Using rotational Invariances of Operators for the Implementation of Discrete Algorithms

P. Jonas, A.K. Louis, A. Rösch

Fachbereich Mathematik, Universität des Saarlandes, D-66041 Saarbrücken

Abstract. In this paper we discuss advantages of using invariances of the operator for the implementation of a discrete algorithm. It turns out that in the case of rotational invariance polar grids play an important role. Since we need cartesian coordinates for the visualisation an interpolation from polar to cartesian coordinates has to be done. We give an error estimate for this transformation. With the help of the error estimate we derive some strategy for the choice of the polar grid points. This strategy is compared with two other strategies in several examples. An application to the inversion of the Radon Transform is included. This generalizes and improves the heuristic approach presented in [1, 2].

1. Introduction

In many applications of imaging the measurement geometries have special invariances. For example in computer tomography one often deals with equidistant angle measurements. Another expample is 2d ultrasonic tomography where circular detectors are used. However these special invariances are rarely taken into account when a reconstruction algorithm is developed although they can make the problem clearer or lead to a faster reconstruction method.

In this paper we take a closer look on problems which have certain invariances and examine what the conditions are for the heredity of these invariances for the discrete algorithm. We notice that we have to be careful with the choice of the discrete grid and the dicretisation of the operator.

It turns out that polar grids play an important role especially for rotation invariant problems. In that case we first carry out all computations in polar coordinates and then interpolate on a cartesian grid for visualisation. To manage the first step we have to choose a discrete grid in the polar coordinate system. A very elementary choice is an equidistant dicretisation in radius r and angle θ, see figure 1. As we can immediately guess the point density at the origin is too high.

We derive error estimates and distribute the interpolation points in such a way, that the interpolation error is equalized over the whole domain.

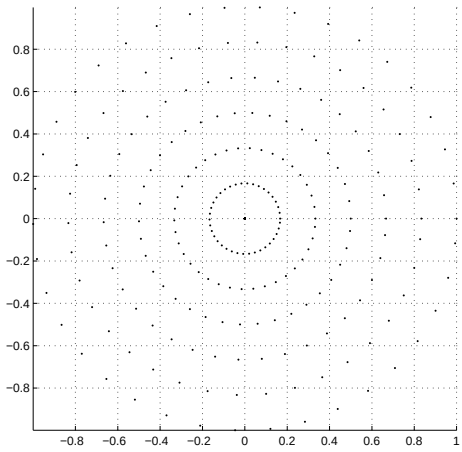

Figure 1. Equidistant polar grid

2. Using Invariances of Operators

Let Ω_1, Ω_2 be two domains and $X(\Omega_1)$, $Y(\Omega_2)$ be Hilbert spaces. Further on we take two bijective transformations $D_1 : \Omega_1 \to \Omega_1$ and $D_2 : \Omega_2 \to \Omega_2$.
We study operator equations $Af = g$ for linear operators between X and Y. Let $T_1 : X \to X$ with $T_1 f(x) = f(D_1 x)$, $T_2 : Y \to Y$ with $T_2 g(y) = g(D_2 y)$ and $\tilde{T} : Y \to Y$ be three operators. We assume the invariances of the operator A with respect to T_1, T_2, $\tilde{T}$ in the following way

$$AT_1 = T_2 A \text{ and } \tilde{T}A = AT_1^*.$$

If T_1 is self-adjoint or unitary then trivially we have $\tilde{T} = T_2^*$.
In the case of a not continuously invertible operator A we have to use a regularisation for the solution of the equation $Af = g$. A very flexible method is the so called approximate inverse, see [4, 5, 6], that is we calculate for a given mollifier e_γ the minimum-norm solution ψ_γ of the equation

$$AA^*\psi_\gamma = Ae_\gamma.$$

Now the regularised solution of the equation $Af = g$ is given by $f_\gamma(x) = \langle \psi_\gamma(x, \cdot), g \rangle_Y$. As the invariances are also valid for the adjoint operator

$$T_1^* A^* = A^* T_2^*$$

we are able to save the invariances for the reconstruction method with an appropriate choice of the mollifier e_γ. In the following we use a special convention: If an operator is applied to a function of two variables it only acts on the second variable, e.g. $T_1 f(x, y) = f(x, D_1 y)$.

Theorem 2.1 *Let $e_\gamma \in X \times X$ and $e_\gamma(D_1 x, w) = T_1^* e_\gamma(x, w)$ for all $x, w \in \Omega_1$ then holds*

$$T_1 f_\gamma(x) = \langle \psi_\gamma(x, \cdot), T_2 g \rangle_Y.$$

Proof: First we show that $\psi_\gamma(D_1 x, y) = T_2^* \psi_\gamma(x, y)$ for all $x \in \Omega_1$ and $y \in \Omega_2$. We calculate

$$\begin{aligned} AA^* \psi_\gamma(D_1 x, w) &= Ae_\gamma(D_1 x, w) \\ &= AT_1^* e_\gamma(x, w) \\ &= \tilde{T} Ae_\gamma(x, w) \\ &= \tilde{T} AA^* \psi_\gamma(x, w) \\ &= AT_1^* A^* \psi_\gamma(x, w) \\ &= AA^* T_2^* \psi_\gamma(x, w). \end{aligned}$$

As ψ_γ and $T_2^* \psi_\gamma$ are minimum-norm solutions we obtain

$$\psi_\gamma(D_1 x, y) = T_2^* \psi_\gamma(x, y) \text{ for all } x \in X \text{ and } y \in Y.$$

Finally we have

$$\begin{aligned} T_1 f_\gamma(x) &= \langle \psi_\gamma(D_1 x, \cdot), g \rangle_Y \\ &= \langle T_2^* \psi_\gamma(x, \cdot), g \rangle_Y \\ &= \langle \psi_\gamma(x, \cdot), T_2 g \rangle_Y. \end{aligned}$$

□

Similar results to theorem 2.1 but with sligthly different requirements can be found in [5] and [9]. Now that we have a regularisation method with special invariances we turn to the numerical implementation. So we define $\mathcal{G}$ as the set of grid points of Ω_1 and $\mathcal{D}$ as the grid points of Ω_2. We substitute the equation $f_\gamma(x) = \langle \psi_\gamma(x, \cdot), g \rangle_Y$ for all $x \in \Omega_1$ by a discrete version

$$f_\gamma(\tilde{x}) = \sum_{\tilde{y} \in \mathcal{D}} c(\tilde{x}, \tilde{y}) g(\tilde{y}) \text{ for all } \tilde{x} \in \mathcal{G}$$

with $c(\tilde{x}, \tilde{y})$ being a certain weight function arsing from the kernel ψ_γ.

Theorem 2.2 *If the following conditions are fulfilled*

(i) $D_1 \mathcal{G} \subseteq \mathcal{G}$

(ii) $D_2 \mathcal{D} \subseteq \mathcal{D}$

(iii) $c(\tilde{x}, \tilde{y}) = c(D_1 \tilde{x}, D_2 \tilde{y})$ *for all* $\tilde{x} \in \mathcal{G}$ *and* $\tilde{y} \in \mathcal{D}$

then holds

$$f_\gamma(D_1 \tilde{x}) = \sum_{\tilde{y} \in \mathcal{D}} c(\tilde{x}, \tilde{y}) g(D_2 \tilde{y}) \text{ for all } \tilde{x} \in \mathcal{G}.$$

Proof: We have for all $\tilde{x} \in \mathcal{G}$

$$\begin{aligned} f_\gamma(D_1 \tilde{x}) &= \sum_{\tilde{y} \in \mathcal{D}} c(D_1 \tilde{x}, \tilde{y}) g(\tilde{y}) \\ &= \sum_{\tilde{y} \in \mathcal{D}} c(\tilde{x}, D_2^{-1} \tilde{y}) g(\tilde{y}) \end{aligned}$$

$$= \sum_{\tilde{y} \in D_2^{-1}\mathcal{D}} c(\tilde{x}, \tilde{y}) g(D_2 \tilde{y})$$
$$= \sum_{\tilde{y} \in \mathcal{D}} c(\tilde{x}, \tilde{y}) g(D_2 \tilde{y}).$$

□

The consequence of this theorem is: If we know the weight function c for one grid point $\tilde{x}_0$ then we already have determined the weight function for all points $\tilde{x} = D_1^n \tilde{x}_0$ with $n \in \mathbb{Z}$, because

$$f_\gamma(D_1^n \tilde{x}_0) = \sum_{\tilde{y} \in \mathcal{D}} c(\tilde{x}_0, \tilde{y}) g(D_2^n \tilde{y}).$$

This means a lot of computation time can be saved in the numerical evaluation.

Example 1 *Let* $\Omega_1 = \Omega_2 = \mathbb{R}$ *and*

$$Af(y) = \int_{\mathbb{R}} k(x - y) f(x) dx.$$

Obviously this operator is translation invariant. So we have

$$g(S^a y) = \int_{\mathbb{R}} k(x - y) f(S^a x) dx$$

with $S^a x = x - a$. *If we take a translation invariant mollifier* $e_\gamma(x, y) = e_\gamma(x - y)$ *then the invariance is saved for the regularisation method, too. Now we define the mesh grid by* $\mathcal{G} = \mathcal{D} = \{x = jh : j \in \mathbb{Z}\}$ *with stepsize* h *and use the transformation* $D_1 = D_2 = S^h$. *The numerical version of the approximate inverse is*

$$f_\gamma(\tilde{x}) = \sum_{\tilde{y} \in \mathcal{D}} \psi_\gamma(\tilde{x} - \tilde{y}) g(\tilde{y})$$

In this setting the theorem 2.2 can be applied and we have to compute the weight function for only one point $\tilde{x}_0$.

Example 2 *Let* $\Omega_1 = \Omega_2 = B_1(0)$ *be the unit disc and*

$$Af(x) = \int_{B_1(0)} k(\langle x, y \rangle) f(y) dy.$$

This operation is rotation invariant. We have

$$g(R^\alpha x) = \int_{B_1(0)} k(\langle x, y \rangle) f(R^\alpha y) dy$$

with R^α *being the rotation around the origin with angle* α. *For this reason we take the polar meshgrids* $\mathcal{G} = \mathcal{D} = \{x = r_j(\cos\varphi_k, \sin\varphi_k) : r_j = (j-1)/(N_r - 1)$ *with* $j \in \{1, .., N_r\}$ *and* $\varphi_k = 2\pi(k-1)/N_\varphi$ *with* $k \in \{1, .., N_\varphi\}\}$ *with the two dicretisation numbers* $N_r \in \mathbb{N}$, $N_\varphi \in \mathbb{N}$ *and the transformation* $D_1 = D_2 = R^{2\pi/N_\varphi}$. *If we take a rotation invariant mollifier,* $e_\gamma(x, y) = e_\gamma(\|x - y\|)$ *the theorem 2.2 can be applied again and we see that the weight function has to be evaluated for all radii* $r_j = (j-1)/(N_r - 1)$. *This means that the weight function has to be computed for* $(N_r + 1)$ *points in order to obtain it for all points.*

These two invariances (translation and rotation) are met very often in applications. While translation invariances suggest a cartesian grid, rotation invariances leads to polar grids. Dealing with polar grids can be a little tricky because a naive discretisation, see figure 1, generates too many points around the origin. We take a closer look at this problem in the next section.

3. Interpolation from Polar to Cartesian Grid and Error Estimate

Now we concentrate on problems with rotation invariances. Considering our idea from the previous section we are led to use a polar meshgrid. But in order to visualize the function we need it in cartesian coordinates. So interpolation is necessary. Because we want to derive a fast method we take an interpolation scheme which only needs few computations namely bilinear interpolation. Let the function f be known at the four points $P_1 = (r, \theta)$, $P_2 = (r + h_r, \theta)$, $P_3 = (r, \theta + h_\theta)$ and $P_4 = (r + h_r, \theta + h_\theta)$ with $r, h_r \in \mathbb{R}^+ \cup \{0\}$ and $\theta, h_\theta \in [0, 2\pi)$. We define the domain

$$\Omega = \{(\tilde{r}, \tilde{\theta}) : r \leq \tilde{r} \leq r + h_r \text{ and } \theta \leq \tilde{\theta} \leq \theta + h_\theta\}.$$

Let I be the interpolation operator, such that $If(\tilde{r}, \tilde{\theta})$ is the interlopation of f at the point $(\tilde{r}, \tilde{\theta}) \in \Omega$, then we have

$$\begin{aligned} If(\tilde{r}, \tilde{\theta}) = (1 - \eta_2)\left((1 - \eta_1)f(r, \theta) + \eta_1 f(r, \theta + h_\theta)\right) + \\ \eta_2\left((1 - \eta_1)f(r + h_r, \theta) + \eta_1 f(r + h_r, \theta + h_\theta)\right) \end{aligned}$$

with $\eta_1 = \frac{\tilde{\theta} - \theta}{h_\theta}$ and $\eta_2 = \frac{\tilde{r} - r}{h_r}$. In order to derive an error estimate we treat the derivatives of f in cartesian coordinates.

Theorem 3.1 *Let $f \in C^2(\Omega)$ with*

$$\max_{\substack{x \in \Omega \\ i \in \{1,2\}}} \left|\frac{\partial f}{\partial x_i}(x)\right| \leq \tilde{M} \text{ and } \max_{\substack{x \in \Omega \\ i,j \in \{1,2\}}} \left|\frac{\partial^2 f}{\partial x_i \partial x_j}(x)\right| \leq M$$

then we have for all points $(\tilde{r}, \tilde{\theta}) \in \Omega$ the inequality

$$|If(\tilde{r}, \tilde{\theta}) - f(\tilde{r}, \tilde{\theta})| \leq \frac{1}{4}h_r^2 M + \frac{1}{4}h_\theta^2\left(\tilde{M}(r + h_r) + M(r + h_r)^2\right).$$

Proof: For the bilinear interpolation we obtain

$$|If(\tilde{r}, \tilde{\theta}) - f(\tilde{r}, \tilde{\theta})| \leq \frac{1}{8}\left(h_r^2 \max_{x \in \Omega}\left|\frac{\partial^2 f}{\partial r^2}(x)\right| + h_\theta^2 \max_{x \in \Omega}\left|\frac{\partial^2 f}{\partial \theta^2}(x)\right|\right).$$

In order to transform the derivatives to cartesian coordinates we need

$$\begin{array}{rclcrcl} \frac{\partial x_1}{\partial r} & = & \cos\theta & ; & \frac{\partial x_1}{\partial \theta} & = & -\sin\theta\ r \\ \frac{\partial x_2}{\partial r} & = & \sin\theta & ; & \frac{\partial x_2}{\partial \theta} & = & \cos\theta\ r. \end{array}$$

Inserting leads to

$$\begin{aligned} \left|\frac{\partial^2 f}{\partial r^2}\right| &\leq (\cos\theta)^2\left|\frac{\partial^2 f}{\partial x_1^2}\right| + 2|\sin\theta\ \cos\theta|\left|\frac{\partial^2 f}{\partial x_1 \partial x_2}\right| + (\sin\theta)^2\left|\frac{\partial^2 f}{\partial x_2^2}\right| \\ &\leq 2M \end{aligned}$$

and

$$\begin{aligned}\left|\frac{\partial^2 f}{\partial\theta^2}\right| &\leq |\cos\theta|\ r\left|\frac{\partial f}{\partial x_1}\right| + (\sin\theta)^2 r^2\left|\frac{\partial^2 f}{\partial x_1^2}\right| + 2|\cos\theta\sin\theta|\ r^2\left|\frac{\partial^2 f}{\partial x_1\partial x_2}\right| + \\ &\quad |\sin\theta|\ r\left|\frac{\partial f}{\partial x_2}\right| + (\cos\theta)^2 r^2\left|\frac{\partial^2 f}{\partial x_2^2}\right| \\ &\leq 2\tilde{M}r + 2Mr^2.\end{aligned}$$

Altogether we obtain

$$|If(\tilde{r},\tilde{\theta}) - f(\tilde{r},\tilde{\theta})| \leq \frac{1}{4}h_r^2 M + \frac{1}{4}h_\theta^2\left(\tilde{M}(r+h_r) + M(r+h_r)^2\right).$$

□

4. Algorithms and Numerical Tests with Polar Grids

In this section we give three choices for the polar grid. First we take an equidistant grid in radius and angle to discretise the unit disc.

Method 1 (Equidistant) *Let N_r and N_θ be two fixed numbers. The grid points are given by*

$$(r_j,\theta_k) = ((j-1)h_r, (k-1)h_\theta)$$

with $h_r = 1/(N_r-1)$, $h_\theta = 2\pi/N_\theta$ and $j \in \{1,..,N_r\}$, $k \in \{1,..,N_\theta\}$.

In our second method we apply the theorem of the previous section. Because the first part of the error estimate in theorem 3.1 does not depend on radius and angle we take a fixed stepsize h_r. However the second part of the error estimate does depend on the radius r. So we take a stepsize h_θ which is dependent on the radius r. Because the term $\tilde{M}r$ dominates Mr^2 for small values of r we choose $h_\theta(r) \sim 1/\sqrt{r}$ for small r . On the other hand we select $h_\theta(r) \sim 1/r$ for large r. To connect these two domains we choose c such that $Mr^2 < \tilde{M}r$ holds for all $r \in [0,c)$.
In the following we use the definition

$$\lfloor x \rfloor := \max\{j \in \mathbb{Z} : j \leq x\}.$$

Method 2 (Variable Angle) *Let N_r and c be two fixed numbers. The grid points are given by*

$$(r_j,\theta_k) = ((j-1)h_r, (k-1)h_\theta(r_j))$$

with $h_r = 1/(N_r-1)$, $j \in \{1,..,N_r\}$ and $h_\theta(r_j) = 2\pi/N_\theta(r_j)$, $k \in \{1,..,N_\theta(r_j)\}$. Here we choose

$$N_\theta(r_j) = \begin{cases} \left\lfloor \sqrt{c}\frac{2\pi\sqrt{r_j}}{h_r} \right\rfloor & \text{for } r_j < c \\ \left\lfloor \sqrt{c}\frac{2\pi\sqrt{c}}{h_r} + \frac{2\pi(r_j-c)}{h_r} \right\rfloor & \text{else.} \end{cases}$$

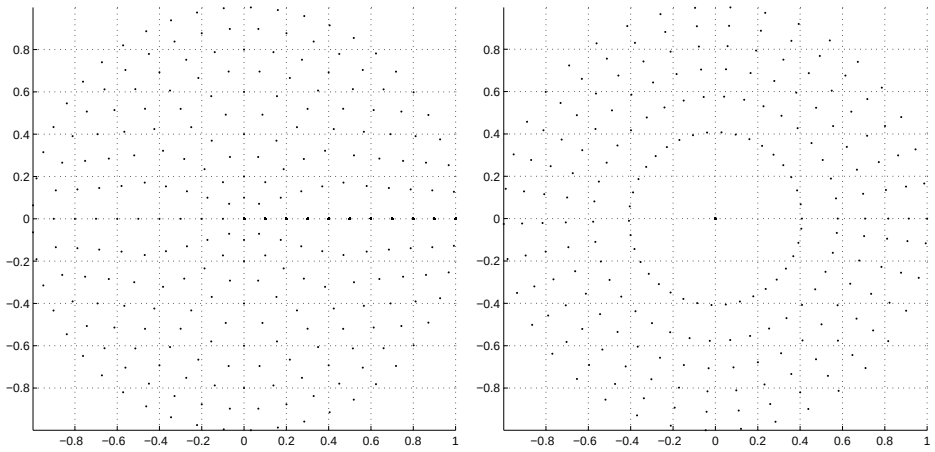

Figure 2. Variable Angle (left) and Variable Radius (right)

In our last approach we argue heuristically. In contrast to the previous method we want to fix the discretisation number for the angle. Our main idea is now that all annulus emerging of the radius discretisation have the same surface area. Concerning this we end up with a radius distribution $r_j \sim \sqrt{j-1}$.

Method 3 (Variable Radius) *Let N_r and N_θ be two fixed numbers. The grid points are given by*

$$(r_j, \theta_k) = \left(\sqrt{\frac{j-1}{N_r - 1}}, (k-1)h_\theta \right)$$

with $h_\theta = 2\pi/N_\theta$ and $j \in \{1, .., N_r\}$, $k \in \{1, .., N_\theta\}$.

Now that we have three strategies we want to test them in some numerical examples. To this end we take ellipses which we put at different positions. The ellipses are given by

Ellipse A	semi axis	a=0.4, b=0.2
	center	$(0, 0)$
	rotation angle	0
Ellipse B	semi axis	a=0.4, b=0.2
	center	$(0.5, 0)$
	rotation angle	$\pi/3$
Ellipse C	semi axis	a=0.4, b=0.2
	center	$(0.35, -0.4)$
	rotation angle	$\pi/4$

In our test we start with the evaluation of these ellipses at the grid points of the three methods. Then we use bilinear interpolation for the visualisation in cartesian coordinates. For the tests we take:

method 1: $N_r = 28$; $N_\theta = 168$
method 2: $N_r = 39$; $c = 0.1$
method 3: $N_r = 28$; $N_\theta = 168$.

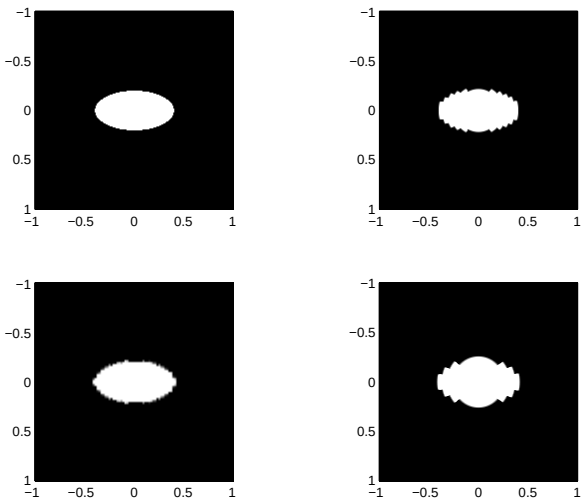

Figure 3. Pictures of Ellipse A
Top left: Original; Top right: Equidistant Dicretisation
Bottom left: Variable Angle; Bottom right: Variable Radius

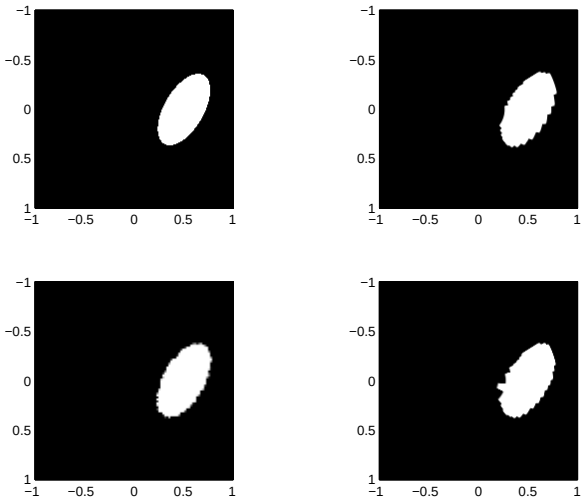

Figure 4. Pictures of Ellipse B
Top left: Original; Top right: Equidistant Dicretisation
Bottom left: Variable Angle; Bottom right: Variable Radius

The reconstructions can be seen in figures 3-5. We notice that method 1 and method 3 produce a frazzeled border. On the other hand method 2 generates a relativly smooth border. This observation is strengthened by the error plot of figure 6 where we varied the total number of discretisation points.

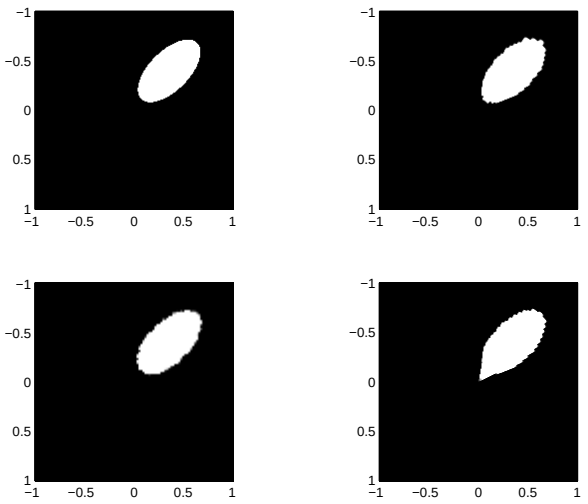

Figure 5. Pictures of Ellipse C
Top left: Original; Top right: Equidistant Dicretisation
Bottom left: Variable Angle; Bottom right: Variable Radius

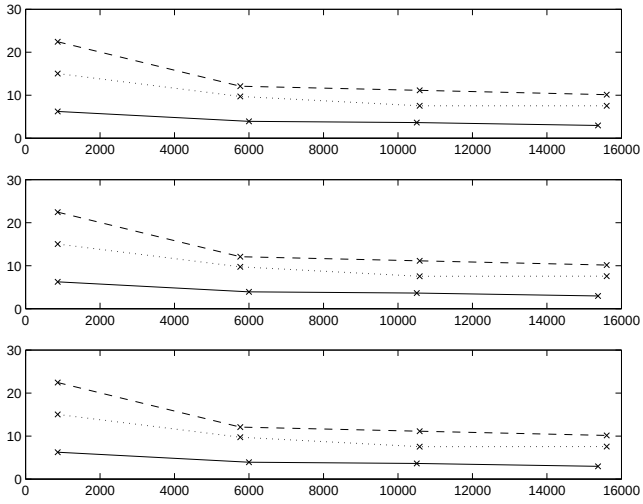

Figure 6. Error of the different methods plotted against total number of discretisation points
Equidistant: pointed line; Variable Angle: solid line; Variable Radius: dashed line
First plot: Ellipse A; second: Ellipse B; third: Ellipse C

5. Application to the Inversion of the Radon Transform

In this section we apply our ideas to the inverison of the 2d Radon Transform. The Radon Transform is defined by

$$\begin{array}{rcccl} \mathcal{R} & : & L_2(\mathbb{R}^2) & \to & L_2(\mathcal{S}^1 \times \mathbb{R}) \\ & & f & \mapsto & \mathcal{R}f : \mathcal{S}^1 \times \mathbb{R} \to \mathbb{R} \\ & & & & (\omega, s) \mapsto \mathcal{R}f(\omega, s) = \int_{\mathbb{R}} f(s\omega + t\omega^{\perp})dt \end{array}$$

It has two important invariances, see [3, 7, 8]. The first one is a translation invariance

$$T_1^a \mathcal{R} = \mathcal{R} T_2^a$$

with $T_1^a g(\omega, s) = g(\omega, s - \langle a, \omega \rangle)$, $a \in \mathbb{R}^2$ and $T_2^a f(x) = f(x - a)$, $a \in \mathbb{R}^2$. The second is a rotation invariance

$$T_3^\alpha \mathcal{R} = \mathcal{R} T_4^\alpha$$

with $T_3^\alpha g(\omega, s) = g(R^\alpha \omega, s)$, $T_4^\alpha f(x) = f(R^\alpha x)$ and R^α being a rotation around the origin with angle α.
The measurement geometry for the detector, that means the dicretisation of $g(\omega, s)$, is fixed in applications by the scanning setup and looks mostly like this: Given the stepsize h and the number of directions p

$$\begin{aligned}\mathcal{D} = \{(\omega, s) : \omega = (\cos(\varphi_k), \sin(\varphi_k)) \text{ with} \\ \varphi_k = 2\pi(k-1)/p, k \in \{1, .., p\} \text{ and } s = mh \text{ with } m \in \mathbb{Z}\}.\end{aligned}$$

Now we want to examine whether the translation invariance can be saved for a numerical algorithm. Refering to theorem 2.2 we need $T_1^a \mathcal{D} \subseteq \mathcal{D}$ that is $(\omega, \langle a, \omega \rangle) \in \mathcal{D}$ for all $\omega = (\cos(\varphi_k), \sin(\varphi_k))$ with $\varphi_k = 2\pi(k-1)/p$ and $k \in \{1, .., p\}$.

Lemma 5.1 *Let $a \neq 0$ and $p > 2$, if $(\omega, \langle a, \omega \rangle) \in \mathcal{D}$ for all $\omega = \omega(\varphi_k) = (\cos(\varphi_k), \sin(\varphi_k))$ with $\varphi_k = 2\pi(k-1)/p$ and $k \in \{1, .., p\}$ then holds*

$$\cos(\varphi_k) \in \mathbb{Q} \text{ for all } k \in \{1, .., p\}.$$

Proof: Let $a = r(\cos\theta, \sin\theta)$. Because of the assumption $(\omega_k, \langle a, \omega \rangle) \in \mathcal{D}$ for all $\omega_k = (\cos(\varphi_k), \sin(\varphi_k))$ with $k \in \{1, .., p\}$, we have

$$\langle a, \omega_k \rangle = r\cos(\theta - \varphi_k) = h\, m_k \quad \text{for certain numbers } m_k \in \mathbb{Z}.$$

Now we assume that $\cos\theta \neq 0$. If this condition does not hold we rotate the coordinate system with an angle $2\pi/p$ and now we assured that $\cos\theta \neq 0$, because $p > 2$.
It follows

$$\frac{r\cos(\theta - \varphi_k)}{r\cos\theta} = \frac{m_k}{m_0}.$$

Simple calculation provides

$$\frac{\cos(\theta - \varphi_k)}{\cos\theta} = \cos\varphi_k + \tan\theta \sin\varphi_k.$$

In combination with the fact that $(\omega(\varphi_k), s) \in \mathcal{D}$ implicates $(\omega(-\varphi_k), s) \in \mathcal{D}$ we obtain

$$\cos(\varphi_k) \in \mathbb{Q} \text{ for all } k \in \{1, .., p\}.$$

□

As a result of this lemma the translation invariance cannot be utilized in a simple way because almost every measurement setup contradicts to: $\cos\varphi_k \in \mathbb{Q}$ for all $k \in \{1, .., p\}$. However it is possible in simple cases like $p = 3, 4, 6$.
Now we turn to the rotation invariance. For a given stepsize h we choose the grid

$\mathcal{G} = \{x = r_j(\cos\varphi_k, \sin\varphi_k) : r_j = (j-1)h$ with $j \in \mathbb{N}$ and $\varphi_k = 2\pi(k-1)/p$ with $k \in \{1, .., p\}\}$ and the operators $D_1 = D_2 = R^{2\pi/p}$. Obviously the first and the second condition of theorem 2.2 are fulfilled. To check the last condition of theorem 2.2 we take a look at the filtered backprojection which is the standard inversion method for the Radon Transform. With the help of a selected filter ψ one calculates

$$f(x) = \int_{S^1} \int_{\mathbb{R}} \psi(\langle x, \omega\rangle - s)\ g(\omega, s) ds d\omega.$$

The discretized version is implemented in two steps. First the inner integration is done by

$$\mathcal{B}(\omega_n, s_m) = \sum_{l\in\mathbb{Z}} \psi(s_m - s_l) g(\omega_n, s_l) \text{ with } s_m = mh \text{ and } m, l \in \mathbb{Z}$$

In the second step the outer integration is performed. This operation is called backprojection.

$$f(\tilde{x}) = \frac{2\pi}{p} \sum_{n=1}^{p} \mathcal{B}(\omega_n, \langle \tilde{x}, \omega_n\rangle).$$

Because the scalar product $\langle \tilde{x}, \omega_n\rangle$ does hardly ever hit a measurement point s_m an interpolation has to be done.

$$f(\tilde{x}) = \frac{2\pi}{p} \sum_{n=1}^{p} \sum_{m\in\mathbb{Z}} I(\langle \omega_n, \tilde{x}\rangle, s_m) \mathcal{B}(\omega_n, s_m)$$

with I being a weight function for the interpolation. So we obtain $c(\tilde{x}, (\omega_n, s_m)) = I(\langle \omega_n, \tilde{x}\rangle, s_m)$. Hence the last condition of theorem 2.2 is fulfilled too.
The idea of using the rotation invariance in a numerical algorithm was first engaged by Buck, Maisl, Reiter [1, 2]. They developed an algorithm for 3d computerized tomography which was much faster than all other reconstruction methods.
To see the advantages in detail we take a closer look on the 2d case. Here we use the special parameters:

N	number of reconstruction points $\tilde{x}$
p	number of projections $\omega_n = (\cos\varphi_n, \sin\varphi_n)$ and $\varphi_n = \frac{2\pi(n-1)}{p}$
$2q+1$	Number of detector elements
h	stepsize in s with $s_m = mh$ and $m \in \{-q, .., q\}$
R^α	Rotation around the origin with angle α

Because the invariance effects only the backprojection we concentrate on this step. The classical backprojection looks like this (see [3], [7]):

Algorithm 5.2 (Backprojection)
Evaluate for every reconstruction point $\tilde{x}$

$$f(\tilde{x}) = \frac{2\pi}{p} \sum_{n=1}^{p} ((1-\eta^{(n)}) g(\omega_n, s_{k^{(n)}}) + \eta^{(n)} g(\omega_n, s_{k^{(n)}+1}))$$

with

$$\begin{aligned} s^{(n)} &= \langle \tilde{x}, \omega_n \rangle \\ k^{(n)} &= \left\lfloor \frac{s^{(n)}}{h} \right\rfloor \\ \eta^{(n)} &= \frac{s^{(n)}}{h} - k^{(n)} \end{aligned}$$

The fast version is now:

Algorithm 5.3 (Fast Version)

(i) Precomputation

- *Evaluate for all reconstruction points* $\tilde{x}$ *and only one projection* ω_1

$$\begin{aligned} s^{(\tilde{x})} &= \langle \tilde{x}, \omega_1 \rangle \\ k^{(\tilde{x})} &= \left\lfloor \frac{s^{(\tilde{x})}}{h} \right\rfloor \\ \eta^{(\tilde{x})} &= \frac{s^{(\tilde{x})}}{h} - k^{(\tilde{x})} \end{aligned}$$

(ii) Main part

- *calculate*

$$\begin{aligned} f(\tilde{x}) = \frac{2\pi}{p} \sum_{n=1}^{p} \Big(\Big(1 - \eta^{(R^{-\varphi_n}\tilde{x})} \Big) \, g\left(\omega_n, s_{k^{(R^{-\varphi_n}\tilde{x})}}\right) \\ + \eta^{(R^{-\varphi_n}\tilde{x})} \, g\left(\omega_n, s_{k^{(R^{-\varphi_n}\tilde{x})}+1}\right) \Big) \end{aligned}$$

(iii) Postcomputation

- *perform interpolation on cartesian grid*

We see that the complexity of the classical backprojection (only multiplication counted) is $6pN$ and of the fast version $2pN$. This means using the invariances results in an algorithm that is 3 times faster. Taking into account that the backprojetion consumes more than 80% of the whole reconstruction algorithm in computerized tomography we have now an algorithm that is more than twice as fast.
This issue can be stated numerically. In our test we use a modified Shepp-Logan phantom in parallel geometry. We select the parameters

$N = 512^2$	number of reconstruction points $\tilde{x}$
$p = 128$	number of projections $\omega_n = (\cos\varphi_n, \sin\varphi_n)$ and $\varphi_n = \frac{2\pi(n-1)}{p}$
$2q+1 = 513$	number of detector elements
$h = 1/256$	stepsize in s on the detector with $s_m = mh$ $m \in \{-q, .., q\}$

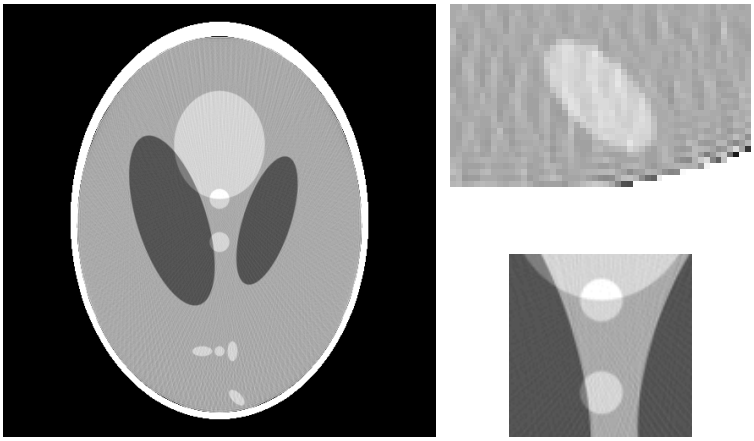

Figure 7. Reconstructions with Standard FBP
Left: total reconstruction ; Top right: magnification of the lower part ; Bottom right: magnification of the center

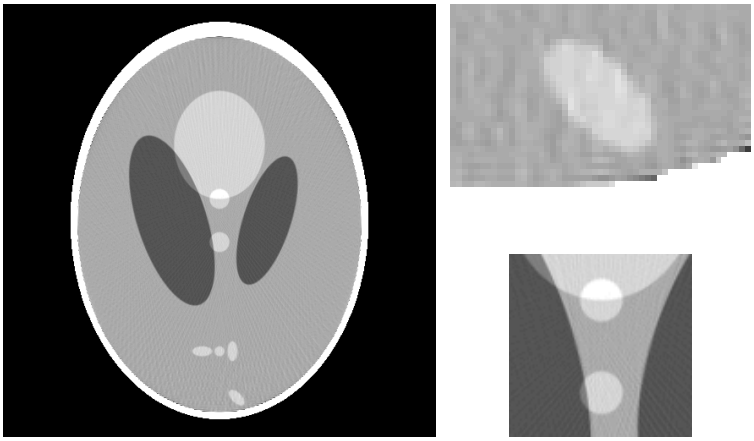

Figure 8. Reconstructions with Improved FBP
Left: total reconstruction ; Top right: magnification of the lower part ; Bottom right: magnification of the center

First we perform the reconstruction with the standard filtered backprojection (FBP). After that we replace the backprojection by the fast backprojection on an equidistant polar grid with $N_r = 256$ and $N_\theta = 768$ (total number of points: 196608). With an ordinary desktop pc we get the following computation times.

	Filtering	Backprojection	Interpolation	Total
Standard FBP	3.82 s	81.33 s	0.00 s	85.15 s
Improved FBP	3.76 s	29.65 s	0.41 s	33.83 s

We notice that the improved FBP needs approximatly 40% of the standard FBP. However the image of the improved reconstruction algorithm see figure 8 is not very good especially in the lower part of the phantom but in the above reconstruction we did not take into account the results of section 4. In section 4 we noticed that method 2 is

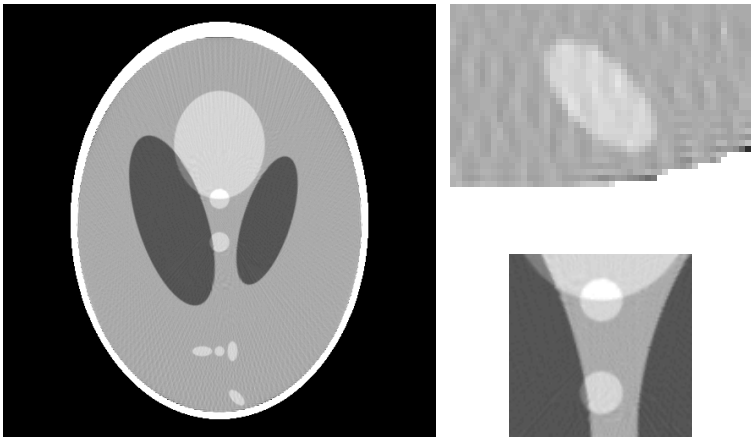

Figure 9. Reconstructions with Second Improved FBP
Left: total reconstruction ; Top right: magnification of the lower part ; Bottom right: magnification of the center

a good choice for a polar grid. But to apply this technique here we have to ensure that the discretisation number of the angle $N_\theta(r_j)$ is a multiple of p or $N_\theta(r_j)$ is a divisor of p. So we have to modify method 2.

Method 4 (Modified Angle) *Let N_r and $N_\theta(r_j)$ be chosen like in method 2. The grid points are given by*

$$(r_j, \theta_k) = ((j-1)h_r, (k-1)h_\theta(r_j))$$

with $h_r = 1/(N_r - 1)$, $j \in \{1, .., N_r\}$ and $h_\theta(r_j) = 2\pi/\tilde{N}_\theta(r_j)$, $k \in \{1, .., \tilde{N}_\theta(r_j)\}$. Here we define $\tilde{N}_\theta(r_j)$ such that $\tilde{N}_\theta(r_j)$ is a multiple or divisor of p and

$$|\tilde{N}_\theta(r_j) - N_\theta(r_j)| = \min\{|N_\theta(r_j) - m| : m \text{ is a multiple or divisor of } p\}$$

Together with this new method we create a second improved FBP. Now we take $N_r = 248$ and $c = 0.1$ (total number of points: 196979). The computation time has now slightly increased but on the other hand the quality of the reconstruction is much better see figure 9.

	Filtering	Backprojection	Interpolation	Total
Second Improved FBP	3.80 s	34.62 s	0.40 s	38.82 s

6. Conclusion

In this paper we took a general look on problems which have certain invariances and discussed whether these invariances can be used for a numerical implementation.
We noticed that rotation invariance leads to polar grids which are difficult to visualize. So we presented three choices for polar grids. Our tests showed that method 2 which we derived from an error estimate is a good choice.
We applied this strategy to a backprojection algorithm and drastically improved

computation time. One has to keep in mind that this approach can only be done having equidistant angle measurements. The error emerging from the interpolation is very small.
The method of using invariances in tomography is not limited to the 2d parallel geometrie as we did in this paper. It is also possible to use it in other geometries like fan or 3d-cone beam provided that the projections are measured with equidistant angle.

References

[1] Buck J 1996 *Schnelles Rekonstruktionsverfahren für die 3D-Röntgen-Computertomographie in der Materialprüfung*, Dissertation, Saarbrücken
[2] Buck J, Maisl M, Reiter H 1995 *Entwicklung eines schnellen Rekonstruktionsverfahrens für die 3D-Röntgen-Computertomographie* Jahrestagung der DGZfP 1995, Aachen, DGZfP Berlin
[3] Louis A K 1989 *Inverse und schlecht gestellte Probleme*, Teubner
[4] Louis A K, Maaß P 1990 *A mollifier method for linear operator equations of the first kind* Inverse Problems 6, S. 427-439
[5] Louis A K 1996 *Approximate inverse for linear and some nonlinear problems* Inverse Problems 12 S. 175-190
[6] Louis A K 1997 *Application of the Approximate Inverse to 3D X-Ray CT and Ultrasound Tomography*, Inverse Problems in Medical Imaging and Nondestructive Testing, Springer-Verlag Wien, New York, Herausgeber: H.W. Engl, A.K. Louis, W. Rundell, S. 120-133,
[7] Natterer F 1986 *Mathematics of Computerized Tomography* Wiley and Teubner
[8] Natterer F, Wübbeling F 2001 *Mathematical Methods in Image Reconstruction* Siam
[9] Schuster T, Rieder A 2003 *The approximate inverse in action III: 3D-Doppler tomography*, preprint Universität des Saarlandes