^T$, then $HAH^T \sim A$;
3. we calculate a scalar λ from the equality of two determinants: $\det((\lambda H)A(\lambda H^T)) = \det A$ or $\lambda^6(\det H)^2\det A = \det A$, then $\lambda = \sqrt[3]{\frac{1}{\det H}}$;
4. with $\lambda H = \bar{H}$
$$\bar{H}A\bar{H}^T = A; \tag{1.35}$$
5. we need to solve this linear equation in the elements of A (for two or more homographies) and extract from A the calibration matrix K via the Cholesky decomposition of A.

This method is simple to implement, but we need to know at least two homographies.

Classification of Hartley and Zisserman. Let us start with basic definitions from [HZ04]. We introduce different classes of transformations of 3D space which form a hierarchy.

Definition 1.8. *A projective transformation is a transformation defined by the matrix* $\begin{pmatrix} A & t \\ v^T & k \end{pmatrix}$, *where A is an invertible* 3×3 *matrix, v is a general* 3*-vector.*

Definition 1.9. *An affine transformation is a transformation defined by the matrix* $\begin{pmatrix} A & t \\ 0^T & 1 \end{pmatrix}$, *where A is an invertible* 3×3 *matrix.*

Definition 1.10. *A similarity transformation is a transformation defined by the matrix* $\begin{pmatrix} \sigma R & t \\ 0^T & 1 \end{pmatrix}$, *where R is a* 3×3 *rotation matrix and* $\sigma \neq 0$.

Definition 1.11. *An Euclidean transformation is a transformation defined by the matrix* $\begin{pmatrix} R & t \\ 0^T & 1 \end{pmatrix}$, *where R is a* 3×3 *rotation matrix.*

From [HZ04] we know that we have a solution to the BA problem up to a projective transformation. The class of such solutions gives us the so-called projective reconstruction of the scene. Sometimes we can have the so-called metric reconstruction of the scene. For the metric reconstruction we have a solution up to a similarity transformation.

Some sophisticated self-calibration methods were summarized in one of the main books for specialists in computer vision [HZ04]. Usually, it's easy to receive a projective reconstruction P^i and Q_j (for example, from bundle adjustment) from the set of interest points. The rectifying homography H is a transformation for which we could obtain a metric reconstruction P^iH, $H^{-1}Q_j$ from the projective reconstruction. Since similarity transformations preserve matrix K, with the metric reconstruction we receive true intrinsic calibration parameters. According to the authors, there are two approaches to camera calibration without markers: those who directly determine the rectifying homography H and stratified.

Let us introduce two examples of algorithms for calibration without markers from two classes. We define the metric reconstruction $P_M^i \sim K^i \begin{pmatrix} R^i & t^i \end{pmatrix}$ for the view number i, $N_{\text{projections}}$ views in total, $P_M^i = P^iH$. If we choose the coordinate system to be connected with the first camera, then $P_M^1 \sim K^1 \begin{pmatrix} I & 0 \end{pmatrix}$, $P^1 \sim \begin{pmatrix} I & 0 \end{pmatrix}$. We need to find the rectifying homography H. Suppose that

$$H = \begin{pmatrix} A & t \\ v^T & k \end{pmatrix}. \tag{1.36}$$

From $P_M^1 = P^1H$ we receive $\begin{pmatrix} K^1 & 0 \end{pmatrix} \sim \begin{pmatrix} I & 0 \end{pmatrix} H$, then $A = K^1$, $t = 0$. Because H should be non-singular, we can consider $k = 1$. Then from $A = K^1$, $t = 0$, $k = 1$ we have

$$H = \begin{pmatrix} K^1 & 0 \\ v^T & 1 \end{pmatrix}. \tag{1.37}$$

Let us define

$$\pi_\infty := H^{-T} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} (K^1)^{-T} & -(K^1)^{-T}v \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -(K^1)^{-T}v \\ 1 \end{pmatrix} = \begin{pmatrix} p \\ 1 \end{pmatrix}, \tag{1.38}$$

where $p = -(K^1)^{-T}v$. Then

$$H = \begin{pmatrix} K^1 & 0 \\ -p^TK^1 & 1 \end{pmatrix}, \tag{1.39}$$

where p consists of coordinates of π_∞ defined in Eq. (1.38). If we know 3 parameters p and 5 parameters K^1 (for the camera with skew, i.e. with $K_{12} \neq 0$), we can find H. If $P^i \sim \begin{pmatrix} A^i & a^i \end{pmatrix}$, then $P^iH \sim \begin{pmatrix} A^i & a^i \end{pmatrix} H \sim K^i \begin{pmatrix} R^i & t^i \end{pmatrix}$ or

$$K^iR^i \sim (A^i - a^ip^T)K^1. \tag{1.40}$$

From $R^i(R^i)^T = I$

$$K^i(K^i)^T \sim (A^i - a^i p^T)K^1(K^1)^T(A^i - a^i p^T)^T. \tag{1.41}$$

Let us replace $K^i(K^i)^T = \omega^{*i}$, then

$$\omega^{*i} \sim (A^i - a^i p^T)\omega^{*1}(A^i - a^i p^T)^T \tag{1.42}$$

with unknown ω^{*i} and p (or it's the same to say with unknown K^i and π_∞). Usually, researchers use constraints on K^i to produce equations on p, ω^{*i} to find them.

An example of the algorithm from the first class. This algorithm is applicable for the situation when we have the same intrinsic parameters or $K^i = K$. Since each side of Eq. (1.41) is a 3×3 symmetric matrix and the equation is homogeneous, each view $i > 1$ gives 5 constraints on K. So, we want $5(N_{\text{projections}} - 1) \geq 8$ or $N_{\text{projections}} \geq 3$ to find eight parameters p and K.

An example of the algorithm from the second class. Let us describe the logical steps of a stratified algorithm also for the case $K^i = K$. We search for H in few steps: we firstly find π_∞, then K. It could be shown that each view pair gives a quartic equation in the coordinates of π_∞ or a modulus constraint. It looks like

$$f_3 f_1^3 = f_2^3, \tag{1.43}$$

$$\det(\lambda I - A^i + a^i p^T) = \lambda^3 - f_1\lambda^2 + f_2\lambda - f_3, \tag{1.44}$$

where the elements of p appear only linearly in the determinant and linearly in each of f_1, f_2, f_3. Thus, it's a quartic equation. The authors told that modulus constraints could be combined with scene information, it helps to kill ambiguity in the solution of Eq. (1.43). Thus, we could obtain p.

We can rewrite Eq. (1.42) with $H^i_\infty = A^i - a^i p^T$:

$$\omega^{*i} \sim H^i_\infty \omega^{*1}(H^i_\infty)^T. \tag{1.45}$$

It's interesting that H^i_∞ represents the homography from a camera $\begin{pmatrix} I & 0 \end{pmatrix}$ (the first camera) to the camera $\begin{pmatrix} A^i & a^i \end{pmatrix}$. For the case of the same intrinsic parameters we can rewrite Eq. (1.45) as

$$\omega^* \sim H^i_\infty \omega^* (H^i_\infty)^T. \tag{1.46}$$

We can normalize H^i_∞ to have $\det(H^i_\infty) = 1$, then we have an equality instead of $\sim$, for that we need to use the similar reasoning as we used to produce Eq. (1.35). Thus, we need to solve a linear system $Ac = 0$ with a 6-vector c formed by elements of ω^*. Some special tricks and a sufficient number of views help to avoid ambiguity in the solution. Thus, from the Cholesky decomposition of the solution ω^* we can obtain K.

Besides bundle adjustment, there are other ways to obtain a projective reconstruction. As an example, one popular method is based on so-called fundamental matrices and is connected with the epipolar geometry [HZ04].

In this section we presented the literature review of the classical calibration methods in computer vision. We can perform the calibration with markers and without markers, when markers are replaced by interest points. We detailed different methods that can be transferred to tomographic systems.

1.4 Geometric calibration of acquisition systems in tomography

1.4.1 Geometric calibration with markers

By analogy with computer vision, researchers in tomography consider image-based calibration methods with markers and without markers (see [Les18] as an example of such division). For the first group, 3D coordinates of marker centers could be known in the world coordinate system or unknown as in the BA problem. In our work in tomography, we bring together for X-ray systems the second subgroup of algorithms where we don't know the geometry of markers and algorithms without markers to self-calibration algorithms.

If we use markers, then we need to design the special calibration cage introduced in the scene only for the calibration. This can add the work to medical practitioners, but also disturb patients during the acquisition, because some X-ray systems require online calibration when the source and detector trajectories aren't reproducible. Note that the calibration can be also an offline procedure when the source and detector trajectories are reproducible, in this case we shouldn't calibrate each acquisition, we can only do it once per some period (a week or a month). Note also that the calibration cage usually should be very precise. Often this is only possible during the experimental studies.

In marker-based calibration methods the centers of spherical markers are usually considered as points and projection properties help to derive calibration procedures. Very often self-calibration procedures end up in this case with numerical nonlinear optimizations as was in the case of bundle adjustment in computer vision. For example, in [WT04] a single point source and few scans helped to end up with the optimization procedure to define unknown geometrical parameters such that the focal distance, the rotation radius of the detector, coordinates of the focal point and the center of the rotation on the detector. The authors used non-classical description of the fan-beam geometry. In their algorithm they need to know the difference in the radius for two scans. In [RGG94] only extrinsic parameters were calibrated with nonlinear minimization in the case of the circular cone-beam geometry. The bundle adjustment with the numerical optimization is used also in the cone-beam geometry, the example can be found in [LW12].

In many works calibration cages with special properties were designed in addition, these properties allow to derive calibration procedures. In [Cho+05] the fact that the projection of the circular pattern of round markers lies on an ellipse was used to estimate all unknown geometrical parameters. The same principle was used for the analytical partial calibration in the previous work [Noo+00]. The special calibration cage was proposed in [MCN09] in order to build the analytical procedure for the full

calibration.

More analytical approaches with markers recently appeared in the literature in [TSH19] for the cone-beam geometry with circular source trajectory and in [Jon18] for fan-beam and cone-beam geometries with general source trajectories. In both articles the authors used non-standard description of geometry and designed calibration cages. In [TSH19] the method required two scans acquired after the precise rotation of the calibration cage by 180 degrees. In [Jon18] proper calibration cages were discussed and it was shown that non-standard acquisition parameters can be identified with linear systems of equations. To perform the conversion to standard parameters, nonlinear inversions must be performed which can cause problems in the noisy situations when we don't have an exact detection of the markers on projected images.

The extensive overview of calibration methods with markers can be found in [YLC16].

1.4.2 Calibration with DCC

Now we describe self-calibration techniques without markers based on data consistency conditions (DCC) and introduce the problem of truncations. The summary of the articles with DCC and corresponding calibration methods is also presented in the table 1.2.

Let start with the case of the 2D Radon transform of a function. It's known that the projection data should satisfy special equations or data consistency conditions. Data consistency conditions or range conditions in this case are known in the literature as Helgason-Ludwig consistency conditions (HLCC) [Hel65; Lud66]:

Theorem 1.1. *A function p is the Radon transform of $f \in \mathscr{S}(\mathbb{R}^2)$ if and only if:*

1. *$p \in \mathscr{S}(S^1 \times \mathbb{R})$, where S^1 is a unit sphere,*
2. *p is even: $\forall s \in \mathbb{R}, \forall \vec{\theta}_\alpha \in S^1 \; p(-\vec{\theta}_\alpha, -s) = p(\vec{\theta}_\alpha, s)$,*
3. *for $k = 0, 1, 2, \ldots$, for all $\vec{\theta}_\alpha \in S^1$ we have the moment conditions:*

$$\int_{-\infty}^{+\infty} p(\vec{\theta}_\alpha, s) s^k ds = \mathscr{P}_k(\vec{\theta}_\alpha), \tag{1.47}$$

$\mathscr{P}_k(\vec{\theta}_\alpha)$ is a homogeneous polynomial of degree at most k in the coordinates of $\vec{\theta}_\alpha$.

Moreover, $p(\vec{\theta}_\alpha, s) = 0$ for all $|s| > a$ and $\vec{\theta}_\alpha \in S^1 \Leftrightarrow f(\vec{x}) = 0$ for $|\vec{x}| > a$.

Note that we want to use these DCC in the 2D case, but they were formulated for an arbitrary dimension.

The notion of truncation is very important in tomography. Obviously, in realistic settings, objects of interest during the acquisition have compact supports. Usually we have a set of source positions (infinitely far for parallel geometry) and for each source or each projection the set of integration lines

Geometry	**Papers with DCC**	**Papers with calibration methods**	**For truncated data**	**Analytical or with optimization**
2D parallel	HLCC: [Hel65; Lud66]	Basu and Bresler: [BB00b; BB00a], Panetta: [Pan+08]	no	optimization
2D parallel	HLCC: [Hel65; Lud66]	Desbat and Spencer: [DS14]	no	analytical
fan-beam on a line	DCC by Clackdoyle: [Cla13]	Nguyen: [NDC20]	no	analytical
fan-beam on a circle	DCC by Yu: [Yu+06; YW07]	no	no	-
fan-beam on a circle	DCC by Clackdoyle: [CD15]	no	yes	-
fan-beam	DCC by Yu: [Yu+15]	no	yes	-
cone-beam	ECC by Aichert: [Aic+15], DCC by Lesaint: [Les+17], DCC by Panetta: [Pan+08]	same	no	optimization
cone-beam on a line	DCC by Nguyen: [NDC20]	same	no	analytical
cone-beam on a circle in the plane perpendicular to the detector	DCC by Clackdoyle: [Cla+16]	no	no	-
cone-beam on a circle in the plane parallel to the detector	DCC by Nguyen: [Ngu21]	no	no	-
cone-beam arbitrary in the plane perpendicular to the detector	DCC by Nguyen: [Ngu21]	no	no	-
cone-beam arbitrary in the plane parallel to the detector	DCC by Clackdoyle and Desbat: [CD13]	no	no	-

Table 1.2: DCC and corresponding calibration methods.

to measure. For each source we associate a field of view, the subspace visible from the source position. The global field of view is the intersection for all sources of such fields of view. We say that the projection is truncated if there is an associated integration line which intersects the support of the object, but isn't measured. It means that in this case not all non-zero projections are known. Then, for example, in the 2D parallel case we can't correctly calculate moments (integrals from $-\infty$ to $+\infty$ over s of the projection data) in the moment conditions Eq. (1.47) and can't exploit HLCC.

In [BB00b; BB00a] Basu and Bresler used HLCC in order to derive the calibration procedure without markers and based on non-truncated data. They estimated the set of unknown projection angles α and the set of shifts s_α of the lattice of s for each projection angle with the numerical optimization technique. In [Pan+08] a partial self-calibration with unknown projection angles was performed with periodicity relations of HLCC and nonlinear optimization. Iterative nonlinear optimization techniques are a priory time-consuming, and it's usually hard to answer which solution from the class of all possible solutions we receive. To overcome this, a partial self-calibration (projection angles were known) with a closed form solution was derived from HLCC and proposed in [DS14]. The disadvantage of these methods using HLCC is the requirement to have non-truncated data which isn't usually satisfied in X-ray systems where a patient isn't completely in the field of view.

The modern data consistency conditions for non-truncated fan-beam data with sources on a line mentioned in the section 1.2 were proposed in [Cla13]:

Theorem 1.2. *Define*

$$\mathscr{P}_k(\lambda) = \int_{-\infty}^{+\infty} g(\lambda, y) y^k dy \tag{1.48}$$

for all $k = 0, 1, 2, \ldots$ The function $\mathscr{P}_k(\lambda)$ is a polynomial in λ of degree k if an only if $g = \mathfrak{D} f$ for some smooth function of compact support f.

The corresponding self-calibration algorithm with a closed form solution was proposed in [NDC20]. Note that this algorithm requires three known source positions and non-truncated data. Fan-beam data consistency conditions applied to the motion correction in the case of the circular trajectory were presented in [Yu+06; YW07]. Some DCC can handle the problem of truncated data, they were presented for 2D parallel geometry and fan-beam geometry with sources on a circle in [CD15] and for the fan-beam geometry with an arbitrary source trajectory - in [Yu+15]. It's difficult to derive an analytical solution to the calibration problem from these conditions.

The good overview of DCC in the case of 3D divergent geometry can be found in [Ngu21]. The author states that there are three types of DCC for the cone-beam geometry: derived from the John's equation, polynomial-type consistency conditions and derived for pairs of projections. Note that for the cone-beam geometry different DCC are also widely used for non-truncated data and help to build calibration procedures with nonlinear optimization, for example, in [Aic+15; Les+17; Pan+08]. Nevertheless, we are mostly interested in the polynomial-type consistency conditions. In [NDC20] such consistency conditions helped to derive an analytical calibration procedure for non-truncated data for

the cone-beam geometry with sources on a line. As for the fan-beam case, some known additional information is needed for such calibration procedure.

The polynomial-type consistency conditions for the cone-beam case for the popular circular source trajectory in the plane perpendicular to the detector plane were presented in [Cla+16], for the popular circular source trajectory in the plane parallel to the detector plane were presented in [Ngu21]. The generalisation of the first DCC to the case of the arbitrary source trajectory in the plane was described in [Ngu21], the generalisation of the second DCC to the case of the arbitrary source trajectory in the plane - in [CD13]. The DCC from [CD13] that we want to use:

Theorem 1.3. *Define*

$$J_k(\lambda_1,\lambda_2,U,V)=\int_{-\infty}^{+\infty} g(\lambda_1,\lambda_2,u,v)(uU+vV)^k dudv \tag{1.49}$$

for all $k=0,1,2,...$ *Then* $J_k(\lambda_1,\lambda_2,U,V)=\mathscr{P}_k(U,V,-\lambda_1U-\lambda_2V)$, $\mathscr{P}_k(U,V,W)$ *is a homogeneous polynomial of degree* k *and* $g(\lambda_1,\lambda_2,\cdot,\cdot)$ *has a compact support for all* (λ_1,λ_2) *if and only if* $g=\mathfrak{D}f$ *with compactly supported* f *in* $z>0$.

The source of truncations could be different, but usually we have truncations because the size of the detector isn't sufficient to cover the body. In 3D, if the patient body is too long compare to the field of view, the detector height isn't sufficient, we have the axial truncation. If the length of the detector isn't sufficient, then we have the transverse truncation (see, for example, [Gin22]). The situation with the axial truncation is common in practice, because the body of the patient is always too long, but it's less critical, because for the reconstruction the FDK algorithm can work with this type of truncation.

1.4.3 The idea of the hybridization

Our work is mostly dedicated to hybrid algorithms, i.e. methods that combine the use of markers and DCC. We were inspired by the hybrid algorithm and the idea to calibrate with truncated cone-beam data presented by Unberath and his colleagues in [Unb+17]. The authors introduced an algorithm to calibrate a cone-beam system with truncated cardiac data and epipolar consistency conditions (ECC). ECC is data consistency conditions for the cone-beam transform derived by Aichert in [Aic+15; Aic+16; Aic19]. ECC was successfully applied for different tasks: calibration and motion compensation, but also later for beam-hardening reduction in [Wur+17; Wur+18; Wur+19] and scatter correction in [Hof+18]. With the beam hardening only higher energy photons contribute to the projection image. This could cause artifacts, but parameters of a polynomial model of the beam hardening could be estimated with ECC. For the scatter correction, we need to estimate scatter components which also cause artifacts such as blurring of images. It's also possible to estimate the scatter components with ECC. Moreover, a new approach to combine ECC with a convolutional neural network for the motion compensation was proposed recently in [Pre+19]. Of course, each time with pure ECC the researchers work with non-truncated data.

For cardiac data, doctors don't acquire the whole patient body, thus only truncated data are available. Data are truncated in cardiology imaging because the chest of a patient is large and the doctor spare the dose. However, we could add a contrast to vessels and assure vessels to be in the field of view. Thus, the main idea of the algorithm from [Unb+17] is to use contrasted vessels as markers and apply known ECC to non-truncated images of the contrasted vessels. For that, the authors need to separate the contrasted vessels from the background on projected images. The Unberath's article suffers from the lack of theoretical explanations, especially for the fact that the projected images with the contrasted vessels extracted by their method with non-linear filtering are in the range of the cone-beam transform (only in this case we can apply ECC).

1.4.4 Description of the problem

We started our research with the aim to improve the mathematical justification of the method from [Unb+17] and to generalize this idea to other images (not just cardiac) and other geometries. Let us illustrate and describe in more details the problem with the application of DCC to truncated data.

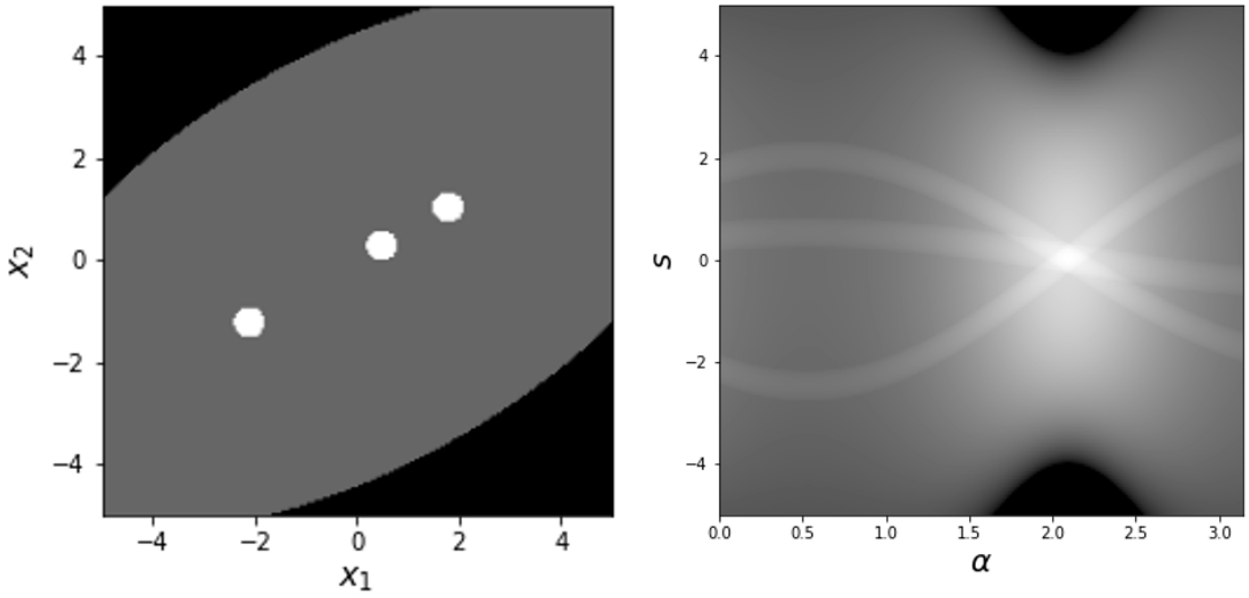

Figure 1.11: Left: truncated object and round markers in $\mathbb{R}^2$, right: their 2D Radon transform.

As we told, there is the huge group of self-calibration methods for X-ray systems with different geometries in tomography based on different DCC. The huge disadvantage of these methods is that they usually can't work when the support of f isn't in the field of view of the system. In this case, we need to work with truncated data. It's possible to see an example of such situation for the 2D Radon transform in the fig. 1.11. The patient body (ellipse) isn't in the field of view completely, so we don't have all non-zero projection data. Thus, we can't exploit the moment conditions Eq. (1.47). The same

problem we have with the integrals in the moment conditions Eq. (1.48) and Eq. (1.49) for the fan-beam and cone-beam geometries.

However, in the fig. 1.11 all the round markers are in the field of view and we can have all non-zero projection information about markers. Thus, in our work we want to build new calibration methods for the case when the whole object is truncated, but markers are not.

In this section we explained the problem of truncated data and the need to use non-truncated markers in the calibration process. We presented the state of the art in the calibration with markers and image-based calibration without markers with DCC. Also, we explained and illustrated the idea of the hybrid algorithm combining the use of markers and DCC.

In the chapter 2 we discuss the detection of the markers on projected images and one self-calibration approach which is based on the bundle adjustment and came from the computer vision theory. In the chapters 3, 4 and 5 we want to generalize existing data consistency conditions to distributions for different X-ray geometries and as in [Unb+17] apply them to the information about non-truncated markers.

2 Cone-beam geometric calibration based on computer vision approach

2.1 French summary of the chapter

Dans ce chapitre nous commençons par décrire le modèle du C-arm. La projection du faisceau conique d'un C-arm est décrite par une matrice de projection similaire à celle d'une caméra. Il existe deux principaux types de C-arm : les C-arms chirurgicaux mobiles (qui se déplacent dans différentes salles d'opération) et les C-arms statiques utilisés, par exemple, en radiologie interventionnelle. En général, les C-arms chirurgicaux mobiles nécessitent un étalonnage en ligne car leur mécanique est moins robuste. Lors de l'étalonnage nous devons trouver l'ensemble des 9 paramètres d'étalonnage pour chaque projection. Dans le cas de marqueurs dont nous ne connaissons pas les coordonnées 3D, nous pouvons appliquer la méthode de l'ajustement de faisceaux. Dans la section 2.2, nous effectuons des simulations numériques pour résoudre le problème d'ajustement de faisceaux pour cette géométrie en faisceau conique. Nous avons mentionné le résultat de la vision par ordinateur dans le chapitre précédent selon lequel si nous avons l'ensemble des matrices de projection comme solutions du problème d'ajustement de faisceaux, elles diffèrent des vraies matrices de projection par une transformation projective. De plus, si on considère que l'inclinaison entre les axes est nulle et si la séquence de vues est suffisamment générale (incluant les vues spécifiques), cette transformation projective doit être une similitude [PKV99]. Cela doit également être vrai pour les C-arms, car ils peuvent être décrits de la même manière que les caméras. Numériquement, nous remarquons que si nous initialisons la solution de notre problème d'optimisation avec une estimation très éloignée de la distance source-détecteur, nous obtenons la solution différente de la vraie par une similitude. La solution est obtenue avec la méthode de Levenberg-Marquardt.

Dans la section 2.3, nous étudions le modèle intégral en faisceau conique afin de trouver le lien entre la théorie de la représentation matricielle et la représentation intégrale. En effet, nous montrons que tout système en faisceau conique décrit par le modèle intégral (y compris C-arm) ne peut être auto-calibré géométriquement qu'à une similitude près avec n'importe quelle méthode d'auto-étalonnage. Nous mentionnons que la même propriété est vraie dans la géométrie divergente 2D pour les systèmes en éventail. Avec la démonstration numérique et les estimations de la section précédente, nous montrons que nous pouvons toujours effectuer la reconstruction avec des paramètres estimés, mais l'image reconstruite, évidemment, diffère de l'image de référence, obtenue avec les vraies paramètres d'étalonnage. Nous vérifions numériquement la formule, le lien entre les images reconstruites et de référence. Dans

le cas réel, nous ne connaissons pas les coordonnées des points 3D, nous ne pouvons donc pas estimer la similitude qui relie les points 3D vrais et estimés comme nous avons fait dans notre démonstration numérique. Notre contribution a été décrite dans l'article [KDG21].

Dans la section 2.4, nous parlons de l'étape de la détection des marqueurs. Lorsqu'on parle de la détection de traces de marqueurs sur les images projetées, on ne peut pas utiliser la définition standard des singularités liées à la continuité des fonctions. Nous devons parler de la discontinuité de Hölder. Nous montrons sur l'exemple de la géométrie parallèle 2D que pour la transformée de Radon de l'indicateur de disque nous avons deux points avec discontinuité de Hölder sur l'image projetée qui nous aident à définir le centre de la projection du marqueur rond. Ces deux points peuvent être détectés avec l'algorithme standard pour trouver les discontinuités de Hölder à l'aide d'ondelettes.

2.2 Bundle adjustment for a cone-beam system

In the next two sections we describe our contribution [KDG21] presented at the 16th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine.

Theory. In the article we considered a C-arm X-ray imaging system (fig. 1.6). Note that there are two types of C-arms: mobile surgical C-arms (which move to different operating rooms) and static C-arms used, for example, in interventional radiology. Usually, mobile surgical C-arms need online calibration because their mechanics is less robust.

Thanks to IEC 61217 standard, the geometry of the C-arm can be described in the similar way as the pinhole camera. As in computer vision, the projection matrix of the C-arm is

$$P = K\begin{pmatrix} R & t \end{pmatrix}, \tag{2.1}$$

$$K = \begin{pmatrix} -f & 0 & u_0 \\ 0 & -f & v_0 \\ 0 & 0 & 1 \end{pmatrix}, f = \frac{SDD}{dim_{\text{pixel}}}, u_0 = \frac{spos_x - dx}{dim_{\text{pixel}}}, v_0 = \frac{spos_y - dy}{dim_{\text{pixel}}}, \tag{2.2}$$

$$R = \begin{pmatrix} c_z & -s_z & 0 \\ s_z & c_z & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_x & -s_x \\ 0 & s_x & c_x \end{pmatrix} \begin{pmatrix} c_y & 0 & s_y \\ 0 & 1 & 0 \\ -s_y & 0 & c_y \end{pmatrix}, \tag{2.3}$$

$$c_\alpha = \cos(-\theta_\alpha),\ s_\alpha = \sin(-\theta_\alpha),\ \alpha \in \{x, y, z\}, \tag{2.4}$$

$$t = \begin{pmatrix} -spos_x \\ -spos_y \\ -SID \end{pmatrix}. \tag{2.5}$$

Here parameters are:

- *SDD*: source-detector distance,
- *SID*: source-isocenter distance,
- $spos_x$: x-coordinate of the position of the source in the rotated frame,
- $spos_y$: y-coordinate of the position of the source in the rotated frame,
- dx: x-coordinate of the center of the detector in the rotated frame,
- dy: y-coordinate of the center of the detector in the rotated frame,
- θ_x: orientation of the rotated frame relative to the world frame along the x axis,
- θ_y: angle of scan,
- θ_z: orientation of the rotated frame relative to the world frame along the z axis.

Notice that the calibration matrix K has zero skew ($s = K_{12} = 0$), because flat panel digital detectors unlike previous analog detectors have no skew. The rotation matrix R is here a rotation around the isocenter, defines the position of the rotated frame, the translation $-t$ is the source position in the rotated frame. The meaning of the parameters of the model could be better understood from the fig. 2.1.

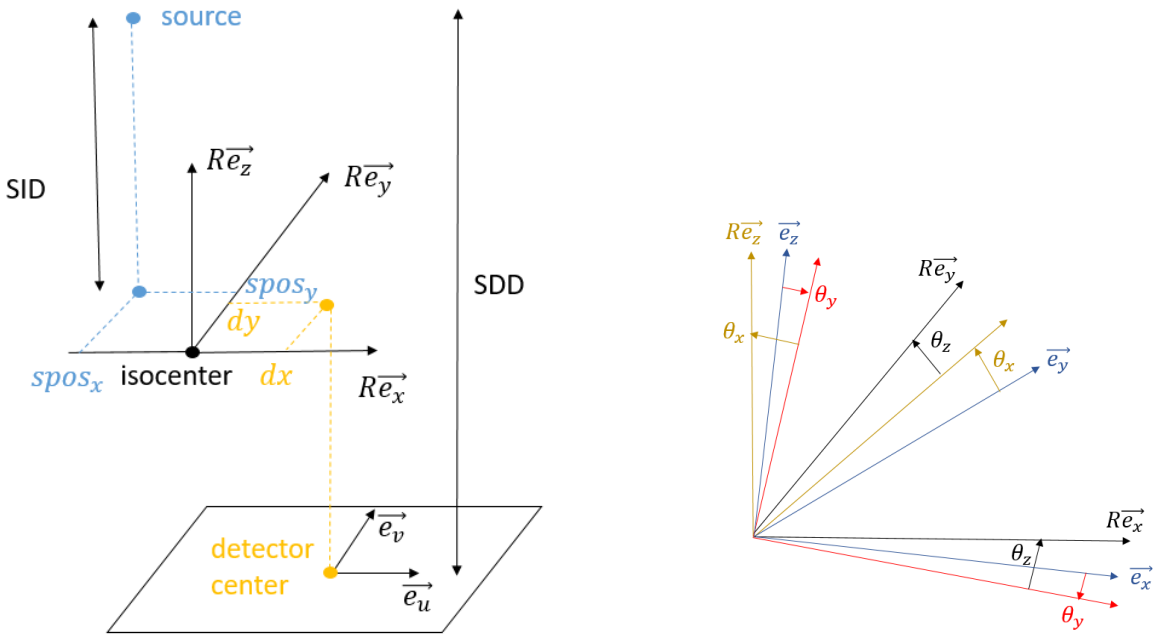

Figure 2.1: Geometric parameters of a C-arm in a schematic view.

In order to derive this model, we need to consider few coordinate systems:

- Firstly, we need to move from the world coordinate system fixed in the isocenter to the virtual coordinate system differing from the world one only by a rotation. Note that this corresponds to the rotation of the C-arm around isocenter. Thus, coordinates of the 3D point in the world coordinate system and the virtual coordinate system are connected:

$$\begin{pmatrix} X^v \\ Y^v \\ Z^v \end{pmatrix} = R \begin{pmatrix} X^w \\ Y^w \\ Z^w \end{pmatrix}. \tag{2.6}$$

- Secondly, we need to move from the virtual coordinate system to the 3D system of the source. They differ only by translation by the vector $-t$, so in homogeneous coordinates we have

$$\begin{pmatrix} X^s \\ Y^s \\ Z^s \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & -spos_x \\ 0 & 1 & 0 & -spos_y \\ 0 & 0 & 1 & -SID \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X^v \\ Y^v \\ Z^v \\ 1 \end{pmatrix}. \tag{2.7}$$

- The projected point belongs to the detector. We can define the coordinate system of the detector, but also the 2D coordinate system of the source, they differ by the vector connecting their centers. In the words of homogeneous coordinates, for the 2D point in different coordinate systems we have:

$$\begin{pmatrix} x^d \\ y^d \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & spos_x - dx \\ 0 & 1 & spos_y - dy \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x^s \\ y^s \\ 1 \end{pmatrix}. \tag{2.8}$$

- Thus, we have the 3D coordinate system of the source and the 2D coordinate system of the source, see connection in the fig. 2.2. We can find the coordinate of the projection q of the point Q in the 2D system of the source. From ratios $\frac{SDD}{-Z^s} = \frac{y^s}{Y^s}$ and $\frac{SDD}{-Z^s} = \frac{x^s}{X^s}$ the connection between the 3D coordinates of Q and 2D coordinates of q in these systems is:

$$\begin{pmatrix} x^s \\ y^s \\ 1 \end{pmatrix} \sim \begin{pmatrix} -SDD & 0 & 0 & 0 \\ 0 & -SDD & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X^s \\ Y^s \\ Z^s \\ 1 \end{pmatrix}. \tag{2.9}$$

- Taking into account also the transition from the 2D detector coordinate system to the 2D pixel coordinate system, we obtain exactly the projection matrix as indicated above.

Since the projection matrix of the C-arm has the same structure as for the pinhole camera, we can apply all calibration algorithms from computer vision to C-arms, for example, we can solve the BA problem if we need to identify projection matrices and 3D coordinates of markers simultaneously. We

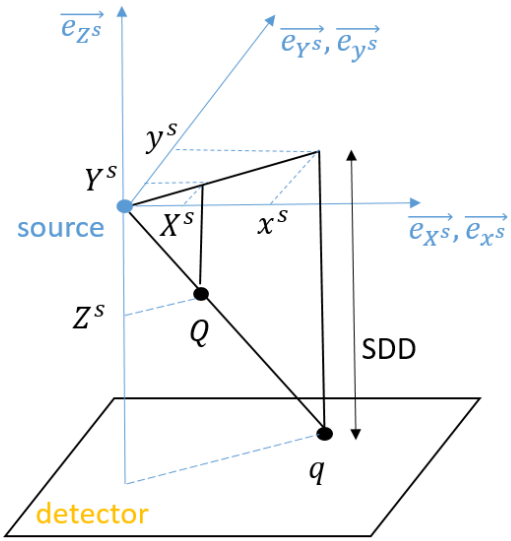

Figure 2.2: Projection of the 3D point Q into q and coordinate systems of the source.

already mentioned that we have a solution to the BA problem in computer vision up to a projective transformation, but sometimes we can have a metric reconstruction up to a similarity transformation. In [PKV99] authors explained in details when we have a metric reconstruction. We have that for special calibration matrices with $K_{12} = 0$ (which is a case for the described pinhole camera model and C-arm), the solution to BA could be calculated up to a similarity transformation. In general case, the camera projection matrix can be factorized as follows

$$P = \begin{pmatrix} H & e \end{pmatrix} = KR \begin{pmatrix} I & -t \end{pmatrix}, \tag{2.10}$$

where $H = KR$, $e = -KRt$, $K = \begin{pmatrix} f_x & s & u \\ 0 & f_y & v \\ 0 & 0 & 1 \end{pmatrix}$ is a calibration matrix, R and t - the camera rotation and translation correspondingly. Here s is the skew.

The authors used in [PKV99] the following lemma to find the equivalent algebraic condition for $s = 0$:

Lemma 2.1. *The absence of skew in K is equivalent with* $(h_1 \times h_3) \cdot (h_2 \times h_3) = 0$*, where* h_i *denotes the i-th row of H.*

After they proved the following theorem:

Theorem 2.1. *The class of transformations which preserves the absence of skew is the group of similarity transformations.*

For that, they showed that the similarity transformations preserve the calibration matrix K. After,

it was sufficient to show that the class of projective transformations of the type $\begin{pmatrix} A & b \\ c^T & d \end{pmatrix}$ which preserve the condition $(h_1 \times h_3) \cdot (h_2 \times h_3) = 0$ is at most the group of similarity transformations. The authors used that the absence of skew should be preserved for all possible views. It is equivalent with the fact that we can transform the set of all possible projection matrices to the set of all possible projection matrices. Thus, it was possible to choose specific R and t in order to translate conditions $(h_1 \times h_3) \cdot (h_2 \times h_3) = 0$ for the elements of the projection matrix to conditions for the projective transformation $\begin{pmatrix} A & b \\ c^T & d \end{pmatrix}$ and to show that this projective transformation should have a form of a similarity transformation.

So, it's true that if we have the set of projection matrices as solutions of the BA problem, they differ from the true projection matrices by a projective transformation. Moreover, if we consider just zero skews and if the sequence of views is general enough (includes the specific views that Pollefeys used in the proof), this projective transformation should be a similarity transformation.

Numerical experiments. For numerical experiments we used the C++ package Ceres [AM+]. We used parameters presented in the table 2.1 in order to simulate data for 181 different projections:

1. we took 20 markers (fig. 2.3) and call them $Q_{j,\text{true}}$;

2. we computed 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 from the parameters in the table 2.1 in order to use them to initialise the optimization algorithm;

3. with the set of initial calibration parameters we build a set of real calibration parameters to which estimations with the algorithm will be compared, we added a uniform noise to each parameter from the table 2.1 for each projection to construct real calibration parameters and real projection matrices P_{real}^i, we used the uniform distribution with bounds as in the table 2.1 and we added noise to the base of realistic values, for the uniform noise we used random.uniform() in Python's numpy with equally likely outcomes from the inteval $[\text{low}, \text{high})$;

4. we simulated projections q_j^i by $P_{\text{real}}^i Q_{j,\text{true}}$, after we simulated detection errors by adding a uniform noise with bounds ± 0.3 pix to the image points $P_{\text{real}}^i Q_{j,\text{true}}$;

5. we calculated with the basic triangulation algorithm (triangulatePoints() in Python's OpenCV) the initial estimations for 3D points from two known initial projection matrices and known projections for 0 and 90 degrees for the initialisation of the algorithm, so if we have projection matrices and projections, then we can reconstruct 3D points from the system of linear equations, see more details about this basic method in [HZ04].

Note that with the presented parameters the isocenter is placed in $(0,0,0)^T$, there are almost no rotations of the rotated frame in the isocenter relative to the world frame along x and z axes, only along

Parameter	**Initial value**	**Base for realistic values**	**Noise bounds**
SDD (mm)	1000	1300	± 3.5
SID (mm)	700	700	± 6.9
$spos_x$ (mm)	0	0	± 6.9
$spos_y$ (mm)	0	0	± 6.9
dx (mm)	0	0	± 13.9
dy (mm)	0	0	± 13.9
θ_x (degrees)	0	0	± 1.4
θ_z (degrees)	0	0	± 1.4
θ_y (degrees)	$i\delta$	$i\delta$	± 0.7
dim_{pixel} (mm)	0.5	-	-
δ (degrees)	2	-	-
$N_{\text{projections}}$	181	-	-

Table 2.1: Initial values of the C-arm parameters, the base for their realistic values, their noise bounds of mechanical vibrations and some parameters for the simulation. Here i is the projection index, δ is the angular step.

y axis, the position of the source is defined by SID, the distance from the isocenter $(0,0,0)^T$ in the positive direction of $R\vec{e}_z$, so we have almost a circular trajectory of the source, then the position of the detector is defined by SDD, the distance from the source in the negative direction of $R\vec{e}_z$, and depends on the position of the source.

Trajectories of projections of 20 markers in the case without noise and in the case with the described noise are presented in the fig. 2.4. It's possible to see that the noise level was high, so we tested the worst case. Note that it's possible to smooth the noise in order to better approximate real systems where we often have a smaller spread of positions.

In order to solve numerically the optimization problem inside the BA we used the Levenberg-Marquardt method. The final cost for the Levenberg-Marquardt method was 0.003. We used the Euclidean distance inside the cost and normalized the cost with $\frac{1}{N_{\text{markers}}N_{\text{projections}}}$, so we solved the following BA:

$$\min_{P^i,Q_j} \frac{1}{N_{\text{markers}}N_{\text{projections}}} \sum_{i,j} d(P^i Q_j, q_j^i)^2. \tag{2.11}$$

We observed high maximum errors in the calibration parameters (especially for the extrinsic parameter t): 9.784 pix for f, 10.279 pix for u_0, 9.650 pix for v_0, 102.085 mm for t. From the error

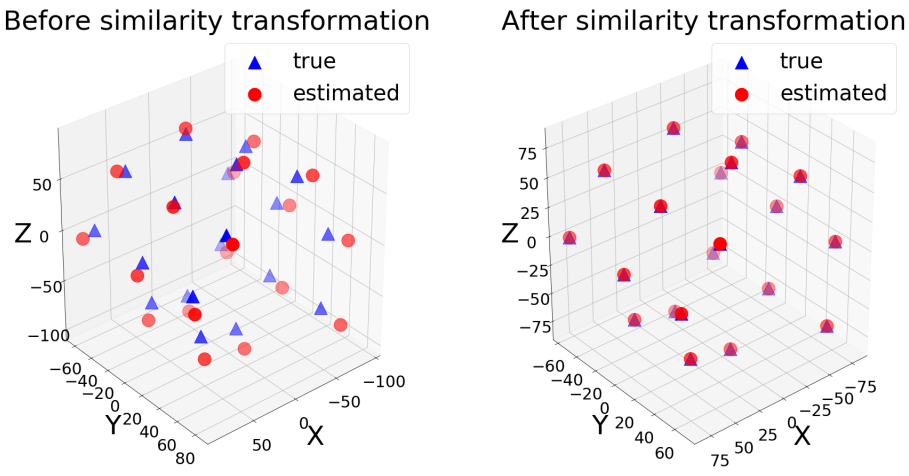

Figure 2.3: True and estimated 3D points (markers) for the Levenberg-Marquardt method in the world coordinate system.

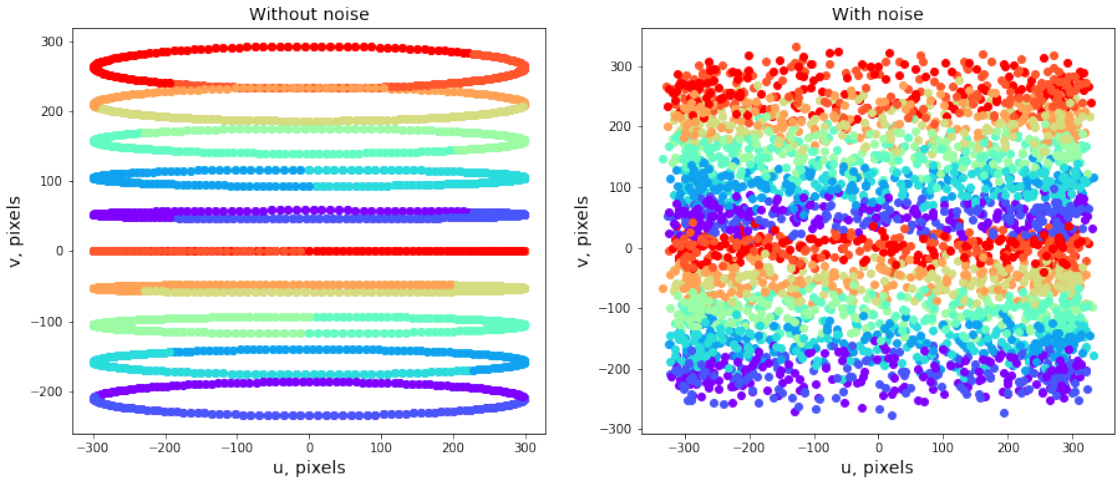

Figure 2.4: Trajectories of projections of markers without noise and for simulations with noise. Each marker out of 20 does the whole circle without noise, but because of the symmetry of markers we can see the half of the trajectory for each marker in the first image.

rotation matrix $R^i_{\text{err}} = (R^i_{\text{real}})^{-1}\hat{R}^i$ for each projection i we calculated ψ_i as in the Euler rotation theorem: $|\psi_i| = \arccos\frac{\text{tr}(R^i_{\text{err}})-1}{2}$. The maximum error defined by $|\psi_i|$ through all projections was 0.790 degrees. Similarly, true and estimated 3D points were far (fig. 2.3) with the maximum error 16.888 mm. But we found that we could obtain the true points from the estimated points with a scaling, a rotation and a translation. For that:

1. we found the barycenters b_{true} and b_{est} of the true and the estimated points, the mean of $\frac{||Q_{j,\text{true}}-b_{\text{true}}||_2}{||\hat{Q}_j-b_{\text{est}}||_2}$

gave us the estimation of the scaling;

2. after the scaling correction, we computed the rotation and translation with the algorithm described in [AHB87].

It is possible to see almost the coincidence of the true 3D points and the estimated 3D points after the similarity transformation in the fig. 2.3, the maximum error is 0.013 mm. There is nothing strange in this result, since we have a non-uniqueness of the solution to the BA problem for this C-arm matrix model as for the pinhole camera. In the next section we will show that this non-uniqueness is also typical for the integral model of the cone-beam system.

2.3 Non-uniqueness in the cone-beam self-calibration

As we told before, researchers in tomography consider image-based calibration methods with markers and without markers. For the first group, 3D coordinates of marker centers could be known in the world coordinate system or unknown as in the BA problem. In the part describing tomography, we unite for X-ray systems the second subgroup of algorithms where we don't know the geometry of markers and algorithms without markers to self-calibration algorithms.

Theory. Cone-beam computed tomography is the basis of many X-ray systems and helps to obtain detailed images of a patient noninvasively. We showed the integral definition of this divergent geometry in the introduction for $f \in \mathscr{S}(\mathbb{R}^3)$. Note here that we want to show our result for the general source trajectory and for $\vec{\zeta}$ as a unit vector inside the definition in Eq. (1.8) $\mathfrak{D}f(\vec{s}_\lambda, \vec{\zeta}) = \int_0^{+\infty} f(\vec{s}_\lambda + l\vec{\zeta})dl$. We use any rotation R and any translation vector $\vec{t} \in \mathbb{R}^3$.

Our main contributions from the article [KDG21] are the following two theorems which show the link between the limits of self-calibration in computer vision and cone-beam computed tomography:

Theorem 2.2. *Let* $f_{R,\vec{t}}(\vec{x}) = f(R\vec{x}+\vec{t})$*, then*

$$\mathfrak{D}f_{R,\vec{t}}(\vec{s}_\lambda, \vec{\zeta}) = \mathfrak{D}f(R\vec{s}_\lambda + \vec{t}, R\vec{\zeta}). \tag{2.12}$$

Proof. We have

$$\mathfrak{D}f_{R,\vec{t}}(\vec{s}_\lambda, \vec{\zeta}) = \int_0^{+\infty} f_{R,\vec{t}}(\vec{s}_\lambda + l\vec{\zeta})dl = \int_0^{+\infty} f(R\vec{s}_\lambda + lR\vec{\zeta} + \vec{t})dl = \mathfrak{D}f(R\vec{s}_\lambda + \vec{t}, R\vec{\zeta}).$$

□

Thus, the cone-beam data $\mathfrak{D}f_{R,\vec{t}}$ of the function $f_{R,\vec{t}}$ from the source $\vec{s}_\lambda$ in the direction $\vec{\zeta}$ are equal to the cone-beam data $\mathfrak{D}f$ of f from the source position $R\vec{s}_\lambda + \vec{t}$ in the direction $R\vec{\zeta}$. If we denote

$\vec{v}_\lambda = R\vec{s}_\lambda + \vec{t}$, then $\vec{s}_\lambda = R^T\left(\vec{v}_\lambda - \vec{t}\right)$, if $\vec{\eta} = R\vec{\zeta}$, then $\vec{\zeta} = R^T\vec{\eta}$. Then for all $\vec{v}_\lambda$ and all unit $\vec{\eta}$ Eq. (2.12) is equivalent to

$$\mathfrak{D}f_{R,\vec{t}}\left(R^T\left(\vec{v}_\lambda - \vec{t}\right), R^T\vec{\eta}\right) = \mathfrak{D}f\left(\vec{v}_\lambda, \vec{\eta}\right). \tag{2.13}$$

Thus, the data $\mathfrak{D}f$ of f from the source position $\vec{v}_\lambda$ in the direction $\vec{\eta}$ are equal to $\mathfrak{D}f_{R,\vec{t}}$ from the source position $R^T\left(\vec{v}_\lambda - \vec{t}\right)$ in the direction $R^T\vec{\eta}$ for any R and $\vec{t}$. In other words, any cone-beam system (including C-arm) can not be geometrically self-calibrated better than up to an Euclidean transformation.

Moreover, we can add the scaling factor σ, then

Theorem 2.3. *Let $f_{\sigma R,\vec{t}}(\vec{x}) = f(\sigma R\vec{x} + \vec{t})$, with $\sigma > 0$, then*

$$\mathfrak{D}\left(\sigma f_{\sigma R,\vec{t}}\right)\left(\vec{s}_\lambda, \vec{\zeta}\right) = \mathfrak{D}f\left(\sigma R\vec{s}_\lambda + \vec{t}, R\vec{\zeta}\right). \tag{2.14}$$

Proof. We have

$$\begin{aligned}\mathfrak{D}f_{\sigma R,\vec{t}}(\vec{s}_\lambda, \vec{\zeta}) = \int_0^{+\infty} f_{\sigma R,\vec{t}}(\vec{s}_\lambda + l\vec{\zeta})dl &= \int_0^{+\infty} f(\sigma R\vec{s}_\lambda + \sigma lR\vec{\zeta} + \vec{t})dl \\ &= \int_0^{+\infty} f(\sigma R\vec{s}_\lambda + nR\vec{\zeta} + \vec{t})d\frac{n}{\sigma} = \frac{1}{\sigma}\mathfrak{D}f(\sigma R\vec{s}_\lambda + \vec{t}, R\vec{\zeta}).\end{aligned}$$

□

Thus, the cone-beam data $\mathfrak{D}(\sigma f_{\sigma R,\vec{t}})$ of $\sigma f_{\sigma R,\vec{t}}$ from the source $\vec{s}_\lambda$ in the direction $\vec{\zeta}$ are equal to the cone-beam data $\mathfrak{D}f$ of f from the source position $\sigma R\vec{s}_\lambda + \vec{t}$ in the direction $R\vec{\zeta}$. If we denote $\vec{v}_\lambda = \sigma R\vec{s}_\lambda + \vec{t}$, then $\vec{s}_\lambda = \frac{1}{\sigma}R^T\left(\vec{v}_\lambda - \vec{t}\right)$, if $\vec{\eta} = R\vec{\zeta}$, then $\vec{\zeta} = R^T\vec{\eta}$. Then for all $\vec{v}_\lambda$ and all unit $\vec{\eta}$ Eq. (2.14) is equivalent to

$$\mathfrak{D}\left(\sigma f_{\sigma R,\vec{t}}\right)\left(\frac{1}{\sigma}R^T\left(\vec{v}_\lambda - \vec{t}\right), R^T\vec{\eta}\right) = \mathfrak{D}f(\vec{v}_\lambda, \vec{\eta}). \tag{2.15}$$

Similarly, we could obtain for $\sigma < 0$:

$$\mathfrak{D}(|\sigma| f_{\sigma R,\vec{t}})\left(\frac{1}{\sigma}R^T\left(\vec{v}_\lambda - \vec{t}\right), -R^T\vec{\eta}\right) = \mathfrak{D}f(\vec{v}_\lambda, \vec{\eta}). \tag{2.16}$$

Thus, we can see that any cone-beam system (including C-arm) can not be geometrically self-calibrated better than up to a similarity transformation with any self-calibration method (including the BA method that we saw in the previous section).

Since the definition of the integral projection in the 2D fan-beam case is described by the similar integral in Eq. (1.7) as in the 3D cone-beam case in Eq. (1.8), if we define in the similar way different transformations in 2D, we can repeat the proof of the Theorem 2.2 and 2.3. Note that we work in the case of non-weighted projections or when $\vec{\zeta}$ is a unit vector. So, in this case we also have the similar

fact in 2D: any fan-beam system can not be geometrically self-calibrated better than up to a scaling, a rotation and a translation. Thus, our result is very general: it's true in the 3D and 2D cases and regardless of the chosen algorithm to self-calibrate.

Numerical experiments. For the numerical demonstration we used the same simulated data from the previous section and the solution showed there to the BA problem (estimated calibration parameters and estimated 3D points). Note that we can use the estimated calibration parameters of the C-arm in the reconstruction process. We took the 3D Shepp–Logan phantom as $f(\vec{x})$. With the help of the Python package RTK [Rit+14] we simulated projections with true calibration parameters and computed a FDK reconstruction $g(\vec{x})$ with the estimated calibration parameters. If we want to compare $f(\vec{x})$ and $g(\vec{x})$ without a similarity correction, we can see that our reconstruction doesn't match with the true Shepp–Logan phantom (see fig. 2.5 and some profiles in the fig. 2.7 and 2.8). Since the initial source-detector distance (SDD) that we use as the first guess for our optimization procedure differs a lot from the true values of SDD, we can achieve a similarity error in the solution obtained with our optimization procedure.

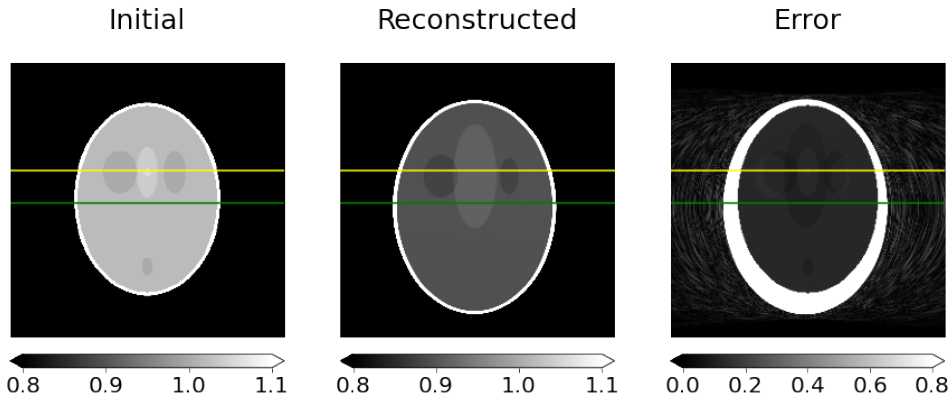

Figure 2.5: Slices $z = 6.5$ mm of: left: the initial 3D Shepp–Logan phantom $f(\vec{x})$, center: the reconstruction $g(\vec{x})$ from the estimated acquisition geometry and right: $|f(\vec{x}) - g(\vec{x})|$. Yellow lines correspond to the profile from 2.7, green lines - to the profile from 2.8.

From Eq. (2.14), the reconstructed image should correspond to the function $\sigma f_{\sigma R,\vec{t}}(\vec{x})$. Since $g(\vec{x})$ 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 $\frac{1}{\sigma} g\left(\frac{1}{\sigma} R^T\left(\vec{x}-\vec{t}\right)\right)$. Since we know the reference and we have from the previous step with the BA estimated positions of markers, we can estimate $\sigma, R, \vec{t}$ as we did in the previous section and we know the similarity correction in this case. Note that in order to compute f in $\vec{x}$ from g we need to interpolate: we used the linear interpolation method from SciPy package. The result of such similarity correction applied to the image g is shown in the fig. 2.6. It's possible to compare profiles after similarity correction for fixed y and z as we did in the fig. 2.7 and 2.8. We also computed the root-mean-square error (RMSE) between the initial 3D image and the reconstructed one after the similarity correction, it was equal to 0.076. To understand the order of the error, we can compute RMSE for the case of the reconstruction with the set of true

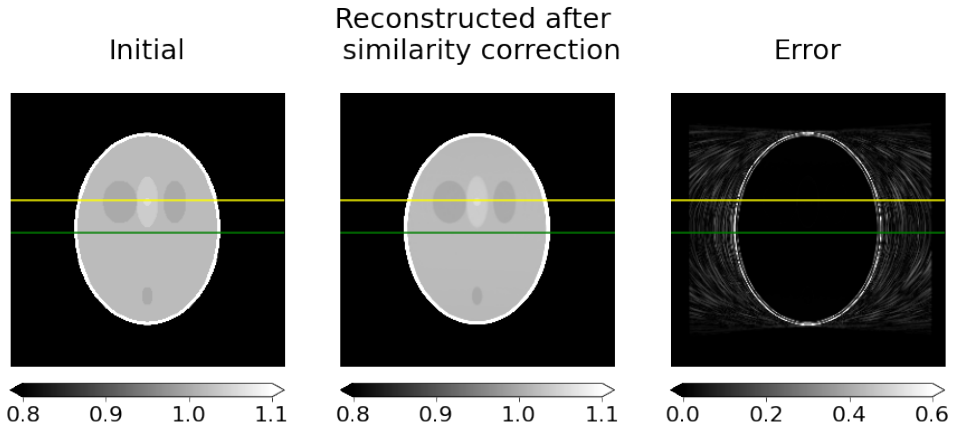

Figure 2.6: Slices $z = 6.5$ mm of: left: the initial 3D Shepp–Logan phantom $f(\vec{x})$, center: the reconstruction from the estimated acquisition geometry after the similarity correction $\frac{1}{\sigma} g\left(\frac{1}{\sigma} R^T\left(\vec{x} - \vec{t}\right)\right)$ and right: $\left| f(\vec{x}) - \frac{1}{\sigma} g\left(\frac{1}{\sigma} R^T\left(\vec{x} - \vec{t}\right)\right)\right|$. Yellow lines correspond to the profile from 2.7, green lines - to the profile from 2.8.

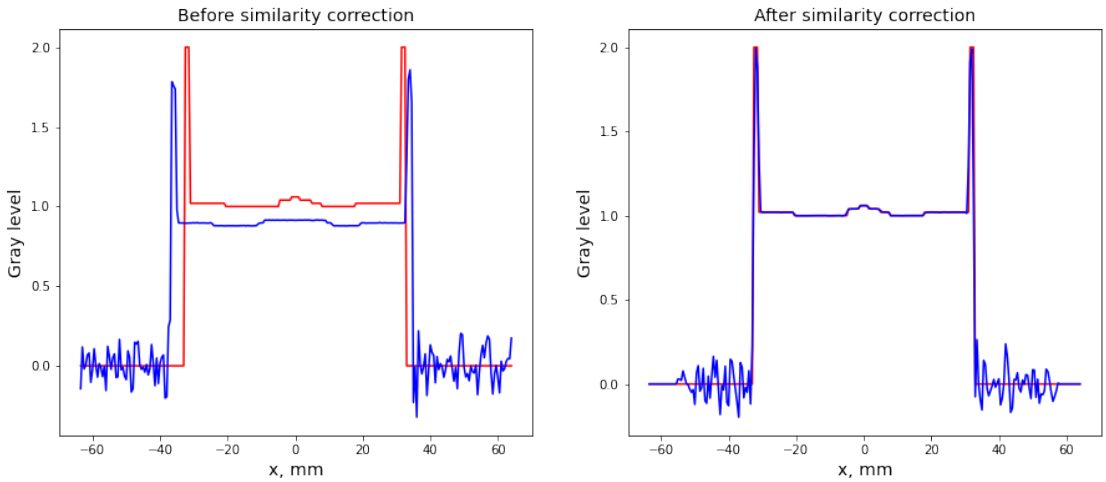

Figure 2.7: Profiles of the reconstruction (blue) before and after similarity correction and the initial phantom (red) for $z = 6.5$ mm, $y = -13.5$ mm.

calibration parameters, in this case the normal reconstruction error of RTK is equal to 0.091. Thus, we can see that the reconstruction with the estimated parameters is correct. Moreover, we verified that in this case the reconstructed image has the form $\sigma f_{\sigma R,\vec{t}}(\vec{x})$. Note that usually we don't have true values of the positions of markers, so we can't make the estimation of the similarity correction.

In these sections we showed that we can apply bundle adjustment to the C-arm calibration, since C-arms can be described with the same algebraic model as cameras. Unfortunately, we can obtain the solution just up to a projective transformation (and up to a similarity transformation in the case of the general sequence of views). From the other side, C-arm is also described by the integral cone-beam

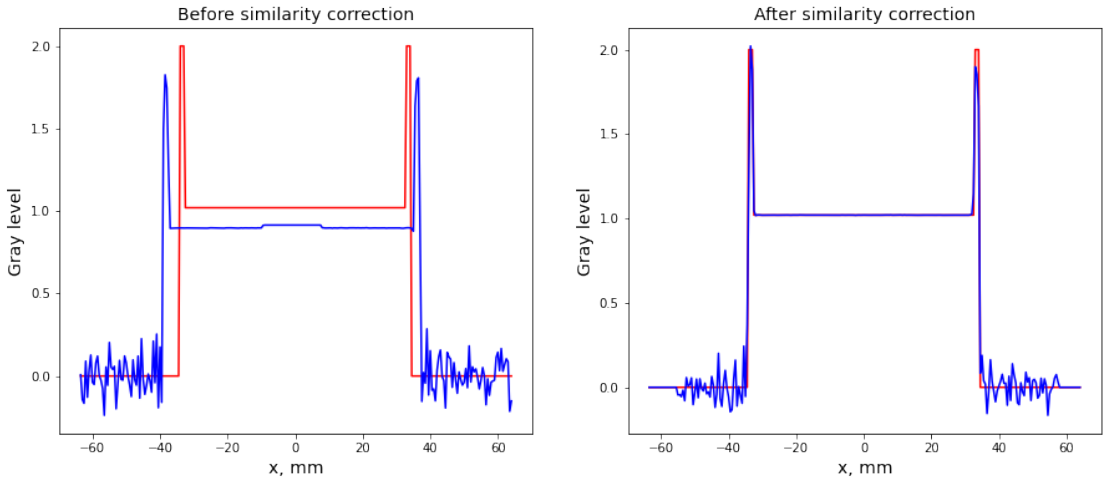

Figure 2.8: Profiles of the reconstruction (blue) before and after similarity correction and the initial phantom (red) for $z = 6.5$ mm, $y = 1.5$ mm.

model. We also showed that the self-calibration problem for any system with the cone-beam geometry and fan-beam geometry can not be solved better than up to a similarity transformation.

2.4 Marker detection difficulty in X-ray projections

In the previous sections we supposed that we have already the information about markers in projected images. But as in computer vision, in self-calibration in tomography we have 3 main tasks:

1. to detect interest points in the projected images (introduced markers or natural singularities in data if we perform the calibration without markers), it's the detection step;
2. to associate same markers in different projected images;
3. to compute the geometric parameters of the model, it's the calibration step.

Here we want to explain why it's not trivial to detect markers in X-ray projection images on the example of the 2D Radon transform. Suppose that markers are small balls. Then, the Radon transform of one marker, i.e. the indicator of a disk of radius R centered at (a,b) denoted by $f_{R,(a,b)}$, for fixed α is the function from Eq. (1.6). Let us denote $f_1 = \mathfrak{R}_\alpha f_{R,(a,b)}$:

$$f_1(s) = \begin{cases} 2\sqrt{R^2 - (s - s_0(\alpha,a,b))^2}, & \text{if } |s - s_0| \le R, \\ 0, & \text{otherwise,} \end{cases} \tag{2.17}$$

where $s_0(\alpha,a,b) = a\cos\alpha + b\sin\alpha$.

We want to study f_1 more carefully. Usually, when we talk about singularities we think about discontinuities. But f_1 is continuous! What kind of singularity of f_1 can we detect? In order to answer to this question, we want to use the Hölder (Lipschitz) continuity. As far as we know, this function has not been investigated before in the context of Hölder regularity, although its other properties were used in the similar task of the watershed segmentation [TDK05].

Let us start with basic definitions.

Definition 2.1. *Let $0 \leq \alpha \leq 1$. A function f is pointwise Hölder (Lipschitz) α at x_0, if there exists $A > 0$ and $h_0 > 0$ such that for any $h < h_0$*

$$|f(x_0+h) - f(x_0)| \leq A|h|^{\alpha}. \tag{2.18}$$

A Hölder (Lipschitz) regularity of f at x_0 is a maximal α: f is Hölder (Lipschitz) α at x_0.

If there exists a constant A such that Eq. (2.18) is satisfied for any x_0 and x_0+h within an interval of $\mathbb{R}$, then the function f is uniformly Hölder (Lipschitz) α over this interval.

Definition 2.2. *A function f is singular at x_0 if it's not Hölder 1 at x_0.*

It's well known that:

- A Hölder α function, $0 < \alpha < 1$, is continuous, but non-differentiable.
- A C^1 function in a neighborhood of x_0 is Hölder 1 at x_0.

The main contribution of this section:

Theorem 2.4. *The function f_1 is Hölder $1/2$ at $\pm R + s_0$ and Hölder 1 otherwise.*

Proof. It's obvious that infinitely differentiable function is Hölder 1. Thus, we need to explain just the situation with the points $\pm R + s_0$. Let us start with the case of $R = 1$ and $s_0 = 0$ in order to better understand the logic of the proof; f_1 becomes

$$f_0(s) = \begin{cases} 2\sqrt{1-s^2}, & \text{if } |s| \leq 1, \\ 0, & \text{otherwise.} \end{cases} \tag{2.19}$$

For the point 1 we have

$$|f_0(s) - f_0(1)| = |f_0(s)| \leq 2\sqrt{1-s}\sqrt{2} = K_1|s-1|^{1/2}. \tag{2.20}$$

Suppose that we can find $K_2 > 0$ and $\varepsilon > 0$: $|f_0(s) - f_0(1)| = |f_0(s)| = 2|s-1|^{1/2}|s+1|^{1/2} \leq K_2|s-1|^{1/2+\varepsilon}$, then for s from $[1-h_0, 1]$, $h_0 > 0$

$$|s+1|^{1/2} \leq \frac{K_2}{2}|s-1|^{\varepsilon}. \tag{2.21}$$

But in the limit $s \to 1^-$ we have $\sqrt{2} \leq 0$. Then Eq. (2.21) can't be true for all s from $[1-h_0, 1]$. Thus, f_0 is Hölder $1/2$ at 1. Similarly, the function f_0 is also Hölder $1/2$ at -1.

Let us consider our function for arbitrary R and s_0. Then, from Eq. (2.20) we automatically have for $R+s_0$:

$$|f_1(s) - f_1(R+s_0)| = |f_1(s)| \leq 2\sqrt{R-(s-s_0)}\sqrt{2R} = K_3|s-(R+s_0)|^{1/2}. \tag{2.22}$$

And as in Eq. (2.21) we have a contradiction if we suppose $|f_1(s) - f_1(R+s_0)| \leq K_4|s-(R+s_0)|^{1/2+\varepsilon}$ for some $K_4 > 0$ and $\varepsilon > 0$, then

$$|s+(R-s_0)|^{1/2} \leq \frac{K_4}{2}|s-(R+s_0)|^{\varepsilon}, \tag{2.23}$$

because in the limit $s \to (R+s_0)^-$ we have $\sqrt{2R} \leq 0$. Thus, $f_1(s)$ is Hölder $1/2$ at $R+s_0$. Similarly, the function f_1 is also Hölder $1/2$ at $-R+s_0$. □

A disk indicator is Hölder 0 at the boundary. We have shown that its Radon projection is Hölder $1/2$ at the projection of the boundary, thus more regular and probably more difficult to detect. The similar behaviour was already described in the case of Sobolev regularity in [Nat01], pp. 42-44. Let $H^{\beta}(\Omega_2)$ of an open set $\Omega_2 \in \mathbb{R}^2$ denotes the Sobolev space, β is real and represent the smoothness of the function. We can also define the Sobolev space for $Z_2 = S^1 \times \mathbb{R}$, S^1 is the unit sphere in the 2-dimensional Euclidean space. Then the author states in the case of the dimension $N = 2$ that if we have the smooth function f of compact support in $H_0^{\beta}(\Omega_2)$, then $\Re f$ belongs to $H^{\beta+1/2}(Z_2)$, where $H_0^{\beta}(\Omega_2)$ is the Sobolev space of order β of functions of support in the closure of Ω_2, $H^{\alpha+1/2}(Z_2)$ is the Sobolev space of the higher order $\beta+1/2$ on Z_2. Thus, the projection is smoother (by $1/2$) than the object.

If we have a patient body described by a function from $\mathscr{D}(\mathbb{R}^2)$, the whole scene is the sum of this function and an indicator of a disk, then the measured Radon transform for each projection has a form $\Phi + f_1$, where $\Phi \in \mathscr{D}(\mathbb{R})$. In this case, we have everywhere Hölder 1 regularity except two special points of f_1 where we still have Hölder $1/2$:

Lemma 2.2. *If $f = \Phi + f_1$, where $\Phi \in \mathscr{D}(\mathbb{R})$, f_1 is from Eq.* (2.17) *(Hölder* 1 *everywhere except 2 points, where it's Hölder* $1/2$*), then f is also Hölder* 1 *everywhere except these two points, where it's Hölder* $1/2$*.*

Proof. It's obvious except the fact about two points. Let's take one $\hat{s}$. For this point in some neighbourhood

$$\begin{aligned} |f(s) - f(\hat{s})| &= |\Phi(s) - \Phi(\hat{s}) + f_1(s) - f_1(\hat{s})| \leq |\Phi(s) - \Phi(\hat{s})| + |f_1(s) - f_1(\hat{s})| \\ &\leq C_1|s-\hat{s}|^1 + C_2|s-\hat{s}|^{1/2} \leq C_1|s-\hat{s}|^{1/2} + C_2|s-\hat{s}|^{1/2} \leq (C_1+C_2)|s-\hat{s}|^{1/2}. \end{aligned}$$

Suppose that we can find $C_3 > 0$ and $\varepsilon > 0$: $|f(s) - f(\hat{s})| \leq C_3 |s - \hat{s}|^{1/2+\varepsilon}$, then in some one-sided neighbourhood (suppose $s < \hat{s}$ for it)

$$\frac{|\Phi(s) - \Phi(\hat{s}) + f_1(s) - f_1(\hat{s})|}{|s - \hat{s}|^{1/2}} \leq C_3 |s - \hat{s}|^{\varepsilon}. \tag{2.24}$$

Then $\lim_{s \to \hat{s}^-} \frac{\Phi(s) - \Phi(\hat{s})}{|s - \hat{s}|^{1/2}} = 0$ from the Taylor expansion, $\lim_{s \to \hat{s}^-} \frac{f_1(s) - f_1(\hat{s})}{|s - \hat{s}|^{1/2}} = \sqrt{2R}$ as before for our special function. Thus, we again have the contradiction, since in the limit $s \to \hat{s}^-$ in Eq. (2.24) we have $\sqrt{2R} \leq 0$. Thus, f is Hölder $1/2$ at $\hat{s}$. □

We can detect these singularities with wavelets, for example, by applying the canonical algorithm of Mallat [MZ92; MH92; Mal09]. The corresponding numerical results are presented in the Appendix A.1. This algorithm gives 2 singular points $-R + s_0$ and $R + s_0$. If we know these 2 points, we can compute the center s_0 of the marker in the projected image. If we have 2 markers, we will detect 4 singular points. Thus, we want to have the markers far from each other in projected images in order to be able to match 2 pairs correctly. To assure this, their centers s_0^1 and s_0^2 should be far from each other, radius of markers R should be as small as possible. But centers s_0^1 and s_0^2 in projected images aren't the same with centers (a^1, b^1) and (a^2, b^2) of the markers in the scene! We know that $s_0^1 = a^1 \cos\alpha + b^1 \sin\alpha$, $s_0^2 = a^2 \cos\alpha + b^2 \sin\alpha$ for each projection angle α. If two markers belong to one horizontal line $b^1 = b^2$, then $|s_0^1 - s_0^2| = |(a^1 - a^2)\cos\alpha|$. It's obvious that we can't have then $\cos\alpha = 0$, where $s_0^1 = s_0^2$. When we are in the discrete setting, we can assure to have $|\cos\alpha| \geq |\cos\alpha_0| \neq 0$, then $|s_0^1 - s_0^2| \geq |(a^1 - a^2)\cos\alpha_0|$. We need $|s_0^1 - s_0^2| > 2R$. If $|\cos\alpha_0| = 2R$, then we should have

$$|a^1 - a^2| > 1. \tag{2.25}$$

If two markers belong to another line, we need to assure $|s_0^1 - s_0^2| = |(a^1 - a^2)\cos\alpha + (b^1 - b^2)\sin\alpha| = |\sqrt{(a^1 - a^2)^2 + (b^1 - b^2)^2}\sin(\alpha + \beta)| > 2R$, β is defined by $\sin\beta = \frac{a^1 - a^2}{\sqrt{(a^1 - a^2)^2 + (b^1 - b^2)^2}}$ and $\cos\beta = \frac{b^1 - b^2}{\sqrt{(a^1 - a^2)^2 + (b^1 - b^2)^2}}$. Thus, we can't have $\sin(\alpha + \beta) = 0$, where $s_0^1 = s_0^2$. If we assure $|\sin(\alpha + \beta)| \geq |\sin(\alpha_0 + \beta)| \neq 0$, take $|\sin(\alpha_0 + \beta)| = 2R$, then we should have

$$\sqrt{(a^1 - a^2)^2 + (b^1 - b^2)^2} > 1. \tag{2.26}$$

If we have 3 or more markers, we need to assure Eq. (2.26) for each pair of markers.

We illustrated that projections in tomography are usually smoothed by the integral transform, thus this makes it difficult to detect the traces of markers in projected images, hence the centers of markers in projected images also. We justified in this section the application of the canonical algorithm of Mallat to detect the Hölder irregularities of f_1 for fixed projection angle. From the detected irregularities we can detect the centers of markers in projected images. This detection can be performed just with some assumptions. However, this gives us hope that the task of markers' detection can be solved, for example, by applying the theory of wavelets.

3 Calibration with DCC on distributions for 2D Radon transform

3.1 French summary of the chapter

Ce chapitre débute par la présentation des motivations de ce travail (la section 3.2). Nous revenons au problème des données tronquées. Les DCC sur les fonctions ne peut pas gérer les données tronquées. Nous prévoyons d'utiliser l'information locale sur les marqueurs, de modéliser les marqueurs avec Diracs et de construire les DCC pour les marqueurs non tronqués. C'est pourquoi nous devons rappeler la théorie de la distribution.

Nous rappelons ensuite les définitions de base liées aux distributions et présentons les notations que nous utiliserons dans ce chapitre. Dans la section 3.3, nous décrivons qu'une distribution agissant sur K est une fonctionnelle linéaire continue sur K. L'espace de toutes les distributions agissant sur l'espace K est noté K'. On rappelle que les distributions générales agissent sur l'espace des fonctions lisses de support compact, mais on ne s'intéresse qu'aux distributions de support compact. Elles sont définies en agissant sur l'espace des fonctions lisses. Aussi, nous mentionnons que dans ce cas la preuve de continuité de la fonctionnelle linéaire peut être remplacée par la preuve que la fonctionnelle linéaire est bornée.

Dans la section 3.4 nous parlons de la définition connue de la transformée de Radon sur les distributions présentée par Ramm et Katsevich dans leur ouvrage [RK96]. Ils y présentent également les conditions de cohérence des données nécessaires et suffisantes pour la transformée de Radon sur les distributions, la généralisation des conditions de cohérence de Helgason-Ludwig bien connues. Dans la section 3.5 nous simplifions cette définition de la transformée de Radon sur les distributions, nous définissons la transformée de Radon sur les distributions de support compact pour l'angle de projection fixe. Cela nous aide à construire les conditions de cohérence simplifiées qui sont faciles à appliquer. L'application de ces DCC pour les moments d'ordre 0, 1 et 2 nous conduit à la nouvelle procédure analytique pour estimer deux ensembles d'inconnues dans le problème de la calibration classique en géométrie en faisceau parallèle : les décalages et les angles de projection. Dans cette procédure, nous concevons également la mire d'étalonnage de deux lignes perpendiculaires de marqueurs qui nous aide à dériver les formules analytiques. Nous discutons également de la non-unicité de la solution et montrons quelle solution de la classe de toutes les solutions possibles nous pouvons trouver avec notre procédure de la calibration. Nous présentons les simulations numériques avec et sans bruit.

3.2 Introduction

In this chapter we return to the problem of truncated data described in the section 1.4.4. We mentioned the example of the problem to calibrate with truncated data with DCC for the 2D Radon transform. When markers are in the field of view, we have non-truncated information about markers, while complete data for the object of interest are truncated. We want to use this advantage of non-truncated markers to build the set of new calibration algorithms.

In order to derive these algorithms, we need to choose how to model markers. In the clinical settings, round markers are usually used. If these markers are sufficiently small, we can model them with Dirac distributions. Let us remind that the Dirac distribution is used to model a narrow spike function. Since we plan to produce hybrid methods, we want to use markers and build DCC for them. In order to describe DCC in mathematically correct way, we need to start in the section 3.3 with basic facts about distributions. More precisely, with the definition of distributions and some useful properties. Distribution can be defined as a continuous linear functional for different spaces of test functions, but for us it will be important to consider distributions acting on spaces of smooth functions or distributions of compact support. We will remind that the Dirac distribution is the distribution of compact support.

For distributions of compact support we can define the Radon transform, the fan-beam and cone-beam transforms as another distributions. In order to understand properly these definitions of transforms, the continuity of the linear functional acting on the space of smooth functions should be defined. For that, we need to consider the space of smooth functions as a Fréchet space equipped with the set of semi-norms. Then in the topology induced by these semi-norms continuity is equivalent with boundedness which is easy to check.

In the section 3.4 we will show the existing definition in the literature for the Radon transform of distribution of compact support described by Ramm and Katsevich. The authors also introduced necessary and sufficient DCC for distributions to be in the range of the Radon transform. These DCC are hard to apply, that's why in the section 3.5 we will introduce our definition of the Radon transform of distribution of compact support for fixed projection angle. In order to apply these DCC to the calibration task, we need only the necessary part. Our DCC can be applied to Diracs, thus they use only the local non-truncated information about markers. DCC will help us as usually to produce the analytical calibration procedure, but the specific calibration cage will be introduced to the scene to use the advantage of markers too. Thus, we will design the hybrid method.

Similar calibration procedures will be designed for the case of the fan-beam transform on a line in the chapter 4, the cone-beam transform on a line and the cone-beam transform on a plane parallel to the detector in the chepter 5. In each case we need to start with proper definitions of transforms of distributions of compact support, to prove necessary DCC and apply these DCC to Dirac distributions (markers) to design calibration procedures. The goal of this chapter is to show how DCC applied to Diracs can be defined and used in the calibration process for the simple case of the 2D Radon transform.

3.3 Distributions

In this section we introduce notations and basic facts about distributions that we use below. We use the following notation in this chapter:

- $\mathscr{S}_N = \mathscr{S}(\mathbb{R}^N)$ for Schwartz spaces of rapidly decreasing smooth functions,
- $\mathscr{D}_N = \mathscr{D}(\mathbb{R}^N) = C_0^\infty(\mathbb{R}^N)$ for spaces of compactly supported smooth functions,
- $\mathscr{E}_N = C^\infty(\mathbb{R}^N)$ for spaces of smooth functions.

We have the inclusions: $\mathscr{D}_N \subset \mathscr{S}_N \subset \mathscr{E}_N$.

Definition 3.1. *A distribution acting on a set of functions K is a continuous linear functional $T : K \to \mathbb{R}$. The space of all distributions acting on the space K is denoted by K'.*

Thus, we can define different spaces of distributions $\mathscr{S}'_N$, $\mathscr{D}'_N$, $\mathscr{E}'_N$. According to [Hel99], we have the inclusions: $\mathscr{E}'_N \subset \mathscr{S}'_N \subset \mathscr{D}'_N$. Since $\mathscr{D}'_N$ is the largest set, we will write "any distribution" with the meaning that it's a distribution belonging to $\mathscr{D}'_N$. Moreover, each function $f \in \mathscr{D}_N$ can be seen as a distribution T_f acting on $\mathscr{E}_N$ with $T_f(\phi) = (f, \phi)$, where $(\cdot,\cdot)$ is the usual scalar product in $L^2(\mathbb{R}^N)$ defined by the integral:

$$(f, \phi) = \int_{\mathbb{R}^N} f(\vec{x})\phi(\vec{x})d\vec{x}, \tag{3.1}$$

where we compute here the Lebesgue integral. Thus, $\mathscr{D}_N \subset \mathscr{E}'_N$. Also, each function $f \in \mathscr{E}_N$ can be seen as a distribution T_f acting on $\mathscr{D}_N$ with $T_f(\phi) = (f, \phi)$, thus $\mathscr{E}_N \subset \mathscr{D}'_N$.

If we have any functional, then it's usually easy to check its linearity. But the situation is more complicated with continuity. Note that we can also define for an arbitrary open set $\Omega_N \subset \mathbb{R}^N$ $\mathscr{D}'(\Omega_N)$ and $\mathscr{E}'(\Omega_N)$. We will focus on distributions from $\mathscr{E}'(\Omega_N)$ in the next sections, thus it's good to understand better the continuity of the functional $T : \mathscr{E}(\Omega_N) \to \mathbb{R}$. We will replace the check of continuity with the check of boundedness. In the literature (see, for example, the basic introduction to distributions in French [Bon01]), $\mathscr{E}(\Omega_N)$ is usually considered as the so-called Fréchet space, the special topological vector space equipped with the set of semi-norms. Note that semi-norms are characterized by similar three properties as norms: for them we have the triangle inequality as for norms, the absolute homogeneity as for norms and non-negativity without point separation inherent in norms. More precisely:

Definition 3.2. *For the vector space X a real-value function $P : X \to \mathbb{R}$ is a semi-norm if it satisfies:*

1. $P(f+g) \leq P(f) + P(g)$, $f, g \in X$,
2. $P(\lambda f) = |\lambda| P(f)$, $f \in X$ *and a scalar* λ,
3. $P(f) \geq 0$, $f \in X$.

The set of semi-norms in our case is defined by

$$P_j(\phi) = \sum_{|\boldsymbol{\beta}|\leq j} \sup_{\vec{x}\in K_j} |\partial^{\boldsymbol{\beta}}\phi(\vec{x})|, \tag{3.2}$$

where $\phi \in \mathscr{E}(\Omega_N)$, $|\boldsymbol{\beta}| = \beta_1 + ... + \beta_N$, $\beta_k \in \mathbb{N}$, $k = 1,\ldots,N$, $\mathbb{N}$ is the set of non-negative integers, $\boldsymbol{\beta} = (\beta_1,...,\beta_N) \in \mathbb{N}^N$ and $\partial^{\boldsymbol{\beta}} = \partial_1^{\beta_1}...\partial_N^{\beta_N}$, $\{K_j\}_{j=0,1,...}$ - a set of compacts with $\cup_j K_j = \Omega_N$, K_j is in the interior of the set K_{j+1}. So, in order to fix the concrete definition of semi-norms, we usually fix the set of compacts. For example, in the case of $\Omega_1 = \mathbb{R}$ researches usually use $K_j = [-j, j]$. This set helps to define a proper topology in $\mathscr{E}(\Omega_N)$.

For spaces with semi-norms we know that the continuity of T is equivalent with the boundedness of T [NB10]. Thus, we can check the boundedness of T or if T maps a bounded subset of $\mathscr{E}(\Omega_N)$ to a bounded subset of $\mathbb{R}$. A subset $\Phi \subset \mathscr{E}(\Omega_N)$ is bounded iff $P_j(\Phi)$ is bounded for each j or we can find $M_j < +\infty$ such that each $P_j(\phi) \leq M_j$ for $\phi \in \Phi$. Moreover, the boundedness of all $P_j(\Phi)$ for all j is equivalent with the boundedness of all $\tilde{P}_{j,\boldsymbol{\beta}}(\Phi)$ for all j and $\boldsymbol{\beta}$:

$$\tilde{P}_{j,\boldsymbol{\beta}}(\phi) = \sup_{\vec{x}\in K_j} |\partial^{\boldsymbol{\beta}}\phi(\vec{x})|. \tag{3.3}$$

Definition 3.3. *A sequence of distributions $\{T_j\}_{j\in\mathbb{N}}$ acting on K converges to T if for each $\phi \in K$* $\lim_{j\to+\infty}(T_j,\phi) = (T,\phi)$.

Definition 3.4. *Any distribution T is null on open set O if $T(\phi) = 0$ for each function ϕ:* $\mathrm{supp}(\phi) \subset O$. *The support of T is the complement of the largest open set for which T is null.*

According to [Hel99], $\mathscr{E}'(\Omega_N)$ as defined before is equal to the subspace of $\mathscr{D}'(\Omega_N)$ of all distributions of compact support. This is also true in the case of $\Omega_N = \mathbb{R}^N$. An example of the distribution of compact support is a Dirac distribution.

Definition 3.5. *A Dirac distribution (or just Dirac) $\delta_{\vec{c}}$ is a distribution from $\mathscr{D}'_N$: $\forall\phi \in \mathscr{D}_N$ $\delta_{\vec{c}}(\phi) = \phi(\vec{c})$.*

It's true that $\delta_{\vec{c}}$ is a distribution of compact support, thus $\delta_{\vec{c}} \in \mathscr{E}'_N$. Also, the finite sum of Diracs is a distribution of compact support (see [GW99]). Let us show these facts in the following theorem:

Theorem 3.1. $\mathrm{supp}(\sum_{i=1}^n \lambda_i\delta_{\vec{c}_i}) = \{\vec{c}_1,...,\vec{c}_n\}$.

Proof. Denote $T = \sum_{i=1}^n \lambda_i\delta_{\vec{c}_i}$. We should show that T is null on $\mathbb{R}^N \setminus \{\vec{c}_1,...,\vec{c}_n\}$. Indeed, for each ϕ with $\mathrm{supp}(\phi)\cap\{\vec{c}_1,...,\vec{c}_n\} = \varnothing$ we have $\phi(\vec{c}_i) = 0$ for all c_i, $i = 1,\ldots,n$, then $T(\phi) = \sum_{i=1}^n \lambda_i\phi(\vec{c}_i) = 0$. Thus, $\mathrm{supp}(T) \subset \{\vec{c}_1,...,\vec{c}_n\}$.

If $\exists k : \vec{c}_k \notin \mathrm{supp}(T)$, thus $\vec{c}_k$ is in the complement of $\mathrm{supp}(T)$, thus in an open set where T is null. Then we can take a neighborhood I of $\vec{c}_k$ in this open set: $\vec{c}_i \notin I$ for all other indices i, T is null on I. We can choose ϕ with $\mathrm{supp}(\phi) \subset I$ and $\phi(\vec{c}_k) = 1$, then $T(\phi) = 0$, because T is null on I. From the other side, $T(\phi) = \sum_{i=1}^n \lambda_i\delta_{\vec{c}_i}(\phi) = \lambda_k\phi(\vec{c}_k) = \lambda_k$. Thus we have a contradiction, since $\lambda_k \neq 0$. Thus, $\mathrm{supp}(T) = \{\vec{c}_1,...,\vec{c}_n\}$. □

According to [Bon01], a convolution of any distribution with a smooth function of compact support can be defined:

Definition 3.6. *For any distribution T and $g \in \mathscr{D}_N$, a convolution $T * g$ is defined and it's a distribution defined by a function $f(\vec{x}) = (T(\vec{t}), g(\vec{x}-\vec{t}))$.*

Note that the notation $g(\vec{x})$ usually means that we consider the value of the function g in the point $\vec{x}$. Here we put the point inside the function to point out that g is a function depending on $\vec{x}-\vec{t}$. We also add the variable $\vec{t}$ to the distribution to point out that the distribution T acts on functions of the variable $\vec{t}$, $\vec{x}$ is a parameter. It's possible to find the following properties of convolution with proofs in [Bon01]:

Theorem 3.2. *Let $T \in \mathscr{D}'_N$, $g \in \mathscr{D}_N$, then* $\operatorname{supp}(T * g) \subset \operatorname{supp}(T) + \operatorname{supp}(g)$.

Theorem 3.3. *Let $T \in \mathscr{D}'_N$, $g \in \mathscr{D}_N$ such that their supports are closed compact sets. Then $T * g$ belongs to $\mathscr{D}_N$.*

Theorem 3.4. *Let $T \in \mathscr{E}'_N$, $g, h \in \mathscr{D}_N$, then $(T * g) * h = T * (g * h)$.*

With this, researchers did the following generalization of the convolution to two distributions of compact support:

Definition 3.7. *$T_1, T_2 \in \mathscr{E}'_N$, then $\exists! T \in \mathscr{E}'_N$: $\forall g \in \mathscr{D}_N$ $T_1 * (T_2 * g) = T * g$. We call T a convolution of T_1 and T_2 and write $T = T_1 * T_2$.*

Moreover, we will use the following definition from [Bon01]:

Definition 3.8. *A translation $\vec{t}$ of any distribution T is defined for each test function ϕ by*

$$(T(\vec{x}-\vec{t}), \phi(\vec{x})) = (T(\vec{x}), \phi(\vec{x}+\vec{t})). \tag{3.4}$$

Here we use the generalisation of the definition of the translation of a function. Since each function $f \in \mathscr{E}_N$ can be seen as a distribution T_f on $\mathscr{D}_N$ with $T_f(\phi) = (f, \phi)$, then after the change of variables from $\vec{x}$ to $\vec{x}'$ with $\vec{x}' = \vec{x}-\vec{t}$

$$(f(\vec{x}-\vec{t}), \phi(\vec{x})) = \int_{\mathbb{R}^N} f(\vec{x}-\vec{t})\phi(\vec{x})d\vec{x} = \int_{\mathbb{R}^N} f(\vec{x}')\phi(\vec{x}'+\vec{t})d\vec{x}' = (f(\vec{x}'), \phi(\vec{x}'+\vec{t})). \tag{3.5}$$

In the next section we recall known results about the Radon transform on distributions using the results presented in this section.

3.4 The work of Ramm and Katsevich

The Radon transform on distributions in arbitrary dimension was firstly defined by Gelfand and colleagues in [GGV66]. But in this section we use the later book by Ramm and Katsevich [RK96], where

the authors introduced DCC for the Radon transform on distributions. We concentrate just on the 2D Radon transform on distributions in this section in order to present the contribution of Ramm and Katsevich, but in their book they described their results in arbitrary dimension.

Definition of Ramm and Katsevich. Let us introduce the adjoint (or dual of the Radon transform $\mathfrak{R}$) operator $\mathfrak{R}^*$ for functions ϕ from the Schwartz space $\mathscr{S}([0,2\pi]\times\mathbb{R})$:

$$\mathfrak{R}^*\phi(\vec{x}) := \int_0^{2\pi} \phi(\alpha, \vec{x}\cdot\vec{\theta}_\alpha)d\alpha \qquad \forall \vec{x}\in\mathbb{R}^2. \tag{3.6}$$

Here as before $\alpha\in[0,2\pi)$, $\vec{\theta}_\alpha = (\cos\alpha,\sin\alpha)$, $\vec{\eta}_\alpha = (-\sin\alpha,\cos\alpha)$. We define the space $\mathscr{S}([0,2\pi]\times\mathbb{R})$ as a restriction of functions in $\mathscr{S}(\mathbb{R}^2)$ to $[0,2\pi]\times\mathbb{R}$. So, the Schwartz space on $[0,2\pi]\times\mathbb{R}$ is the space of smooth functions ϕ in all variables and such that ϕ and all its derivatives are fast decreasing according to the second variable.

Note that for functions $f\in\mathscr{S}_2, \phi\in\mathscr{S}([0,2\pi]\times\mathbb{R})$ we have with the change of variables from (s,l), $s\in\mathbb{R}$, $l\in\mathbb{R}$ to $\vec{x}\in\mathbb{R}^2$ with $s\vec{\theta}_\alpha + l\vec{\eta}_\alpha = \vec{x}$:

$$\begin{aligned}(\mathfrak{R}f,\phi) &= \int_0^{2\pi}\int_{-\infty}^{\infty} \mathfrak{R}f(\alpha,s)\phi(\alpha,s)dsd\alpha = \int_0^{2\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(s\vec{\theta}_\alpha + l\vec{\eta}_\alpha)dl\phi(\alpha,s)dsd\alpha \\ &= \int_0^{2\pi}\int_{\mathbb{R}^2} f(\vec{x})\phi(\alpha,\vec{x}\cdot\vec{\theta}_\alpha)d\vec{x}d\alpha = \int_{\mathbb{R}^2} f(\vec{x})\int_0^{2\pi}\phi(\alpha,\vec{x}\cdot\vec{\theta}_\alpha)d\alpha d\vec{x} = \langle f,\mathfrak{R}^*\phi\rangle, \end{aligned}\tag{3.7}$$

where $(\cdot,\cdot)$ is the scalar product in $L^2([0,2\pi]\times\mathbb{R})$, $\langle\cdot,\cdot\rangle$ is the scalar product in $L^2(\mathbb{R}^2)$.

We can generalize the definition in Eq. (3.6) to $\phi\in C^\infty(Z_2)$, $Z_2=[0,2\pi]\times\mathbb{R}$. Note that we usually consider functions defined on open sets, but we decided to keep the notation from the book where the authors considered also distributions acting on $C^\infty(Z_2)$. This space can be also seen as a Fréchet space with the set of semi-norms of the type Eq. (3.2), where $\cup_j K_j = Z_2$ with $K_j = [0,2\pi]\times K'_j$, $\cup_j K'_j = \mathbb{R}$, K'_j is in the interior of the set K'_{j+1}.

If $\phi\in C^\infty(Z_2)$, then $\mathfrak{R}^*\phi\in\mathscr{E}_2$. To show the last fact, let us use the following basic theorems from analysis:

Theorem 3.5. *If I is the interval in $\mathbb{R}$, $J\subset\mathbb{R}^N$ is the measurable set, f is the measurable function defined on $I\times J$. Suppose that we can properly define the Lebesgue integrals*

$$F(t) = \int_J f(t,\vec{y})d\vec{y}.$$

- *Continuity. If two hypotheses are true:*

  1. *f is continuous on t on I for $\vec{y}$ almost everywhere,*
  2. *$\exists g\in L^1(J) : \forall t\in I\ |f(t,\vec{y})|\le g(\vec{y})$ for $\vec{y}$ almost everywhere.*

  Then F is continuous on I.

- *Differentiability. If two hypotheses are true:*

1. *f is differentiable on t on I for $\vec{y}$ almost everywhere,*
2. *$\exists h \in L^1(J) : \forall t \in I \, |\frac{\partial f}{\partial t}(t,\vec{y})| \leq h(\vec{y})$ for $\vec{y}$ almost everywhere.*

Then F is differentiable on I and

$$F'(t) = \int_J \frac{\partial f}{\partial t}(t,\vec{y})d\vec{y}.$$

For each $[0,2\pi] \times A_2$, $A_2 = [a_1,a_2] \times [b_1,b_2]$ with $a_1,a_2,b_1,b_2 \in \mathbb{R}$ we have:

1. $\phi(\alpha,\vec{x}\cdot\vec{\theta}_\alpha)$ is continuous on $\vec{x}$ for each α,
2. $\max_{\alpha,\vec{x}\in[0,2\pi]\times A_2} |\phi(\alpha,\vec{x}\cdot\vec{\theta}_\alpha)| = M < +\infty$, since $\phi(\alpha,\vec{x}\cdot\vec{\theta}_\alpha)$ is continuous on the compact set $[0,2\pi]\times A_2$. The constant M is integrable on the compact $[0,2\pi]$, thus we have an integrable majorant M: $|\phi(\alpha,\vec{x}\cdot\vec{\theta}_\alpha)| \leq M$ for each parameter $\vec{x}$,

then the parameter-dependent integral $\mathfrak{R}^*\phi$ is a continuous function of $\vec{x}$ on each A_2 by the theorem about the continuity of parameter-dependent Lebesgue integrals $\Rightarrow \mathfrak{R}^*\phi$ is continuous on $\mathbb{R}^2$. Similarly:

1. $\phi(\alpha,\vec{x}\cdot\vec{\theta}_\alpha)$ is differentiable on $\vec{x}$ for each α,
2. since each partial derivative is continuous on the compact set $[0,2\pi]\times A_2$, then it attains its maximum and minimum values, then for each partial derivative we have a majorant $M < +\infty$ (a constant, the maximum of the absolute values for $\alpha,\vec{x} \in [0,2\pi]\times A_2$), it's integrable on the compact $[0,2\pi]$, since it's a constant,

then we can differentiate the parameter-dependent integral $\mathfrak{R}^*\phi$ as $\int_0^{2\pi}\frac{\partial\phi}{\partial x_1}d\alpha$ and $\int_0^{2\pi}\frac{\partial\phi}{\partial x_2}d\alpha$ by the theorem about the differentiability of parameter-dependent Lebesgue integrals. After, we can apply the theorem about the continuity and differentiability to obtain continuous and differentiable integrals again and again. Thus, we showed that $\mathfrak{R}^*\phi \in \mathscr{E}_2$.

Firstly, Ramm and Katsevich justified in their book that the Radon transform can be defined for $f \in \mathscr{S}_2'$ and $f \in \mathscr{D}_2'$:

Definition 3.9. *The Radon transform $\mathfrak{R}f$ for $f \in \mathscr{S}_2'$ ($f \in \mathscr{D}_2'$) is a bounded linear functional acting on the space $K\mathfrak{R}\mathscr{S}_2$ ($K\mathfrak{R}\mathscr{D}_2$) of test functions according to*

$$(\mathfrak{R}f,\phi) = \langle f,\mathfrak{R}^*\phi\rangle, \tag{3.8}$$

where $K = -\frac{1}{4\pi}\mathscr{H}\frac{\partial}{\partial s}$ with a Hibert transform $\mathscr{H}g(s) = \int_{-\infty}^{\infty}\frac{g(q)}{q-s}dq$.

In this case, $\langle f,\mathfrak{R}^*\phi\rangle$ is well defined for any $f \in \mathscr{S}_2'$ ($f \in \mathscr{D}_2'$) just when $\phi = K\mathfrak{R}\tilde{\phi}$, $\tilde{\phi} \in \mathscr{S}_2$ (or $\mathscr{D}_2$), when from the property of K that $\mathfrak{R}^*K\mathfrak{R} = I$: $\mathfrak{R}^*\phi = \mathfrak{R}^*K\mathfrak{R}\tilde{\phi} = \tilde{\phi} \in \mathscr{S}_2$ (or $\mathscr{D}_2$).

Since $\mathfrak{R}^*\phi \in \mathscr{E}_2$ for $\phi \in C^\infty(Z_2)$, then $\langle f,\mathfrak{R}^*\phi\rangle$ is well defined for $f \in \mathscr{E}_2'$:

Definition 3.10. *The Radon transform $\Re f$ for $f \in \mathscr{E}_2'$ is a bounded linear functional acting on the space $C^\infty(Z_2)$ of test functions according to*

$$(\Re f, \phi) = \langle f, \Re^* \phi \rangle. \tag{3.9}$$

The linearity of such functional is obvious. Moreover, $\Re f$ is bounded. Let us introduce logical steps of the proof. We need to start with the fact that any bounded subset $\Phi \subset C^\infty(Z_2)$ becomes a bounded subset $\Re^*(\Phi) \subset \mathscr{E}_2$. In brief, the function $\phi(\alpha, \vec{x} \cdot \vec{\theta}_\alpha)$ corresponds to each $\phi \in \Phi$, then it's true that the obtained smooth functions of the type $\phi(\alpha, \vec{x} \cdot \vec{\theta}_\alpha)$ depending on $\vec{x}$ form a bounded subset of $\mathscr{E}_2$, since each semi-norm $\tilde{P}_{j,\boldsymbol{\beta}}$ corresponding to the supremum on the compact set K_j ($\cup_j K_j = \mathbb{R}^2$, K_j is in the interior of the set K_{j+1}) of the $\boldsymbol{\beta}$-derivatives of the obtained functions is still bounded. The detailed proof of this fact will be presented later during the proofs of our definition of the 2D Radon transform for fixed α. The next logical step: after the integration of each $\phi(\alpha, \vec{x} \cdot \vec{\theta}_\alpha)$ in $\Re^* \phi(\vec{x}) = \int_0^{2\pi} \phi(\alpha, \vec{x} \cdot \vec{\theta}_\alpha) d\alpha$, it's true that the subset $\Re^*(\Phi)$ is bounded, again because of the boundedness of semi-norms of functions of the type $\Re^* \phi$. Lastly, since f is bounded, thus $f(\Re^*(\Phi))$ is bounded for the bounded $\Re^*(\Phi)$. Then $\Re f(\Phi)$ is bounded for any bounded Φ using Eq. (3.9), so $\Re f$ is a bounded functional.

We will denote $\mathscr{E}(Z_2) = C^\infty(Z_2)$ and the set of distributions on this space as $\mathscr{E}'(Z_2)$ in the following.

DCC of Ramm and Katsevich. As far as we know, the authors first introduced DCC for the Radon transform on distributions. We again consider just the 2D case. The following analogue of HLCC is true:

Theorem 3.6. *A distribution p is the Radon transform of $f \in \mathscr{E}_2'$ if and only if:*

1. *$p \in \mathscr{E}'(Z_2)$,*
2. *p is even: the distributions $p(\alpha + \pi, -s)$ and $p(\alpha, s)$ act identically on $\mathscr{E}(Z_2)$ (or the corresponding distributions $p(-\vec{\theta}_\alpha, -s)$ and $p(\vec{\theta}_\alpha, s)$ act identically),*
3. *for $k = 0, 1, 2, \ldots$, $\forall \psi \in C^\infty([0, 2\pi])$ we have the moment conditions:*

$$(p(\alpha, s), s^k \psi(\alpha)) = \int_0^{2\pi} \mathscr{P}_k(\alpha) \psi(\alpha) d\alpha, \tag{3.10}$$

where $\mathscr{P}_k(\alpha)$ is a homogeneous polynomial of degree at most k in $\cos\alpha$, $\sin\alpha$.

Moreover, if $p(\alpha, s) = 0$ for $|s| > a$, then $\operatorname{supp}(f) \subset B_a$, B_a - the ball of the radius a centered at the origin.

The logical steps of the proof can be found in the sketch of the proof presented in the book of Ramm and Katsevich [RK96], pp. 313-314. This result is theoretically beautiful, but it's hard to use the moment conditions Eq. (3.10) in practice.

In this section we presented the known definition of the 2D Radon transform on distributions from Ramm and Katsevich. The Theorem 3.6 gives necessary and sufficient conditions for the distribution p to be in the range of $\mathfrak{R}$. But for practical applications only necessary conditions are required. In the next section we will give necessary conditions more practical than Eq. (3.10). Indeed, they can be used if the number of projections $\mathfrak{R}f$ is finite or if only the finite number of α_i, $i = 0, ..., P-1$, is available or when $\mathfrak{R}f(\alpha, s)$ is discretized in α.

3.5 Radon transform on distributions

3.5.1 Definition and DCC

Our definition for fixed α. Let us consider $\mathfrak{R}_\alpha f(s) := \mathfrak{R}f(\alpha, s)$ for $f \in \mathscr{S}_2$, thus $\mathfrak{R}_\alpha f$ is a function of one variable s for fixed α. Let us introduce the adjoint (or dual of the Radon transform $\mathfrak{R}_\alpha$) operator $\mathfrak{R}^*_\alpha$ for functions from $\mathscr{S}_1$:

$$\mathfrak{R}^*_\alpha \phi(\vec{x}) := \phi(\vec{x} \cdot \vec{\theta}_\alpha) = \phi(x_1 \cos\alpha + x_2 \sin\alpha) \qquad \forall \phi \in \mathscr{S}_1. \tag{3.11}$$

For $f \in \mathscr{S}_2, \phi \in \mathscr{S}_1$, we have with the change of variables from (s, l), $s \in \mathbb{R}$, $l \in \mathbb{R}$ to $\vec{x} \in \mathbb{R}^2$ with $s\vec{\theta}_\alpha + l\vec{\eta}_\alpha = \vec{x}$:

$$\begin{aligned}(\mathfrak{R}_\alpha f, \phi) = \int_{-\infty}^{\infty} \mathfrak{R}_\alpha f(s)\phi(s)ds = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(s\vec{\theta}_\alpha + l\vec{\eta}_\alpha)dl\phi(s)ds \\ = \int_{\mathbb{R}^2} f(\vec{x})\phi(\vec{x}\cdot\vec{\theta}_\alpha)d\vec{x} = \langle f, \mathfrak{R}^*_\alpha\phi\rangle,\end{aligned} \tag{3.12}$$

where $(\cdot,\cdot)$ is the scalar product in $L^2(\mathbb{R})$, $\langle\cdot,\cdot\rangle$ is the scalar product in $L^2(\mathbb{R}^2)$.

We can generalize the definition in Eq. (3.11) to $\phi \in \mathscr{E}_1$. In this case, $\mathfrak{R}^*_\alpha\phi \in \mathscr{E}_2$. Thus, we can define:

Definition 3.11. *The Radon transform $\mathfrak{R}_\alpha f$ for $f \in \mathscr{E}_2'$ is a bounded linear functional acting on the space $\mathscr{E}_1$ of test functions according to*

$$(\mathfrak{R}_\alpha f, \phi) = \langle f, \mathfrak{R}^*_\alpha\phi\rangle. \tag{3.13}$$

The linearity is obvious. Moreover, let us detail why $\mathfrak{R}_\alpha f$ is bounded. For any bounded subset $\Phi \subset \mathscr{E}_1$ each $\tilde{P}^1_{j,\beta}(\Phi)$ from Eq. (3.3) is bounded with $M_{j,\beta}$, where $\tilde{P}^1_{j,\beta}(\phi)$ corresponds to the supremum on the compact set K^1_j of the β-derivative of ϕ of one variable s, $\cup_j K^1_j = \mathbb{R}$, K^1_j is in the interior of the set K^1_{j+1}, and it doesn't depend on the test function $\phi \in \Phi$. We need to show that the operation $\mathfrak{R}^*_\alpha$ preserves boundedness. Let us show that for any bounded subset $\Phi \subset \mathscr{E}_1$ $\mathfrak{R}^*_\alpha(\Phi) \subset \mathscr{E}_2$ is also bounded. Since α is fixed, then for each compact set K^2_j ($\cup_j K^2_j = \mathbb{R}^2$, K^2_j is in the interior of the set K^2_{j+1}) we can find j': $\vec{x}\cdot\vec{\theta}_\alpha = x_1\cos\alpha + x_2\sin\alpha$ is in $K^1_{j'}$ for all $\vec{x} \in K^2_j$. Thus

$$\sup_{\vec{x}\in K^2_j} |\phi^{(\beta)}(\vec{x}\cdot\vec{\theta}_\alpha)| \le \sup_{s\in K^1_{j'}} |\phi^{(\beta)}(s)| \le M_{j',\beta}. \tag{3.14}$$

For each function $\psi \in \mathfrak{R}^*_\alpha(\Phi)$: $\psi(\vec{x}) = \phi(\vec{x}\cdot\vec{\theta}_\alpha)$ with $\boldsymbol{\beta} = (\beta_1,\beta_2) \in \mathbb{N}^2$ we obtain that $\partial_1^{\beta_1}\partial_2^{\beta_2}\psi(\vec{x}) = (\cos\alpha)^{\beta_1}(\sin\alpha)^{\beta_2}\phi^{(|\boldsymbol{\beta}|)}(\vec{x}\cdot\vec{\theta}_\alpha)$. Thus, we have that each $\tilde{P}^2_{j,\boldsymbol{\beta}}(\mathfrak{R}^*_\alpha(\Phi))$ is bounded with $|(\cos\alpha)^{\beta_1}(\sin\alpha)^{\beta_2}|M_{j',|\boldsymbol{\beta}|}$. Since f is a bounded functional, thus $f(\mathfrak{R}^*_\alpha(\Phi))$ is bounded for the bounded $\mathfrak{R}^*_\alpha(\Phi)$. $\mathfrak{R}_\alpha f(\Phi)$ is bounded for any bounded Φ using Eq. (3.13), so $\mathfrak{R}_\alpha f$ is a bounded functional.

We can't do a definition of the adjoint operator for an arbitrary $f \in \mathscr{D}'_2$: if we take $\phi \in \mathscr{D}_1$, then $\mathfrak{R}^*_\alpha\phi$ is not of compact support and $\langle f, \mathfrak{R}^*_\alpha\phi\rangle$ isn't defined. Let us consider $\phi \in \mathscr{D}_1$: $\phi(s) = 0$ for $|s| > M$, $\phi(s) \neq 0$ for $|s| \leq M$. If we consider $\phi(x_1\cos\alpha + x_2\sin\alpha)$, then $\phi(x_1\cos\alpha + x_2\sin\alpha) \neq 0$ for $-M \leq x_1\cos\alpha + x_2\sin\alpha \leq M$ or for (x_1,x_2) such that

$$x_1\cos\alpha + x_2\sin\alpha \leq M \Rightarrow x_2 \leq \frac{M}{\sin\alpha} - x_1\cot\alpha \tag{3.15}$$

and

$$x_1\cos\alpha + x_2\sin\alpha \geq -M \Rightarrow x_2 \geq -\frac{M}{\sin\alpha} - x_1\cot\alpha. \tag{3.16}$$

From Eq. (3.15) and Eq. (3.16) we receive that (x_1,x_2) is in the band between two parallel lines (at least for the case $\alpha \in (0,\pi/2)$), thus we can take any arbitrarily large x_1 and x_2 and have that $\mathfrak{R}^*_\alpha\phi(\vec{x}) = \phi(x_1\cos\alpha + x_2\sin\alpha) \neq 0 \Rightarrow \mathfrak{R}^*_\alpha\phi \notin \mathscr{D}_2$, because $\mathfrak{R}^*_\alpha\phi$ is not of compact support.

DCC for our definition for fixed α. Let us prove the following necessary condition for the case of our definition:

Theorem 3.7. *For $f \in \mathscr{E}'_2$, if $p_\alpha := \mathfrak{R}_\alpha f$ is the Radon transform of f for fixed α, then:*

1. *$p_\alpha \in \mathscr{E}'_1$,*
2. *p_α is even: the distributions $p_{\alpha+\pi}(-s)$ and $p_\alpha(s)$ act identically on $\mathscr{E}'_1$ (or the corresponding distributions $p_{-\vec{\theta}_\alpha}(-s)$ and $p_{\vec{\theta}_\alpha}(s)$ act identically),*
3. *for $k = 0,1,2,\ldots$ we have the necessary moment conditions:*

$$(p_\alpha(s), s^k) = \mathscr{P}_k(\alpha) \qquad \forall\alpha, \tag{3.17}$$

where $\mathscr{P}_k(\alpha)$ is a homogeneous polynomial of degree at most k in $\cos\alpha$, $\sin\alpha$.

Moreover, $p_\alpha(s) = 0$ for $|s| > a$, if $\operatorname{supp}(f) \subset B_a$.

Proof. Take any $f \in \mathscr{E}'_2$ with $\operatorname{supp}(f) \subset B_a$. Let us take the same smooth positive function of compact support $W_1 \in C^\infty_0(B_1)$, $\int_{B_1} W_1(\vec{x})d\vec{x} = 1$, $W_\varepsilon(\vec{x}) = \varepsilon^{-2}W_1(\vec{x}/\varepsilon)$ as Ramm and Katsevich took in their proof of their theorem for the similar convolution step. Then $f_\varepsilon = W_\varepsilon * f$ and $f_\varepsilon \in C^\infty_0(B_{a+\varepsilon})$ from the Theorem 3.2 and Theorem 3.3. According to [Bon01], for such regularisation function W_ε we have $f_\varepsilon \xrightarrow[\varepsilon\to 0]{} f$ in the sense of distributions. From HLCC (Theorem 1.1) for the function f_ε we have that $\mathfrak{R}f_\varepsilon \in C^\infty_0(Z_{2,a+\varepsilon})$, where $Z_{2,a+\varepsilon} = [0,2\pi]\times[-(a+\varepsilon),a+\varepsilon]$. If we fix α, we build $\mathfrak{R}_\alpha f_\varepsilon \in C^\infty_0([-(a+\varepsilon),a+\varepsilon])$. We have $\forall\phi \in \mathscr{E}_1$

$$(\mathfrak{R}_\alpha f_\varepsilon, \phi) = \langle f_\varepsilon, \mathfrak{R}^*_\alpha\phi\rangle. \tag{3.18}$$

It's true that

$$\lim_{\varepsilon\to 0}\langle f_\varepsilon, \mathfrak{R}^*_\alpha\phi\rangle = \langle f, \mathfrak{R}^*_\alpha\phi\rangle. \tag{3.19}$$

Form the definition $(\mathfrak{R}_\alpha f,\phi) = \langle f, \mathfrak{R}^*_\alpha\phi\rangle$, Eq. (3.18) and Eq. (3.19) $\lim_{\varepsilon\to 0}\mathfrak{R}_\alpha f_\varepsilon = \mathfrak{R}_\alpha f$ in $\mathscr{E}_1'$. Thus, we can transfer properties of $\mathfrak{R}_\alpha f_\varepsilon$: $\mathfrak{R}_\alpha f(s) = 0$ for $|s| > a$, $\mathfrak{R}_\alpha f$ is even (in the sense as described before).

Now we want to derive the moment conditions. For $s^k \in \mathscr{E}_1$

$$(\mathfrak{R}_\alpha f(s), s^k) = \langle f(\vec{x}), \mathfrak{R}^*_\alpha(s^k)(\vec{x})\rangle = \langle f(\vec{x}), (\vec{x}\cdot\vec{\theta}_\alpha)^k\rangle = \langle f(\vec{x}), (x_1\cos\alpha + x_2\sin\alpha)^k\rangle$$
$$= \langle f(\vec{x}), \sum_{i=0}^{k} C(k,i)(x_1\cos\alpha)^{k-i}(x_2\sin\alpha)^i\rangle = \sum_{i=0}^{k} C(k,i)\langle f(\vec{x}), x_1^{k-i}x_2^i\rangle(\cos\alpha)^{k-i}(\sin\alpha)^i = \mathscr{P}_k(\alpha).$$

□

We can see the connection between our definition and the definition of Ramm and Katsevich. For any $\phi \in \mathscr{E}(Z_2)$, $\phi_\alpha \in \mathscr{E}_1$, where $\phi_\alpha(s) = \phi(\alpha, s)$, let us denote

$$F_\phi(\alpha) = (\mathfrak{R}_\alpha f(s), \phi_\alpha(s)), \tag{3.20}$$

$$F_{\phi,\varepsilon}(\alpha) = \int_{\mathbb{R}} \mathfrak{R}_\alpha f_\varepsilon(s)\phi_\alpha(s)ds. \tag{3.21}$$

Thus, we showed in the proof that

$$F_\phi(\alpha) = \lim_{\varepsilon\to 0}(\mathfrak{R}_\alpha f_\varepsilon(s), \phi_\alpha(s)) = \lim_{\varepsilon\to 0}\int_{\mathbb{R}} \mathfrak{R}_\alpha f_\varepsilon(s)\phi_\alpha(s)ds = \lim_{\varepsilon\to 0} F_{\phi,\varepsilon}(\alpha). \tag{3.22}$$

In the proof of the Theorem 3.6 Ramm and Katsevich used the same functions f_ε. They showed the similar equality:

$$(\mathfrak{R}f(\alpha,s), \phi(\alpha,s)) = \lim_{\varepsilon\to 0}(\mathfrak{R}f_\varepsilon(\alpha,s), \phi(\alpha,s))$$
$$= \lim_{\varepsilon\to 0}\int_0^{2\pi}\int_{\mathbb{R}} \mathfrak{R}f_\varepsilon(\alpha,s)\phi(\alpha,s)dsd\alpha = \lim_{\varepsilon\to 0}\int_0^{2\pi} F_{\phi,\varepsilon}(\alpha)d\alpha,$$

where we use that $\mathfrak{R}_\alpha f_\varepsilon(s) = \mathfrak{R}f_\varepsilon(\alpha,s)$.

Let us consider $F_{\phi,\varepsilon}(\alpha) = \int_{\mathbb{R}} \mathfrak{R}_\alpha f_\varepsilon(s)\phi_\alpha(s)ds = \int_{-(a+\varepsilon)}^{a+\varepsilon} \mathfrak{R}_\alpha f_\varepsilon(s)\phi_\alpha(s)ds$ for fixed ε. Using the Theorem 3.5:

1. $\mathfrak{R}_\alpha f_\varepsilon(s)\phi_\alpha(s)$ is continuous on α for each s as a product of two continuous functions,

2. $\mathfrak{R}_\alpha f_\varepsilon \cdot \phi_\alpha$ is continuous on the compact set $[0,2\pi]\times[-(a+\varepsilon), a+\varepsilon]$, thus there is an integrable majorant (a constant, the maximum of the absolute values for $\alpha, s \in [0,2\pi]\times[-(a+\varepsilon), a+\varepsilon]$),

then the parameter-dependent integral $F_{\phi,\varepsilon}(\alpha)$ is a continuous function on $[0,2\pi]$ by the theorem about the continuity of parameter-dependent Lebesgue integrals. For the differentiability:

1. $\mathfrak{R}_\alpha f_\varepsilon(s)\phi_\alpha(s)$ is differentiable on α for each s as a product of two differentiable functions,
2. $\frac{\partial}{\partial\alpha}(\mathfrak{R}_\alpha f_\varepsilon \cdot \phi_\alpha)$ is continuous on the compact set $[0,2\pi]\times[-(a+\varepsilon),a+\varepsilon]$, thus there is an integrable majorant (a constant, the maximum of the absolute values for $\alpha, s \in [0,2\pi]\times[-(a+\varepsilon),a+\varepsilon]$),

then the parameter-dependent integral $F_{\phi,\varepsilon}(\alpha)$ is differentiable on $[0,2\pi]$ by the theorem about the differentiability of parameter-dependent Lebesgue integrals. In the same way, we can show with the basic analysis the smoothness of $F_{\phi,\varepsilon}(\alpha)$.

Thus, for each function ϕ (and the corresponding ϕ_α) we have the sequence of smooth functions $F_{\phi,\varepsilon}(\alpha)$ on $[0,2\pi]$ defined by Eq. (3.21), thus they are Riemann integrable. Also, $\lim_{\varepsilon\to 0} F_{\phi,\varepsilon}(\alpha) = F_\phi(\alpha)$ for each α. In the case when this convergence is uniform, we can change the order of the limit and the integral, we will have

$$(\mathfrak{R}f(\alpha,s),\phi(\alpha,s)) = \lim_{\varepsilon\to 0}\int_0^{2\pi} F_{\phi,\varepsilon}(\alpha)d\alpha = \int_0^{2\pi}\lim_{\varepsilon\to 0} F_{\phi,\varepsilon}(\alpha)d\alpha$$
$$= \int_0^{2\pi} F_\phi(\alpha)d\alpha = \int_0^{2\pi}(\mathfrak{R}_\alpha f(s),\phi_\alpha(s))d\alpha. \quad (3.23)$$

In Eq. (3.23) we can see the connection between two definitions of the Radon transform on distributions. Some thoughts about the reverse implication of the Theorem 3.7 can be found in the Appendix A.2.

3.5.2 Calibration algorithm

Now we want to show how we can apply the moment conditions Eq. (3.17) from our DCC to calibrate with the specific calibration cage with unknown position in the world. This algorithm was described in our work in preparation [KD23]. We want to apply our algorithm in the case when projection data are truncated as in the fig. 3.1, but projections of markers from the calibration cage are not truncated. Suppose that we work with a lattice of s, but the system is moving, so we have errors. For P projections for unknown angles $\alpha_0,...,\alpha_{P-1}$ our measurements are

$$m_i(s) = \mathfrak{R}_{\alpha_i} f(s - s_{\alpha_i}), \quad (3.24)$$

where s_{α_i} are unknowns too.

We want to model the centers of markers with Dirac distributions, thus we want to understand what will be the projection of the center of one marker on the projected image. For one marker $f = \delta_{\vec{c}}$ we can calculate:

$$(\mathfrak{R}_\alpha\delta_{\vec{c}}(s),\phi(s)) = \langle\delta_{\vec{c}},\mathfrak{R}^*_\alpha\phi\rangle = \langle\delta_{\vec{c}}(\vec{x}),\phi(\vec{x}\cdot\vec{\theta}_\alpha)\rangle = \phi(\vec{c}\cdot\vec{\theta}_\alpha) = \delta_{\vec{c}\cdot\vec{\theta}_\alpha}(\phi). \quad (3.25)$$

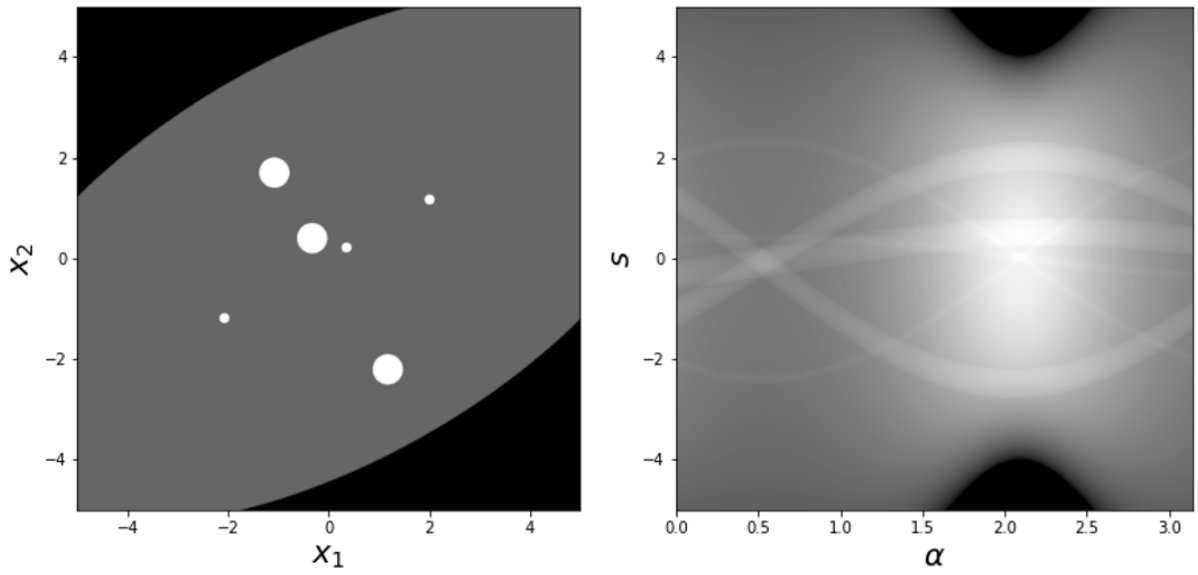

Figure 3.1: Left: truncated object and our specific calibration cage in $\mathbb{R}^2$, right: their 2D Radon transform.

We can see that the Radon transform for fixed α of one Dirac is again a Dirac in the projected image. Note that markers are usually small balls, not points. We can compute analytically the Radon transform for the indicator function $f_{R,\vec{c}}$ of the closed circular disc of radius $R > 0$ centered at $\vec{c} = (c_1, c_2)^T$ with Eq. (1.6) (the function corresponding to the ball of the marker). From this theoretical result we see that the support of $\mathfrak{R}_\alpha f_{R,\vec{c}}(s)$ is $[-R+s_0, R+s_0]$, where $s_0 = c_1 \cos\alpha + c_2 \sin\alpha = \vec{c} \cdot \vec{\theta}_\alpha$. Thus, the obtained projected Dirac $\delta_{\vec{c}\cdot\vec{\theta}_\alpha}$ is in the center s_0 of the α-projection of the marker.

Since we have few markers in the scene, then $f = \sum_{j=1}^n \delta_{\vec{c}_j}$, where n is the number of markers, $\vec{c}_j$ is the 2D center of the marker j. We admit that we can detect the centers of round markers for each projection, then in practice our measurements are of the form $m_i(s) = \sum_{j=1}^n \delta_{q_{ij}}(s)$.

With the linearity for $f = \sum_{j=1}^n \delta_{\vec{c}_j}$ we can compute that $\mathfrak{R}_{\alpha_i} f = \sum_{j=1}^n \delta_{\vec{c}_j \cdot \vec{\theta}_{\alpha_i}}$. So, the Radon transform for fixed α of finite sums of Diracs is well-defined, thus the moment conditions Eq. (3.17) from our DCC can be applied. We also can show the following beautiful property:

Theorem 3.8. *The center of mass c_α of the Radon transform for fixed α of the distribution f is the projection $\vec{c} \cdot \vec{\theta}_\alpha$ of the center of mass $\vec{c}$ of the distribution f.*

Proof. Let us define $\vec{c} = (\frac{c_{11}+\ldots+c_{n1}}{n}, \frac{c_{12}+\ldots+c_{n2}}{n})$, $c_\alpha = \frac{M_1(\alpha)}{M_0(\alpha)} = \frac{(\mathfrak{R}_\alpha f, s)}{(\mathfrak{R}_\alpha f, 1)}$, where $M_k(\alpha)$ denotes the moment of order k from Eq. (3.17). Then:

$$c_\alpha = \frac{(\mathfrak{R}_\alpha f, s)}{(\mathfrak{R}_\alpha f, 1)} = \frac{(\sum_{i=1}^n \delta_{\vec{c}_i \cdot \vec{\theta}_\alpha}, s)}{n} = \frac{\vec{c}_1 \cdot \vec{\theta}_\alpha + \ldots + \vec{c}_n \cdot \vec{\theta}_\alpha}{n} = \frac{(\vec{c}_1 + \ldots + \vec{c}_n) \cdot \vec{\theta}_\alpha}{n} = \vec{c} \cdot \vec{\theta}_\alpha.$$

□

For the calibration cage we chose two perpendicular lines of markers. An example of such calibration cage is presented in the fig. 3.1. Suppose that we perform the detection step separately: we have n centers of "horizontal" markers q_{ij}^h and n centers of "vertical" markers q_{ij}^v for each projection i. For our figure $n = 3$. Note that we chose different radii of markers for different groups in order to guarantee the detection step. As we can see in Eq. (1.6), the trace of the round marker has the same radius in the projected image. Thus, we can guarantee that we can correctly associate each marker with its group using radii of traces.

As explained in the section 2.4, in this detection task we have two forbidden integration directions or four forbidden angles, each of them is perpendicular to the line with markers. We must avoid these projection angles, because it's not possible for them to distinguish different markers from one group in the projected image.

Non-uniqueness of the solution. The complete explanation of non-uniqueness can be found in [BB00b; BB00a]. Here we only want to remind that after a rotation R_γ and a translation $\vec{t}$ of the object we obtain the same data, but for another parameters of the system, so we can't calibrate better than up to an Euclidean transformation. It was shown before for functions. So, with $f_{R_\gamma,\vec{t}}(\vec{x}) := f(R_\gamma\vec{x}+\vec{t})$, $R_\gamma := \begin{pmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{pmatrix}$, $\alpha' = \alpha+\gamma$ we have

$$\begin{aligned}\Re f_{R_\gamma,\vec{t}}(\alpha,s) &= \int_{-\infty}^{+\infty} f_{R_\gamma,\vec{t}}(s\vec{\theta}_\alpha + l\vec{\eta}_\alpha)dl = \int_{-\infty}^{+\infty} f(R_\gamma(s\vec{\theta}_\alpha + l\vec{\eta}_\alpha)+\vec{t})dl \\ &= \int_{-\infty}^{+\infty} f(sR_\gamma\vec{\theta}_\alpha + lR_\gamma\vec{\eta}_\alpha + \vec{t})dl = \int_{-\infty}^{+\infty} f(s\vec{\theta}_{\alpha'} + l\vec{\eta}_{\alpha'} + (\vec{\theta}_{\alpha'}\cdot\vec{t})\vec{\theta}_{\alpha'} + (\vec{\eta}_{\alpha'}\cdot\vec{t})\vec{\eta}_{\alpha'})dl \\ &= \int_{-\infty}^{+\infty} f((s+\vec{\theta}_{\alpha'}\cdot\vec{t})\vec{\theta}_{\alpha'} + l'\vec{\eta}_{\alpha'})dl' = \Re f(\alpha', s+\vec{\theta}_{\alpha'}\cdot\vec{t}).\end{aligned}$$

The equivalent form is

$$\Re f_{R_\gamma,\vec{t}}(\alpha-\gamma, s-\vec{\theta}_\alpha\cdot\vec{t}) = \Re f(\alpha,s). \tag{3.26}$$

This fact is also true for one Dirac $f = \delta_{\vec{c}}$ and for finite sums of Diracs. Let us prove this. We can define from the Dirac f a new distribution $f_{R_\gamma,\vec{t}}$: $\langle f_{R_\gamma,\vec{t}}(\vec{x}), \phi(\vec{x})\rangle = \langle f(\vec{x}), \phi(R_\gamma^{-1}(\vec{x}-\vec{t})\rangle$. We have this definition for distributions from the generalization of the same fact for functions. For function $f \in \mathscr{D}(\mathbb{R}^2)$ and $\phi \in \mathscr{E}(\mathbb{R}^2)$ with the change of variables from $\vec{x}$ to $\vec{x}'$ with $\vec{x}' = R_\gamma\vec{x}+\vec{t}$ and the determinant of the Jacobian matrix equal to 1:

$$(f(R_\gamma\vec{x}+\vec{t}), \phi(\vec{x})) = \int_{\mathbb{R}^2} f(R_\gamma\vec{x}+\vec{t})\phi(\vec{x})d\vec{x} = \int_{\mathbb{R}^2} f(\vec{x}')\phi(R_\gamma^{-1}(\vec{x}'-\vec{t}))d\vec{x}' = (f(\vec{x}'), \phi(R_\gamma^{-1}(\vec{x}'-\vec{t})). \tag{3.27}$$

We can show that $f_{R_\gamma,\vec{t}}$ is a distribution $\delta_{R_\gamma^{-1}(\vec{c}-\vec{t})}$ for $f = \delta_{\vec{c}}$:

$$\langle f_{R_\gamma,\vec{t}}(\vec{x}), \phi(\vec{x})\rangle = \langle f(\vec{x}), \phi(R_\gamma^{-1}(\vec{x}-\vec{t})\rangle = \langle \delta_{\vec{c}}(\vec{x}), \phi(R_\gamma^{-1}(\vec{x}-\vec{t})\rangle = \phi(R_\gamma^{-1}(\vec{c}-\vec{t})) = \langle \delta_{R_\gamma^{-1}(\vec{c}-\vec{t})}(\vec{x}), \phi(\vec{x})\rangle. \tag{3.28}$$

Then for the rotation and the translation of one Dirac:

$$(\mathfrak{R}_\alpha(\delta_{\vec{c}})_{R_\gamma,\vec{t}}(s),\phi(s)) = \langle(\delta_{\vec{c}})_{R_\gamma,\vec{t}}(\vec{x}),\phi(\vec{x}\cdot\vec{\theta}_\alpha)\rangle = \langle\delta_{R_\gamma^{-1}(\vec{c}-\vec{t})}(\vec{x}),\phi(\vec{x}\cdot\vec{\theta}_\alpha)\rangle = \phi((R_\gamma^{-1}\vec{c}-R_\gamma^{-1}\vec{t})\cdot\vec{\theta}_\alpha)$$
$$= \phi((\vec{c}-\vec{t})\cdot R_\gamma\vec{\theta}_\alpha) = \phi((\vec{c}-\vec{t})\cdot\vec{\theta}_{\alpha'}) = (\delta_{\vec{c}\cdot\vec{\theta}_{\alpha'}}(s),\phi(s-\vec{t}\cdot\vec{\theta}_{\alpha'}))$$
$$= (\mathfrak{R}_{\alpha'}\delta_{\vec{c}}(s),\phi(s-\vec{t}\cdot\vec{\theta}_{\alpha'})) = (\mathfrak{R}_{\alpha'}\delta_{\vec{c}}(s+\vec{t}\cdot\vec{\theta}_{\alpha'}),\phi(s)).$$

The equivalent form is

$$\mathfrak{R}_{\alpha-\gamma}(\delta_{\vec{c}})_{R_\gamma,\vec{t}}(s-\vec{\theta}_\alpha\cdot\vec{t}) = \mathfrak{R}_\alpha\delta_{\vec{c}}(s). \tag{3.29}$$

So, we have the same projection data after the rotation and the translation of a Dirac with another parameters of the system, thus with the access to projection data only we have non-uniqueness of the self-calibration problem, we can't say from which configuration of the system we obtained our data. We can try to find only one configuration of the system among all possibilities. Since we have these degrees of freedom for functions and for Diracs (the solution is up to a rotation and a translation), we can fix the coordinate system in the scene as we want, and by this we wish to find just one solution among all possible solutions.

Our calibration algorithm requires few coordinate systems. Firstly, let us define the aligned coordinate system: suppose that the horizontal coordinate axis is parallel to the "horizontal" line of markers, the vertical coordinate axis - to the "vertical" line of markers, the center of mass of the system of all markers is in $(0,0)^T$. We will estimate shifts in this coordinate system. Secondly, we will also consider two coordinate systems with the center in the center of mass of each group of markers and same direction of axes, so the first coordinate system with the center in the center of mass of the "horizontal" group, the second - in the center of mass of the "vertical" group. We will estimate angles in these coordinate systems. Each coordinate system differs from the aligned coordinate system by a translation, so in them we will have the same set of projection angles. Thus, the calibration parameters will be estimated in relation to the aligned coordinate system. In the following, we will use superscript $l = h$ or $l = v$, h will correspond to the "horizontal" group of markers, v - to the "vertical" group of markers.

Shift correction. Here we want to adapt the logic of the self-calibration algorithm from [DS14] for functions to Diracs to find all s_{α_i} as defined in Eq. (3.24). If we consider all markers, so $2n = 6$, we can compute the moments of order 0 and 1 for $i = 0,...,P-1$:

$$M_0(i) = (m_i(s),1) = (\mathfrak{R}_{\alpha_i}f(s-s_{\alpha_i}),1) = (\sum_{j=1}^{2n}\delta_{\vec{c}_j\cdot\vec{\theta}_{\alpha_i}}(s-s_{\alpha_i}),1) = (\sum_{j=1}^{2n}\delta_{\vec{c}_j\cdot\vec{\theta}_{\alpha_i}}(s),1) = 2n, \tag{3.30}$$

$$M_1(i) = (m_i(s),s) = (\mathfrak{R}_{\alpha_i}f(s-s_{\alpha_i}),s) = (\mathfrak{R}_{\alpha_i}f(s),s+s_{\alpha_i}) = (\sum_{j=1}^{2n}\delta_{\vec{c}_j\cdot\vec{\theta}_{\alpha_i}},s+s_{\alpha_i}) = \sum_{j=1}^{2n}\vec{c}_j\cdot\vec{\theta}_{\alpha_i}+s_{\alpha_i}2n. \tag{3.31}$$

The last equation gives us

$$s_{\alpha_i} = \frac{M_1(i) - \sum_{j=1}^{2n} \vec{c}_j \cdot \vec{\theta}_{\alpha_i}}{2n}. \tag{3.32}$$

We plan to calibrate in the aligned coordinate system. In this system the center of mass of all markers in the scene is in zero and $\sum_{j=1}^{2n} \vec{c}_j = \vec{0}$, then

$$s_{\alpha_i} = \frac{M_1(i)}{2n} = \frac{(\sum_{j=1}^{2n} \delta_{q_{ij}}(s), s)}{2n} = \frac{\sum_{j=1}^{2n} q_{ij}}{2n}, \tag{3.33}$$

where we can compute $M_1(i)$ from the detected centers of all markers q_{ij}.

Note that we can apply the same logic for each group of markers separately and compute $s^l_{\alpha_i}$, $l \in \{h, v\}$. For each group l we can have:

$$s^l_{\alpha_i} = \frac{M^l_1(i) - \sum_{j=1}^{n} \vec{c}^l_j \cdot \vec{\theta}_{\alpha_i}}{n}. \tag{3.34}$$

Now we can fix coordinate systems connected to each group of markers and compute two corresponding set of shifts. Let us fix for $l = h$ a coordinate system in the center of mass of the first group of markers, x_1 axis should be aligned with the first line of markers. After let us fix the coordinate system in the center of mass of the second group $l = v$ of markers, x_2 axis should be aligned with the second line of markers. Then we have:

$$s^l_{\alpha_i} = \frac{M^l_1(i)}{n} = \frac{(\sum_{j=1}^{n} \delta_{q^l_{ij}}(s), s)}{n} = \frac{\sum_{j=1}^{n} q^l_{ij}}{n}, \tag{3.35}$$

where q^l_{ij} states for the detected center of the marker j of the line l in the projection i. For the "horizontal" and "vertical" groups of markers we perform the shift corrections $s^h_{\alpha_i}$ and $s^v_{\alpha_i}$ separately. So, during the next step of finding of angles we will use two sets of corrected input data in two different coordinate systems.

Finding of angles. Here we start with the measurements $m^l_i(s) = \mathfrak{R}_{\alpha_i} f^l(s)$. We need to find unknown α_i. Note that we want to propose a new algorithm, the previous algorithm [DS14] designed for non-truncated data covered only shift corrections. We can compute the moments of order 2 for each l, $l \in \{h, v\}$:

$$\begin{aligned} M^l_2(i) = (m^l_i(s), s^2) = (\mathfrak{R}_{\alpha_i} f^l(s), s^2) = (\sum_{j=1}^{n} \delta_{\vec{c}^l_j \cdot \vec{\theta}_{\alpha_i}}, s^2) = \sum_{j=1}^{n} (\vec{c}^l_j \cdot \vec{\theta}_{\alpha_i})^2 = \sum_{j=1}^{n} (c^l_{j1} \cos\alpha_i + c^l_{j2} \sin\alpha_i)^2 \\ = \sum_{j=1}^{n} ((c^l_{j1})^2 \cos^2\alpha_i + 2c^l_{j1}c^l_{j2} \cos\alpha_i \sin\alpha_i + (c^l_{j2})^2 \sin^2\alpha_i) \\ = a^l_{20} \cos^2\alpha_i + 2a^l_{11} \cos\alpha_i \sin\alpha_i + a^l_{02} \sin^2\alpha_i, \end{aligned} \tag{3.36}$$

where $a^l_{20} = \sum_{j=1}^{n} (c^l_{j1})^2$, $a^l_{11} = \sum_{j=1}^{n} c^l_{j1}c^l_{j2}$, $a^l_{02} = \sum_{j=1}^{n} (c^l_{j2})^2$. We know that $\vec{c}^h_j = (c^h_{j1}, 0)^T$ for the first group of markers in the corresponding coordinate system, $\vec{c}^v_j = (0, c^v_{j2})^T$ for the second group of

markers in the corresponding coordinate system. Thus, we can simplify the second moments:

$$M_2^h(i) = a_{20}^h \cos^2 \alpha_i, \qquad M_2^v(i) = a_{02}^v \sin^2 \alpha_i. \tag{3.37}$$

As before we can compute from the available data q_{ij}^l and know the moments $M_2^l(i) = (\sum_{j=1}^n \delta_{q_{ij}^l}(s), s^2) = \sum_{j=1}^n (q_{ij}^l)^2 > 0$. For two different angles α_0 and α_1 from $(0, \pi/2)$ from the system of 4 equations we can compute 4 unknowns a_{20}^h, a_{02}^v, α_0, α_1:

$$\begin{cases} M_2^h(0) = a_{20}^h \cos^2 \alpha_0 \\ M_2^v(0) = a_{02}^v \sin^2 \alpha_0 \\ M_2^h(1) = a_{20}^h \cos^2 \alpha_1 \\ M_2^v(1) = a_{02}^v \sin^2 \alpha_1 \end{cases} \Leftrightarrow \begin{cases} M_2^h(0) = \cos^2 \alpha_0 \cdot M_2^h(1) / \cos^2 \alpha_1 \\ M_2^v(0) = \sin^2 \alpha_0 \cdot M_2^v(1) / \sin^2 \alpha_1 \\ a_{20}^h = M_2^h(1) / \cos^2 \alpha_1 \\ a_{02}^v = M_2^v(1) / \sin^2 \alpha_1 \end{cases}. \tag{3.38}$$

From the second equation

$$\cos^2 \alpha_1 = 1 - \sin^2 \alpha_1 = 1 - \frac{M_2^v(1) \sin^2 \alpha_0}{M_2^v(0)} = \frac{M_2^v(0) - M_2^v(1) \sin^2 \alpha_0}{M_2^v(0)},$$

then from the first equation

$$\begin{aligned} M_2^h(0) \frac{M_2^v(0) - M_2^v(1) \sin^2 \alpha_0}{M_2^v(0)} &= M_2^h(1) \cos^2 \alpha_0 \Rightarrow \\ M_2^h(0) M_2^v(0) - M_2^h(0) M_2^v(1) \sin^2 \alpha_0 &= M_2^h(1) M_2^v(0) (1 - \sin^2 \alpha_0) \Rightarrow \\ (M_2^h(0) M_2^v(1) - M_2^h(1) M_2^v(0)) \sin^2 \alpha_0 &= M_2^h(0) M_2^v(0) - M_2^h(1) M_2^v(0), \end{aligned}$$

then

$$\sin \alpha_0 = \left(\frac{M_2^h(0) M_2^v(0) - M_2^h(1) M_2^v(0)}{M_2^h(0) M_2^v(1) - M_2^h(1) M_2^v(0)} \right)^{1/2} \tag{3.39}$$

uniquely defines $\alpha_0 \in (0, \pi/2)$. Here $M_2^h(0) M_2^v(1) - M_2^h(1) M_2^v(0) \neq 0$. Indeed, $q_{ij}^l = \vec{c}_j^{\,l} \cdot \vec{\theta}_{\alpha_i}$ and

$$\begin{aligned} M_2^h(0) M_2^v(1) - M_2^h(1) M_2^v(0) &= \sum_{j=1}^n (\vec{c}_j^{\,h} \cdot \vec{\theta}_{\alpha_0})^2 \cdot \sum_{j=1}^n (\vec{c}_j^{\,v} \cdot \vec{\theta}_{\alpha_1})^2 - \sum_{j=1}^n (\vec{c}_j^{\,h} \cdot \vec{\theta}_{\alpha_1})^2 \cdot \sum_{j=1}^n (\vec{c}_j^{\,v} \cdot \vec{\theta}_{\alpha_0})^2 \\ &= \sum_{j=1}^n (c_{j1}^h \cos \alpha_0)^2 \cdot \sum_{j=1}^n (c_{j2}^v \sin \alpha_1)^2 - \sum_{j=1}^n (c_{j1}^h \cos \alpha_1)^2 \cdot \sum_{j=1}^n (c_{j2}^v \sin \alpha_0)^2 \\ &= \sum_{j=1}^n (c_{j1}^h)^2 \sum_{j=1}^n (c_{j2}^v)^2 (\cos^2 \alpha_0 \sin^2 \alpha_1 - \cos^2 \alpha_1 \sin^2 \alpha_0) \\ &= -\sum_{j=1}^n (c_{j1}^h)^2 \sum_{j=1}^n (c_{j2}^v)^2 \sin(\alpha_0 - \alpha_1) \sin(\alpha_0 + \alpha_1) \neq 0, \end{aligned}$$

because we have two positive sums, $\sin(\alpha_0 - \alpha_1)$ and $\sin(\alpha_0 + \alpha_1)$ aren't zeros, because $\alpha_0 \neq \alpha_1$ and they are in the same quadrant.

Then the coefficients can be computed

$$a^h_{20} = \frac{M^h_2(0)}{\cos^2\alpha_0}, \qquad a^v_{02} = \frac{M^v_2(0)}{\sin^2\alpha_0}. \tag{3.40}$$

From the moments of order 3

$$\begin{aligned} M^l_3(i) = (m^l_i(s), s^3) = (\mathfrak{R}_{\alpha_i} f^l(s), s^3) = (\sum_{j=1}^{n} \delta_{\vec{c}^{\,l}_j \cdot \vec{\theta}_{\alpha_i}}, s^3) = \sum_{j=1}^{n} (\vec{c}^{\,l}_j \cdot \vec{\theta}_{\alpha_i})^3 = \sum_{j=1}^{n} (c^l_{j1}\cos\alpha_i + c^l_{j2}\sin\alpha_i)^3 \\ = \sum_{j=1}^{n} ((c^l_{j1})^3\cos^3\alpha_i + 3(c^l_{j1})^2 c^l_{j2}\cos^2\alpha_i\sin\alpha_i + 3c^l_{j1}(c^l_{j2})^2\cos\alpha_i\sin^2\alpha_i + (c^l_{j2})^3\sin^3\alpha_i) \\ = a^l_{30}\cos^3\alpha_i + 3a^l_{21}\cos^2\alpha_i\sin\alpha_i + 3a^l_{12}\cos\alpha_i\sin^2\alpha_i + a^l_{03}\sin^3\alpha_i \end{aligned} \tag{3.41}$$

we have

$$M^h_3(i) = a^h_{30}\cos^3\alpha_i, \qquad M^v_3(i) = a^v_{03}\sin^3\alpha_i, \tag{3.42}$$

where $a^h_{30} = \sum_{j=1}^{n}(c^h_{j1})^3$, $a^v_{03} = \sum_{j=1}^{n}(c^v_{j2})^3$. It's easy to obtain a^h_{30}, a^v_{03}:

$$a^h_{30} = \frac{M^h_3(0)}{\cos^3\alpha_0}, \qquad a^v_{03} = \frac{M^v_3(0)}{\sin^3\alpha_0} \tag{3.43}$$

and uniquely compute α_i from the second and the third moments:

$$\cos\alpha_i = \frac{a^h_{20}M^h_3(i)}{a^h_{30}M^h_2(i)}, \qquad \sin\alpha_i = \frac{a^v_{02}M^v_3(i)}{a^v_{03}M^v_2(i)}, \tag{3.44}$$

where $M^l_3(i)$ can be computed with q^l_{ij} as $M^l_3(i) = (\sum_{j=1}^{n}\delta_{q^l_{ij}}(s), s^3) = \sum_{j=1}^{n}(q^l_{ij})^3$. Note that to apply these formulas we should have $a^h_{30} \neq 0$ and $a^v_{03} \neq 0$. We should take it into account during the design of the calibration cage.

Thus, our algorithm consists of Eq. (3.33), (3.39), (3.40), (3.43) and (3.44). We designed the analytical procedure to compute all unknown acquisition parameters. Our approach is based on the local information of the centers of non-truncated markers, the rest of data can be truncated. So, the algorithm is suitable for the situation of data truncation.

Numerical experiments. We chose the calibration cage as it's presented in the fig. 3.1. The coordinates of the centers of the markers in the aligned coordinate system in cm are:

- $(-2.4, 0)^T$, $(0.4, 0)^T$, $(2.3, 0)^T$ for the horizontal line,
- $(-0.1, -2.5)^T$, $(-0.1, 0.5)^T$, $(-0.1, 2)^T$ for the vertical line.

Thus, our calibration cage satisfies the conditions: we have two perpendicular lines of markers, the sums of cubes of coordinates aren't zeros, thus $a^h_{30} \neq 0$ and $a^v_{03} \neq 0$. The initial coordinate system is produced from the aligned coordinate system by the rotation of $\pi/6$. So, we operate with the following coordinates of the centers of the markers in the initial coordinate system:

- $(-2.079,-1.200)^T$, $(0.346,0.200)^T$, $(1.992,1.150)^T$ for the first line,
- $(1.163,-2.215)^T$, $(-0.337,0.383)^T$, $(-1.087,1.682)^T$ for the second line.

The coordinates above are presented with 3 digits precision. We simulated the calibration parameters to be estimated.

1. We took $P = 80$ random different projection angles in the initial coordinate system in $[\pi/6, \pi + \pi/6]$ as true values, but we avoided the forbidden directions discussed before and in the section 2.4. We must avoid integration directions perpendicular to the line with markers, because it's not possible for them to distinguish markers from the same group in the projected image. To avoid these directions, the first half of projections was from $(\pi/6, \pi/2 + \pi/6)$, the second half - from $(\pi/2 + \pi/6, \pi + \pi/6)$.
2. We had the grid on $s \in [-5,5]$ with the step 0.01. We simulated s_{α_i} with the random uniform noise on the interval $[-0.05, 0.05)$.

In realistic settings, the information that we have is usually the combination of the projections of the truncated body and non-truncated markers, but in our experiment we only used as an input the information about the centers of the traces of the markers on projected images. Thus, as an input we had two arrays for each marker line with projections of markers of the size $(3, P)$ each.

All computations were performed in Python 3. If we know projections q_{ij}^l exactly, then our algorithm was almost perfect (see the first line in the table 3.1). Note that we subtracted $\pi/6$ from the true angles to be able to compare them with the estimated angles, because we found estimations in the aligned coordinate system which differs from the initial one with true angles. For the case of the detection of q_{ij}^l as in the realistic setting, we have some errors after the detection of q_{ij}^l. We modelled these detection errors as a Gaussian noise $N(0,\sigma)$ added to q_{ij}^l. Here $\sigma = 0.01 \cdot Nl$, where Nl is the noise level, 0.01 is the pixel size of the initial image in cm. In the table 3.1 we can also see the results of noisy experiments. We present the mean absolute errors of 100 realisations of the Gaussian noise for each Nl. Thus, for shifts we computed

$$\frac{1}{100}\sum_{k=1}^{100}\frac{1}{P}\sum_{i=0}^{P-1}\left|\hat{s}_{\alpha_i,k} - s_{\alpha_i}\right|, \tag{3.45}$$

where s_{α_i} is the true value for the projection i, $\hat{s}_{\alpha_i,k}$ is the estimation with our algorithm of the shift for the projection i with the realisation of the Gaussian noise number k added to projections q_{ij}^l. For projection angles we computed

$$\frac{1}{100}\sum_{k=1}^{100}\frac{1}{P}\sum_{i=0}^{P-1}\left|\hat{\alpha}_{i,k} - \alpha_i\right|, \tag{3.46}$$

where α_i is the true value for the projection i, $\hat{\alpha}_{i,k}$ is the estimation with our algorithm of the projection angle i with the realisation of the Gaussian noise number k added to projections q_{ij}^l. We show that the errors are proportional to Nl.

In this section we presented our definition for fixed α of the 2D Radon transform on distributions, we proved DCC for our definition. We also showed that our DCC combined with the proper calibration cage allow to perform the full analytical self-calibration with truncated data when markers are not truncated. In the next chapter we will extend this approach to the fan-beam geometry with sources on a line.

Noise level *Nl*	**Noise std in cm**	**Error for shifts in cm**	**Error for angles in rad**
0%	0	5.08×10^{-17}	3.15×10^{-16}
10%	0.001	3.24×10^{-4}	2.20×10^{-3}
50%	0.005	1.61×10^{-3}	1.10×10^{-2}
100%	0.01	3.26×10^{-3}	2.31×10^{-2}
200%	0.02	6.53×10^{-3}	6.52×10^{-2}

Table 3.1: Mean absolute errors of calibration parameters (shifts and angles) for numerical experiments with non-noisy and noisy projections.

4 Calibration with DCC on distributions for fan-beam transform

4.1 French summary of the chapter

Dans ce chapitre nous appliquons la même logique pour la géométrie en éventail avec des sources alignées que pour la géométrie parallèle 2D. Nous commençons par la définition de la transformée en éventail sur les distributions avec des sources alignées pour une position de source fixe. Après nous pouvons généraliser les DCC de [Cla13] pour les fonctions aux distributions. Ces nouvelles DCC avec les moments d'ordre 0, 1 et 2 et la mire d'étalonnage de deux ensembles de marqueurs sur deux lignes parallèles nous donnent la procédure de calibrage analytique. Cette procédure nous aide à construire l'algorithme en trois étapes simples pour estimer les paramètres d'étalonnage et les positions inconnues des centres de masse des lignes de la mire. Nous supposons que nous connaissons la structure de la mire d'étalonnage. Puisque nos DCC ne fonctionnent qu'avec les marqueurs non tronqués, nous pouvons appliquer notre procédure dans la situation où les données complètes sont tronquées.

Nous testons numériquement notre algorithme pour estimer les paramètres d'étalonnage inconnus (les positions de la source $\hat{\lambda}_i$ et les décalages du détecteur $\hat{y}_{\lambda_i}$) avec et sans bruit. Nous discutons de la non-unicité de la solution de notre problème de calibration. Nous expliquons quelle solution de la classe de toutes les solutions nous trouvons avec notre algorithme (la solution dans le système de coordonnées alignées).

Pour comprendre comment les erreurs de calibration peuvent influencer le processus de la reconstruction, nous calculons l'erreur suivante. Puisque nous avons les estimations pour chaque projection i dans le système de coordonnées aligné $\hat{\lambda}_i$ et $\hat{y}_{\lambda_i}$, dans le système de coordonnées initial, ils peuvent être recalculés comme $\hat{\lambda}'_i = \hat{\lambda}_i + \lambda_0$ et $\hat{y}'_{\lambda_i} = \hat{y}_{\lambda_i} + y_{\lambda_0}$. Connaissant la position de la source $\hat{\lambda}'_i$ et la projection du marqueur 2D $\vec{c}^{\,l}_j$ corrigée $\hat{q}^l_{ij} = q^l_{ij} - \hat{y}'_{\lambda_i}$ pour chaque ligne avec les marqueurs l et chaque marqueur j nous pouvons tracer la ligne reliant $\hat{\lambda}'_i$ et $\hat{q}^l_{ij}$. Si nous avons l'estimation exacte des paramètres d'étalonnage, la distance entre le marqueur 2D $\vec{c}^{\,l}_j$ dans le système de coordonnées initial et cette ligne est essentiellement nulle, car $\hat{q}^l_{ij}$ doit être la projection perspective 2D exacte de $\vec{c}^{\,l}_j$ sur cette ligne. Nous pouvons voir ce comportement pour les simulations sans bruit. Dans le cas de données d'entrée bruitées, on observe le transfert des erreurs d'estimation des paramètres de calibration au décalage de la droite de projection par rapport à sa vraie position dans le plan. Ce décalage influencera également le processus de la reconstruction.

Notre algorithme pour la géométrie parallèle 2D du chapitre précédent et l'algorithme pour la géométrie en éventail de ce chapitre sont décrits dans notre article en cours [KD23].

4.2 Fan-beam transform on distributions with sources on a line

4.2.1 Definition and DCC

In this section about the fan-beam transform with sources on a line, we refer to the geometry presented in the fig. 1.3. The fan-beam transform on functions for this case was described by the integral model Eq. (1.9).

Our definition for fixed λ. Let us denote $\mathfrak{D}_\lambda f(y) := \mathfrak{D} f(\lambda, y)$ for $f \in \mathscr{D}_2$ with support in $X_2 = (D_1, D_2) \times \mathbb{R}$, $0 < D_1 < D_2 < D$. We can consider $\mathfrak{D}_\lambda f$ as a function of one variable y. Firstly, we want to define the adjoint operator $\mathfrak{D}_\lambda^*$ for functions from $\mathscr{E}_1$:

$$\mathfrak{D}_\lambda^* \phi(\vec{x}) := \frac{1}{D - x_1} \phi\left(\frac{x_2 D - x_1 \lambda}{D - x_1}\right). \tag{4.1}$$

For the described functions f and ϕ we have:

$$\begin{aligned}(\mathfrak{D}_\lambda f, \phi) &= \int_{-\infty}^{+\infty} \mathfrak{D}_\lambda f(y)\phi(y)dy = \int_{-\infty}^{+\infty}\int_0^{+\infty} f(D - lD, \lambda + ly - l\lambda)dl\phi(y)dy \\ &= \frac{1}{D}\int_{-\infty}^{+\infty}\int_{-\infty}^{D} f\left(u, \frac{u\lambda}{D} + \frac{(D-u)y}{D}\right)du\phi(y)dy = \int_{-\infty}^{D}\int_{-\infty}^{+\infty} f(u,v)\phi\left(\frac{vD - u\lambda}{D-u}\right)\frac{1}{D-u}dvdu \\ &= \int_{D_1}^{D_2}\int_{-\infty}^{+\infty} f(u,v)\phi\left(\frac{vD - u\lambda}{D-u}\right)\frac{1}{D-u}dvdu = \langle f, \mathfrak{D}_\lambda^*\phi\rangle, \end{aligned} \tag{4.2}$$

where $(\cdot,\cdot)$ is the scalar product in $L^2(\mathbb{R})$, $\langle\cdot,\cdot\rangle$ is the scalar product in $L^2(X_2)$, we used the change of variables $u = D - lD$, $dl = -\frac{du}{D}$, $\lambda + l(y-\lambda) = \frac{u\lambda}{D} + \frac{(D-u)y}{D}$ and $v = \frac{u\lambda}{D} + \frac{(D-u)y}{D}$, $dv = \frac{D-u}{D}dy$, $y = \frac{vD-u\lambda}{D-u}$.

If $\phi \in \mathscr{E}_1$, then from Eq. (4.1) we have $\mathfrak{D}_\lambda^*\phi \in \mathscr{E}(X_2)$ and can define:

Definition 4.1. *A fan-beam transform on a line $\mathfrak{D}_\lambda f$ for $f \in \mathscr{E}'(X_2)$ is a bounded linear functional acting on the space $\mathscr{E}_1$ of test functions according to*

$$(\mathfrak{D}_\lambda f, \phi) = \langle f, \mathfrak{D}_\lambda^*\phi\rangle. \tag{4.3}$$

We justify why this definition is properly formulated in the Appendix A.3.

If we model markers with Dirac distributions, then we can obtain weighted Diracs in projections. For $f = \delta_{\vec{c}}$ we have:

$$\begin{aligned}(\mathfrak{D}_\lambda \delta_{\vec{c}}(y), \phi(y)) = \langle \delta_{\vec{c}}, \mathfrak{D}_\lambda^*\phi\rangle &= \left\langle \delta_{\vec{c}}(\vec{x}), \phi\left(\frac{x_2 D - x_1\lambda}{D - x_1}\right)\frac{1}{D-x_1}\right\rangle \\ &= \phi\left(\frac{c_2 D - c_1\lambda}{D-c_1}\right)\frac{1}{D-c_1} = \frac{1}{D-c_1}\delta_{\tilde{c}}(\phi), \qquad \tilde{c} = \frac{c_2 D - c_1\lambda}{D-c_1}. \end{aligned} \tag{4.4}$$

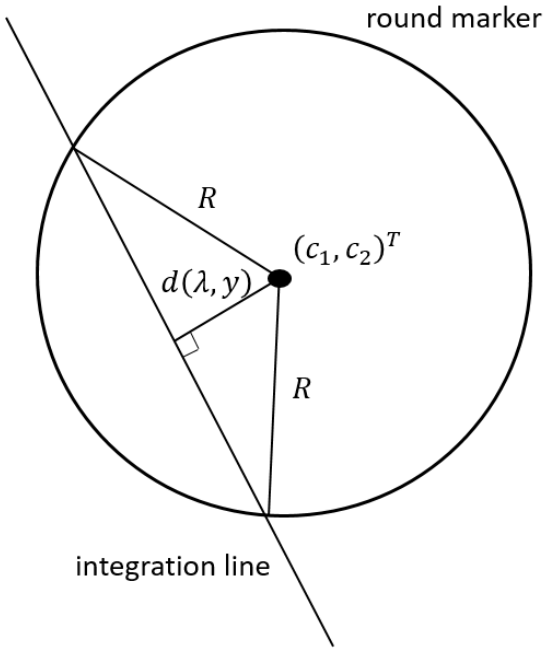

Figure 4.1: Intersection of the integration line with the round marker.

Note that this $\tilde{c}$ is also the usual 2D perspective projection of the point $\vec{c}_j$, see [DC19].

If we consider the non-weighted fan-beam transform with sources on a line with the normalized vector $\vec{\zeta}$ in the definition Eq. (1.7), then we can analytically calculate the non-weighted fan-beam transform of the indicator function of a disk. It's interesting, since a marker can be considered as a small disk. The fan-beam integral in Eq. (1.7) over the integration line $(y-\lambda)x_1 + Dx_2 - Dy = 0$ for the indicator function of a disk is equal to the length of the chord in the intersection of the integration line with the disk of the center $(c_1, c_2)^T$ and the radius R. The distance between the center of the disk and this integration line can be expressed with the standard formula from geometry as $d(\lambda, y) = \frac{|(y-\lambda)c_1 + Dc_2 - Dy|}{\sqrt{(y-\lambda)^2 + D^2}}$. Thus, the non-weighted fan-beam transform of the indicator function of a disk is the length of the chord in the intersection that can be found from the Pythagorean theorem (fig. 4.1):

$$\mathfrak{D} f_{R,\vec{c}}(\lambda, y) = \begin{cases} 2\sqrt{R^2 - d^2(\lambda, y)} & \text{if } d(\lambda, y) \leq R \\ 0 & \text{otherwise.} \end{cases} \tag{4.5}$$

Let us find for which y $d^2(\lambda, y) = R^2$. We need to solve

$$((y-\lambda)c_1 + Dc_2 - Dy)^2 = R^2((y-\lambda)^2 + D^2) \Leftrightarrow Ay^2 + By + C = 0,$$

where $A = (c_1 - D)^2 - R^2$ and it's positive, because the marker is between the source line and the detector line, $B = 2(c_1 - D)(c_2 D - c_1\lambda) + 2\lambda R^2$, $C = (c_2 D - c_1\lambda)^2 - R^2\lambda^2 - R^2 D^2$. The discriminant can be computed $\Delta = \frac{B^2}{4} - AC = D^2R^2(c_1^2 - 2c_1 D + D^2 + c_2^2 - 2c_2\lambda + \lambda^2 - R^2) = D^2R^2((c_1 - D)^2 +$

$(\lambda - c_2)^2 - R^2)$. It's positive, because the marker is between the source line and the detector line. So, we have 2 different solutions $y_1 = \frac{-B/2-\sqrt{\Delta}}{A}$ and $y_2 = \frac{-B/2+\sqrt{\Delta}}{A}$. The support of the non-weighted fan-beam transform $\mathfrak{D}_\lambda f_{R,\vec{c}}(y)$ is (y_1, y_2), thus we can say that $-\frac{B}{2A}$ is the center of the marker's trace on the projected image. In the limit we have

$$\lim_{R\to 0} \frac{-B}{2A} = \lim_{R\to 0} \frac{-2(c_1 - D)(c_2 D - c_1\lambda) - 2\lambda R^2}{2((c_1 - D)^2 - R^2)} = \frac{c_2 D - c_1\lambda}{D - c_1} = \tilde{c}, \tag{4.6}$$

so we can approximate $\tilde{c}$ with the detected center of the marker's trace on the projected image when R is small.

DCC for our definition for fixed λ. As in the case of the 2D Radon transform, we plan to use only necessary consistency conditions. For that, we want to prove the generalisation to distributions of known for functions DCC from the Theorem 1.2 mentioned in the first section:

Theorem 4.1. *If $f \in \mathscr{E}'(X_2)$, $g_\lambda := \mathfrak{D}_\lambda f$ is the fan-beam transform on a line of f for fixed λ, then:*

1. $g_\lambda \in \mathscr{E}'_1$,
2. *for $k = 0, 1, 2, \ldots$ we have the necessary moment conditions:*

$$(g_\lambda(y), y^k) = \mathscr{P}_k(\lambda), \tag{4.7}$$

where $\mathscr{P}_k(\lambda)$ is a polynomial of degree at most k in λ.

Proof. The first point is showed in the Appendix A.3. Now we prove the moment conditions. For $y^k \in \mathscr{E}_1$

$$\begin{aligned}(\mathfrak{D}_\lambda f(y), y^k) = \langle f(\vec{x}), \mathfrak{D}^*_\lambda(y^k)\rangle &= \left\langle f(\vec{x}), \left(\frac{x_2 D - x_1\lambda}{D - x_1}\right)^k \frac{1}{D - x_1}\right\rangle \\ &= \left\langle f(\vec{x}), \frac{1}{(D-x_1)^{k+1}} \sum_{i=0}^{k} C(k,i)(x_2 D)^{k-i}(-x_1\lambda)^i \right\rangle \\ &= \sum_{i=0}^{k} C(k,i) \left\langle f(\vec{x}), \frac{(x_2 D)^{k-i}(-x_1)^i}{(D-x_1)^{k+1}} \right\rangle \lambda^i = \mathscr{P}_k(\lambda).\end{aligned}$$

□

4.2.2 Calibration algorithm

Now we want to show how we can apply the moment conditions Eq. (4.7) from our DCC to calibrate with the specific calibration cage with unknown position in the world. This algorithm was also described in our work [KD23]. The purpose of this algorithm is to calibrate with truncated data as in the fig. 4.2. Suppose that we work with a lattice of y, but the system is moving, so we have errors. For

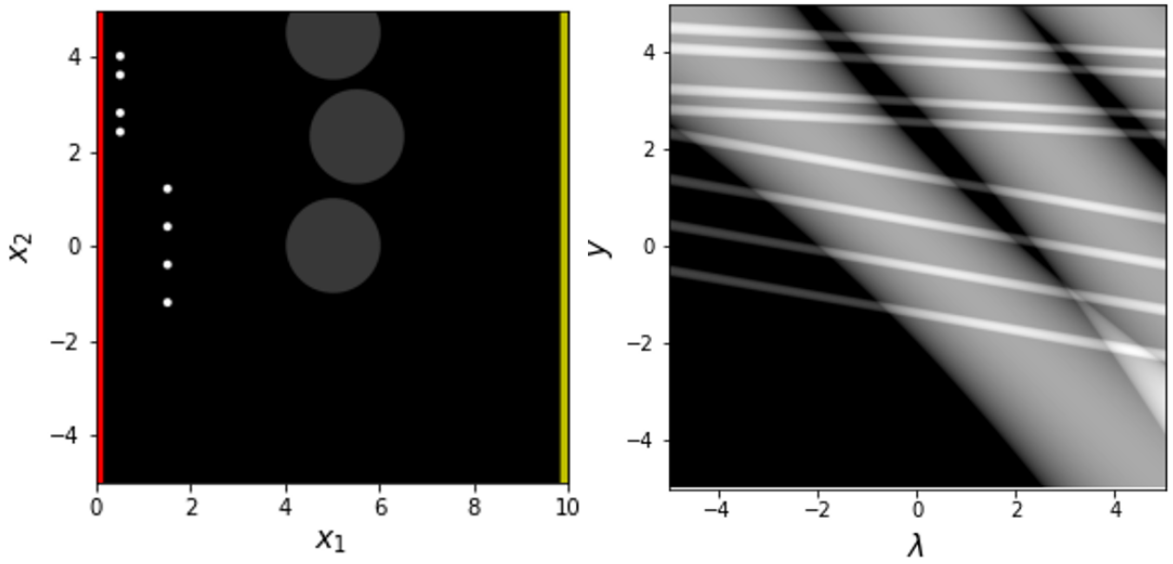

Figure 4.2: Left: truncated object (the union of big disks) and our specific calibration cage (8 small disks) in $\mathbb{R}^2$, the detector is in the red line, the line with source positions is yellow, right: their fan-beam transform.

P projections for unknown source positions $\lambda_0, ..., \lambda_{P-1}$ our measurements are $m_i(y) = \mathfrak{D}_{\lambda_i} f(y - y_{\lambda_i})$, where y_{λ_i} are unknowns too.

If our calibration cage is modelled by $f = \sum_{j=1}^{n} \delta_{\vec{c}_j}$, then $\mathfrak{D}_{\lambda_i} f = \sum_{j=1}^{n} \frac{1}{D - c_{j1}} \delta_{\tilde{c}(\vec{c}_j, \lambda_i, D)}$, where $\tilde{c}(\vec{c}_j, \lambda_i, D)$ defined by Eq. (4.4) is the projection onto the detector of the 2D center $\vec{c}_j$ of the marker j. We again want to solve the calibration task with the specific calibration cage (fig. 4.2) consisting of two lines of 4 markers. These two lines should be parallel to the detector and source lines. Also, we want to place the calibration cage as close as possible to the detector in order to have non-truncated projections of the markers. Moreover, the calibration cage should be attached to the support of the main object and move with the object (if there is any movement of the object). The patterns of two lines are known, but we don't know positions of these lines in the scene. Suppose that the first line has the coordinates of the centers of markers $\vec{c}_1^1 = (C_1, p_1 - k_1 L)^T$, $\vec{c}_2^1 = (C_1, p_1 - L)^T$, $\vec{c}_3^1 = (C_1, p_1 + L)^T$, $\vec{c}_4^1 = (C_1, p_1 + k_1 L)^T$, the centers for the second line are $\vec{c}_1^2 = (C_2, p_2 - k_2 L)^T$, $\vec{c}_2^2 = (C_2, p_2 - k_3 L)^T$, $\vec{c}_3^2 = (C_2, p_2 + k_3 L)^T$, $\vec{c}_4^2 = (C_2, p_2 + k_2 L)^T$. Then the center of mass ordinates of each marker line p_1, p_2 and abscissas of lines C_1, C_2 are unknown, but we know L, $k_1 > 0$, $k_2 > 0$, $k_3 > 0$. Thus, we want to exploit some known properties of the calibration cage. Thus, by knowing the detected i-projection of the center of the marker j from the group l ($l \in \{1, 2\}$) q_{ij}^l we want to compute unknowns λ_i, y_{λ_i}, D, p_1, p_2 and C_1, C_2.

Non-uniqueness of the solution. As in the case of the 2D Radon transform for functions, we don't have a unique solution to the self-calibration task for functions in this fan-beam geometry. The explanation of some ambiguities was described in [DC19]. For now, it's important to see that after

the special shearing and translation of the object with $f_{M,\vec{t}}(\vec{x}) := f(M\vec{x}+\vec{t})$, $M := \begin{pmatrix} 1 & 0 \\ (y'+\lambda')/D & 1 \end{pmatrix}$, $\vec{t} := (0,-y')^T$, we obtain the same data for another calibration parameters. In [DC19] the shearing and the translation were described separately, here we want to combine these two results in one formula. So, we have for functions:

$$\begin{aligned}\mathfrak{D}f_{M,\vec{t}}(\lambda,y) &= \int_0^{+\infty} f_{M,\vec{t}}(D-lD,\lambda+ly-l\lambda)dl \\ &= \int_0^{+\infty} f(D-lD,\frac{y'+\lambda'}{D}(D-lD)+\lambda+ly-l\lambda-y')dl \\ &= \int_0^{+\infty} f(D-lD,y'-ly'+\lambda'-l\lambda'+\lambda+ly-l\lambda-y')dl \\ &= \int_0^{+\infty} f(D-lD,\lambda+\lambda'+l(y-y')-l(\lambda+\lambda'))dl = \mathfrak{D}f(\lambda+\lambda',y-y'), \quad (4.8)\end{aligned}$$

We see that projection data for $f_{M,\vec{t}}$ are the same as for f with the translated source by λ' and the translated line of detector by y'. Thus, we can't estimate the calibration parameters and f using only projected images better than up to M and $\vec{t}$. It's also true for Dirac distributions. Let us show this. Let us define for $f = \delta_{\vec{c}} \in \mathscr{E}'(X_2)$ a new distribution $f_{M,\vec{t}}$ as $\langle f_{M,\vec{t}}(\vec{x}),\phi(\vec{x})\rangle = \langle f(\vec{x}),\phi(M^{-1}(\vec{x}-\vec{t})\rangle$, where $M^{-1} = \begin{pmatrix} 1 & 0 \\ -(y'+\lambda')/D & 1 \end{pmatrix}$, thus $M^{-1}(\vec{x}-\vec{t}) = \begin{pmatrix} x_1 \\ -(y'+\lambda')x_1/D+x_2+y' \end{pmatrix}$. It's easy to see that $f_{M,\vec{t}}$ is a distribution $\delta_{M^{-1}(\vec{c}-\vec{t})} \in \mathscr{E}'(X_2)$. Then we can show for one Dirac $f = \delta_{\vec{c}}$:

$$\begin{aligned}(\mathfrak{D}_\lambda f_{M,\vec{t}}(y),\phi(y)) &= \left\langle f_{M,\vec{t}}(\vec{x}),\phi\left(\frac{x_2D-x_1\lambda}{D-x_1}\right)\frac{1}{D-x_1}\right\rangle = \left\langle \delta_{M^{-1}(\vec{c}-\vec{t})}(\vec{x}),\phi\left(\frac{x_2D-x_1\lambda}{D-x_1}\right)\frac{1}{D-x_1}\right\rangle \\ &= \frac{1}{D-c_1}\phi\left(\frac{(-(y'+\lambda')c_1/D+c_2+y')D-c_1\lambda}{D-c_1}\right) = \frac{1}{D-c_1}\phi\left(\frac{c_2D-c_1(\lambda+\lambda')}{D-c_1}+y'\right) \\ &= (\mathfrak{D}_{\lambda+\lambda'}\delta_{\vec{c}}(y),\phi(y+y')) = (\mathfrak{D}_{\lambda+\lambda'}\delta_{\vec{c}}(y-y'),\phi(y)) = (\mathfrak{D}_{\lambda+\lambda'}f(y-y'),\phi(y)). \quad (4.9)\end{aligned}$$

Thus, we can fix the aligned coordinate system in the scene such that in this system $\lambda_0 = 0$ and $y_{\lambda_0} = 0$. Thus, we want to find just one solution in the aligned coordinate system among all possible solutions. This solution will correspond to another object obtained from the initial one by a transformation including a sharing and a translation.

Moreover, we can't uniquely compute D. As in 3D divergent geometry, we also have the scaling ambiguity in the 2D case. Indeed, the fan-beam data $\mathfrak{D}f(\lambda,y) = \int_0^{+\infty} f(kD-lkD,\lambda+ly-l\lambda)dl$ for the function $f(x_1,x_2)$ and the source-detector distance kD, $k \neq 0$ are the same with $\mathfrak{D}\tilde{f}(\lambda,y) = \int_0^{+\infty} f(k(D-lD),\lambda+ly-l\lambda)dl$ for the function $\tilde{f}(x_1,x_2) = f(kx_1,x_2)$ and the source-detector distance D. We can easily see the same for Diracs: the 2D perspective projection $\frac{c_2kD-c_1\lambda}{kD-c_1}$ of $(c_1,c_2)^T$ in the geometry with kD is equal to $\frac{c_2D-c_1\lambda/k}{D-c_1/k}$ of $(c_1/k,c_2)^T$ in the geometry with D. Thus, we can fix any D for the solution. Suppose that we fix D equal to the initial value that we know after the installation of the X-ray system.

Our algorithm. We will compute separately the moments Eq. (4.7) for each line of markers, we will use the superscript $l = 1$ for the first line, $l = 2$ for the second line. The moments of order 0 for the markers (Diracs) on the line l corresponding to the projection data $m_i^l(y) = \mathfrak{D}_{\lambda_i} f^l(y - y_{\lambda_i})$, $f^l = \sum_{j=1}^4 \delta_{\vec{c}_j^l}$:

$$M_0^l(i) = (m_i^l(y), 1) = (\mathfrak{D}_{\lambda_i} f^l(y - y_{\lambda_i}), 1) = \left(\sum_{j=1}^4 \frac{1}{D - c_{j1}^l} \delta_{\tilde{c}(\vec{c}_j^l, \lambda_i, D)}(y - y_{\lambda_i}), 1 \right) = \frac{4}{D - C_l}. \quad (4.10)$$

The moments of order 1 using the fact that $c_{j1}^l = C_l$ for each l:

$$\begin{aligned} M_1^l(i) &= (m_i^l(y), y) = (\mathfrak{D}_{\lambda_i} f^l(y - y_{\lambda_i}), y) = (\mathfrak{D}_{\lambda_i} f^l(y), y + y_{\lambda_i}) \\ &= \left(\sum_{j=1}^4 \frac{1}{D - c_{j1}^l} \delta_{\tilde{c}(\vec{c}_j^l, \lambda_i, D)}(y), y + y_{\lambda_i} \right) = \sum_{j=1}^4 \frac{\tilde{c}(\vec{c}_j^l, \lambda_i, D)}{D - c_{j1}^l} + \sum_{j=1}^4 \frac{y_{\lambda_i}}{D - c_{j1}^l} \\ &= \sum_{j=1}^4 \frac{c_{j2}^l D - c_{j1}^l \lambda_i}{(D - c_{j1}^l)^2} + \sum_{j=1}^4 \frac{y_{\lambda_i}}{D - c_{j1}^l} = -\lambda_i \frac{4C_l}{(D - C_l)^2} + y_{\lambda_i} M_0^l(i) + \sum_{j=1}^4 \frac{c_{j2}^l D}{(D - C_l)^2}. \end{aligned} \quad (4.11)$$

In the chosen aligned coordinate system in the scene $\lambda_0 = 0$, $y_{\lambda_0} = 0$, then

$$M_1^1(0) = \sum_{j=1}^4 \frac{c_{j2}^1 D}{(D - C_1)^2}, \qquad M_1^2(0) = \sum_{j=1}^4 \frac{c_{j2}^2 D}{(D - C_2)^2}. \quad (4.12)$$

Let us remind that we know the detected centers of the markers q_{ij}^l, then

$$M_1^l(i) = \left(\sum_{j=1}^4 \frac{1}{D - c_{j1}^l} \delta_{q_{ij}^l}, y \right) = \sum_{j=1}^4 \frac{q_{ij}^l}{D - c_{j1}^l} = \frac{1}{D - C_l} \sum_{j=1}^4 q_{ij}^l. \quad (4.13)$$

Let us use Eq. (4.12) and (4.13) in Eq. (4.11) and multiple each side of the equation by $D - C_l$. Then we have

$$-\lambda_i \frac{4C_l}{D - C_l} + 4 y_{\lambda_i} = \sum_{j=1}^4 q_{ij}^l - \sum_{j=1}^4 q_{0j}^l. \quad (4.14)$$

If we introduce new unknown variables $r_1 > 0$ and $r_2 > 0$ with

$$r_1 = \frac{C_1}{D - C_1}, \qquad r_2 = \frac{C_2}{D - C_2}, \quad (4.15)$$

then we can rewrite Eq. (4.14) as

$$y_{\lambda_i} - \lambda_i r_l = \Delta \tilde{M}_1^l(i), \qquad \Delta \tilde{M}_1^l(i) := \frac{1}{4} \left(\sum_{j=1}^4 q_{ij}^l - \sum_{j=1}^4 q_{0j}^l \right). \quad (4.16)$$

Here we see that $\Delta \tilde{M}_1^l(i)$ can be computed from the projection data. Now we have the set of unknowns: y_{λ_i}, λ_i, r_1, r_2, p_1, p_2. Note that Eq. (4.16) contains only y_{λ_i}, λ_i, r_1, r_2, but not p_1, p_2.

Let us exploit the moment conditions of order 2 to complete our algorithm using $c_{j1}^l = C_l$:

$$M_2^l(i) = (m_i^l(y), y^2) = (\mathfrak{D}_{\lambda_i} f^l(y), (y+y_{\lambda_i})^2) = \sum_{j=1}^4 \frac{(\tilde{c}(\vec{c}_j^l, \lambda_i, D) + y_{\lambda_i})^2}{D - c_{j1}^l}$$
$$= \frac{1}{D-C_l} \sum_{j=1}^4 \left(\frac{c_{j2}^l D - c_{j1}^l \lambda_i}{D - c_{j1}^l} + y_{\lambda_i} \right)^2 = \frac{1}{D-C_l} \sum_{j=1}^4 \left(\frac{c_{j2}^l D}{D-C_l} + y_{\lambda_i} - \lambda_i r_l \right)^2$$
$$= \frac{1}{D-C_l} \sum_{j=1}^4 \left(\frac{c_{j2}^l D}{D-C_l} + \Delta\tilde{M}_1^l(i) \right)^2. \quad (4.17)$$

From Eq. (4.12), (4.13) and (4.15) we obtain

$$\sum_{j=1}^4 \frac{c_{j2}^l D}{D-C_l} = \frac{D}{D-C_l} \sum_{j=1}^4 c_{j2}^l = (1+r_l) \sum_{j=1}^4 c_{j2}^l = \sum_{j=1}^4 q_{0j}^l, \quad (4.18)$$

then we have

$$\sum_{j=1}^4 \frac{c_{j2}^l D}{D-C_l} = \sum_{j=1}^4 q_{0j}^l, \quad (4.19)$$

but also from $\sum_{j=1}^4 c_{j2}^l = 4p_l$ and Eq. (4.18)

$$(1+r_l) 4p_l = \sum_{j=1}^4 q_{0j}^l. \quad (4.20)$$

For the chosen markers $\sum_{j=1}^4 (c_{j2}^1)^2 = 4p_1^2 + (2+2k_1^2)L^2$ and $\sum_{j=1}^4 (c_{j2}^2)^2 = 4p_2^2 + (2k_2^2 + 2k_3^2)L^2$, then we can compute using Eq. (4.15):

$$\sum_{j=1}^4 \left(\frac{c_{j2}^l D}{D-C_l} \right)^2 = (1+r_l)^2 \sum_{j=1}^4 (c_{j2}^l)^2 = \begin{cases} (1+r_1)^2 (4p_1^2 + (2+2k_1^2)L^2) & \text{if } l=1 \\ (1+r_1)^2 (4p_2^2 + (2k_2^2+2k_3^2)L^2) & \text{if } l=2. \end{cases} \quad (4.21)$$

We can develop Eq. (4.17), after the multiplication of each side of the equation by $D - C_l$:

$$\begin{cases} \sum_{j=1}^4 (q_{ij}^1)^2 = (1+r_1)^2 (4p_1^2 + (2+2k_1^2)L^2) + 2\Delta\tilde{M}_1^1(i) \sum_{j=1}^4 q_{0j}^1 + 4[\Delta\tilde{M}_1^1(i)]^2 \\ \sum_{j=1}^4 (q_{ij}^2)^2 = (1+r_2)^2 (4p_2^2 + (2k_2^2+2k_3^2)L^2) + 2\Delta\tilde{M}_1^2(i) \sum_{j=1}^4 q_{0j}^2 + 4[\Delta\tilde{M}_1^2(i)]^2. \end{cases}$$

Using Eq. (4.20), we have the following system to find r_1 and r_2:

$$\begin{cases} \sum_{j=1}^4 (q_{ij}^1)^2 = (1+r_1)^2 (2+2k_1^2)L^2 + \frac{1}{4} \left[\sum_{j=1}^4 q_{0j}^1 \right]^2 + 2\Delta\tilde{M}_1^1(i) \sum_{j=1}^4 q_{0j}^1 + 4[\Delta\tilde{M}_1^1(i)]^2 \\ \sum_{j=1}^4 (q_{ij}^2)^2 = (1+r_2)^2 (2k_2^2+2k_3^2)L^2 + \frac{1}{4} \left[\sum_{j=1}^4 q_{0j}^2 \right]^2 + 2\Delta\tilde{M}_1^2(i) \sum_{j=1}^4 q_{0j}^2 + 4[\Delta\tilde{M}_1^2(i)]^2. \end{cases} \quad (4.22)$$

Using only one projection i, we can estimate from Eq. (4.22) $(1+r_1)^2$ and $(1+r_2)^2$. In the case of noisy data we can perform the estimation from few projections. If the solution of the system is $R_l =$

$(1+r_l)^2$, then $r_l = -1 \pm R_l^{1/2}$. We know that $r_l > 0$. Then we can uniquely calculate $r_l = -1 + R_l^{1/2}$. From that, Eq. (4.15) gives us C_1 and C_2 for fixed D, Eq. (4.20) gives us p_1 and p_2. For each $i \geq 1$ we have two lines of markers, so we have the system of two equations from Eq. (4.16), $l = 1$ or 2. From this system we can compute y_{λ_i}, λ_i. This system gives a unique solution, since:

$$\det \begin{pmatrix} 1 & -r_1 \\ 1 & -r_2 \end{pmatrix} = r_1 - r_2 = \frac{C_1}{D-C_1} - \frac{C_2}{D-C_2} = \frac{C_1(D-C_2) - C_2(D-C_1)}{(D-C_1)(D-C_2)}$$
$$= \frac{D(C_1-C_2)}{(D-C_1)(D-C_2)} \neq 0.$$

With Eq. (4.22), (4.15), (4.20), (4.16) we can analytically compute all calibration parameters and the location of two lines with markers from the local information about detected markers on the projected images.

Numerical experiments. We chose the calibration cage as it's presented in the fig. 4.2. We measure distances in cm. For this cage in the initial coordinate system the known parameters of the pattern are $L = 0.4$, $k_1 = 3$, $k_2 = 1$, $k_3 = 2$.

We generated true values of unknown parameters:

1. $p_1 = 0$, $p_2 = 3.2$, $C_1 = 1.5$, $C_2 = 0.5$,
2. for geometric parameters of the system λ_i and y_{λ_i} (calibration parameters) we fixed $P = 30$ random values for source positions λ_i in $[-5,5]$, the grid on $y \in [-5,5]$ had the step 0.01, y_{λ_i} were generated as random uniform noise on the interval $[-0.05, 0.05)$.

Note that the x_2-distance between the last point of the first group and the first point of the second group was $3L$. We suppose that D is fixed to its initial value 10.

We want to calibrate in the aligned coordinate system and compare our estimations with true values. In order to perform this comparison correctly in the aligned coordinate system, we need to subtract λ_0 from the true λ_i and y_{λ_0} from the true y_{λ_i}, because in the aligned coordinate system we have $\lambda_0 = 0$ and $y_{\lambda_0} = 0$. We also need to perform some transformation to p_1 and p_2 in order to compare correctly estimated values with true values. We know the connection between ordinates in the aligned and in the initial coordinate systems from the discussion of the non-uniqueness of the solution of the current calibration problem $x_2^{\text{aligned}} = x_2^{\text{init}} - (\lambda_0/D + y_{\lambda_0}/D)x_1^{\text{init}} + y_{\lambda_0}$. With this formula we can compare correctly initial and estimated ordinates of the centers of mass p_1 and p_2 of markers on each line in the aligned coordinate system.

All computations were performed in Python 3. For exact projections q_{ij}^l our algorithm was almost perfect (see the first line of the table 4.1). To mimic realistic settings, we introduced the detection noise as a Gaussian noise $N(0,\sigma)$ added to q_{ij}^l. Here $\sigma = 0.01 \cdot Nl$, where Nl is the noise level, 0.01 is the pixel size of the initial image. In the table 4.1 we can also see the results of noisy experiments.

We present the mean absolute errors of 100 realisations of the Gaussian noise for each Nl. Thus, for calibration parameters, for example, for source positions we computed

$$\frac{1}{100}\sum_{k=1}^{100}\frac{1}{P}\sum_{i=0}^{P-1}\left|\hat{\lambda}_{i,k}-\lambda_i\right|, \tag{4.23}$$

where λ_i is the true value for the projection i, $\hat{\lambda}_{i,k}$ is the estimation with our algorithm of the source position for the projection i with the realisation of the Gaussian noise number k added to projections q^l_{ij}, so k states for the experiment number. For marker positions, for example, for abscissas we computed

$$\frac{1}{100}\sum_{k=1}^{100}\frac{1}{2}\sum_{l=1}^{2}\left|\hat{C}_{l,k}-C_l\right|, \tag{4.24}$$

where C_l is the true value, $\hat{C}_{l,k}$ is the estimation with our algorithm of the position of the marker line number l with the realisation of the Gaussian noise number k added to projections. We show that the errors are proportional to Nl.

Noise level Nl	**Noise std**	**Error for λ_i**	**Error for y_{λ_i}**	**Error for p_l**	**Error for C_l**
0%	0	2.10×10^{-14}	3.62×10^{-15}	1.17×10^{-15}	3.39×10^{-15}
10%	0.001	2.32×10^{-2}	3.40×10^{-3}	1.26×10^{-3}	4.58×10^{-3}
50%	0.005	1.01×10^{-1}	1.52×10^{-2}	5.86×10^{-3}	2.12×10^{-2}
100%	0.01	2.20×10^{-1}	3.13×10^{-2}	1.15×10^{-2}	4.32×10^{-2}
200%	0.02	5.19×10^{-1}	7.45×10^{-2}	2.63×10^{-2}	9.66×10^{-2}

Table 4.1: Mean absolute errors for calibration parameters and the positions of the markers with non-noisy and noisy projections, all errors are in cm.

In order to understand how these errors of the calibration can influence the reconstruction process, let us compute the following error. Since we estimated for each projection i in the aligned coordinate system $\hat{\lambda}_i$ and $\hat{y}_{\lambda_i}$, in the initial coordinate system they can be recalculated as $\hat{\lambda}'_i=\hat{\lambda}_i+\lambda_0$ and $\hat{y}'_{\lambda_i}=\hat{y}_{\lambda_i}+y_{\lambda_0}$. Knowing the source position $\hat{\lambda}'_i$ and the corrected projection $\hat{q}^l_{ij}=q^l_{ij}-\hat{y}'_{\lambda_i}$ for each marker line l for each point j we can draw the line connecting $\hat{\lambda}'_i$ and $\hat{q}^l_{ij}$. If we have the exact estimation of calibration parameters, then the distance from the 2D marker point $\vec{c}^{\,l}_j$ in the initial coordinate system to this line is essentially zero, because $\hat{q}^l_{ij}$ should be the exact 2D perspective projection of $\vec{c}^{\,l}_j$ along this line. We can see this behaviour in the first line of the table 4.2 when we don't have any noise in the input values of our algorithm. In the case of noisy input data, we see the transfer of errors of the estimation of calibration parameters to the shift of the projection line from its true position in the plane. This shift will also influence the reconstruction process.

In this section we presented our definition for fixed λ of the fan-beam transform on distributions with sources on a line, we proved necessary DCC for our definition. As in the case of the 2D Radon

Noise level *Nl*	**Noise std**	**Distance between the marker point and the projection line**
0%	0	1.11×10^{-15}
10%	0.001	1.70×10^{-3}
50%	0.005	7.96×10^{-3}
100%	0.01	1.62×10^{-2}
200%	0.02	3.76×10^{-2}

Table 4.2: Mean distances between the 2D marker point $\vec{c}_j^{\,l}$ and the projection line through $\hat{\lambda}_i'$ and $\hat{q}_{ij}^{l}$, all distances are in cm.

transform, our DCC combined with a proper calibration cage allow to perform the full analytical self-calibration with truncated data when markers are not truncated.

5 Calibration with DCC on distributions for cone-beam transform

5.1 French summary of the chapter

Ce chapitre est consacré à la géométrie en faisceau conique avec des sources alignées et des sources dans le plan parallèle au plan du détecteur. Dans le premier cas, nous utilisons le fait présenté en [NDC20] que la tâche de calibration dans ce cas peut être réduite à la calibration 2D en éventail dans un plan oblique. Donc, nous pouvons réutiliser l'algorithme 2D de la section 4.2 pour calibrer dans le plan oblique dans ce cas. Nous discutons également de la conception de la mire d'étalonnage correspondante afin d'effectuer cet étalonnage en situation réelle et d'avoir la disposition de marqueurs souhaités dans le plan oblique. Cette contribution a été décrite dans l'article [KDG22].

Dans le second cas, nous définissons la transformée en faisceau conique sur les distributions pour la position de la source fixe et les DCC correspondantes, la généralisation des DCC de [CD13]. La mire d'étalonnage spéciale nous permet de séparer la tâche en deux tâches indépendantes pour chaque coordonnée. Nous pouvons à nouveau obtenir les procédures d'étalonnage analytiques.

Nous présentons également les simulations numériques dans ce chapitre et montrons que nos approches peuvent être appliquées dans les situations réelles. Nous résolvons en fait dans chaque cas la tâche d'auto-étalonnage, car nous utilisons les mires d'étalonnage avec des marqueurs avec des modèles connus, mais des coordonnées inconnues dans le système de coordonnées mondial. De plus, nos algorithmes sont basés sur les informations locales sur les marqueurs non tronqués, nous pouvons donc appliquer nos approches dans la situation où les données complètes sont tronquées.

5.2 Cone-beam transform on distributions with sources on a line

5.2.1 Definition

In this section we describe our contribution presented at the IEEE Nuclear Science Symposium and Medical Imaging Conference 2022 [KDG22]. The case of 3D divergent geometry we start with the simple case when the source trajectory is a line parallel to the detector plane as you can see in the fig. 5.1. Suppose that u and v are parameters of the detector placed in the plane $x_1 = 0$. The position of the source $(D, \lambda, 0)^T$ is described with the parameter $\lambda \in \mathbb{R}$. Thus, the source position is moving along a

line parallel to the x_2 axis at a distance $D > 0$ to the detector. The u axis is the same as the x_2 axis and the v axis is the same as the x_3 axis, see fig. 5.1. Then we can define:

Definition 5.1. *Cone-beam data with sources on a line*

$$\mathfrak{D}_\lambda f(u,v) = \int_0^{+\infty} f(D - lD, \lambda + l(u-\lambda), lv)dl, \tag{5.1}$$

where $f \in \mathscr{D}_3$.

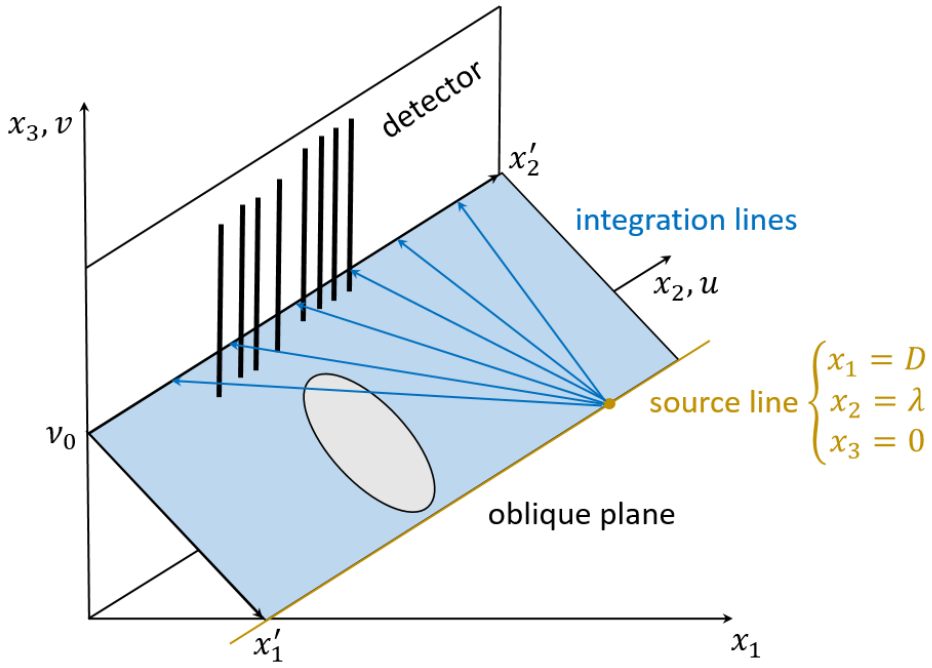

Figure 5.1: The 3D cone-beam geometry with sources on a line with the calibration cage of 8 vertical sticks.

Suppose that $f \in \mathscr{D}_3$ with support in $X_3 = (D_1, D_2) \times \mathbb{R}^2$, $0 < D_1 < D_2 < D$. We consider $\mathfrak{D}_\lambda f$ as a function of two variables. In order to define the adjoint operator $\mathfrak{D}_\lambda^*$, we need to note that for the described function f and $\phi \in \mathscr{E}_2$:

$$\begin{aligned}(\mathfrak{D}_\lambda f, \phi) &= \int_{\mathbb{R}^2} \mathfrak{D}_\lambda f(u,v)\phi(u,v)dudv = \int_{\mathbb{R}^2}\int_0^{+\infty} f(D-lD, \lambda + lu - l\lambda, lv)dl\phi(u,v)dudv \\ &= \frac{1}{D}\int_{\mathbb{R}^2}\int_{-\infty}^{D} f\left(t_1, \lambda + \frac{D-t_1}{D}(u-\lambda), \frac{D-t_1}{D}v\right)dt_1\phi(u,v)dudv \\ &= \int_{-\infty}^{D}\int_{\mathbb{R}^2} f(t_1,t_2,t_3)\phi\left(\frac{t_2D - t_1\lambda}{D-t_1}, \frac{Dt_3}{D-t_1}\right)\frac{D}{(D-t_1)^2}dt_2dt_3dt_1 = \langle f, \mathfrak{D}_\lambda^*\phi\rangle, \end{aligned} \tag{5.2}$$

where $(\cdot,\cdot)$ is the scalar product in $L^2(\mathbb{R}^2)$, $\langle\cdot,\cdot\rangle$ is the scalar product in $L^2(X_3)$, we used the change of variables $t_1 = D - lD$, $dl = -\frac{dt_1}{D}$, $l = \frac{D-t_1}{D}$; $t_2 = \lambda + \frac{D-t_1}{D}(u-\lambda)$, $dt_2 = \frac{D-t_1}{D}du$; $t_3 = \frac{D-t_1}{D}v$, $dt_3 = \frac{D-t_1}{D}dv$.

Now we can define the adjoint operator for functions from $\mathscr{E}_2$, it should be equal to

$$\mathfrak{D}^*_\lambda \phi(\vec{x}) := \frac{D}{(D-x_1)^2}\phi\left(\frac{x_2 D - x_1\lambda}{D-x_1}, \frac{Dx_3}{D-x_1}\right). \tag{5.3}$$

Definition 5.2. *The cone-beam transform on a line of a compactly supported distribution $f \in \mathscr{E}'(X_3)$ at fixed λ is a distribution from $\mathscr{E}'_2$ defined by the same duality Eq.* (4.3) *with the adjoint operator Eq.* (5.3)*.*

We justify why this definition is properly formulated in the Appendix A.4.

If we again model markers with Dirac distributions, then

$$\begin{aligned}(\mathfrak{D}_\lambda \delta_{\vec{c}}(u,v), \phi(u,v)) = \langle \delta_{\vec{c}}, \mathfrak{D}^*_\lambda \phi\rangle = \left\langle \delta_{\vec{c}}(\vec{x}), \frac{D}{(D-x_1)^2}\phi\left(\frac{x_2 D - x_1\lambda}{D-x_1}, \frac{Dx_3}{D-x_1}\right)\right\rangle \\ = \frac{D}{(D-c_1)^2}\phi\left(\frac{c_2 D - c_1\lambda}{D-c_1}, \frac{Dc_3}{D-c_1}\right) = \frac{D}{(D-c_1)^2}\delta_{\tilde{c}}(\phi), \qquad \tilde{c} = \left(\frac{c_2 D - c_1\lambda}{D-c_1}, \frac{Dc_3}{D-c_1}\right).\end{aligned} \tag{5.4}$$

So, the cone-beam projection with sources on a line of a Dirac is again a weighted Dirac. The point $\tilde{c}$ that we found is actually the usual perspective projection of the point $\vec{c}$. To understand this, let us look at similar triangles in the fig. 5.2. From the left picture we have two similar triangles $\triangle AH_2Q'_{\text{par}}$ and $\triangle Q_{\text{par}}H'_2Q'_{\text{par}}$, where A is the source position, Q_{par} is the projection of the marker point Q with coordinates $(c_1,c_2,c_3)^T$ onto the plane Ox_1x_2, Q'_{par} is the projection of the projected to the detector point Q' onto the line Ox_2. Then we have $\frac{AH_2}{Q_{\text{par}}H'_2} = \frac{Q'_{\text{par}}H_2}{Q'_{\text{par}}H'_2}$ or $\frac{D}{c_1} = \frac{\lambda-u}{c_2-u}$, thus we have for the projected point the first coordinate $u = \frac{c_2D-c_1\lambda}{D-c_1}$. From the right picture we have two similar triangles $\triangle AQH_1$ and $\triangle AQ'O_{\text{par}}$, where as before Q' is the projection of the point Q along the line AQ. Then we have $\frac{QH_1}{Q'O_{\text{par}}} = \frac{AH_1}{AO_{\text{par}}}$ or $\frac{c_3}{v} = \frac{D-c_1}{D}$, thus we have for the projected point the second coordinate $v = \frac{Dc_3}{D-c_1}$.

5.2.2 Calibration procedure

Suppose that the detector is moving along the x_2 axis, so v is known, but shifts u_{λ_i} for each unknown source position λ_i are unknown. Let us fix known $v = v_0$. Moreover, we have an ambiguity with the estimation of D as we showed previously for the calibration of the cone-beam system, then let us fix D as known. We want to use a specific calibration cage of 8 sticks parallel to the detector plane (see fig. 5.1). We want again to find unknown calibration parameters. The pattern of sticks is known, but we don't know the position of the calibration cage in the scene.

Let us show that we can reduce this 3D task of the self-calibration to the 2D fan-beam self-calibration that we described in the previous chapter. We want to propose a similar method as in [NDC20], but suitable for the case of truncation, since we plan to use only the information about markers.

Let us define an oblique plane as a plane passing through the source line and the line $v = v_0$. Thus, this plane should pass through the point $(D,0,0)^T$ and should be perpendicular to $(v_0,0,D)^T$, where $(v_0,0,D)^T$ is the vector product of two vectors $(0,1,0)^T$ and $(0,u,v_0)^T - (D,\lambda,0)^T = (-D,u-\lambda,v_0)^T$.

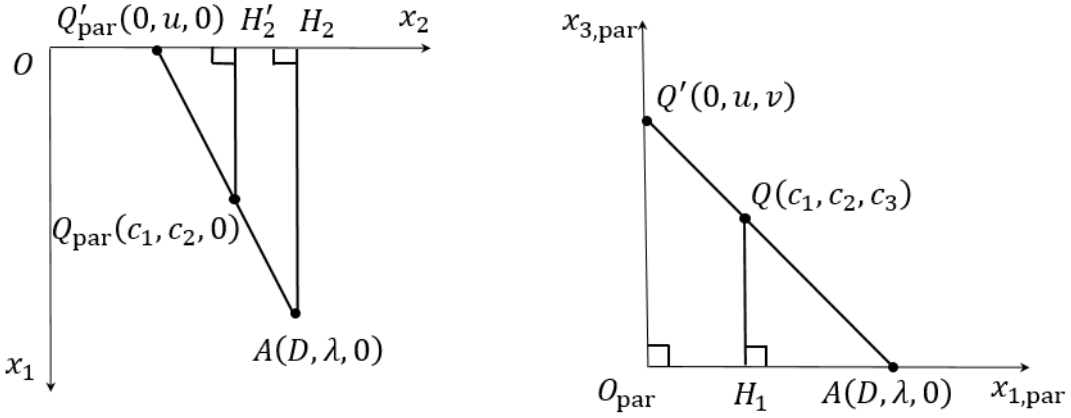

Figure 5.2: Left: the view in the plane Ox_1x_2, right: the view in the plane passing through A and Q, parallel to Ox_1x_3. A is the source position, Q_{par} is the projection of the marker point Q with coordinates $(c_1,c_2,c_3)^T$ onto the plane Ox_1x_2, Q' is the perspective projection of Q onto the detector plane, Q'_{par} is the projection of Q' onto the line Ox_2.

Thus, the equation of the oblique plane is $v_0x_1+0+Dx_3+d=0$, where d is unknown. Since $(D,0,0)^T$ belongs to the plane, then $d=-v_0D$ and the equation of the oblique plane is $v_0x_1+Dx_3-v_0D=0$. Thus, each point in the plane can be presented as $(x_1,x_2,v_0-\frac{v_0}{D}x_1)^T$.

With similar computations as in [NDC20], we can show that for our different coordinate systems we also have that the cone-beam data of some function f is the fan-beam data of another function f_{ob} in the oblique plane. To see that, we need to note that the 3D point $(x_1,x_2,v_0-\frac{v_0}{D}x_1)^T$ in the oblique plane goes to the 2D point $(x_1\sqrt{1+\frac{v_0^2}{D^2}},x_2)^T$ in the coordinate system of the oblique plane (see fig. 5.3). Then in the oblique plane:

$$f\left(x_1,x_2,v_0-\frac{v_0}{D}x_1\right)=f_{\text{ob}}\left(x_1\sqrt{1+\frac{v_0^2}{D^2}},x_2\right). \tag{5.5}$$

Then for the projection data:

$$\begin{aligned}\mathfrak{D}_\lambda f(u,v_0)&=\int_0^{+\infty}f(D-lD,\lambda+lu-l\lambda,lv_0)dl\\&=\int_0^{+\infty}f\left(D-lD,\lambda+lu-l\lambda,v_0-\frac{v_0}{D}D(1-l)\right)dl=\int_0^{+\infty}f_{\text{ob}}\left(D(1-l)\sqrt{1+\frac{v_0^2}{D^2}},\lambda+lu-l\lambda\right)dl\\&=\int_0^{+\infty}f_{\text{ob}}(D_{v_0}-lD_{v_0},\lambda+lu-l\lambda)=\mathfrak{D}_\lambda f_{\text{ob}}(u),\end{aligned} \tag{5.6}$$

where

$$D_{v_0}=D\sqrt{1+\frac{v_0^2}{D^2}}, \tag{5.7}$$

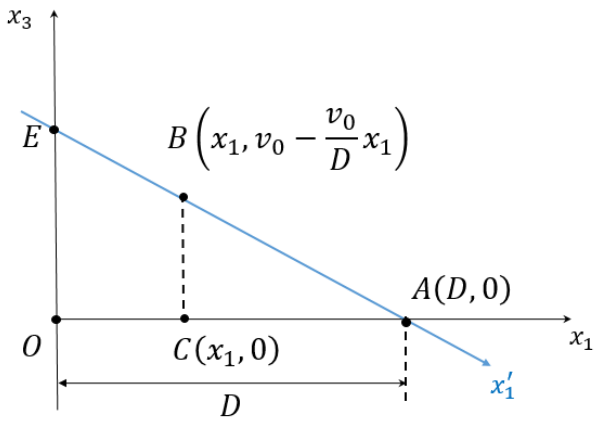

Figure 5.3: The blue line is the line in the intersection of the oblique plane $v_0x_1 + Dx_3 - v_0D = 0$ and Ox_1x_3. We can compute $AB = \sqrt{1+\frac{v_0^2}{D^2}}(D-x_1)$ and from two similar triangles $\triangle ABC$ and $\triangle AEO$ we find $BE = x_1\sqrt{1+\frac{v_0^2}{D^2}}$. BE will be the first coordinate of the point in the oblique plane in the coordinate system $Ex_1'x_2'$ connected with the oblique plane and introduced in the fig. 5.1.

so the fan-beam and the cone-beam data in this equation correspond to different source-detector distances. We need to find the same type of connection for Diracs. Note that if we know v_0, thus we know the second coordinate of the projection $\tilde{c}$ in Eq. (5.4). The first coordinate of $\tilde{c}$ in Eq. (5.4) has exactly the same form as in the fan-beam case, then it corresponds to the fan-beam projection for fixed λ of some point.

To be more precise, let us introduce the coordinate system in the oblique plane. Suppose that the center is in $(0,0,v_0)^T$. We know that the source point $(D,0,0)^T$ belongs to the plane, then the first coordinate vector in the oblique plane (along x_1') is $\frac{1}{\sqrt{D^2+v_0^2}}(D,0,-v_0)^T$, the second coordinate vector is $(0,1,0)^T$. Then the point with 3D coordinates $(c_1,c_2,c_3)^T$ at the intersection of the calibration stick with the oblique plane has coordinates $(t_1,t_2)^T$ in the coordinate system of the oblique plane such that

$$\begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ v_0 \end{pmatrix} + t_1 \frac{1}{\sqrt{D^2+v_0^2}} \begin{pmatrix} D \\ 0 \\ -v_0 \end{pmatrix} + t_2 \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}.$$

Then $c_1 = \frac{t_1}{\sqrt{1+v_0^2/D^2}}$, $c_2 = t_2$, $c_3 = v_0 - \frac{v_0 t_1}{\sqrt{D^2+v_0^2}}$. From Eq. (5.4) our data $\frac{c_2 D - c_1\lambda}{D-c_1}$ depending on 3D coordinates of the point on the stick become $\frac{t_2 D - \frac{t_1}{\sqrt{1+v_0^2/D^2}}\lambda}{D-\frac{t_1}{\sqrt{1+v_0^2/D^2}}} = \frac{t_2 D_{v_0} - t_1\lambda}{D_{v_0}-t_1}$ with D_{v_0} from Eq. (5.7). So, we reduced our 3D calibration task to the 2D fan-beam calibration task with the same type of data, but the source-detector distance D_{v_0}. Moreover, we showed the connection between the marker point in the 3D coordinate system in the scene and in the 2D coordinate system of the oblique plane.

Non-uniqueness of the solution. As in the case of the 2D fan-beam transform, we have the non-

uniqueness of the solution and the ambiguity represented by the special shearing and translation of the object with $f_{M,\vec{t}}(\vec{x}) := f(M\vec{x}+\vec{t})$, $M := \begin{pmatrix} 1 & 0 & 0 \\ (u'+\lambda')/D & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$, $\vec{t} := (0,-u',0)^T$. It's true for functions:

$$\begin{aligned}
\mathfrak{D}_\lambda f_{M,\vec{t}}(u,v_0) &= \int_0^{+\infty} f_{M,\vec{t}}(D-lD,\lambda+lu-l\lambda,lv_0)dl \\
&= \int_0^{+\infty} f(D-lD,\frac{u'+\lambda'}{D}(D-lD)+\lambda+lu-l\lambda-u',lv_0)dl \\
&= \int_0^{+\infty} f(D-lD,u'-lu'+\lambda'-l\lambda'+\lambda+lu-l\lambda-u',lv_0)dl \\
&= \int_0^{+\infty} f(D-lD,\lambda+\lambda'+l(u-u')-l(\lambda+\lambda'),lv_0)dl = \mathfrak{D}_{\lambda+\lambda'} f(u-u',v_0). \quad (5.8)
\end{aligned}$$

With the projection data of f, if the detector is shifted by u' and the source positions are all shifted by $-\lambda'$, then there exists an object $f_{M,\vec{t}}$ having the same projection data from the original source and detector positions. Thus, we can't estimate the calibration parameters and f better than up to M and $\vec{t}$. It's also true for Dirac distributions. For $f \in \mathscr{E}'(X_3)$ we define $f_{M,\vec{t}}$ as $\langle f_{M,\vec{t}}(\vec{x}),\phi(\vec{x})\rangle = \langle f(\vec{x}),\phi(M^{-1}(\vec{x}-\vec{t})\rangle$ with $M^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ -(u'+\lambda')/D & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$, thus $M^{-1}(\vec{x}-\vec{t}) = \begin{pmatrix} x_1 \\ -(u'+\lambda')x_1/D+x_2+u' \\ x_3 \end{pmatrix}$. This is the extension to distributions of the same equality for function f, true for any invertible matrix M and vector $\vec{t}$. Thus $(\delta_{\vec{c}})_{M,\vec{t}}$ is the distribution $\delta_{M^{-1}(\vec{c}-\vec{t})} \in \mathscr{E}'(X_3)$. Then we can show for $\delta_{\vec{c}}$:

$$\begin{aligned}
(\mathfrak{D}_\lambda\,(\delta_{\vec{c}})_{M,\vec{t}}(u,v),\phi(u,v)) &= \left\langle (\delta_{\vec{c}})_{M,\vec{t}}(\vec{x}),\phi\left(\frac{x_2D-x_1\lambda}{D-x_1},\frac{Dx_3}{D-x_1}\right)\frac{D}{(D-x_1)^2}\right\rangle \\
&= \left\langle \delta_{M^{-1}(\vec{c}-\vec{t})}(\vec{x}),\phi\left(\frac{x_2D-x_1\lambda}{D-x_1},\frac{Dx_3}{D-x_1}\right)\frac{D}{(D-x_1)^2}\right\rangle \\
&= \frac{D}{(D-c_1)^2}\phi\left(\frac{(-(u'+\lambda')c_1/D+c_2+u')D-c_1\lambda}{D-c_1},\frac{Dc_3}{D-c_1}\right) \\
= \frac{D}{(D-c_1)^2}\phi\left(\frac{c_2D-c_1(\lambda+\lambda')}{D-c_1}+u',\frac{Dc_3}{D-c_1}\right) &= (\mathfrak{D}_{\lambda+\lambda'}\delta_{\vec{c}}(u,v),\phi(u+u',v)) \\
&= (\mathfrak{D}_{\lambda+\lambda'}\delta_{\vec{c}}(u-u',v),\phi(u,v)). \quad (5.9)
\end{aligned}$$

Since we have the similar ambiguity as in the fan-beam calibration, we can choose the solution to the self-calibration problem such that $\lambda_0 = 0$ and $u_{\lambda_0} = 0$.

Numerical experiments. We fixed $v_0 = 1$, $D = 10$, then we can compute the new source-detector distance that we should use in the fan-beam task $D_{v_0} = 10.050$. We chose the same type of the calibration cage as we used for the fan-beam case. For this cage $L = 0.4$, $k_1 = 3$, $k_2 = 1$, $k_3 = 2$, the x_2-distance in the oblique plane between the last point of the first group and the first point of the second group was $3L$. For other parameters in the coordinate system of the oblique plane: $p_1 = 5$, $p_2 = 8.2$, $C_1 = 1.5$,

$C_2 = 0.3$. We fixed $P = 30$ random values for source positions λ_i in $[0, 10]$, in order to simplify the final comparison of results we fixed $\lambda_0 = 0$. We chose the grid on $u \in [0, 10]$ with the sampling step 0.01, u_{λ_i} were generated as random uniform noise on the interval $[-0.05, 0.05)$, in order to simplify the final comparison of results we fixed $u_{\lambda_0} = 0$.

We reused our algorithm written in Python 3 for the fan-beam case. For exact projections our algorithm was almost perfect (see the table 5.1). As in the fan-beam case, we introduced the detection noise as a Gaussian noise $N(0, \sigma)$ added to projections, $\sigma = 0.01 \cdot Nl$, where Nl is the noise level, 0.01 is the pixel size of the initial image. The table 5.1 also represents the results of noisy experiments. We present the mean absolute errors of 100 realisations of the Gaussian noise for each Nl. Thus, for calibration parameters, for example, for source positions we computed

$$\frac{1}{100} \sum_{k=1}^{100} \frac{1}{P} \sum_{i=0}^{P-1} \left| \hat{\lambda}_{i,k} - \lambda_i \right|, \tag{5.10}$$

where λ_i is the true value for the projection i, $\hat{\lambda}_{i,k}$ is the estimation with our algorithm of the source position for the projection i with the realisation of the Gaussian noise number k added to projections. For marker positions, for example, for abscissas in the oblique plane we computed

$$\frac{1}{100} \sum_{k=1}^{100} \frac{1}{2} \sum_{l=1}^{2} \left| \hat{C}_{l,k} - C_l \right|, \tag{5.11}$$

where C_l is the true value, $\hat{C}_{l,k}$ is the estimation with our algorithm of the position of the marker group number l with the realisation of the Gaussian noise number k added to projections. We show that the errors are proportional to Nl. In the table we compared the calibration parameters, but also estimated coordinates of the markers in the oblique plane. Note that with the theory introduced in this section we can recompute 3D coordinates of the markers with known D and v_0 if it's needed.

Noise level Nl	**Noise std**	**Error for λ_i**	**Error for u_{λ_i}**	**Error for p_l**	**Error for C_l**
0%	0	1.34×10^{-13}	9.27×10^{-15}	1.02×10^{-14}	1.61×10^{-14}
10%	0.001	2.19×10^{-2}	2.98×10^{-3}	2.98×10^{-3}	3.94×10^{-3}
50%	0.005	1.37×10^{-1}	2.04×10^{-2}	1.91×10^{-2}	2.51×10^{-2}
100%	0.01	2.63×10^{-1}	3.91×10^{-2}	3.65×10^{-2}	4.79×10^{-2}
200%	0.02	4.59×10^{-1}	6.61×10^{-2}	6.37×10^{-2}	8.50×10^{-2}

Table 5.1: Mean absolute errors for calibration parameters and the positions of the markers in the oblique plane with non-noisy and noisy projections, all errors are in cm.

5.2.3 Design of a calibration cage and numerical experiments

In this subsection we simulate a calibration cage and test our algorithm from the previous subsection. Before any production of a calibration cage we provide numerical simulations including the construction of projections, detection of traces of markers on projected images, estimation of calibration parameters and verification of the result of the calibration step.

For our numerical experiments we simulated the use of an X-ray system installed in the INSA Lyon. This system consists of a flat detector and a source which can move horizontally in order to adjust the source-detector distance (see fig. 5.4). We propose that we fix positions of the detector and the source in numerical experiments (just like in X-ray baggage screening systems). Moreover, we have a translation table that can be moved vertically. We want to calibrate the translation of the table with the calibration object. The movement of the translation table is equivalent with the common movement of the source and the detector in the problem statement we have described in the previous section. Thus, we want to calibrate as before two sets of parameters u_{λ_i} and λ_i, but each set should represent the same movement of the translation table.

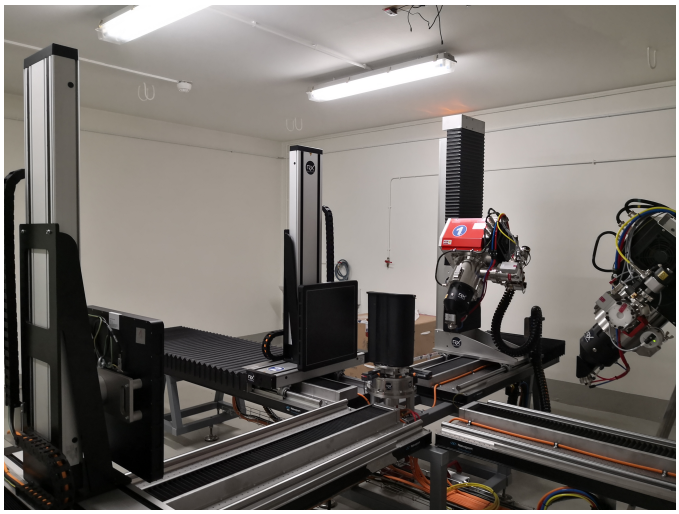

Figure 5.4: The X-ray system proposed by INSA Lyon for experiments with real data.

In order to mimic the situation with real measurements, we use the following real parameters of the X-ray system:

- the source-detector distance is fixed and equal to 100 cm,
- the size of the detector is 43 cm$\times$43 cm,
- the pixel size is 0.015 cm.

In order to produce projections of the calibration cage, we used the RTK package in Python [Rit+14] and the random set of $P = 10$ source positions λ_i from 0 cm to 7 cm, we chose $\lambda_0 = 0$ to simplify the

verification step. We used the same values for the shifts of the detector u_{λ_i} as source positions, since they correspond to the same shift of the translation table that we want to estimate.

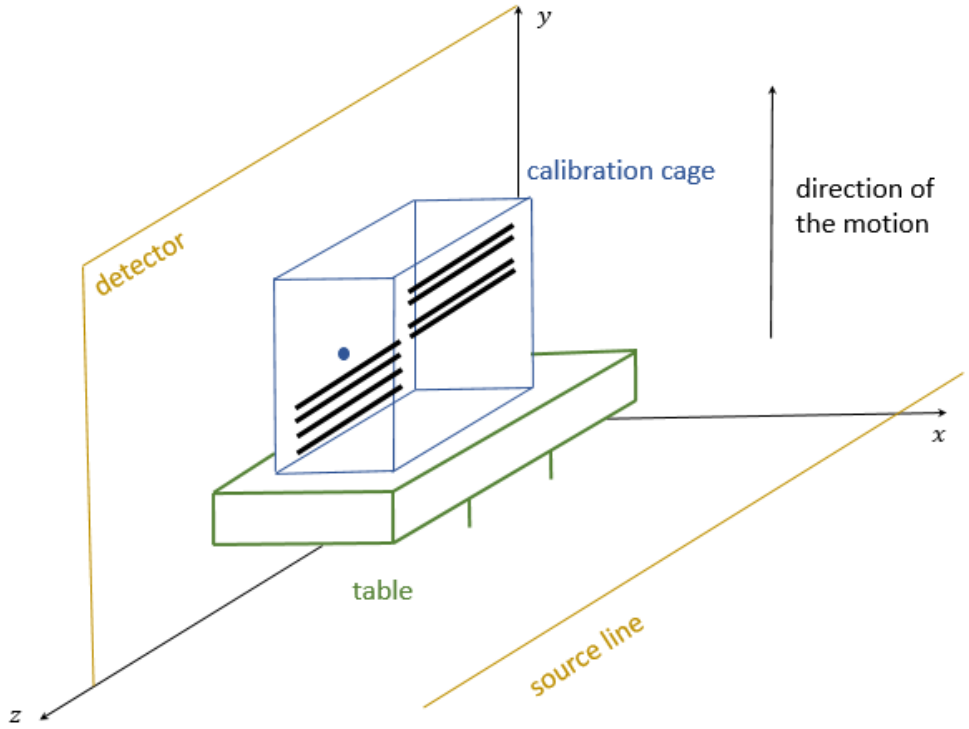

Figure 5.5: Configuration of the experiment.

The calibration cage as before consists of two sets of parallel sticks in two parallel planes (see fig. 5.5), it's possible to see dimensions of the calibration cage in the fig. 5.6. This cage should consist of the plexiglass plate or few plates assembled to each other in order to satisfy dimensions in the figure. Each line can be produced from the metal cylindrical stick, the diameter of each cylinder is 0.1 cm. In the middle between two sets of sticks we add a small ball in order to check after the calibration step the reprojection error, we will talk about this verification later. The chosen radius of the metal ball is 0.2 cm. For the RTK package we described our phantom (which is the calibration cage) in the file, so the initial position of the object is the following:

- for the line l_1 $x = 10$ cm, $y = 14$ cm, $z = 0$ cm,
- for the line l_2 $x = 10$ cm, $y = 15$ cm, $z = 0$ cm,
- for the line l_3 $x = 10$ cm, $y = 16$ cm, $z = 0$ cm,
- for the line l_4 $x = 10$ cm, $y = 17$ cm, $z = 0$ cm,
- for the point p_1 $x = 10$ cm, $y = 20$ cm, $z = 15$ cm,
- for the line l_5 $x = 20$ cm, $y = 23$ cm, $z = 0$ cm,

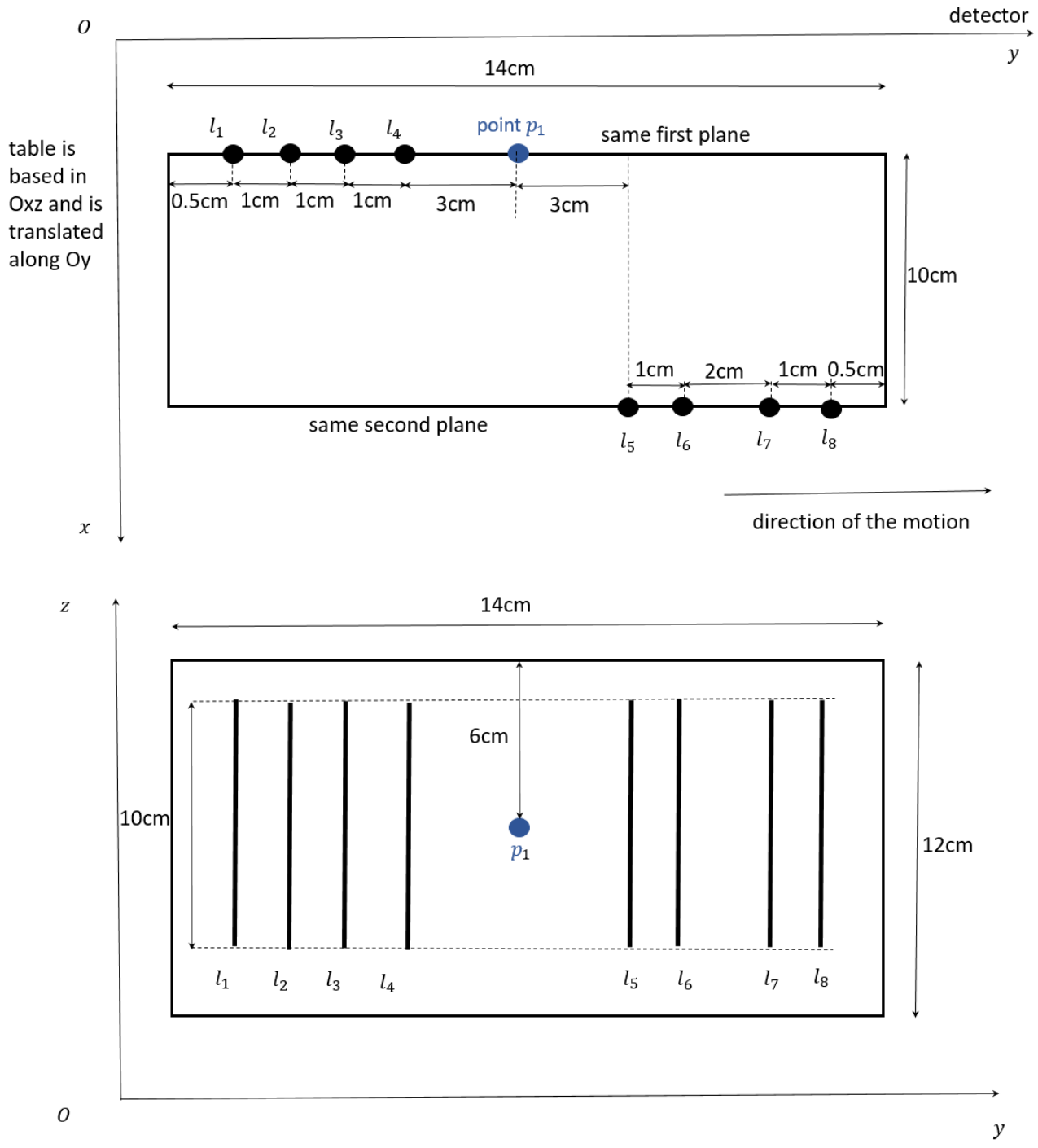

Figure 5.6: The calibration cage with all dimensions.

- for the line l_6 $x = 20$ cm, $y = 24$ cm, $z = 0$ cm,
- for the line l_7 $x = 20$ cm, $y = 26$ cm, $z = 0$ cm,
- for the line l_8 $x = 20$ cm, $y = 27$ cm, $z = 0$ cm.

For our calibration algorithm it means that we know the pattern of the calibration cage, thus for this

cage $L = 0.5$ cm, $k_1 = 3$ cm, $k_2 = 2$ cm, $k_3 = 4$ cm when there is no inclination. We know that for the calibration we need to solve our task just in one oblique plane or we need to fix v_0, we can fix this value for the algorithm manually, we took the detector line number 10.

For the step of the generation of projections we started with the case when the cage and lines are parallel to the axis Oz, it corresponds to the angle of inclination of 0 degrees in the table 5.2. Often it can be false, thus there is a small non-zero angle of inclination (see fig. 5.7). We simulated with RTK both situations in our numerical experiments.

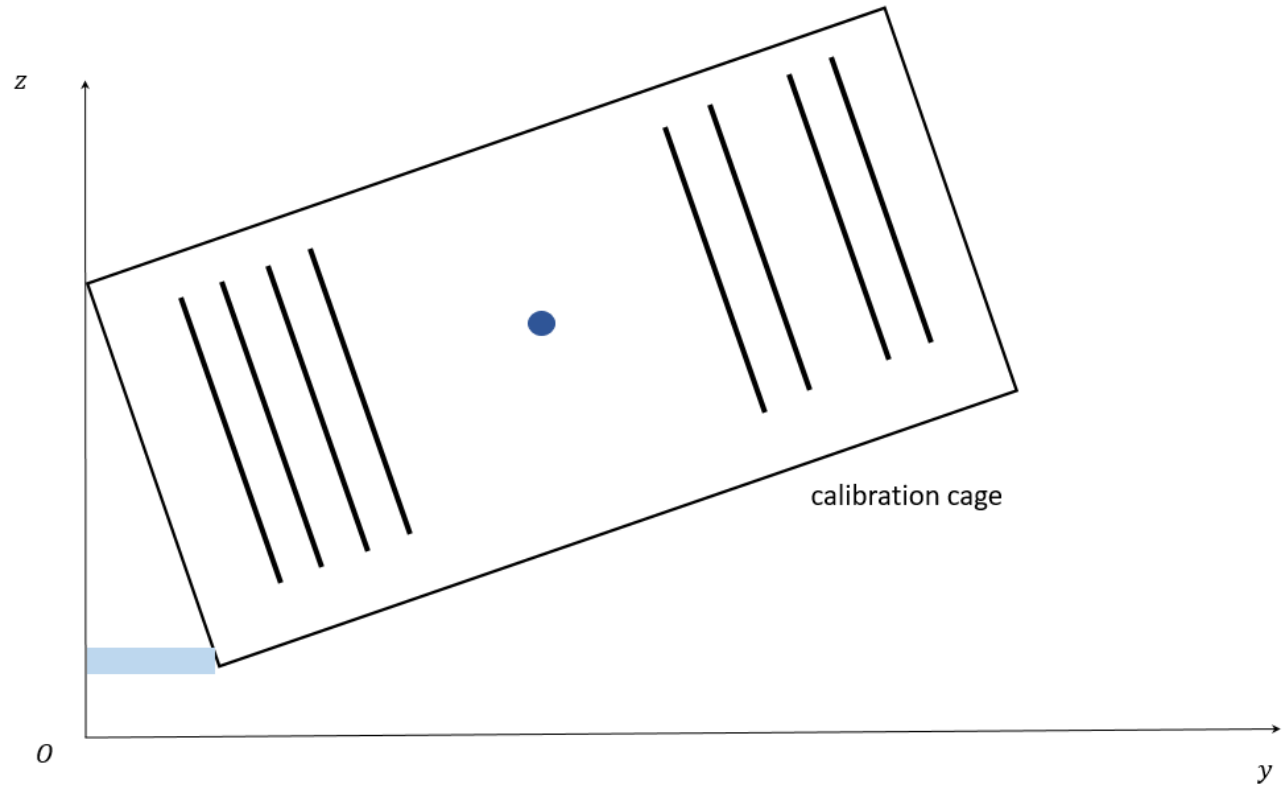

Figure 5.7: The possible position with the inclination of the calibration cage.

For the second step of the detection of the traces of markers in projected images we can work directly with gray values in projected images. We know that the projection of the set of these sticks is the set of stripes, each stripe has non-zero gray value. For the point we have the ellipse in the projection with non-zero gray value, while the rest (the background) is zero. By knowing this, we can simply extract by checking the gray value these non-zero strips and the ellipse. Moreover, we know that the line order is preserved, so we can simply numerate each stripe from 1 to 8 from the left to the right.

We need to build one value as the input of the calibration algorithm for each stripe for each projection. We can compute the direction of each stripe. For that, we use the built-in function of Python's skimage.transform to compute the well-known Hough transform. This transform helps us to identify two lines in the boundary for each stripe with a voting procedure. We describe each line with two

parameters: the distance between the line and the center of coordinates and the angle of the line. For each line defined by these two parameters we check if the input picture has non-zero pixels on this line. Thus, the Hough transform returns the set of more probable lines in the input pictures for which we voted a lot. It also helps to estimate the angle of inclination or the direction of the stripe. Then we can find the center of mass of each stripe and with the direction of the stripe we can compute for the necessary line of the detector the estimation of the Dirac in the projection. If this estimation is good or not we can see during the verification step.

The detection of lines in images can be done with Hough transform, but also with other techniques, for example, with the help of the Radon transform. The set of corresponding algorithms was described in [Spe15] and they performed better than the standard combination of the Canny edge detector plus the Hough transform to detect lines in projections of edges of colimators.

Since we can estimate the angle of inclination, we can correct the value of L needed for the calibration algorithm. Since we still take the oblique plane through the line parallel to Oz, then L should be replaced with $L/\cos\gamma$, where γ is the angle of inclination estimated with the Hough transform for each line and set as the mean value for all stripes for all projections.

We show the errors of estimations of the calibration parameters in the table 5.2 for different angles of inclination. We show mean absolute errors, thus, for example, for source positions we computed $\frac{1}{P}\sum_{i=0}^{P-1}\left|\hat{\lambda}_i-\lambda_i\right|$, where λ_i is the true value for the projection i, $\hat{\lambda}_i$ is the estimation with our algorithm of the source position for the projection i. We put both results for λ_i and u_{λ_i}, but for the final calibration of the translation of the table we propose to use the estimations for u_{λ_i}, since they are slightly better. Since the translation of the table can be known very precisely, then it could be possible to verify the estimated shifts with their true values in realistic settings.

The second verification is the reprojection error. For that, we need to use the detected projection of the introduced to the calibration cage point p_1. We use only one value for each projection for this point, we chose this value as the center of mass of the corresponding detected ellipse. We know from [Des+06] that the cone-beam projection of the center of mass of a ball is not exactly the center of mass of the projection but the error is very small unless the cone angle is huge. We model the detected point as a Dirac. For each projection we know that the 3D point $(c_1,c_2,c_3)^T$ should go to the 2D point $(u,v)^T$ with $u=\frac{c_2D-c_1\hat{\lambda}}{D-c_1}+\hat{u}_\lambda$, $v=\frac{Dc_3}{D-c_1}$, we use here the estimated calibration parameters. From the first equation we have the equation for c_1 and c_2 $(u-\hat{u}_\lambda-\hat{\lambda})c_1+Dc_2=(u-\hat{u}_\lambda)D$. We can estimate $\hat{c}_1$ and $\hat{c}_2$ with 10 projections with the least square method. For the second equation we can estimate $\hat{c}_3$. Then with the estimated calibration parameters we can reproject the 3D point that we computed with such method, this reprojected 2D point we can compare with the true 2D point in order to evaluate the precision of the calibration parameters. We computed the mean reprojection error over all projections. We described how we calculated the reprojection error from the table 5.2.

From all numerical experiments, we can see that the results of the calibration are good, thus we can use this method for the real system. It's also possible to see that the calibration is better in the case of

non-zero angle of inclination. We suppose that this situation occurs because of the better compensation of discretization errors during the computation of the center of mass of each stripe on the projected image. In the case of zero angle of inclination we have systematic errors of discretization while non-zero angle of inclination implies non-regular errors, thus we have less error during the computation of the center of mass. One disadvantage of this algorithm is that we need to place the calibration cage to have planes with sticks parallel to the detector plane (and the line of sources).

Angle γ, degrees	**Error for λ_i**	**Error for u_{λ_i}**	**Reprojection error**
-10	1.33×10^{-3}	2.94×10^{-4}	2.45×10^{-5}
-7	6.82×10^{-4}	9.71×10^{-5}	3.47×10^{-5}
-5	1.76×10^{-4}	6.87×10^{-5}	4.05×10^{-5}
0	2.17×10^{-2}	5.20×10^{-3}	3.24×10^{-4}
5	2.37×10^{-4}	9.23×10^{-5}	5.67×10^{-5}
7	4.57×10^{-4}	1.43×10^{-4}	5.08×10^{-5}
10	4.71×10^{-4}	2.89×10^{-4}	4.50×10^{-5}

Table 5.2: Mean absolute errors for calibration parameters and reprojection errors for different angles of inclination γ, all errors are in cm.

In this section we presented our definition for fixed λ of the cone-beam transform on distributions with sources on a line. We showed that as in the case of functions, the cone-beam calibration problem can be reduced to the fan-beam calibration problem in one oblique plane. Using a proper calibration cage, we can perform the analytical self-calibration with truncated data when the set of markers is not truncated. This calibration procedure can be tested with the real X-ray system.

5.3 Cone-beam transform on distributions with sources on a plane

5.3.1 Definition and DCC

In this section about cone-beam transform with sources on a plane parallel to the detector, we refer to the geometry presented in the fig. 1.4. The cone-beam transform on functions for this case was described by the integral model Eq. (1.10).

Our definition for fixed λ_1, λ_2. In this section we use the same coordinate system introduced in the fig. 1.4. The geometry can be also seen in the fig. 5.8. There the source position has coordinates $\vec{s}_\lambda = (\lambda_1, \lambda_2, 0)^T$, the detector is at $x_3 = D$, u and v are the parameters of the detector. We want to generalize this definition to distributions and provide the generalization of known DCC of the Theorem 1.3 to distributions.

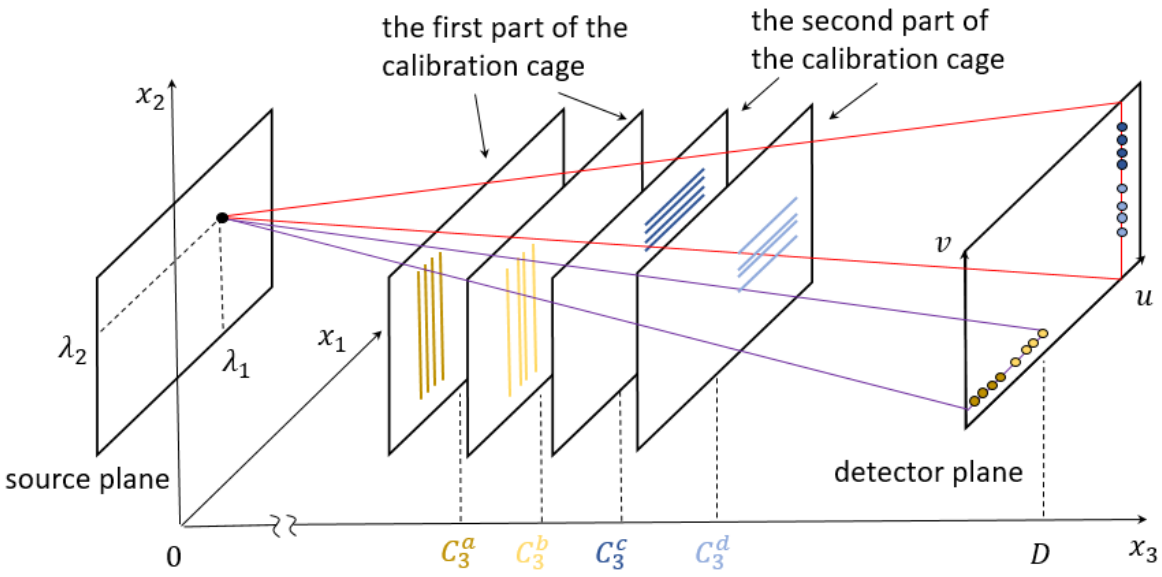

Figure 5.8: The 3D cone-beam geometry with sources on a plane with the calibration cage of two groups of 8 parallel sticks each.

As before we can denote the cone-beam transform for fixed λ_1 and λ_2 with $\mathfrak{D}_{\lambda_1,\lambda_2} f(u,v) := \mathfrak{D} f(\lambda_1,\lambda_2,u,v)$ for $f \in \mathscr{D}_3$ with support in $Y_3 = \mathbb{R}^2 \times (D_1, D_2)$, $0 < D_1 < D_2 < D$. Then we consider $\mathfrak{D}_{\lambda_1,\lambda_2} f$ as a function of two variables. We need to define the adjoint operator $\mathfrak{D}^*_{\lambda_1,\lambda_2}$. Note that for $f \in \mathscr{D}_3$ and $\phi \in \mathscr{E}_2$:

$$
\begin{aligned}
(\mathfrak{D}_{\lambda_1,\lambda_2} f, \phi) &= \int_{\mathbb{R}^2} \mathfrak{D}_{\lambda_1,\lambda_2} f(u,v)\phi(u,v)dudv \\
&= \int_{\mathbb{R}^2}\int_0^{+\infty} f(\lambda_1 + l(u-\lambda_1), \lambda_2 + l(v-\lambda_2), lD)dl\,\phi(u,v)dudv \\
&= \frac{1}{D}\int_{\mathbb{R}^2}\int_0^{+\infty} f\left(\lambda_1 + \frac{t_3}{D}(u-\lambda_1), \lambda_2 + \frac{t_3}{D}(v-\lambda_2), t_3\right) dt_3\,\phi(u,v)dudv \\
= \int_0^{+\infty}\int_{\mathbb{R}^2} f(t_1,t_2,t_3)\phi&\left(\frac{Dt_1 - \lambda_1(D-t_3)}{t_3}, \frac{Dt_2 - \lambda_2(D-t_3)}{t_3}\right)\frac{D}{t_3^2}dt_1dt_2dt_3 = \langle f, \mathfrak{D}^*_{\lambda_1,\lambda_2}\phi\rangle, \quad (5.12)
\end{aligned}
$$

where $(\cdot,\cdot)$ is the scalar product in $L^2(\mathbb{R}^2)$, $\langle\cdot,\cdot\rangle$ is the scalar product in $L^2(Y_3)$, we used the change of variables $t_3 = lD$, $dl = \frac{dt_3}{D}$; $t_1 = \lambda_1 + \frac{t_3}{D}(u-\lambda_1)$, $du = \frac{D}{t_3}dt_1$; $t_2 = \lambda_2 + \frac{t_3}{D}(v-\lambda_2)$, $dv = \frac{D}{t_3}dt_2$.

Now we can define the adjoint operator for functions from $\mathscr{E}_2$:

$$
\mathfrak{D}^*_{\lambda_1,\lambda_2}\phi(\vec{x}) := \frac{D}{x_3^2}\phi\left(\frac{Dx_1 - \lambda_1(D-x_3)}{x_3}, \frac{Dx_2 - \lambda_2(D-x_3)}{x_3}\right). \quad (5.13)
$$

Definition 5.3. *The cone-beam transform on a plane of a compactly supported distribution $f \in \mathscr{E}'(Y_3)$ at fixed λ_1 and λ_2 is a distribution from $\mathscr{E}'_2$ defined by the standard duality*

$$
(\mathfrak{D}_{\lambda_1,\lambda_2} f, \phi) = \langle f, \mathfrak{D}^*_{\lambda_1,\lambda_2}\phi\rangle \quad (5.14)
$$

with the adjoint operator from Eq. (5.13).

We justify why this definition is properly formulated in the Appendix A.5.

If we model markers with Dirac distributions, then

$$(\mathfrak{D}_{\lambda_1,\lambda_2}\delta_{\vec{c}}(u,v),\phi(u,v)) = \langle \delta_{\vec{c}}(\vec{x}), \mathfrak{D}^*_{\lambda_1,\lambda_2}\phi(\vec{x})\rangle = \left\langle \delta_{\vec{c}}(\vec{x}), \frac{D}{x_3^2}\phi\left(\frac{Dx_1-\lambda_1(D-x_3)}{x_3}, \frac{Dx_2-\lambda_2(D-x_3)}{x_3}\right)\right\rangle$$
$$= \frac{D}{c_3^2}\phi\left(\frac{Dc_1-\lambda_1(D-c_3)}{c_3}, \frac{Dc_2-\lambda_2(D-c_3)}{c_3}\right) = \frac{D}{c_3^2}\delta_{\tilde{c}}(\phi),$$
$$\text{where } \tilde{c} = \left(\frac{Dc_1-\lambda_1(D-c_3)}{c_3}, \frac{Dc_2-\lambda_2(D-c_3)}{c_3}\right). \quad (5.15)$$

So, the cone-beam projection with sources on a plane of a Dirac is again a weighted Dirac. The point $\tilde{c}$ that we found is actually the usual perspective projection of the point $\vec{c}$. To understand this, let us look at similar triangles in the fig. 5.9. From the left picture we have two similar triangles $\triangle A_{2\text{par}}H_2Q_{2\text{par}}$ and $\triangle A_{2\text{par}}H_2'Q_{2\text{par}}'$, where $A_{2\text{par}}$ is the projection of the source position with coordinates $(\lambda_1,\lambda_2,0)^T$ onto the plane Ox_1x_3, $Q_{2\text{par}}$ is the projection of the marker point Q with coordinates $(c_1,c_2,c_3)^T$ onto the plane Ox_1x_3, $Q_{2\text{par}}'$ is the projection of the projected to the detector point Q' onto Ox_1x_3. Then we have $\frac{A_{2\text{par}}H_2}{A_{2\text{par}}H_2'} = \frac{Q_{2\text{par}}H_2}{Q_{2\text{par}}'H_2'}$ or $\frac{c_1-\lambda_1}{u-\lambda_1} = \frac{c_3}{D}$, thus we have for the projected point the first coordinate $u = \frac{Dc_1-\lambda_1(D-c_3)}{c_3}$. From the right picture we have two similar triangles $\triangle A_{1\text{par}}H_1Q_{1\text{par}}$ and $\triangle A_{1\text{par}}H_1'Q_{1\text{par}}'$, where $A_{1\text{par}}$ is the projection of the source position with coordinates $(\lambda_1,\lambda_2,0)^T$ onto the plane Ox_2x_3, $Q_{1\text{par}}$ is the projection of the marker point Q with coordinates $(c_1,c_2,c_3)^T$ onto the plane Ox_2x_3, $Q_{1\text{par}}'$ is the projection of the projected to the detector point Q' onto Ox_2x_3. Then we have $\frac{A_{1\text{par}}H_1}{A_{1\text{par}}H_1'} = \frac{Q_{1\text{par}}H_1}{Q_{1\text{par}}'H_1'}$ or $\frac{c_2-\lambda_2}{v-\lambda_2} = \frac{c_3}{D}$, thus we have for the projected point the second coordinate $v = \frac{Dc_2-\lambda_2(D-c_3)}{c_3}$.

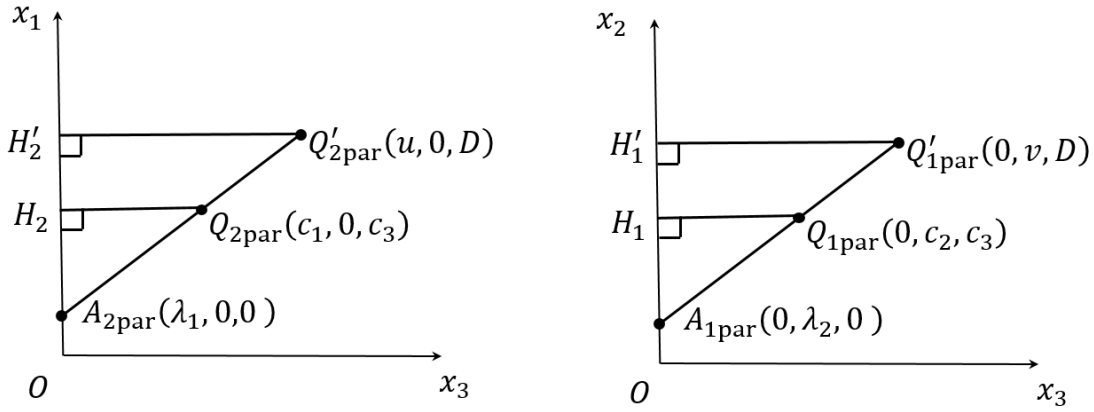

Figure 5.9: Left: the view in the plane Ox_1x_3, right: the view in the plane Ox_2x_3.

DCC for our definition. As before, we plan to use only the necessary part of DCC. So, we want to show the generalisation to distributions of DCC from the Theorem 1.3:

Theorem 5.1. *If $f \in \mathscr{E}'(Y_3)$, $g_{\lambda_1,\lambda_2} := \mathfrak{D}_{\lambda_1,\lambda_2}f$ is the cone-beam transform on a plane of f for fixed λ_1, λ_2, then:*

1. $g_{\lambda_1,\lambda_2} \in \mathscr{E}_2'$,

2. *for* $k = 0,1,2,\ldots$ *we have the moment conditions:*

$$(g_{\lambda_1,\lambda_2}(u,v),(uU+vV)^k) = \mathscr{P}_k(U,V,-\lambda_1 U-\lambda_2 V), \tag{5.16}$$

where $\mathscr{P}_k(U,V,W)$ *is a homogeneous polynomial of degree* k.

Proof. The first point is showed in the Appendix A.5. Now we prove the moment conditions. Obviously $(u,v) \longmapsto (uU+vV)^k \in \mathscr{E}_2$, then

$$\begin{aligned}
(\mathfrak{D}_{\lambda_1,\lambda_2} f(u,v),(uU+vV)^k) &= \langle f(\vec{x}), \mathfrak{D}^*_{\lambda_1,\lambda_2}((uU+vV)^k)(\vec{x})\rangle \\
&= \left\langle f(\vec{x}), \frac{D}{x_3^2}\left(\frac{Dx_1-\lambda_1(D-x_3)}{x_3}U + \frac{Dx_2-\lambda_2(D-x_3)}{x_3}V\right)^k \right\rangle \\
&= \left\langle f(\vec{x}), \frac{D}{x_3^{k+2}}(Dx_1U+Dx_2V+(D-x_3)(-\lambda_1U-\lambda_2V))^k \right\rangle \\
&= \left\langle f(\vec{x}), \frac{D}{x_3^{k+2}} \sum_{\substack{i,j,l \\ i+j+l=k}} \frac{k!}{i!j!l!}(Dx_1U)^i(Dx_2V)^j((D-x_3)(-\lambda_1U-\lambda_2V))^l \right\rangle \\
= \sum_{\substack{i,j,l \\ i+j+l=k}} \frac{k!}{i!j!l!} U^i V^j (-\lambda_1U-\lambda_2V)^l &\left\langle f(\vec{x}), \frac{D}{x_3^{k+2}}(Dx_1)^i(Dx_2)^j(D-x_3)^l \right\rangle = \mathscr{P}_k(U,V,-\lambda_1U-\lambda_2V).
\end{aligned}$$

□

5.3.2 Calibration algorithm

We can again try to use the moment conditions to provide a calibration procedure. We want to use the same logic in the derivation as for the fan-beam case. We also want to calibrate with the specific calibration cage with unknown position in the world. The purpose of our new algorithm is to calibrate with truncated data. Suppose that we work with a lattice of u,v, but the system is moving. For P projections for unknown coordinates of source positions λ_{1i}, λ_{2i} our measurements are $m_i(u,v) = \mathfrak{D}_{\lambda_{1i},\lambda_{2i}} f(u-u_i, v-v_i)$, where u_i, v_i are unknown detector jitters.

Our purpose now is to show that our task can be separated into two independent tasks to find the couple of λ_{1i} and u_i for the first task and λ_{2i} and v_i for the second task. Since we have same shifts u_i and v_i for each detector point for fixed source position, then we can solve each task in one oblique plane containing the current source position and the appropriate detector line as for the case with sources on a plane.

Our calibration cage consists of two parts: the first part of 8 vertical sticks for the first calibration task, the second part of 8 horizontal sticks for the second calibration task (see fig. 5.8). We placed these

sticks as close as possible to the detector and these two groups are separated such that it's possible to find one detector line for each part (horizontal detector line for the vertical sticks and vertical detector line for horizontal sticks) containing only projections of one part of our calibration cage. In this case, we can work separately with the projection of vertical sticks for the fixed detector line and the projection of horizontal sticks for another fixed detector line. So, as the first step the selection of two appropriate detector lines (one horizontal and one vertical) for the current source position should be done.

Let us talk only about the first task to estimate λ_{1i} and u_i with the group of vertical sticks. For each group of 8 sticks we have the known pattern, the same as we used for the case with sources on a line. Thus, each group of 4 sticks belongs to one plane. We will use the superscript $l = a$ or $l = b$ to label the each group of 4 sticks. So, the first 4 sticks are in the unknown plane $x_3 = C_3^a$, the second 4 sticks are in the unknown plane $x_3 = C_3^b$. Moreover, since sticks are perpendicular to the x_1-axis, then from the known pattern of sticks we know for the first coordinates of the marker points (points in the intersection of sticks and the fixed oblique plane) that $c_{11}^a = p_a - k_1 L$, $c_{21}^a = p_a - L$, $c_{31}^a = p_a + L$, $c_{41}^a = p_a + k_1 L$, $c_{11}^b = p_b - k_2 L$, $c_{21}^b = p_b - k_3 L$, $c_{31}^b = p_b + k_3 L$, $c_{41}^b = p_b + k_2 L$, where p_a, p_b are unknown, but we know L, $k_1 > 0$, $k_2 > 0$, $k_3 > 0$. Note that unknowns C_3^a, C_3^b, p_a, p_b doesn't depend on i. Thus, we want to exploit these known properties of the calibration cage and known detected points of the sticks q_{ij}^l in the projection for one fixed detector line, where i is the projection number, j is the index of the point, l is the group of sticks, $l \in \{a, b\}$.

Non-uniqueness of the solution. As before, we have a non-uniqueness of the solution. As before, we will search for the solution such that $\lambda_{10} = 0$ and $u_0 = 0$. The ambiguity is represented by the special shearing and translation of the object with $f_{M,\vec{t}}(\vec{x}) := f(M\vec{x} + \vec{t})$, where

$$M := \begin{pmatrix} 1 & 0 & -(u' + \lambda_1')/D \\ 0 & 1 & -(v' + \lambda_2')/D \\ 0 & 0 & 1 \end{pmatrix}, \qquad \vec{t} := (\lambda_1', \lambda_2', 0)^T. \tag{5.17}$$

It's true for functions:

$$\begin{aligned} \mathfrak{D} f_{M,\vec{t}}(\lambda_1, \lambda_2, u, v) &= \int_0^{+\infty} f_{M,\vec{t}}(\lambda_1 + l(u - \lambda_1), \lambda_2 + l(v - \lambda_2), lD) dl \\ &= \int_0^{+\infty} f(\lambda_1 + lu - l\lambda_1 - \frac{u' + \lambda_1'}{D} Dl + \lambda_1', \lambda_2 + lv - l\lambda_2 - \frac{v' + \lambda_2'}{D} Dl + \lambda_2', lD) dl \\ &= \int_0^{+\infty} f(\lambda_1 + \lambda_1' + l(u - u') - l(\lambda_1 + \lambda_1'), \lambda_2 + \lambda_2' + l(v - v') - l(\lambda_2 + \lambda_2'), lD) dl \\ &= \mathfrak{D} f(\lambda_1 + \lambda_1', \lambda_2 + \lambda_2', u - u', v - v'). \end{aligned} \tag{5.18}$$

With the projection data of f, if the detector is shifted by u', v' and the source positions are all shifted by $-\lambda_1'$, $-\lambda_2'$, then there exists an object $f_{M,\vec{t}}$ having the same projection data from the original source and detector positions. Thus, we can't estimate the calibration parameters and f better than up to M and $\vec{t}$. It's also true for $f = \delta_{\vec{c}} \in \mathscr{E}'(Y_3)$. Let us consider for $f \in \mathscr{E}'(Y_3)$ the distribution

$f_{M,\vec{t}} \in \mathscr{E}'(Y_3)$ defined as $\langle f_{M,\vec{t}}(\vec{x}), \phi(\vec{x})\rangle = \langle f(\vec{x}), \phi(M^{-1}(\vec{x}-\vec{t})\rangle$, where $M^{-1} = \begin{pmatrix} 1 & 0 & (u'+\lambda_1')/D \\ 0 & 1 & (v'+\lambda_2')/D \\ 0 & 0 & 1 \end{pmatrix}$, thus $M^{-1}(\vec{x}-\vec{t}) = \begin{pmatrix} x_1 - \lambda_1' + \frac{u'+\lambda_1'}{D}x_3 \\ x_2 - \lambda_2' + \frac{v'+\lambda_2'}{D}x_3 \\ x_3 \end{pmatrix}$. It's true that $(\delta_{\vec{c}})_{M,\vec{t}}$ is the distribution $\delta_{M^{-1}(\vec{c}-\vec{t})} \in \mathscr{E}'(Y_3)$. Then we can show

$$
\begin{aligned}
(\mathfrak{D}_{\lambda_1,\lambda_2}(\delta_{\vec{c}})_{M,\vec{t}}(u,v), \phi(u,v)) &= \left\langle (\delta_{\vec{c}})_{M,\vec{t}}(\vec{x}), \frac{D}{x_3^2}\phi\left(\frac{Dx_1-\lambda_1(D-x_3)}{x_3}, \frac{Dx_2-\lambda_2(D-x_3)}{x_3}\right)\right\rangle \\
&= \left\langle \delta_{M^{-1}(\vec{c}-\vec{t})}(\vec{x}), \frac{D}{x_3^2}\phi\left(\frac{Dx_1-\lambda_1(D-x_3)}{x_3}, \frac{Dx_2-\lambda_2(D-x_3)}{x_3}\right)\right\rangle \\
&= \frac{D}{c_3^2}\phi\left(\frac{D(c_1-\lambda_1'+c_3(u'+\lambda_1')/D)-(D-c_3)\lambda_1}{c_3}, \frac{D(c_2-\lambda_2'+c_3(v'+\lambda_2')/D)-(D-c_3)\lambda_2}{c_3}\right) \\
&= \frac{D}{c_3^2}\phi\left(\frac{Dc_1-(D-c_3)(\lambda_1+\lambda_1')}{c_3}+u', \frac{Dc_2-(D-c_3)(\lambda_2+\lambda_2')}{c_3}+v'\right) \\
&= (\mathfrak{D}_{\lambda_1+\lambda_1',\lambda_2+\lambda_2'}\delta_{\vec{c}}(u,v), \phi(u+u',v+v')) = (\mathfrak{D}_{\lambda_1+\lambda_1',\lambda_2+\lambda_2'}\delta_{\vec{c}}(u-u',v-v'), \phi(u,v)). \quad (5.19)
\end{aligned}
$$

Thus, if the distribution is sheared and translated with M and $\vec{t}$ given in Eq. (5.17), then it's equivalent to translate sources by $-\lambda_1'$, $-\lambda_2'$ and the detector by u', v'.

Our algorithm. We compute moments from Eq. (5.16) for each group of 4 markers separately. We will provide the similar formulas with the fan-beam case for this cone-beam geometry. We will provide all formulas for the first task to estimate λ_{1i}, u_i, p_a, p_b, C_3^a, C_3^b. But these formulas are also applicable for the second task of the estimation of λ_{2i}, v_i, p_c, p_d, C_3^c, C_3^d, we only need to replace λ_1 with λ_2, u with v and indices a, b with indices c, d.

The moments of order 1 for $m_i^l(u,v) = \mathfrak{D}_{\lambda_{1i},\lambda_{2i}} f^l(u-u_i, v-v_i)$ and $f^l = \sum_{j=1}^4 \delta_{\vec{c}_j^{\,l}}$ using $c_{j3}^l = C_3^l$:

$$
\begin{aligned}
M_1^l(i) = (m_i^l(u,v),u) &= (\mathfrak{D}_{\lambda_{1i},\lambda_{2i}} f^l(u-u_i,v-v_i),u) = (\mathfrak{D}_{\lambda_{1i},\lambda_{2i}} f^l(u,v), u+u_i) \\
&= \left(\sum_{j=1}^4 \frac{D}{(c_{j3}^l)^2}\delta_{\tilde{c}(\vec{c}_j^{\,l},\lambda_{1i},\lambda_{2i})}(u,v), u+u_i\right) = \sum_{j=1}^4 \frac{D^2 c_{j1}^l - \lambda_{1i}D(D-c_{j3}^l)}{(c_{j3}^l)^3} + \sum_{j=1}^4 \frac{u_i D}{(c_{j3}^l)^2} \\
&= \sum_{j=1}^4 \frac{D^2 c_{j1}^l - \lambda_{1i}D(D-C_3^l)}{(C_3^l)^3} + \sum_{j=1}^4 \frac{u_i D}{(C_3^l)^2}. \quad (5.20)
\end{aligned}
$$

Since for the solution $\lambda_{10} = 0$, $u_0 = 0$, then

$$
M_1^a(0) = \sum_{j=1}^4 \frac{D^2 c_{j1}^a}{(C_3^a)^3}, \qquad M_1^b(0) = \sum_{j=1}^4 \frac{D^2 c_{j1}^b}{(C_3^b)^3}. \quad (5.21)
$$

Since we know abscissas q_{ij}^l of the detected points (q_{ij}^l, s_{ij}^l), then we have

$$M_1^l(i) = \left(\sum_{j=1}^{4} \frac{D}{(c_{j3}^l)^2} \delta_{(q_{ij}^l, s_{ij}^l)}, u \right) = \sum_{j=1}^{4} \frac{D q_{ij}^l}{(c_{j3}^l)^2} = \frac{D}{(C_3^l)^2} \sum_{j=1}^{4} q_{ij}^l. \tag{5.22}$$

Let us use Eq. (5.21) and Eq. (5.22) in Eq. (5.20) and multiple each side of the equation by $\frac{(C_3^l)^2}{D}$. If we introduce new unknown variables $r_a > 0$ and $r_b > 0$ with

$$r_a = \frac{D - C_3^a}{C_3^a}, \qquad r_b = \frac{D - C_3^b}{C_3^b}, \tag{5.23}$$

then we have

$$u_i - \lambda_{1i} r_l = \Delta \tilde{M}_1^l(i), \qquad \Delta \tilde{M}_1^l(i) := \frac{1}{4} \left(\sum_{j=1}^{4} q_{ij}^l - \sum_{j=1}^{4} q_{0j}^l \right). \tag{5.24}$$

We need to add the moment conditions of order 2 to complete our algorithm. Using $c_{j3}^l = C_3^l$, we have

$$\begin{aligned} M_2^l(i) = (m_i^l(u,v), u^2) = (\mathfrak{D}_{\lambda_{1i}, \lambda_{2i}} f^l(u,v), (u+u_i)^2) = \sum_{j=1}^{4} \frac{D}{(c_{j3}^l)^2} (\tilde{c}_1(\vec{c}_j^{\,l}, \lambda_{1i}, \lambda_{2i}) + u_i)^2 \\ = \frac{D}{(C_3^l)^2} \sum_{j=1}^{4} \left(\frac{D c_{j1}^l - \lambda_{1i}(D - c_{j3}^l)}{c_{j3}^l} + u_i \right)^2 = \frac{D}{(C_3^l)^2} \sum_{j=1}^{4} \left(\frac{D c_{j1}^l}{C_3^l} + u_i - \lambda_{1i} r_l \right)^2 \\ = \frac{D}{(C_3^l)^2} \sum_{j=1}^{4} \left(\frac{D c_{j1}^l}{C_3^l} + \Delta \tilde{M}_1^l(i) \right)^2. \end{aligned} \tag{5.25}$$

From Eq. (5.21), (5.22) and (5.23) we obtain

$$\sum_{j=1}^{4} \frac{D c_{j1}^l}{C_3^l} = \frac{D}{C_3^l} \sum_{j=1}^{4} c_{j1}^l = (1 + r_l) \sum_{j=1}^{4} c_{j1}^l = \sum_{j=1}^{4} q_{0j}^l, \tag{5.26}$$

then we have

$$\sum_{j=1}^{4} \frac{D c_{j1}^l}{C_3^l} = \sum_{j=1}^{4} q_{0j}^l, \tag{5.27}$$

but also from $\sum_{j=1}^{4} c_{j1}^l = 4 p_l$ and Eq. (5.26)

$$(1 + r_l) 4 p_l = \sum_{j=1}^{4} q_{0j}^l. \tag{5.28}$$

For chosen markers $\sum_{j=1}^{4} (c_{j1}^a)^2 = 4p_a^2 + (2 + 2k_1^2)L^2$ and $\sum_{j=1}^{4} (c_{j1}^b)^2 = 4p_b^2 + (2k_2^2 + 2k_3^2)L^2$. Then we can compute using Eq. (5.23):

$$\sum_{j=1}^{4} \left(\frac{D c_{j1}^l}{C_3^l} \right)^2 = (1 + r_l)^2 \sum_{j=1}^{4} (c_{j1}^l)^2 = \begin{cases} (1 + r_a)^2 (4p_a^2 + (2 + 2k_1^2)L^2) & \text{if } l = a \\ (1 + r_b)^2 (4p_b^2 + (2k_2^2 + 2k_3^2)L^2) & \text{if } l = b. \end{cases} \tag{5.29}$$

We can develop Eq. (5.25), multiply each side of the equation by $\frac{(C_3^l)^2}{D}$, use Eq. (5.28), then we obtain the same system of equations to find r_a and r_b as for the fan-beam case:

$$\begin{cases} \sum_{j=1}^4 (q_{ij}^a)^2 = (1+r_a)^2(2+2k_1^2)L^2 + \frac{1}{4}\left[\sum_{j=1}^4 q_{0j}^a\right]^2 + 2\Delta\tilde{M}_1^a(i)\sum_{j=1}^4 q_{0j}^a + 4[\Delta\tilde{M}_1^a(i)]^2 \\ \sum_{j=1}^4 (q_{ij}^b)^2 = (1+r_b)^2(2k_2^2+2k_3^2)L^2 + \frac{1}{4}\left[\sum_{j=1}^4 q_{0j}^b\right]^2 + 2\Delta\tilde{M}_1^b(i)\sum_{j=1}^4 q_{0j}^b + 4[\Delta\tilde{M}_1^b(i)]^2. \end{cases} \tag{5.30}$$

As before, with only one projection i we can estimate from Eq. (5.30) r_a and r_b, then C_3^a and C_3^b for fixed D from Eq. (5.23). Then Eq. (5.28) gives us p_a and p_b. For each $i \geq 1$ from the system of two equations Eq. (5.24) we can compute u_i, λ_{1i}. This system gives a unique solution, since:

$$\det\begin{pmatrix} 1 & -r_a \\ 1 & -r_b \end{pmatrix} = r_a - r_b = \frac{D-C_3^a}{C_3^a} - \frac{D-C_3^b}{C_3^b} = \frac{C_3^b(D-C_3^a) - C_3^a(D-C_3^b)}{C_3^a C_3^b} = \frac{D(C_3^b - C_3^a)}{C_3^a C_3^b} \neq 0.$$

Thus, the equations Eq. (5.30), (5.23), (5.28), (5.24) form the analytical procedure to estimate calibration parameters u_i, λ_{1i} and the locations C_3^a, C_3^b, p_a and p_b of the first part of the calibration cage. The same procedure can be developed for the second part of the calibration cage and the second group of calibration parameters v_i, λ_{2i}.

Numerical experiments. We made numerical simulations for the described task to estimate the half of parameters, since simulations for the second half should be essentially the same. We measure distances in cm. For the calibration cage we chose as before $L = 0.4$, $k_1 = 3$, $k_2 = 1$, $k_3 = 2$, the x_1-distance between the last point of the first group and the first point of the second group was $3L$. For other parameters: $p_a = 5$, $p_b = 8.2$, $C_3^a = 8$, $C_3^b = 9.5$. We fixed $P = 30$ random values for source positions λ_{1i} in $[0, 10]$. We chose the grid on $u \in [0, 10]$ with the sampling step 0.01. The detector jitters u_i were generated as random uniform noise on the interval $[-0.05, 0.05)$. In order to simplify the final comparison of results we fixed $\lambda_{10} = 0$, $u_0 = 0$.

We wrote our algorithm in Python 3. For exact projections our algorithm was almost perfect (see the first part of the table 5.3). For noisy experiments, we used a Gaussian noise $N(0, \sigma)$ added to q_{ij}^l, $\sigma = 0.01 \cdot Nl$, where Nl is the noise level, 0.01 is the pixel size of the initial image. In the first part of the table 5.3 we can also see the results of noisy experiments. We present the mean absolute errors of 100 realisations of the Gaussian noise for each Nl. Thus, for calibration parameters, for example, for source positions we computed

$$\frac{1}{100}\sum_{k=1}^{100}\frac{1}{P}\sum_{i=0}^{P-1}\left|\hat{\lambda}_{1i,k} - \lambda_{1i}\right|, \tag{5.31}$$

where λ_{1i} is the true value for the projection i, $\hat{\lambda}_{1i,k}$ is the estimation with our algorithm for the projection i with the realisation of the Gaussian noise number k added to projections, so k states for the experiment number. For marker positions, for example, for x_3-coordinates we computed

$$\frac{1}{100}\sum_{k=1}^{100}\frac{1}{2}\sum_{l\in\{a,b\}}\left|\hat{C}_{3,k}^l - C_3^l\right|, \tag{5.32}$$

where C_3^l is the true value, $\hat{C}_{3,k}^l$ is the estimation with our algorithm of the position of the marker group l with the realisation of the Gaussian noise number k added to projections. The numerical experiments show that the errors are proportional to Nl.

Note that the estimation of λ_{2i} and v_i can be made with the same code. We also fixed $P = 30$ random values for source positions λ_{2i} in $[0,10]$, the grid on $v \in [0,10]$ with the sampling step 0.01. The detector jitters v_i were generated as random uniform noise on the interval $[-0.05, 0.05)$. We also fixed $\lambda_{20} = 0$, $v_0 = 0$. We chose $p_c = 4$, $p_d = 7.2$, $C_3^c = 8$, $C_3^d = 9.5$. The numerical results are represented in the second part of the table 5.3.

Noise level Nl	**Noise std**	**Error for λ_{1i}**	**Error for u_i**	**Error for p_l**	**Error for C_3^l**
0%	0	6.95×10^{-13}	1.74×10^{-13}	1.18×10^{-13}	1.37×10^{-13}
10%	0.001	1.92×10^{-2}	4.02×10^{-3}	3.66×10^{-3}	4.61×10^{-3}
50%	0.005	1.03×10^{-1}	2.08×10^{-2}	1.81×10^{-2}	2.28×10^{-2}
100%	0.01	1.91×10^{-1}	3.76×10^{-2}	3.34×10^{-2}	4.31×10^{-2}
200%	0.02	3.84×10^{-1}	8.12×10^{-2}	7.20×10^{-2}	9.17×10^{-2}

Noise level Nl	**Noise std**	**Error for λ_{2i}**	**Error for v_i**	**Error for p_l**	**Error for C_3^l**
0%	0	7.71×10^{-14}	1.93×10^{-14}	1.20×10^{-14}	1.60×10^{-14}
10%	0.001	1.67×10^{-2}	3.55×10^{-3}	2.86×10^{-3}	4.22×10^{-3}
50%	0.005	1.01×10^{-1}	2.07×10^{-2}	1.51×10^{-2}	2.22×10^{-2}
100%	0.01	1.87×10^{-1}	3.79×10^{-2}	2.30×10^{-2}	4.43×10^{-2}
200%	0.02	3.89×10^{-1}	7.82×10^{-2}	5.73×10^{-2}	8.65×10^{-2}

Table 5.3: Mean absolute errors for calibration parameters and the positions of the markers with non-noisy and noisy projections, all errors are in cm.

In this section we presented our definition for fixed source position of the cone-beam transform on distributions with sources on a plane parallel to the detector plane. We showed that the calibration task can be separated to two independent calibration procedures. Each procedure consists of analytical equations similar to the fan-beam calibration solution. Thus, with the special calibration cage we can solve analytically this 3D calibration problem based only on the local information about markers when the complete data are truncated.

6 Conclusions

6.1 French version

Dans notre travail nous avons élargi les connaissances sur l'auto-étalonnage des systèmes radiologiques. Dans les deux parties avec les résultats, nous avons présenté deux approches différentes de l'auto-calibration, mais les résultats montrent des liens entre l'imagerie médicale et la vision par ordinateur.

- Dans le chapitre 2 nous avons présenté l'application de la méthode d'ajustement de faisceaux de la vision par ordinateur aux systèmes d'imagerie rayons X en faisceau conique tels que les C-arms. Nous avons montré que nous n'avons pas l'unicité de la solution à ce problème résolue par une optimisation non-linéaire et nous ne pouvons auto-calibrer qu'à une similitude près. Ce résultat est le même avec la vision par ordinateur. De plus, nous avons montré quelles données d'entrée nous donnent la solution différant par une similitude. Nous avons également montré numériquement que nous pouvons toujours effectuer la reconstruction dans ce cas. En revanche, nous ne pouvons généralement pas trouver cette similitude dans la situation réelle car nous ne connaissons évidemment pas les vraies valeurs des coordonnées 3D des marqueurs. Notre expérience avec les simulations numériques indique que si nous commençons notre procédure d'optimisation avec des valeurs initiales proches des vraies, nous devons obtenir une solution exactement vraie. Mais la question des valeurs initiales qui nous donnent la vraie solution exacte devrait être étudiée plus en profondeur.

- Dans les chapitres 3, 4 et 5 nous avons présenté une série d'algorithmes hybrides basés sur la même idée de combiner les DCC et les marqueurs dans les mires d'étalonnage. Grâce aux DCC, nous pouvons obtenir les procédures analytiques pour calibrer à l'aide des moments. Grâce aux marqueurs, nous pouvons réduire l'application de DCC aux informations locales sur les marqueurs, ce qui est courant en vision par ordinateur. Ainsi, nous pouvons utiliser les DCC pour les données tronquées. Notons également que nos procédures sont basées sur les DCC appliquées à des distributions prouvées dans ce travail, correspondant à la généralisation de DCC connues. Contrairement aux algorithmes précédents, nous n'avons généralement pas besoin d'une connaissance partielle des paramètres d'étalonnage, mais nous devons connaître les modèles de mires d'étalonnage. Un inconvénient est que pour les géométries divergentes dans les chapitres 4 et 5, nous devons placer les mires d'étalonnage parallèlement au plan ou à la ligne du détecteur. La possibilité d'appliquer ces algorithmes dans des contextes réalistes nécessite des tests supplémentaires.

6.2 English version

In our work we expanded the knowledge about the self-calibration of radiological systems. In the two parts with results we presented two different approaches to self-calibrate, but both results show connections between the world of medical imaging and computer vision.

- In the chapter 2 we presented the application of the bundle adjustment method from computer vision to the cone-beam system C-arm. We showed that we don't have the uniqueness of the solution to this problem completed with the non-linear optimization and we can't self-calibrate better than up to a similarity transformation. This result is the same with computer vision world. Moreover, we showed what input data give us the solution differing by a similarity transformation. We also showed numerically that we can still perform the reconstruction in this case. The bad news is that we usually can't find this similarity transformation in the realistic settings when we obviously don't know true values of 3D coordinates of markers. Our experience with numerical simulations says that if we start our optimization procedure with the initial values close to true ones, we should obtain exactly a true solution. But the question of initial values that give us the exact true solution should be studied more deeply.

- In the chapters 3, 4 and 5 we presented a series of hybrid algorithms based on the same idea to combine DCC and markers in the calibration cages. Because of DCC, we could obtain the analytical procedures to calibrate using moments. Because of markers, we could reduce the application of DCC to the local information about markers which is common in computer vision. Thus, we could use DCC for truncated data. Note also that our procedures are based on DCC on distributions proved in this work, the generalisation of known DCC. Unlike previous algorithms, we usually don't require the partial knowledge of calibration parameters, but we need to know the patterns of calibration cages. One disadvantage is that for the divergent geometries in the chapters 4 and 5 we need to place the calibration cages as parallel as possible to the detector plane or line. The possibility of applying these algorithms in realistic settings requires further testing.

A Appendix

A.1 Marker detection on projected images with wavelets in 2D parallel geometry

The goal of this section is to show how we can detect Hölder singularities of the Radon transform of the indicator of a disk from Eq. (2.17) with the help of wavelet theory.

Theory. The basic theory of wavelets was developed around 1980s. A continuous wavelet transform was defined by Grossmann and Morlet [GM84]. Later Mallat introduced a canonical method to detect singularities from the continuous wavelet transform of signals. Let us use the main publications of Mallat [MZ92; MH92], his famous book [Mal09] in this section and start with the definition of a wavelet in 1D:

Definition A.1. *A wavelet or mother wavelet $\psi(s)$ is a function whose average is zero $\int_{-\infty}^{\infty} \psi(s)ds = 0$.*

Let us denote a dilation of $\psi(s)$ or a wavelet $\psi_u(s) = \frac{1}{u}\psi(\frac{s}{u})$, $u > 0$.

Definition A.2. *A continuous wavelet transform of a function $f(s) \in L^2(\mathbb{R})$ at the scale u is defined by a 1D convolution:*

$$Wf(s,u) = W_u f(s) = f \circledast \psi_u(s). \tag{A.1}$$

Sometimes we consider $u = 2^j$, $j \in \mathbb{Z}$, then $\psi_{2^j}(s) = \frac{1}{2^j}\psi(\frac{s}{2^j})$ and we can define in 1D:

Definition A.3. *A dyadic wavelet transform of $f(s)$ is a set of functions*

$$Wf = \{W_{2^j} f(s)\}_{j\in\mathbb{Z}}. \tag{A.2}$$

In practice, we also have a discretization for s, because we usually know the signal f in some discrete lattice. But in order to have a basis of wavelets on $L^2(\mathbb{R})$, we need to consider a proper discretization of $\psi_{2^j}(s)$ on s. Thus, the theory of discrete wavelet transform and construction of bases of wavelets appeared along with the theory of wavelets in other dimensions.

Wavelets in different dimensions were applied in different scientific fields, including tomography. In 1994, Berenstein and Walnut [BW94] used the theory of the continuous wavelet transform to receive inversion formulas for the Radon transform in even dimensions. Received inversion formulas were local. Thus, it was possible to recover a function f from the continuous wavelet transform of its Radon transform in a ball about $\vec{x}_0$ with the error $\varepsilon > 0$ just by knowing projections only on lines through a ball of some bigger radius about the same $\vec{x}_0$. Another local inversion formulas for the 2D Radon transform

were obtained later with the discrete wavelet transform: in 1994 by Olson and DeStefano [OD94], in 1995 by Delaney and Bresler [DB95], in 1996 by Berenstein and Walnut [BW96], in 1997 by Rashid-Farrokhi and colleagues [Ras+97], in 2000 by Bonnet and colleagues [Bon+00]. You can see that the history of the use of wavelets in tomography is extensive.

Now we want to show that we can detect markers with wavelets by applying the canonical algorithm of Mallat to detect irregularities of signals. In the Theorem 2.4 we showed that we have only 2 Hölder singularities for f_1. We can apply the canonical algorithm of Mallat to detect them.

From [Mal09] we know:

Theorem A.1. *Suppose that ψ is C^1 wavelet with a compact support, $\psi = \theta'$ for θ with non-zero average. Let $f \in L^1(\mathbb{R})$. If there exists a scale u_0 such that $|Wf(x,u)|$ has no local maximum for $x \in [a,b]$ and $u < u_0$, then f is uniformly Hölder 1 on $[a+\varepsilon, b-\varepsilon]$ for any $\varepsilon > 0$.*

From [MH92] we know:

Theorem A.2. *Let use a function $\theta(x)$ whose integral is non-zero and $\psi = \theta'$. We suppose that the wavelet $\psi(x)$ is continuously differentiable and has a compact support. Let $f \in L^2(\mathbb{R})$ and $\alpha \leq 1$. If $f(x)$ is Hölder α at x_0, then there exists a constant A such that for all x in the neighborhood of x_0 and any scale $u = 2^j$*

$$|W_{2^j} f(x)| \leq A((2^j)^\alpha + |x - x_0|^\alpha). \tag{A.3}$$

Thus, for $x = x_0$ we should have:

$$|W_{2^j} f(x)| \leq A(2^j)^\alpha. \tag{A.4}$$

According to the Theorem A.1, we can detect all singularities of f_1 by choosing local maximum points of $|W_{2^j} f_1|$ for the smallest for us resolution $j = 1$. Since the algorithm of Mallat computes just the dyadic wavelet transform, we don't have any information for scales $u < 2^1$, thus we can speak just about singularities at the resolution 1. If we detect no local maximum of $|W_{2^1} f_1|$ for some neighbourhood of x_0, then x_0 should be Hölder 1 at the resolution 1, otherwise we have a singularity (it could be Hölder α, $\alpha < 1$, or not Hölder at all). The good thing is that Mallat and Zhong [MZ92] built in their article a special wavelet ψ in such a way that they normalize the finest scale to be 2^1. It could be useful to know a regularity of each singular point and keep just Hölder $1/2$ points. According to the Theorem A.2, we can try to estimate the regularity α for such points with a linear regression from the dyadic wavelet transform, since

$$\log_2 |W_{2^j} f(x)| \leq \log_2 A + \alpha j. \tag{A.5}$$

Simulations. We constructed a special wavelet ψ and calculated the dyadic wavelet transform as described in [MZ92]. We implemented everything in Python.

Let us define a reconstructing wavelet $\chi(x)$ as any function whose Fourier transform satisfies $\sum_{j=-\infty}^{+\infty} \hat{\psi}(2^j\omega)\hat{\chi}(2^j\omega) = 1$. We use the definition of the Fourier transform as $\hat{f}(\omega) = Ff(\omega) = F(\omega) = \int_{-\infty}^{+\infty} f(x)e^{-i\omega x}dx$ and the inverse Fourier transform as $F^{-1}(F(\omega)) = f(x) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} F(\omega)e^{i\omega x}d\omega$. Then, the function $f(x)$ could be recovered with $f(x) = \sum_{j=-\infty}^{+\infty} W_{2^j}f * \chi_{2^j}(x)$. Let us define a smoothing function $\phi(x)$ to normalize the finest scale to be 2^1 as

$$|\hat{\phi}(\omega)|^2 = \sum_{j=1}^{+\infty} \hat{\psi}(2^j\omega)\hat{\chi}(2^j\omega). \tag{A.6}$$

Firstly, Mallat and Zhong managed to express the smoothing function in the Fourier space through a 2π-periodic differentiable function $H(\omega)$ such that $|H(\omega)|^2 + |H(\omega+\pi)|^2 \leq 1$, $|H(0)| = 1$. Then $\psi(x)$ was associated with $G(\omega)$:

$$\hat{\psi}(2\omega) = e^{-i\omega w}G(\omega)\hat{\phi}(\omega), \tag{A.7}$$

where $w = 1/2$ is the sampling shift. And $\chi(x)$ was associated with $K(\omega)$:

$$\hat{\chi}(2\omega) = e^{i\omega w}K(\omega)\hat{\phi}(\omega). \tag{A.8}$$

Then the authors calculated H, G and K from different conditions that our wavelet needs to fulfill:

$$H(\omega) = e^{i\omega/2}(\cos(\omega/2))^3,\ G(\omega) = 4ie^{i\omega/2}\sin(\omega/2),\ K(\omega) = \frac{1-|H(\omega)|^2}{G(\omega)}. \tag{A.9}$$

And finally:

$$\hat{\phi}(\omega) = \left(\frac{\sin(\omega/2)}{\omega/2}\right)^3,\ \hat{\psi}(\omega) = i\omega\left(\frac{\sin(\omega/4)}{\omega/4}\right)^4. \tag{A.10}$$

Thus, they received a quadratic spline with a compact support $\psi(x)$, it's the first-order derivative of the function $\theta(x)$ defined by $\hat{\theta}(\omega) = \left(\frac{\sin(\omega/4)}{\omega/4}\right)^4$. It's possible to see the constructed with the inverse Fourier transform wavelet $\psi(x)$ and its primitive $\theta(x)$ in the fig. A.1.

We started with the discrete signal d_n, $n \in \mathbb{Z}$. Let $S_1f(n) = d_n$. If $S_{2^j}f(x) = f \circledast \phi_{2^j}(x)$, then $W^d_{2^j}f = \{W_{2^j}f(n+w)\}_{n\in\mathbb{Z}}$, $S^d_{2^j}f = \{S_{2^j}f(n+w)\}_{n\in\mathbb{Z}}$.

For the implementation, the original discrete signal has a finite number N of values. Denote H_p, G_p the discrete filters obtained by putting $2p-1$ zeros between each of the coefficients of the filters H and G. It's possible to implement these filters with the table of impulse responses from [MZ92]. To compute $W^d_{2^{j+1}}f$ and $S^d_{2^{j+1}}f$ from the resolution level j to level $j+1$ we used:

$$W^d_{2^{j+1}}f = \frac{1}{\lambda_j}S^d_{2^j}f * G_j,\ S^d_{2^{j+1}}f = S^d_{2^j}f * H_j. \tag{A.11}$$

Values λ_j we again took from the corresponding table of the article [MZ92]. We detected correctly singularities with local maximum points of $|W_{2^1}f|$ for different functions. We present one example

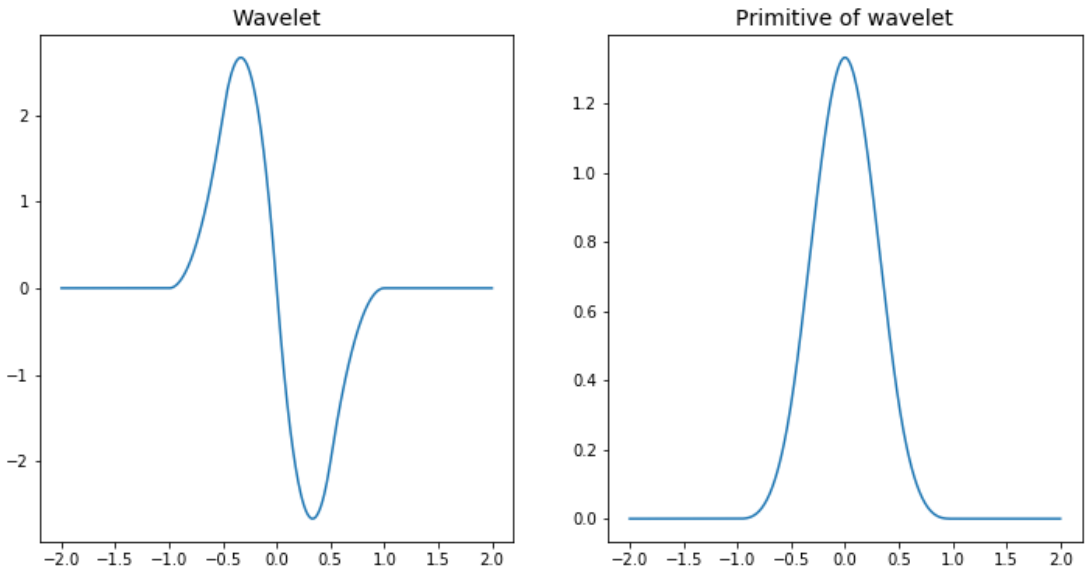

Figure A.1: The wavelet ψ and its primitive θ.

in the figure A.2. It was computed for the sum of two functions of the type f_1. The first one with $s_0 = -1.5$, $R = 0.1$ mimics a marker, the second one with $s_0 = 0$, $R = 5$ mimics a smooth function on $(-2,2)$, since its support is $(-5,5)$ and we played with data just for $(-2,2)$. We can check by eye and we checked in the code local maximum points corresponding to the marker, we found them correctly: $x_1 = -1.598$ (for the true one $x_1 = -1.6$) and $x_2 = -1.402$ (for the true one $x_2 = -1.4$) with 3 digits precision. Nevertheless, the method to estimate α didn't work perfectly, with the linear regression estimation for x_1 we received that α is equal to 0.146 and for x_2 α is equal to 0.402. Indeed, there is no contradiction, since in Eq. (A.5) we have just the inequality. More exact method to estimate α with other wavelets (foveal wavelets) can be found in [Mal03].

In this section we showed that we can easily detect points with $1/2$ Hölder regularities for the function f_1 with the help of the canonical algorithm from Mallat including dyadic wavelet transform. This can help to detect the centers of traces of round markers in projected images in 2D parallel geometry.

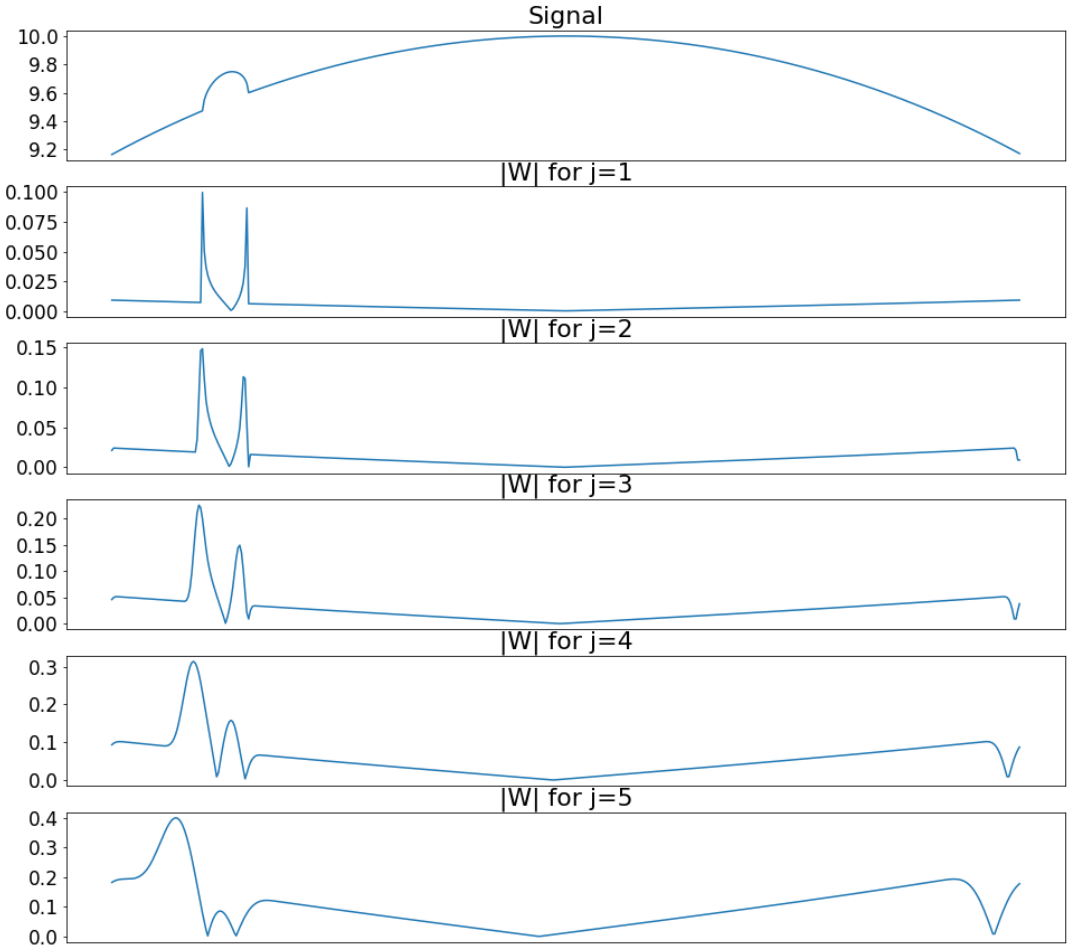

Figure A.2: Absolute values of the dyadic wavelet transform for few first levels ($j \leq 5$) for the test signal.

A.2 Sufficiency of DCC for the Radon transform on distributions for fixed projection angle

In this section we introduce some thoughts when we can have the sufficiency for our DCC for the Radon transform on distributions for fixed projection angle. Since our Theorem 3.7 is less strict than the Theorem 3.6 of Ramm and Katsevich, we can't prove the reverse implication in our case without additional assumptions. Moreover, Ramm and Katsevich showed in [RK96] the following fact used in the proof of their DCC:

Theorem A.3. *$\mathfrak{R}^* C^\infty(Z_2)$ is dense in $\mathscr{E}_2$ in the topology of $\mathscr{E}_2$.*

Let us remind that the subset $A \subset \mathscr{E}_2$ is dense in $\mathscr{E}_2$ if $\forall \varepsilon > 0, \forall \tilde{\phi} \in \mathscr{E}_2$ $B(\tilde{\phi}, \varepsilon) \cap A \neq \varnothing$, where $B(\tilde{\phi}, \varepsilon)$ is the ball in the topology of $\mathscr{E}_2$ with the center $\tilde{\phi}$ and the radius ε.

The analogue of the Theorem A.3 isn't true for arbitrary $\alpha \in [0, 2\pi]$ in our case, thus we have: $\mathfrak{R}^*_\alpha \mathscr{E}_1$ isn't dense in $\mathscr{E}_2$ in the topology of $\mathscr{E}_2$. If it's dense, then we need to have that $\forall \varepsilon > 0, \forall \tilde{\phi} \in \mathscr{E}_2$, $\forall$ compact $U \in \mathbb{R}^2$, $\forall j$ ($j = 0, 1, ...$) we can find $\phi \in \mathscr{E}_1$ such that the element $\mathfrak{R}^*_\alpha \phi$ is inside the ball of radius ε or

$$\sup_{\vec{x} \in U, |\boldsymbol{\beta}| \leq j} |\partial^{\boldsymbol{\beta}} \tilde{\phi}(\vec{x}) - \partial^{\boldsymbol{\beta}} (\mathfrak{R}^*_\alpha \phi(\vec{x}))| \leq \varepsilon. \tag{A.12}$$

We can show the counterexample: $\exists \varepsilon = 1, \exists \tilde{\phi} \in \mathscr{E}_2 : \tilde{\phi}(\vec{x}) = x_1 + x_2$, $\exists j = 1$: $\forall \phi \in \mathscr{E}_1$ we have

$$|\partial^2 \tilde{\phi} - \partial^2 (\mathfrak{R}^*_\alpha \phi)| \geq \varepsilon. \tag{A.13}$$

Indeed, $\frac{\partial \tilde{\phi}}{\partial x_1} = \frac{\partial \tilde{\phi}}{\partial x_2} = 1$. If $\alpha = 0$, then $\mathfrak{R}^*_\alpha \phi(\vec{x}) = \phi(x_1)$ and $\frac{\partial (\mathfrak{R}^*_\alpha \phi)}{\partial x_2} = 0 \Rightarrow |\frac{\partial \tilde{\phi}}{\partial x_2} - \frac{\partial (\mathfrak{R}^*_\alpha \phi)}{\partial x_2}| = 1 \geq \varepsilon$.

Thus, we can't reuse directly the proof of Ramm and Katsevich in our case, but we want to present under which hypotheses we can reduce the proof of sufficiency in our case to the proof of Ramm and Katsevich. We will reuse few logical steps from the proof of Ramm and Katsevich. Note that we also want to use the following theorem form [Bon01]:

Theorem A.4. *If $T \in \mathscr{E}'_N$, $g, \phi \in \mathscr{D}_N$, then*

$$(T * g, \phi) = (T, \tilde{g} * \phi), \tag{A.14}$$

where $\tilde{g}(\vec{x}) = g(-\vec{x})$.

Let us start with $p_\alpha \in \mathscr{E}'_1$ which is even (in the sense as described in the Theorem 3.7), $p_\alpha(s) = 0$ for $|s| > a$. For $\phi \in \mathscr{E}(Z_2)$, $\phi_\alpha \in \mathscr{E}_1$ with $\phi_\alpha(s) = \phi(\alpha, s)$ (p_α, ϕ_α) defines a function on α. We want to be able to define a distribution p:

$$(p(\alpha, s), \phi(\alpha, s)) = \int_0^{2\pi} (p_\alpha, \phi_\alpha) d\alpha. \tag{A.15}$$

We also suppose that $\forall k = 0,1,2,\dots$ the moment conditions (3.17) hold, then we want to understand when $\exists f \in \mathscr{E}_2'$ with $\mathrm{supp}(f) \subset B_a$ and $\mathfrak{R}f = p$.

The first stage is to define functions $p_{\alpha,\varepsilon}$ and p_ε, $p_{\alpha,\varepsilon}(s) = p_\varepsilon(\alpha,s)$, from p_α such that we can have all 3 properties of HLCC (Theorem 1.1) for p_ε and have $\mathfrak{R}f_\varepsilon = p_\varepsilon$ for some functions f_ε. Moreover, we need $p_\varepsilon \xrightarrow[\varepsilon\to 0]{} p$ in $\mathscr{E}'(Z_2)$. Then from the density in the Theorem A.3 we have the limit $\lim_{\varepsilon\to 0} f_\varepsilon$ in $\mathscr{E}_2'$. This limit will give us exactly f such that $\mathfrak{R}f = p$.

1. Let us consider $w_\varepsilon(s) = \varepsilon^{-1} w(s/\varepsilon)$, $w \in C_0^\infty([-1,1])$ is a positive even function with $\int_{-1}^1 w(s)ds = 1$. Since $p_\alpha \in \mathscr{E}_1'$, w_ε is a continuous function of compact support, then $\exists w_\varepsilon * p_\alpha$: $p_{\alpha,\varepsilon} = w_\varepsilon * p_\alpha \in \mathscr{D}_1$ from the Theorem 3.3, by the definition $p_{\alpha,\varepsilon}(s) = (p_\alpha(t), w_\varepsilon(s-t))$. Let us introduce **the first hypothesis**: suppose that (p_α, ϕ_α) defines a smooth function on $[0,2\pi]$ for each test function. Then $p_{\alpha,\varepsilon}$ is smooth on α. Then we can define a smooth and even $p_\varepsilon(\alpha,s) \in C_0^\infty(Z_{2,a+\varepsilon})$, $Z_{2,a+\varepsilon} = [0,2\pi] \times [-(a+\varepsilon), a+\varepsilon]$, where $p_{\alpha,\varepsilon}(s) = p_\varepsilon(\alpha,s)$. Here we also used the Theorem 3.2 about the support of the convolution.

   According to [Bon01], for such regularisation function w_ε $\lim_{\varepsilon\to 0} p_{\alpha,\varepsilon} = p_\alpha$. Now we need to show that the moment conditions from HLCC are fulfilled for p_ε. We want to calculate $(p_{\alpha,\varepsilon}, \phi)$. Let us start just with the smooth test function of compact support ψ, thus we can apply the Theorem A.4:

$$(p_{\alpha,\varepsilon}, \psi) = (w_\varepsilon * p_\alpha, \psi) = (p_\alpha, \tilde{w}_\varepsilon * \psi),$$

   where $\tilde{w}_\varepsilon(s) = w_\varepsilon(-s)$. We can calculate directly

$$\tilde{w}_\varepsilon * \psi(s) = \int_{\mathbb{R}} \psi(s')\tilde{w}_\varepsilon(s-s')ds' = \int_{\mathbb{R}} \psi(s') w_\varepsilon(s'-s)ds' = \frac{1}{\varepsilon}\int_{\mathbb{R}} \psi(s') w((s'-s)/\varepsilon)ds'$$
$$= \int_{\mathbb{R}} \psi(s+\varepsilon t)w(t)dt = \int_{-1}^1 \psi(s+\varepsilon t)w(t)dt \Rightarrow (p_{\alpha,\varepsilon}(s), \psi(s)) = (p_\alpha(s), \int_{-1}^1 \psi(s+\varepsilon t)w(t)dt).$$

   For $\phi \in \mathscr{E}_1$: since $p_{\alpha,\varepsilon}$ is a function with support in $[-(a+\varepsilon), a+\varepsilon]$, then we can consider

$$(p_{\alpha,\varepsilon}, \phi) = \int_{-(a+\varepsilon)}^{a+\varepsilon} p_{\alpha,\varepsilon}(s)\phi(s)ds = \int_{\mathbb{R}} p_{\alpha,\varepsilon}(s)\psi(s)ds = (p_{\alpha,\varepsilon}, \psi),$$

   where $\psi = \chi_1\phi$ is with support in $[-(a+3\varepsilon), a+3\varepsilon]$, $\chi_1(s) = 1$ in $[-(a+2\varepsilon), a+2\varepsilon]$ and χ_1 is defined on the remaining intervals to be a smooth function with compact support. This choice of 2ε is important in the further computations in Eq. (A.17). We have

$$(p_{\alpha,\varepsilon}(s), \phi(s)) = (p_{\alpha,\varepsilon}(s), \psi(s)) = (p_\alpha(s), \int_{-1}^1 \psi(s+\varepsilon t)w(t)dt) = (p_\alpha(s), \theta_{\psi,\varepsilon}(s)), \quad \text{(A.16)}$$

   where $\theta_{\psi,\varepsilon}(s) = \int_{-1}^1 \psi(s+\varepsilon t)w(t)dt$, we will write $\theta_{\phi,\varepsilon}(s)$ and $\theta_{\psi,\varepsilon}(s)$ to denote the same integral. Let us show the smoothness $\theta_{\psi,\varepsilon} \in \mathscr{E}_1$. Note that ψ has the support in $[-(a+3\varepsilon), a+3\varepsilon]$. We fix the neighbourhood of 0 for ε, suppose that $|\varepsilon| \le \varepsilon_M$. Then we know that $|t| \le 1$, then $\psi(s+\varepsilon t)$ is non-zero when $-(a+3\varepsilon) \le s+\varepsilon t \le a+3\varepsilon$ or $-(a+4\varepsilon_M) \le s \le a+4\varepsilon_M$, then

(a) $\psi(s+\varepsilon t)w(t)$ is continuous on s for all t,

(b) $\max_{s,t\in[-(a+4\varepsilon_M),a+4\varepsilon_M]\times[-1,1]}|\psi(s+\varepsilon t)| = M < +\infty$, since $\psi(s+\varepsilon t)$ is continuous on the compact set $[-(a+4\varepsilon_M),a+4\varepsilon_M]\times[-1,1]$. The function $Mw(t)$ is integrable on the compact $[-1,1]$, thus we have an integrable majorant $Mw(t)$ for each parameter s,

then the parameter-dependent integral $\theta_{\psi,\varepsilon}$ is a continuous function on $[-(a+4\varepsilon_M),a+4\varepsilon_M]$ by the theorem about the continuity of parameter-dependent Lebesgue integrals. Similarly:

(a) $\psi(s+\varepsilon t)w(t)$ is differentiable on s for all t,

(b) since each partial derivative of $\psi(s+\varepsilon t)$ on s is continuous on the compact set $[-(a+4\varepsilon_M),a+4\varepsilon_M]\times[-1,1]$, then it attains its maximum and minimum values, then for each partial derivative of $\psi(s+\varepsilon t)w(t)$ we have a majorant $Mw(t)$ (M is a constant, the maximum of the absolute values of the partial derivative of $\psi(s+\varepsilon t)$ for $s,t\in[-(a+4\varepsilon_M),a+4\varepsilon_M]\times[-1,1]$), it's integrable on the compact $[-1,1]$,

then we can differentiate the parameter-dependent integral $\theta_{\psi,\varepsilon}$ by the theorem about the differentiability of parameter-dependent Lebesgue integrals. We can apply the same theorem about the continuity and differentiability to show the existence and the continuity of higher derivatives.

Note that p_α has a compact support in $[-a,a]$. Then in order to compute $(p_\alpha,\theta_{\psi,\varepsilon})$ we need to know the function $\theta_{\psi,\varepsilon}$ only in the interval $[-(a+\varepsilon),a+\varepsilon]$. Indeed:

$$(p_\alpha,\theta_{\psi,\varepsilon}) = (p_\alpha,\theta_{\psi,\varepsilon}\chi_2+\theta_{\psi,\varepsilon}(1-\chi_2)) = (p_\alpha,\theta_{\psi,\varepsilon}\chi_2) + (p_\alpha,\theta_{\psi,\varepsilon}(1-\chi_2)) = (p_\alpha,\theta_{\psi,\varepsilon}\chi_2)+0,$$

where $\chi_2(s)=1$ in $[-(a+\varepsilon),a+\varepsilon]$ and smooth outside this interval. Moreover, the value $(p_\alpha,\theta_{\psi,\varepsilon}\chi_2)$ doesn't depend on the continuation of χ_2 outside the interval $[-(a+\varepsilon),a+\varepsilon]$ (see [Hel99]). Since for $-(a+\varepsilon)\le s\le a+\varepsilon$ we have $-(a+2\varepsilon)\le s+\varepsilon t\le a+2\varepsilon$, then

$$\psi(s+\varepsilon t)=\phi(s+\varepsilon t),\qquad \int_{-1}^{1}\psi(s+\varepsilon t)w(t)dt=\int_{-1}^{1}\phi(s+\varepsilon t)w(t)dt,\qquad s\in[-(a+\varepsilon),a+\varepsilon]. \tag{A.17}$$

Then we justified that we can write for the function $\phi\in\mathscr{E}_1$: $\phi(s)=s^k$ that $(p_{\alpha,\varepsilon}(s),s^k) = (p_\alpha(s),\int_{-1}^{1}(s+\varepsilon t)^k w(t)dt)$, since the value of $(p_\alpha(s),\int_{-1}^{1}(s+\varepsilon t)^k w(t)dt)$ doesn't depend on $\int_{-1}^{1}(s+\varepsilon t)^k w(t)dt$ outside the interval $[-(a+\varepsilon),a+\varepsilon]$. Thus, using the fact that w is even and each integral of the type $\int_{-1}^{1}t^{k-j}w(t)dt$ is zero for each odd $k-j$, we have

$$\int_{\mathbb{R}} p_\varepsilon(\alpha,s)s^k ds = \int_{\mathbb{R}} p_{\alpha,\varepsilon}(s)s^k ds = (p_{\alpha,\varepsilon}(s),s^k) = (p_\alpha(s),\int_{-1}^{1}(s+\varepsilon t)^k w(t)dt)$$
$$= (p_\alpha(s),\sum_{j=0,k-j \text{ even}}^{k} K_j s^j) = \sum_{j=0,k-j \text{ even}}^{k} K_j (p_\alpha(s),s^j) = \sum_{j=0,k-j \text{ even}}^{k} K_j \mathscr{P}_j(\alpha)$$
$$= \sum_{j=0,k-j \text{ even}}^{k} K_j \mathscr{P}_j(\alpha)(\cos^2\alpha+\sin^2\alpha)^{\frac{k-j}{2}} = \mathscr{Q}_k(\alpha),$$

where $\mathscr{Q}_k(\alpha)$ is a homogeneous polynomial of degree at most k in $\cos\alpha$, $\sin\alpha$. Here we used the moment conditions Eq. (3.17) that $(p_\alpha(s),s^j) = \mathscr{P}_j(\alpha)\ \forall\alpha$, $j = 0,1,2,\ldots$, $\mathscr{P}_j(\alpha)$ is a homogeneous polynomial of degree at most j in $\cos\alpha$, $\sin\alpha$.

We need to show $p_\varepsilon \xrightarrow[\varepsilon\to 0]{} p$ in $\mathscr{E}'(Z_2)$. Under the first hypothesis $\int_0^{2\pi}(p_\alpha,\phi_\alpha)d\alpha$ is properly defined. Let us introduce **the second hypothesis**: if we can change the limit and the integral for each α for each test function ϕ_α

$$\lim_{\varepsilon\to 0}\int_0^{2\pi}(p_{\alpha,\varepsilon},\phi_\alpha)d\alpha = \int_0^{2\pi}\lim_{\varepsilon\to 0}(p_{\alpha,\varepsilon},\phi_\alpha)d\alpha, \tag{A.18}$$

then

$$\lim_{\varepsilon\to 0}\int_0^{2\pi}(p_{\alpha,\varepsilon},\phi_\alpha)d\alpha = \int_0^{2\pi}\lim_{\varepsilon\to 0}(p_{\alpha,\varepsilon},\phi_\alpha)d\alpha = \int_0^{2\pi}(p_\alpha,\phi_\alpha)d\alpha = (p(\alpha,s),\phi(\alpha,s)) \Rightarrow$$
$$\lim_{\varepsilon\to 0}(p_\varepsilon(\alpha,s),\phi(\alpha,s)) = \lim_{\varepsilon\to 0}\int_0^{2\pi}(p_{\alpha,\varepsilon},\phi_\alpha)d\alpha = (p(\alpha,s),\phi(\alpha,s)).$$

2. Thus, we showed that we have properties $1-3$ of HLCC (Theorem 1.1) for each $p_\varepsilon(\alpha,s)$, so $\exists f_\varepsilon \in C_0^\infty(B_{a+\varepsilon})$: $\mathfrak{R}f_\varepsilon = p_\varepsilon$.

3. As in the proof of Ramm and Katsevich, from the previous step and $p_\varepsilon \xrightarrow[\varepsilon\to 0]{} p$ in $\mathscr{E}'(Z_2)$ we have $\mathfrak{R}f_\varepsilon \xrightarrow[\varepsilon\to 0]{} p$ in $\mathscr{E}'(Z_2)$.

4. Since $\mathfrak{R}^*\mathscr{E}(Z_2)$ is dense in $\mathscr{E}_2$, then we can apply the Theorem A.3. Let us detail this application mentioned briefly in the book of Ramm and Katsevich. For $\phi \in \mathscr{E}(Z_2)$ $\langle f_\varepsilon,\mathfrak{R}^*\phi\rangle = (\mathfrak{R}f_\varepsilon,\phi) \xrightarrow[\varepsilon\to 0]{} (p,\phi)$, $\langle f_\varepsilon,\mathfrak{R}^*\phi\rangle = \int_{\mathbb{R}^2} f_\varepsilon(\vec{x})\mathfrak{R}^*\phi(\vec{x})d\vec{x} = \int_{B_{a+\varepsilon}} f_\varepsilon(\vec{x})\mathfrak{R}^*\phi(\vec{x})d\vec{x}$. For this compact $B_{a+\varepsilon}$ $\forall\mu \in \mathscr{E}_2$ $\exists\phi \in \mathscr{E}(Z_2)$: $\sup_{B_{a+\varepsilon}}|\mu-\mathfrak{R}^*\phi| \le \varepsilon$ or $-\varepsilon+\mathfrak{R}^*\phi \le \mu \le \varepsilon+\mathfrak{R}^*\phi$. Then for each ε

$$\int_{\mathbb{R}^2} f_\varepsilon(\vec{x})(-\varepsilon+\mathfrak{R}^*\phi(\vec{x}))d\vec{x} \le \int_{\mathbb{R}^2} f_\varepsilon(\vec{x})\mu(\vec{x})d\vec{x} \le \int_{\mathbb{R}^2} f_\varepsilon(\vec{x})(\varepsilon+\mathfrak{R}^*\phi(\vec{x}))d\vec{x}.$$

The left and the right sides in the limit go to (p,ϕ), thus $\exists\lim_{\varepsilon\to 0} f_\varepsilon$ in $\mathscr{E}_2'$.

5. Denote $f = \lim_{\varepsilon \to 0} f_\varepsilon$ in $\mathscr{E}_2'$, then $\operatorname{supp}(f) \subset B_a$ and $\mathfrak{R}f = p$ for it.

In this section we showed under which hypotheses we have the reduction of the sufficiency for our definition of the Radon transform on distributions for fixed α to the proof of sufficiency of Ramm and Katsevich presented earlier for their definition.

A.3 Justification of the definition of the fan-beam transform on distributions with sources on a line

In this section we explain why our definition Eq. (4.3) of the fan-beam transform on distributions with fixed source position and sources on a line is properly formulated. Inside this definition we use the specific adjoint operator defined in Eq. (4.1). To justify our definition, we need to show that Eq. (4.3) defines a distribution or we need to show that $\mathfrak{D}_\lambda f$ for $f \in \mathscr{E}'(X_2)$ is a bounded linear functional. Note that we discussed before that we replace everywhere the check of continuity of the linear functional with the check of boundedness.

The linearity of such functional is obvious. We only need to show that $\mathfrak{D}_\lambda f$ is bounded. For that, we will prove that any bounded subset $\Phi \subset \mathscr{E}_1$ becomes a bounded subset $\mathfrak{D}^*_\lambda(\Phi) \subset \mathscr{E}(X_2)$. In this case from the boundedness of f we will have that $f(\mathfrak{D}^*_\lambda(\Phi))$ is bounded. Then we will have that $\mathfrak{D}_\lambda f(\Phi)$ is bounded for any bounded Φ using Eq. (4.3) or $\mathfrak{D}_\lambda f$ is bounded.

For any bounded $\Phi \subset \mathscr{E}_1$ each $\tilde{P}^1_{j,\beta}(\Phi)$ from Eq. (3.3) is bounded with $M_{j,\beta}$, where $\cup_j K^1_j = \mathbb{R}$ with compacts K^1_j, K^1_j is in the interior of the set K^1_{j+1}, $M_{j,\beta}$ doesn't depend on $\phi \in \Phi$. If we consider $K^2_j = [d^j_1, d^j_2] \times [-b^j, b^j]$ with $\cup_j K^2_j = X_2$, $0 < d^j_1, d^j_2 < D$, $b^j > 0$, $b^j \in \mathbb{R}$ with $d^{j+1}_1 \le d^j_1$, $d^j_2 \le d^{j+1}_2$, $b^j \le b^{j+1}$, then we need to prove that for each function $\psi \in \mathfrak{D}^*_\lambda(\Phi)$, $\psi(\vec{x}) = \frac{1}{D-x_1}\phi\left(\frac{x_2 D - x_1\lambda}{D-x_1}\right)$, we have the boundedness of $\tilde{P}^2_{j,\boldsymbol{\beta}}(\psi) = \sup_{\vec{x}\in K^2_j} |\partial^{\boldsymbol{\beta}}\psi(\vec{x})| \le \tilde{M}_{j,\boldsymbol{\beta}}$, where $\tilde{M}_{j,\boldsymbol{\beta}}$ doesn't depend on ψ.

- When $\boldsymbol{\beta} = (0,0)$:

$$\tilde{P}^2_{j,(0,0)}(\psi) = \sup_{\vec{x}\in K^2_j} |\psi(\vec{x})| = \sup_{\vec{x}\in K^2_j} \left|\phi\left(\frac{x_2 D - x_1\lambda}{D-x_1}\right)\frac{1}{D-x_1}\right|$$
$$\le \sup_{\vec{x}\in K^2_j} \left|\phi\left(\frac{x_2 D - x_1\lambda}{D-x_1}\right)\right| \frac{1}{D-D_2} \le \sup_{y\in K^1_{j'}} |\phi(y)| \frac{1}{D-D_2} \le M_{j',0}\frac{1}{D-D_2},$$

  where

$$\left|\frac{x_2 D - x_1\lambda}{D-x_1}\right| = \left|\lambda + D\frac{x_2-\lambda}{D-x_1}\right| \le |\lambda| + D\left|\frac{x_2-\lambda}{D-x_1}\right| \le |\lambda| + D\frac{|x_2|+|\lambda|}{D-D_2} \le |\lambda| + D\frac{b^j+|\lambda|}{D-D_2},$$

  thus we can find j' and $K^1_{j'}$ such that $\{y : |y| \le |\lambda| + D\frac{b^j+|\lambda|}{D-D_2}\} \subset K^1_{j'}$.

  Note that for each $\beta \ge 1$, for each K^2_j we can find $K^1_{j'}$ such that

$$\sup_{\vec{x}\in K^2_j} \left|\phi^{(\beta)}\left(\frac{x_2 D - x_1\lambda}{D-x_1}\right)\frac{(x_2-\lambda)^{p_1}}{(D-x_1)^{p_2}}\right| \le \sup_{y\in K^1_{j'}} |\phi^{(\beta)}(y)| \frac{(b^j+|\lambda|)^{p_1}}{(D-D_2)^{p_2}}$$
$$\le M_{j',\beta}\frac{(b^j+|\lambda|)^{p_1}}{(D-D_2)^{p_2}}, \qquad p_1, p_2 \ge 0. \quad \text{(A.19)}$$

- The case $\boldsymbol{\beta} = (0,\beta)$. When $\beta = 1$:

$$\partial_2^1 \psi(\vec{x}) = \phi'\left(\frac{x_2 D - x_1\lambda}{D-x_1}\right)\frac{D}{D-x_1}\frac{1}{D-x_1}.$$

For $\beta > 1$ by induction:

$$\partial_2^\beta \psi(\vec{x}) = \phi^{(\beta)}\left(\frac{x_2 D - x_1\lambda}{D-x_1}\right)\frac{D^\beta}{(D-x_1)^{\beta+1}}.$$

Since $\tilde{P}^2_{j,(0,\beta)}(\psi)$ is of the type Eq. (A.19), thus it's bounded with some constant which doesn't depend on ψ.

- The case $\boldsymbol{\beta} = (\beta,0)$. When $\beta = 1$:

$$\begin{aligned}\partial_1^1 \psi(\vec{x}) &= \phi'\left(\lambda + D\frac{x_2-\lambda}{D-x_1}\right)\frac{D(x_2-\lambda)}{(D-x_1)^2}\frac{1}{D-x_1} + \phi\left(\lambda + D\frac{x_2-\lambda}{D-x_1}\right)\frac{1}{(D-x_1)^2} \\ &= \phi'\left(\lambda + D\frac{x_2-\lambda}{D-x_1}\right)\frac{D(x_2-\lambda)}{(D-x_1)^3} + \phi\left(\lambda + D\frac{x_2-\lambda}{D-x_1}\right)\frac{1}{(D-x_1)^2}.\end{aligned}$$

When $\beta = 2$:

$$\begin{aligned}\partial_1^2 \psi(\vec{x}) &= \phi''\left(\lambda + D\frac{x_2-\lambda}{D-x_1}\right)\frac{D(x_2-\lambda)}{(D-x_1)^2}\frac{D(x_2-\lambda)}{(D-x_1)^3} + \phi'\left(\lambda + D\frac{x_2-\lambda}{D-x_1}\right)\cdot\frac{3D(x_2-\lambda)}{(D-x_1)^4} \\ &\quad + \phi'\left(\lambda + D\frac{x_2-\lambda}{D-x_1}\right)\frac{D(x_2-\lambda)}{(D-x_1)^2}\frac{1}{(D-x_1)^2} + \phi\left(\lambda + D\frac{x_2-\lambda}{D-x_1}\right)\frac{2}{(D-x_1)^3} \\ &= \phi''\left(\lambda + D\frac{x_2-\lambda}{D-x_1}\right)\cdot\frac{D^2(x_2-\lambda)^2}{(D-x_1)^{2\cdot 2+1}} + c_1^2\phi'\left(\lambda + D\frac{x_2-\lambda}{D-x_1}\right)\frac{D^1(x_2-\lambda)^1}{(D-x_1)^{2+1+1}} \\ &\quad + \phi\left(\lambda + D\frac{x_2-\lambda}{D-x_1}\right)\frac{2!}{(D-x_1)^{2+1}}.\end{aligned}$$

We can use this equation as the base of induction with $c_1^2 = 4$, then for $\beta \geq 2$ we have:

$$\begin{aligned}\partial_1^\beta \psi(\vec{x}) &= \phi^{(\beta)}\left(\lambda + D\frac{x_2-\lambda}{D-x_1}\right)\frac{D^\beta(x_2-\lambda)^\beta}{(D-x_1)^{2\beta+1}} + \sum_{i=1}^{\beta-1} c_i^\beta \phi^{(i)}\left(\lambda + D\frac{x_2-\lambda}{D-x_1}\right) \\ &\quad \cdot\frac{D^i(x_2-\lambda)^i}{(D-x_1)^{\beta+1+i}} + \phi\left(\lambda + D\frac{x_2-\lambda}{D-x_1}\right)\frac{\beta!}{(D-x_1)^{\beta+1}}, \quad \text{(A.20)}\end{aligned}$$

where $c_1^{\beta+1} = c_1^\beta(\beta+1+1) + \beta!$, $c_\beta^{\beta+1} = 2\beta + 1 + c_{\beta-1}^\beta$ and $c_i^{\beta+1} = c_i^\beta(\beta+1+i) + c_{i-1}^\beta$ for $i = 2,...,\beta-1$. Indeed, from the differentiation of Eq. (A.20), we have the expression of the same type:

$$\partial_1^{\beta+1}\psi(\vec{x}) = \phi^{(\beta+1)}\left(\lambda + D\frac{x_2-\lambda}{D-x_1}\right)\frac{D(x_2-\lambda)}{(D-x_1)^2}\frac{D^\beta(x_2-\lambda)^\beta}{(D-x_1)^{2\beta+1}}$$
$$+\phi^{(\beta)}\left(\lambda + D\frac{x_2-\lambda}{D-x_1}\right)\frac{D^\beta(x_2-\lambda)^\beta(2\beta+1)}{(D-x_1)^{2\beta+2}}$$
$$+\sum_{i=1}^{\beta-1}\left[\phi^{(i+1)}\left(\lambda + D\frac{x_2-\lambda}{D-x_1}\right)\frac{D(x_2-\lambda)}{(D-x_1)^2}c_i^\beta\frac{D^i(x_2-\lambda)^i}{(D-x_1)^{\beta+1+i}}\right.$$
$$\left.+\phi^{(i)}\left(\lambda + D\frac{x_2-\lambda}{D-x_1}\right)c_i^\beta\frac{D^i(x_2-\lambda)^i(\beta+1+i)}{(D-x_1)^{\beta+2+i}}\right] + \phi'\left(\lambda + D\frac{x_2-\lambda}{D-x_1}\right)\frac{D(x_2-\lambda)}{(D-x_1)^2}\frac{\beta!}{(D-x_1)^{\beta+1}}$$
$$+\phi\left(\lambda + D\frac{x_2-\lambda}{D-x_1}\right)\frac{(\beta+1)!}{(D-x_1)^{\beta+2}} = \phi^{(\beta+1)}\left(\lambda + D\frac{x_2-\lambda}{D-x_1}\right)\frac{D^{\beta+1}(x_2-\lambda)^{\beta+1}}{(D-x_1)^{2(\beta+1)+1}}$$
$$+\sum_{i=1}^{\beta}c_i^{\beta+1}\phi^{(i)}\left(\lambda + D\frac{x_2-\lambda}{D-x_1}\right)\frac{D^i(x_2-\lambda)^i}{(D-x_1)^{(\beta+1)+1+i}} + \phi\left(\lambda + D\frac{x_2-\lambda}{D-x_1}\right)\frac{(\beta+1)!}{(D-x_1)^{(\beta+1)+1}}.$$

Thus, $\tilde{P}^2_{j,(\beta,0)}(\psi)$ is bounded with some constant which doesn't depend on ψ, since $\partial_1^\beta\psi(\vec{x})$ is the finite sum of expressions of the type Eq. (A.19).

- To handle mixed derivatives of ψ we need to apply two previous steps. Since each $\tilde{P}^2_{j,(\beta_1,\beta_2)}(\psi)$ is the sum expressions of the type Eq. (A.19), then it's bounded with some constant which doesn't depend on ψ.

Thus, we properly defined the fan-beam transform for fixed λ on distributions of compact support.

A.4 Justification of the definition of the cone-beam transform on distributions with sources on a line

In this section we explain why our definition of the cone-beam transform on distributions with fixed source position and sources on a line is properly formulated. To justify our definition, we need to show that the duality Eq. (4.3) defines a distribution or we need to show that $\mathfrak{D}_\lambda f$ for $f \in \mathscr{E}'(X_3)$ is a bounded linear functional.

The linearity of such functional is obvious. We only need to show that $\mathfrak{D}_\lambda f$ is bounded. For each bounded $\Phi \subset \mathscr{E}_2$ we have each $\tilde{P}^1_{j,\boldsymbol{\beta}}(\Phi)$ from Eq. (3.3) is bounded with $M_{j,\boldsymbol{\beta}}$, where $\cup_j K^1_j = \mathbb{R}^2$ with compacts K^1_j, K^1_j is in the interior of the set K^1_{j+1}. Let us consider $K^2_j = [d^j_1, d^j_2] \times [-a^j, a^j] \times [-b^j, b^j]$ with $\cup_j K^2_j = X_3$, $0 < d^j_1, d^j_2 < D$, $a^j, b^j > 0$, $a^j, b^j \in \mathbb{R}$ with $d^{j+1}_1 \le d^j_1$, $d^j_2 \le d^{j+1}_2$, $a^j \le a^{j+1}$, $b^j \le b^{j+1}$, then we need to prove for each function $\psi \in \mathfrak{D}^*_\lambda(\Phi)$: $\psi(\vec{x}) = \frac{D}{(D-x_1)^2}\phi\left(\frac{x_2 D - x_1\lambda}{D-x_1}, \frac{Dx_3}{D-x_1}\right)$ the boundedness of $\tilde{P}^2_{j,\boldsymbol{\beta}}(\psi) = \sup_{\vec{x}\in K^2_j} |\partial^{\boldsymbol{\beta}}\psi(\vec{x})| \le \tilde{M}_{j,\boldsymbol{\beta}}$, where $\tilde{M}_{j,\boldsymbol{\beta}}$ doesn't depend on ψ.

- When $\boldsymbol{\beta} = (0,0,0)$:

$$\tilde{P}^2_{j,(0,0,0)}(\psi) = \sup_{\vec{x}\in K^2_j} |\psi(\vec{x})| = \sup_{\vec{x}\in K^2_j} \left|\phi\left(\frac{x_2 D - x_1\lambda}{D-x_1}, \frac{Dx_3}{D-x_1}\right)\frac{D}{(D-x_1)^2}\right|$$

$$\le \sup_{\vec{x}\in K^2_j} \left|\phi\left(\frac{x_2 D - x_1\lambda}{D-x_1}, \frac{Dx_3}{D-x_1}\right)\right| \frac{D}{(D-D_2)^2} \le \sup_{(y_1,y_2)\in K^1_{j'}} |\phi(y_1,y_2)| \frac{D}{(D-D_2)^2} \le M_{j',(0,0)} \frac{D}{(D-D_2)^2},$$

where

$$\left|\frac{x_2 D - x_1\lambda}{D-x_1}\right| = \left|\lambda + D\frac{x_2-\lambda}{D-x_1}\right| \le |\lambda| + D\frac{a^j + |\lambda|}{D-D_2},$$

$$\left|\frac{Dx_3}{D-x_1}\right| \le \frac{Db^j}{D-D_2},$$

thus we can find j' and $K^1_{j'}$ such that $\{(y_1,y_2) : |y_1| \le |\lambda| + D\frac{a^j+|\lambda|}{D-D_2}, |y_2| \le \frac{Db^j}{D-D_2}\} \subset K^1_{j'}$.

Note that for each $|\boldsymbol{\beta}| \ge 1$, for each K^2_j we can find $K^1_{j'}$ such that

$$\sup_{\vec{x}\in K^2_j} \left|\partial^{\boldsymbol{\beta}}\phi\left(\frac{x_2 D - x_1\lambda}{D-x_1}, \frac{Dx_3}{D-x_1}\right)\frac{(x_2-\lambda)^{p_1} x_3^{p_2}}{(D-x_1)^{p_3}}\right| \le \sup_{(y_1,y_2)\in K^1_{j'}} |\partial^{\boldsymbol{\beta}}\phi(y_1,y_2)| \frac{(a^j+|\lambda|)^{p_1}(b^j)^{p_2}}{(D-D_2)^{p_3}}$$

$$\le M_{j',\boldsymbol{\beta}} \frac{(a^j+|\lambda|)^{p_1}(b^j)^{p_2}}{(D-D_2)^{p_3}}, \qquad p_1,p_2,p_3 \ge 0. \quad \text{(A.21)}$$

- The case $\boldsymbol{\beta} = (0,\beta,0)$. When $\beta = 1$:

$$\partial^1_2 \psi(\vec{x}) = \partial^1_1 \phi\left(\frac{x_2 D - x_1\lambda}{D-x_1}, \frac{Dx_3}{D-x_1}\right)\frac{D}{D-x_1}\frac{D}{(D-x_1)^2}.$$

For $\beta > 1$ by induction:

$$\partial_2^\beta \psi(\vec{x}) = \partial_1^\beta \phi \left(\frac{x_2 D - x_1 \lambda}{D - x_1}, \frac{D x_3}{D - x_1} \right) \frac{D^{\beta+1}}{(D-x_1)^{\beta+2}}.$$

Since $\tilde{P}^2_{j,(0,\beta,0)}(\psi)$ is of the type Eq. (A.21), thus it's bounded with some constant which doesn't depend on ψ.

- The case $\boldsymbol{\beta} = (0,0,\beta)$. When $\beta = 1$:

$$\partial_3^1 \psi(\vec{x}) = \partial_2^1 \phi \left(\frac{x_2 D - x_1 \lambda}{D - x_1}, \frac{D x_3}{D - x_1} \right) \frac{D}{D-x_1} \frac{D}{(D-x_1)^2}.$$

For $\beta > 1$ by induction:

$$\partial_3^\beta \psi(\vec{x}) = \partial_2^\beta \phi \left(\frac{x_2 D - x_1 \lambda}{D - x_1}, \frac{D x_3}{D - x_1} \right) \frac{D^{\beta+1}}{(D-x_1)^{\beta+2}}.$$

Since $\tilde{P}^2_{j,(0,0,\beta)}(\psi)$ is of the type Eq. (A.21), thus it's bounded with some constant which doesn't depend on ψ.

- The case $\boldsymbol{\beta} = (\beta,0,0)$. When $\beta = 1$:

$$\begin{aligned}
&\partial_1^1 \psi(\vec{x}) = \partial_1^1 \phi \left(\lambda + D\frac{x_2 - \lambda}{D - x_1}, \frac{D x_3}{D - x_1} \right) \frac{D(x_2 - \lambda)}{(D-x_1)^2} \frac{D}{(D-x_1)^2} \\
&+ \partial_2^1 \phi \left(\lambda + D\frac{x_2 - \lambda}{D - x_1}, \frac{D x_3}{D - x_1} \right) \frac{D x_3}{(D-x_1)^2} \frac{D}{(D-x_1)^2} + \phi \left(\lambda + D\frac{x_2 - \lambda}{D - x_1}, \frac{D x_3}{D - x_1} \right) \frac{2D}{(D-x_1)^3}.
\end{aligned}$$

For the first derivative we have the sum of expressions of the type Eq. (A.21), thus we have boundedness of $\tilde{P}^2_{j,(1,0,0)}(\psi)$. Let us show that all higher derivatives will be finite sums of expressions of the same type Eq. (A.21). For that, we can show that the derivative of Eq. (A.21) is the finite sum of expressions of the type Eq. (A.21):

$$\begin{aligned}
&\partial_1^1 \left(\partial_1^{\beta_1} \partial_2^{\beta_2} \phi \left(\lambda + D\frac{x_2 - \lambda}{D - x_1}, \frac{D x_3}{D - x_1} \right) \frac{(x_2 - \lambda)^{p_1} x_3^{p_2}}{(D-x_1)^{p_3}} \right) \\
&\qquad = \partial_1^{\beta_1+1} \partial_2^{\beta_2} \phi \left(\lambda + D\frac{x_2 - \lambda}{D - x_1}, \frac{D x_3}{D - x_1} \right) \frac{D(x_2 - \lambda)}{(D-x_1)^2} \frac{(x_2 - \lambda)^{p_1} x_3^{p_2}}{(D-x_1)^{p_3}} \\
&\qquad + \partial_1^{\beta_1} \partial_2^{\beta_2+1} \phi \left(\lambda + D\frac{x_2 - \lambda}{D - x_1}, \frac{D x_3}{D - x_1} \right) \frac{D x_3}{(D-x_1)^2} \frac{(x_2 - \lambda)^{p_1} x_3^{p_2}}{(D-x_1)^{p_3}} \\
&\qquad\qquad + \partial_1^{\beta_1} \partial_2^{\beta_2} \phi \left(\lambda + D\frac{x_2 - \lambda}{D - x_1}, \frac{D x_3}{D - x_1} \right) \frac{p_3 (x_2 - \lambda)^{p_1} x_3^{p_2}}{(D-x_1)^{p_3+1}}.
\end{aligned}$$

- To handle mixed derivatives of ψ we need to apply three previous steps. Since each $\tilde{P}^2_{j,(\beta_1,\beta_2,\beta_3)}(\psi)$ is the sum expressions of the type Eq. (A.21), then it's bounded with some constant which doesn't depend on ψ.

As for the fan-beam case, we will have from the boundedness of f that $f(\mathfrak{D}_\lambda^*(\Phi))$ is bounded for bounded $\mathfrak{D}_\lambda^*(\Phi)$. Then $\mathfrak{D}_\lambda f$ is bounded from the definition.

Thus, we properly defined the cone-beam transform for fixed λ on distributions of compact support with sources on a line.

A.5 Justification of the definition of the cone-beam transform on distributions with sources on a plane

In this section we explain why our definition of the cone-beam transform on distributions with fixed source position and sources on a plane parallel to the detector is properly formulated. To justify our definition, we need to show that the duality Eq. (4.3) defines a distribution or we need to show that $\mathfrak{D}_{\lambda_1,\lambda_2} f$ for $f \in \mathscr{E}'(Y_3)$ is a bounded linear functional.

The linearity of such functional is obvious. We need to show that $\mathfrak{D}_{\lambda_1,\lambda_2} f$ is bounded. For each bounded $\Phi \subset \mathscr{E}_2$ we have each $\tilde{P}^1_{j,\boldsymbol{\beta}}(\Phi)$ from Eq. (3.3) is bounded with $M_{j,\boldsymbol{\beta}}$, where $\cup_j K^1_j = \mathbb{R}^2$ with compacts K^1_j, K^1_j is in the interior of the set K^1_{j+1}. Let us consider $K^2_j = [-a^j, a^j] \times [-b^j, b^j] \times [d^j_1, d^j_2]$ with $\cup_j K^2_j = Y_3$, $0 < d^j_1, d^j_2 < D$, $a^j, b^j > 0$, $a^j, b^j \in \mathbb{R}$ with $d^{j+1}_1 \le d^j_1$, $d^j_2 \le d^{j+1}_2$, $a^j \le a^{j+1}$, $b^j \le b^{j+1}$, then we need to prove for each function $\psi \in \mathfrak{D}^*_{\lambda_1,\lambda_2}(\Phi)$: $\psi(\vec{x}) = \frac{D}{x_3^2}\phi\left(\frac{Dx_1-\lambda_1(D-x_3)}{x_3}, \frac{Dx_2-\lambda_2(D-x_3)}{x_3}\right)$ the boundedness of $\tilde{P}^2_{j,\boldsymbol{\beta}}(\psi) = \sup_{\vec{x}\in K^2_j} |\partial^{\boldsymbol{\beta}}\psi(\vec{x})| \le \tilde{M}_{j,\boldsymbol{\beta}}$, where $\tilde{M}_{j,\boldsymbol{\beta}}$ doesn't depend on ψ.

- When $\boldsymbol{\beta} = (0,0,0)$:

$$\begin{aligned}\tilde{P}^2_{j,(0,0,0)}(\psi) = \sup_{\vec{x}\in K^2_j} |\psi(\vec{x})| &= \sup_{\vec{x}\in K^2_j} \left|\phi\left(\frac{Dx_1-\lambda_1(D-x_3)}{x_3}, \frac{Dx_2-\lambda_2(D-x_3)}{x_3}\right)\frac{D}{x_3^2}\right| \\ &\le \sup_{\vec{x}\in K^2_j} \left|\phi\left(\frac{Dx_1-\lambda_1(D-x_3)}{x_3}, \frac{Dx_2-\lambda_2(D-x_3)}{x_3}\right)\right| \frac{D}{D_1^2} \\ &\le \sup_{(y_1,y_2)\in K^1_{j'}} |\phi(y_1,y_2)| \frac{D}{D_1^2} \le M_{j',(0,0)} \frac{D}{D_1^2},\end{aligned}$$

where

$$\left|\frac{Dx_1-\lambda_1(D-x_3)}{x_3}\right| = \left|D\frac{x_1-\lambda_1}{x_3}+\lambda_1\right| \le D\frac{a^j+|\lambda_1|}{D_1}+|\lambda_1|,$$
$$\left|\frac{Dx_2-\lambda_2(D-x_3)}{x_3}\right| \le D\frac{b^j+|\lambda_2|}{D_1}+|\lambda_2|,$$

thus we can find j' and $K^1_{j'}$ such that $\{(y_1,y_2) : |y_1| \le D\frac{a^j+|\lambda_1|}{D_1}+|\lambda_1|, |y_2| \le D\frac{b^j+|\lambda_2|}{D_1}+|\lambda_2|\} \subset K^1_{j'}$. Note that for each $|\boldsymbol{\beta}| \ge 1$, for each K^2_j we can find $K^1_{j'}$ such that

$$\begin{aligned}&\sup_{\vec{x}\in K^2_j} \left|\partial^{\boldsymbol{\beta}}\phi\left(\frac{Dx_1-\lambda_1(D-x_3)}{x_3}, \frac{Dx_2-\lambda_2(D-x_3)}{x_3}\right)\frac{(x_1-\lambda_1)^{p_1}(x_2-\lambda_2)^{p_2}}{x_3^{p_3}}\right| \\ &\le \sup_{(y_1,y_2)\in K^1_{j'}} |\partial^{\boldsymbol{\beta}}\phi(y_1,y_2)| \frac{(a^j+|\lambda|)^{p_1}(b^j+|\lambda_2|)^{p_2}}{D_1^{p_3}} \\ &\le M_{j',\boldsymbol{\beta}} \frac{(a^j+|\lambda_1|)^{p_1}(b^j+|\lambda_2|)^{p_2}}{D_1^{p_3}}, \qquad p_1,p_2,p_3 \ge 0. \quad \text{(A.22)}\end{aligned}$$

- The case $\boldsymbol{\beta} = (\beta, 0, 0)$. When $\beta = 1$:

$$\partial_1^1 \psi(\vec{x}) = \partial_1^1 \phi \left(\frac{Dx_1 - \lambda_1(D - x_3)}{x_3}, \frac{Dx_2 - \lambda_2(D - x_3)}{x_3} \right) \frac{D}{x_3} \frac{D}{x_3^2}.$$

  For $\beta > 1$ by induction:

$$\partial_1^\beta \psi(\vec{x}) = \partial_1^\beta \phi \left(\frac{Dx_1 - \lambda_1(D - x_3)}{x_3}, \frac{Dx_2 - \lambda_2(D - x_3)}{x_3} \right) \frac{D^{\beta+1}}{x_3^{\beta+2}}.$$

  Since $\tilde{P}^2_{j,(\beta,0,0)}(\psi)$ is of the type Eq. (A.22), thus it's bounded with some constant which doesn't depend on ψ.

- The same is true for $\boldsymbol{\beta} = (0, \beta, 0)$. When $\beta = 1$:

$$\partial_2^1 \psi(\vec{x}) = \partial_2^1 \phi \left(\frac{Dx_1 - \lambda_1(D - x_3)}{x_3}, \frac{Dx_2 - \lambda_2(D - x_3)}{x_3} \right) \frac{D}{x_3} \frac{D}{x_3^2}.$$

  And for $\beta > 1$ by induction:

$$\partial_2^\beta \psi(\vec{x}) = \partial_2^\beta \phi \left(\frac{Dx_1 - \lambda_1(D - x_3)}{x_3}, \frac{Dx_2 - \lambda_2(D - x_3)}{x_3} \right) \frac{D^{\beta+1}}{x_3^{\beta+2}}.$$

- The case $\boldsymbol{\beta} = (0, 0, \beta)$. When $\beta = 1$:

$$\begin{aligned}
\partial_3^1 \psi(\vec{x}) = {} & \partial_1^1 \phi \left(\lambda_1 + D\frac{x_1 - \lambda_1}{x_3}, \lambda_2 + D\frac{x_2 - \lambda_2}{x_3} \right) \frac{-D(x_1 - \lambda_1)}{x_3^2} \frac{D}{x_3^2} \\
& + \partial_2^1 \phi \left(\lambda_1 + D\frac{x_1 - \lambda_1}{x_3}, \lambda_2 + D\frac{x_2 - \lambda_2}{x_3} \right) \frac{-D(x_2 - \lambda_2)}{x_3^2} \frac{D}{x_3^2} \\
& + \phi \left(\lambda_1 + D\frac{x_1 - \lambda_1}{x_3}, \lambda_2 + D\frac{x_2 - \lambda_2}{x_3} \right) \frac{-2D}{x_3^3}.
\end{aligned}$$

  The first derivative is the sum of expressions of the type Eq. (A.22), thus we have boundedness of $\tilde{P}^2_{j,(0,0,1)}(\psi)$. As for the case of sources on a line, let us show that the derivative of Eq. (A.22) is the finite sum of expressions of the type Eq. (A.22):

$$\begin{aligned}
& \partial_1^1 \left(\partial_1^{\beta_1} \partial_2^{\beta_2} \phi \left(\lambda_1 + D\frac{x_1 - \lambda_1}{x_3}, \lambda_2 + D\frac{x_2 - \lambda_2}{x_3} \right) \frac{(x_1 - \lambda_1)^{p_1} (x_2 - \lambda_2)^{p_2}}{x_3^{p_3}} \right) \\
& \quad = \partial_1^{\beta_1+1} \partial_2^{\beta_2} \phi \left(\lambda_1 + D\frac{x_1 - \lambda_1}{x_3}, \lambda_2 + D\frac{x_2 - \lambda_2}{x_3} \right) \frac{-D(x_1 - \lambda_1)}{x_3^2} \frac{(x_1 - \lambda_1)^{p_1} (x_2 - \lambda_2)^{p_2}}{x_3^{p_3}} \\
& \quad + \partial_1^{\beta_1} \partial_2^{\beta_2+1} \phi \left(\lambda_1 + D\frac{x_1 - \lambda_1}{x_3}, \lambda_2 + D\frac{x_2 - \lambda_2}{x_3} \right) \frac{-D(x_2 - \lambda_2)}{x_3^2} \frac{(x_1 - \lambda_1)^{p_1} (x_2 - \lambda_2)^{p_2}}{x_3^{p_3}} \\
& \quad + \partial_1^{\beta_1} \partial_2^{\beta_2} \phi \left(\lambda_1 + D\frac{x_1 - \lambda_1}{x_3}, \lambda_2 + D\frac{x_2 - \lambda_2}{x_3} \right) \frac{-p_3 (x_1 - \lambda_1)^{p_1} (x_2 - \lambda_2)^{p_2}}{x_3^{p_3+1}}.
\end{aligned}$$

  Then all higher derivatives are sums of expressions of the type Eq. (A.22).

- To handle mixed derivatives of ψ we need to apply three previous steps. Since each $\tilde{P}^2_{j,(\beta_1,\beta_2,\beta_3)}(\psi)$ is the sum expressions of the type Eq. (A.22), then it's bounded with some constant which doesn't depend on ψ.

Then we have from the boundedness of f that $f(\mathfrak{D}^*_{\lambda_1,\lambda_2}(\Phi))$ is bounded for bounded $\mathfrak{D}^*_{\lambda_1,\lambda_2}(\Phi)$. Then $\mathfrak{D}_{\lambda_1,\lambda_2} f$ is bounded from the definition.

Thus, we properly defined the cone-beam transform for fixed λ_1 and λ_2 on distributions of compact support with sources on a plane parallel to the detector.

Bibliography

[AHB87] 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.

[Aic+15] A. Aichert, M. Berger, J. Wang, N. Maass, A. Doerfler, J. Hornegger, and A. Maier. “Epipolar consistency in transmission imaging”. *IEEE Transactions on Medical Imaging*. 34.10 (2015), pp. 1–15. DOI: 10.1109/TMI.2015.2426417.

[Aic+16] A. Aichert, K. Breininger, T. Kohler, and A. Maier. “Efficient epipolar consistency”. In: *Proceedings of the Fourth CT-Meeting*. 2016, pp. 383–386. URL: https://www5.informatik.uni-erlangen.de/Forschung/Publikationen/2016/Aichert16-EEC.pdf.

[Aic19] A. Aichert. “Epipolar consistency in transmission imaging”. PhD thesis. 2019.

[AM+] S. Agarwal, K. Mierle, et al. *Ceres Solver*. URL: http://ceres-solver.org.

[BB00a] S. Basu and Y. Bresler. “Feasibility of tomography with unknown view angles”. *IEEE Transactions on Image Processing*. 9.6 (2000), pp. 1107–1122. DOI: 10.1109/83.846252.

[BB00b] S. Basu and Y. Bresler. “Uniqueness of tomography with unknown view angles”. *IEEE Transactions on Image Processing*. 9.6 (2000), pp. 1094–1106. DOI: 10.1109/83.846251.

[Bon+00] S. Bonnet, F. Peyrin, F. Turjman, and R. Prost. “Tomographic reconstruction using non-separable wavelets”. *IEEE Transactions on Image Processing*. 9.8 (2000), pp. 1445–1450. DOI: 10.1109/83.855441.

[Bon01] J.-M. Bony. *Cours d’analyse: théorie des distributions et analyse de Fourier*. Les éditions de l’Ecole polytechnique, 2001.

[BW94] C. Berenstein and D. Walnut. “Local inversion of the Radon transform in even dimensions using wavelets”. In: *Proceedings of the Conference 75 years of the Radon Transform*. 1994, pp. 38–58.

[BW96] C. Berenstein and D. Walnut. In: *Wavelets in medicine and biology*. Ed. by A. Aldroubi and M. Unser. CRC Press, 1996. Chap. Wavelets and local tomography, pp. 231–262.

[CD13] R. Clackdoyle and L. Desbat. “Full data consistency conditions for cone-beam projections with sources on a plane”. *Physics in Medicine and Biology*. 58.23 (2013), pp. 8437–8456. DOI: 10.1088/0031-9155/58/23/8437.

[CD15] R. Clackdoyle and L. Desbat. "Data consistency conditions for truncated fanbeam and parallel projections". *Medical Physics*. 42.2 (2015), pp. 831–845. DOI: 10.1118/1.4905161.

[Cho+05] Y. Cho, D. J. Moseley, J. H. Siewerdsen, and D. A. Jaffray. "Accurate technique for complete geometric calibration of cone-beam computed tomography systems". *Medical Physics*. 32.4 (2005), pp. 968–983. DOI: 10.1118/1.1869652.

[CL13] J. Chow and D. Lichti. "Photogrammetric bundle adjustment with self-calibration of the PrimeSense 3D camera technology: Microsoft Kinect". *IEEE Access*. 1 (2013), pp. 465–474. DOI: 10.1109/ACCESS.2013.2271860.

[Cla+16] R. Clackdoyle, L. Desbat, J. Lesaint, and S. Rit. "Data consistency conditions for cone-beam projections on a circular trajectory". *IEEE Signal Processing Letters*. 23.12 (2016), pp. 1746–1750. DOI: 10.1109/LSP.2016.2616026.

[Cla13] R. Clackdoyle. "Necessary and sufficient consistency conditions for fanbeam projections along a line". *IEEE Transactions on Nuclear Science*. 60.3 (2013), pp. 1560–1569. DOI: 10.1109/TNS.2013.2251901.

[DB95] A. H. Delaney and Y. Bresler. "Multiresolution tomographic reconstruction using wavelets". *IEEE Transactions on Image Processing*. 4.6 (1995), pp. 799–813. DOI: 10.1109/83.388081.

[DC19] L. Desbat and R. Clackdoyle. "Calibration and data consistency in parallel and fan-beam linogram geometries". In: *Proceedings of the IEEE Nuclear Science Symposium and Medical Imaging Conference*. 2019, pp. 1–5. DOI: 10.1109/NSS/MIC42101.2019.9059826.

[Des+06] L. Desbat, R. Clackdoyle, L. Grezes-Besset, C. Mennessier, and I. Bricault. "Cone-beam imaging of delta functions". In: *IEEE Nuclear Science Symposium Conference Record*. Vol. 5. Dec. 2006, pp. 2859–2863. DOI: 10.1109/NSSMIC.2006.356473.

[DG03] J. Dobbins and D. Godfrey. "Digital X-ray tomosynthesis: current state of the art and clinical potential". *Physics in Medicine and Biology*. 48 (2003), R65–R106. DOI: 10.1088/0031-9155/48/19/R01.

[DP14] A. Delaunoy and M. Pollefeys. "Photometric bundle adjustment for dense multi-view 3D modeling". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1486–1493. DOI: 10.1109/CVPR.2014.193.

[DS14] L. Desbat and B. Spencer. "Calibration with DCC in tomography". In: *Proceedings of the SURGETICA conference*. 2014, pp. 255–257. URL: https://hal.archives-ouvertes.fr/hal-01100508.

[ESN06] C. Engels, H. Stewénius, and D. Nistér. "Bundle adjustment rules". *Photogrammetric Computer Vision*. 2.32 (2006).

[FDK84] L. Feldkamp, L. C. Davis, and J. Kress. "Practical cone-beam algorithm". *Journal of the Optical Society of America*. 1 (1984), pp. 612–619. DOI: 10.1364/JOSAA.1.000612.

[Fee10] T. G. Feeman. *The mathematics of medical imaging: a beginner's guide*. Springer, 2010. DOI: 10.1007/978-0-387-92712-1.

[Flo+06] T. Flohr, C. McCollough, H. Bruder, M. Petersilka, K. Gruber, C. Suess, M. Grasruck, K. Stierstorfer, B. Krauss, R. Raupach, A. Primak, A. Küttner, S. Achenbach, C. Becker, A. Kopp, and B. Ohnesorge. "First performance evaluation of a dual-source CT (DSCT) system". *European radiology*. 16 (2006), pp. 256–268. DOI: 10.1007/s00330-005-2919-2.

[GGV66] I. M. Gelfand, M. I. Graev, and N. Ya. Vilenkin. *Integral geometry and representation theory*. Academic Press, 1966.

[Gin22] N. Gindrier. "Reconstruction cone-beam à partir de projections avec troncations transverses". PhD thesis. 2022.

[GM84] A. Grossmann and J. Morlet. "Decomposition of Hardy functions into square integrable wavelets of constant shape". *SIAM Journal on Mathematical Analysis*. 15 (1984), pp. 723–736. DOI: 10.1137/0515056.

[Gon+13] B. Gonzales, D. Spronk, Y. Cheng, Z. Zhang, X. Pan, M. Beckmann, O. Zhou, and J. Lu. "Rectangular computed tomography using a stationary array of CNT emitters: initial experimental results". *Progress in Biomedical Optics and Imaging - Proceedings of SPIE*. 8668 (2013), 86685K. DOI: 10.1117/12.2008030.

[Gon+18] C. Gong, L. Zeng, C. Wang, and L. Ran. "Design and simulation study of a CNT-based multisource cubical CT system for dynamic objects". *Scanning*. 2018 (2018), pp. 1–15. DOI: 10.1155/2018/6985698.

[GW99] C. Gasquet and P. Witomski. *Fourier analysis and applications: filtering, numerical computation, wavelets*. New York, NY: Springer, 1999.

[Hel+05] L. Helfen, T. Baumbach, P. Mikulik, D. Kiel, P. Pernot, P. Cloetens, and J. Baruchel. "High-resolution three-dimensional imaging of flat objects by synchrotron-radiation computed laminography". *Applied Physics Letters*. 86 (2005), pp. 071915/1–3. DOI: 10.1063/1.1854735.

[Hel65] S. Helgason. "The Radon transform on Euclidean spaces, compact two-point homogeneous spaces and Grassmann manifolds". *Acta Mathematica*. 113 (1965), pp. 153–180. DOI: 10.1007/BF02391776.

[Hel99] S. Helgason. *The Radon transform*. Springer US, 1999. DOI: 10.1007/978-1-4757-1463-0.

[Hof+17] M. Hoffmann, A. Ernst, T. Bergen, S. Hettenkofer, and J.-U. Garbas. “A robust chessboard detector for geometric camera calibration”. In: *Proceedings of VISIGRAPP*. 2017, pp. 34–43. DOI: 10.5220/0006104300340043.

[Hof+18] M. Hoffmann, T. Wurfl, N. Maass, F. Dennerlein, A. Aichert, and A. Maier. “Empirical scatter correction using the Epipolar Consistency Condition”. In: *Proceedings of the Fifth International Conference on Image Formation in X-Ray Computed Tomography (CT-Meeting)*. 2018, pp. 193–197. URL: https://www5.informatik.uni-erlangen.de/Forschung/Publikationen/2018/Hoffmann18-ESC.pdf.

[HZ04] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2004. DOI: 10.1017/CBO9780511811685.

[Jon18] P. Jonas. “Design criteria for geometrical calibration phantoms in fan and cone beam CT systems”. *Journal of Inverse and Ill-posed Problems*. 26.6 (2018), pp. 729–753. DOI: 10.1515/jiip-2017-0084.

[Kal06] W. Kalender. “X-ray computed tomography”. *Physics in Medicine and Biology*. 51 (2006), R29–R43. DOI: 10.1088/0031-9155/51/13/R03.

[KD23] A. Konik and L Desbat. “Self-calibration with range conditions for fan-beam on distributions with sources on a line” (2023). In preparation.

[KDG21] A. Konik, L. Desbat, and Y. Grondin. “Non-uniqueness in C-arm calibration with bundle adjustment”. In: *Proceedings of the 16th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*. 2021, pp. 299–303. URL: https://hal.archives-ouvertes.fr/hal-03419265/.

[KDG22] A. Konik, L. Desbat, and Y. Grondin. “Hybrid calibration in 3D cone-beam geometry with sources on a line”. In: *Proceedings of the IEEE Nuclear Science Symposium and Medical Imaging Conference*. 2022.

[KS01] A. Kak and M. Slaney. *Principles of computerized tomographic imaging*. SIAM edition, 2001. DOI: 10.1137/1.9780898719277.

[Les+17] J. Lesaint, S. Rit, R. Clackdoyle, and L. Desbat. “Calibration for circular cone-beam CT based on consistency conditions”. *IEEE Transactions on Radiation and Plasma Medical Sciences*. 1.6 (2017), pp. 517–526. DOI: 10.1109/TRPMS.2017.2734844.

[Les18] J. Lesaint. “Data consistency conditions in X-ray transmission imaging and their application to the self-calibration problem”. PhD thesis. 2018.

[Lud66] D. Ludwig. “The Radon transform on Euclidean space”. *Communications on Pure and Applied Mathematics*. 19.1 (1966), pp. 49–81. DOI: 10.1002/cpa.3160190105.

[LW12] A. Ladikos and W. Wein. “Geometric calibration using bundle adjustment for cone-beam computed tomography devices”. In: *Medical Imaging 2012: Physics of Medical Imaging. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*. Vol. 8313. 2012, p. 83132T. DOI: 10.1117/12.906238.

[Mal03] S. Mallat. “Foveal detection and approximation for singularities”. *Applied and Computational Harmonic Analysis*. 14 (2003), pp. 133–180. DOI: 10.1016/S1063-5203(03)00021-6.

[Mal09] S. Mallat. *A wavelet tour of signal processing*. Academic Press, 2009. DOI: 10.1016/B978-0-12-374370-1.00004-5.

[MCN09] C. Mennessier, R. Clackdoyle, and F. Noo. “Direct determination of geometric alignment parameters for cone-beam scanners”. *Physics in Medicine and Biology*. 54.6 (2009), pp. 1633–1660. DOI: 10.1088/0031-9155/54/6/016.

[MH92] S. Mallat and W. Hwang. “Singularity detection and processing with wavelets”. *IEEE Transactions on Information Theory*. 38.2 (1992), pp. 617–643. DOI: 10.1109/18.119727.

[MZ92] S. Mallat and S. Zhong. “Characterization of signals from multiscale edges”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 14.7 (1992), pp. 710–732. DOI: 10.1109/34.142909.

[Nat01] F. Natterer. *The mathematics of computerized tomography*. SIAM edition, 2001. DOI: 10.1137/1.9780898719284.

[NB10] L. Narici and E. Beckenstein. *Topological vector spaces*. Chapman and Hall/CRC, 2010. DOI: 10.1201/9781584888673.

[NDC20] H. Nguyen, L. Desbat, and R. Clackdoyle. “Automatic geometric calibration in 3D cone-beam geometry with sources on a line”. In: *Proceedings of the 6th International Conference on Image Formation in X-ray Computed Tomography*. 2020, pp. 530–533.

[Ngu21] H. Nguyen. “Data consistency conditions in 3D tomography and scanner calibration using analytic approaches”. PhD thesis. 2021.

[Noo+00] F. Noo, R. Clackdoyle, C. Mennessier, T. A. White, and T. J. Roney. “Analytic method based on identification of ellipse parameters for scanner calibration in cone-beam tomography”. *Physics in Medicine and Biology*. 45.11 (2000), pp. 3489–3508. DOI: 10.1088/0031-9155/45/11/327.

[OD94] T. Olson and J. DeStefano. “Wavelet localization of the Radon transform”. *IEEE Transactions on Signal Processing*. 42.8 (1994), pp. 2055–2067. DOI: 10.1109/78.301841.

[Pan+08] D. Panetta, N. Belcari, A. Del Guerra, and S. Moehrs. "An optimization-based method for geometrical calibration in cone-beam CT without dedicated phantoms". *Physics in Medicine and Biology*. 53 (2008), pp. 3841–3861. DOI: 10.1088/0031-9155/53/14/009.

[PKB14] V. Pradeep, K. Konolige, and E. Berger. "Calibrating a multi-arm multi-sensor robot: a bundle adjustment approach". *Experimental Robotics*. 79 (2014), pp. 211–225. DOI: 10.1007/978-3-642-28572-1_15.

[PKV99] 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.

[Pre+19] A. Preuhs, M. Manhart, P. Roser, B. Stimpel, C. Syben, M. Psychogios, M. Kowarschik, and A. Maier. *Deep autofocus with cone-beam CT consistency constraint*. 2019. arXiv: 1911.13162 [cs.LG].

[Ras+97] F. Rashid-Farrokhi, K. J. Ray Liu, C. Berenstein, and D. Walnut. "Wavelet-based multiresolution local tomography". *IEEE Transactions on Image Processing*. 6.10 (1997), pp. 1412–1429. DOI: 10.1109/83.624961.

[RGG94] P. Rizo, P. Grangeat, and R. Guillemaud. "Geometric calibration method for multiple-head cone-beam SPECT system". *IEEE Transactions on Nuclear Science*. 41.6 (1994), pp. 2748–2757. DOI: 10.1109/23.340643.

[Rit+14] S. Rit, M. Vila Oliva, S. Brousmiche, R. Labarbe, D. Sarrut, and G. Sharp. "The Reconstruction Toolkit (RTK), an open-source cone-beam CT reconstruction toolkit based on the Insight Toolkit (ITK)". *Journal of Physics: Conference Series*. 489 (2014). DOI: 10.1088/1742-6596/489/1/012079.

[RK96] A. G. Ramm and A. I. Katsevich. *The Radon transform and local tomography*. CRC Press, 1996. DOI: 10.1201/9781003069331.

[Spe15] B. Spencer. "On-line C-arm intrinsic calibration by means of an accurate method of line detection using the Radon transform". PhD thesis. 2015.

[Stu12] P. Sturm. "Some lecture notes on geometric computer vision". In: 2012.

[TDK05] F. B. Tek, A. G. Dempster, and I. Kale. "Blood cell segmentation using minimum area watershed and circle Radon transformations". In: *Proceedings of the 7th International Symposium on Mathematical Morphology*. 2005, pp. 441–454. DOI: 10.1007/1-4020-3443-1_40.

[Tri+00] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. “Bundle adjustment - a modern synthesis”. In: *Proceedings of the International Workshop on Vision Algorithms*. 2000, pp. 298–372. DOI: 10.1007/3-540-44480-7_21.

[TSH19] O. Tischenko, N. Saeid Nezhad, and C. Hoeschen. “A method of determining geometry of cone beam CT scanner”. *Biomedical Physics & Engineering Express*. 5.5 (2019), p. 055006. DOI: 10.1088/2057-1976/ab367a.

[Unb+17] M. Unberath, A. Aichert, S. Achenbach, and A. Maier. “Consistency-based respiratory motion estimation in rotational angiography”. *Medical Physics*. 44.9 (2017), pp. e113–e124. DOI: 10.1002/mp.12021.

[Urb+17] S. Urban, S. Wursthorn, J. Leitloff, and S. Hinz. “MultiCol bundle adjustment: a generic method for pose estimation, simultaneous self-calibration and reconstruction for arbitrary multi-camera systems”. *International Journal of Computer Vision*. 121 (2017), pp. 234–252. DOI: 10.1007/s11263-016-0935-0.

[WT04] M. Wilk and L. Tsukerman. “Method of fan-beam imaging calibration”. In: *Proceedings of the IEEE Nuclear Science Symposium and Medical Imaging Conference*. Vol. 6. 2004, pp. 3793–3795. DOI: 10.1109/NSSMIC.2004.1466706.

[Wur+17] T. Wurfl, N. Maass, F. Dennerlein, X. Huang, and A. Maier. “Epipolar consistency guided beam hardening reduction - ECC2”. In: *Proceedings of the 14th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*. 2017, pp. 181–185.

[Wur+18] T. Wurfl, N. Maass, F. Dennerlein, A. Aichert, and A. Maier. “Physical constraints for beam hardening reduction using polynomial models”. In: *Proceedings of the Fifth International Conference on Image Formation in X-Ray Computed Tomography (CT-Meeting)*. 2018, pp. 356–359. URL: https://www5.informatik.uni-erlangen.de/Forschung/Publikationen/2018/Wurfl18-PCF.pdf.

[Wur+19] T. Wurfl, M. Hoffmann, A. Aichert, A. Maier, N. Maass, and F. Dennerlein. “Calibration-free beam hardening reduction in X-ray CBCT using the epipolar consistency condition and physical constraints”. *Medical Physics*. 46.12 (2019), pp. e810–e822. DOI: 10.1002/mp.13625.

[YLC16] Y. Yang, L. Li, and Z. Chen. “A review of geometric calibration for different 3-D X-ray imaging systems”. *Nuclear Science and Techniques*. 27.3 (2016), pp. 1–11. DOI: 10.1007/s41365-016-0073-y.

[Yu+06] H. Yu, Y. Wei, J. Hsieh, and G. Wang. “Data consistency based translational motion artifact reduction in fan-beam CT”. *IEEE Transactions on Medical Imaging*. 25 (2006), pp. 792–803. DOI: 10.1109/TMI.2006.875424.

[Yu+15] H. Yu, G. Wang, J. Yang, J. Pack, M. Jiang, and B. De Man. “Data consistency condition for truncated projections in fan-beam geometry”. *Journal of X-Ray Science and Technology*. 23 (2015), pp. 627–638. DOI: 10.3233/XST-150515.

[YW07] H. Yu and G. Wang. “Data consistency based rigid motion artifact reduction in fan-beam CT”. *IEEE transactions on medical imaging*. 26 (2007), pp. 249–260. DOI: 10.1109/TMI.2006.889717.