

The Generalized Radon Transform: Sampling, Accuracy and Memory Considerations

Cris L. Luengo Hendriks^{*,1}, Michael van Ginkel²,
Piet W. Verbeek and Lucas J. van Vliet

*Quantitative Imaging Group, Delft University of Technology,
Lorentzweg 1, 2628 CJ Delft, The Netherlands*

Abstract

The generalized Radon (or Hough) transform is a well-known tool for detecting parameterized shapes in an image. The Radon transform is a mapping between the image space and a parameter space. The coordinates of a point in the latter correspond to the parameters of a shape in the image. The amplitude at that point corresponds to the amount of evidence for that shape. In this paper we discuss three important aspects of the Radon transform. The first aspect is discretization. Using concepts from sampling theory we derive a set of sampling criteria for the generalized Radon transform. The second aspect is accuracy. For the specific case of the Radon transform for spheres, we examine how well the location of the maxima matches the true parameters. We derive a correction term to reduce the bias in the estimated radii. The third aspect concerns a projection-based algorithm to reduce memory requirements.

Key words: Radon transform, Hough transform, parameterized shape detection, multi-dimensional image, error estimation, efficient implementation, image analysis.

* Corresponding author. Tel: +1-510-4865359; fax: +1-510-4865730; e-mail: cluengo@lbl.gov

¹ Current address: Life Science Division, Lawrence Berkeley National Laboratory, One Cyclotron Road, mail stop 84R171, Berkeley, CA 94720, USA

² Current address: Measurement Science, Unilever Research Colworth House, Sharnbrook, MK44 1LQ Bedford, United Kingdom

1 Introduction

One of the first stages in image analysis is the extraction of primitives, such as lines, edges, curves or simple textures, from an image. In this paper we focus on curve detection, or more precisely, *shape* detection. In three- and higher-dimensional spaces, manifolds (N -dimensional), such as a spherical membrane, are as interesting as curves (one-dimensional). In general we are interested in a given *family of shapes*. Our assumption is that the members of this family can be described by a set of parameters. The task, then, is to find the parameters corresponding to the best fitting member of the family of shapes. The standard method for detecting parameterized shapes is based on a family of transformations, which includes the Radon [1] and Hough [2] transforms.

The organization of the paper is as follows: we continue the introduction by briefly discussing shape representation, followed by a short tutorial overview of the different transformations. The main point of this overview is to show that the different transformations are in fact different manifestations of a single unifying transform as has been described earlier, with some minor differences, in [3–6]. To clarify the definitions and results obtained, we will use the detection of hollow hyper-spheres in D -dimensional space (circles in 2D) throughout the paper.

The transform we describe is continuous and should operate on continuous images, but to allow computer processing we must work with sampled images. Likewise, the transform should be discretized. This is one of the main points of the paper and is discussed in Section 2. To allow discretization we must, in most cases, replace the original transform by a regularized version. The consequences of the regularization are discussed briefly in Section 3. We study the particular case of a Radon transform for hollow spheres in more detail: the regularization in combination with a normalization for the surface area of the sphere leads to a bias in the estimated radius. This is the second contribution of the paper. The third contribution of the paper is a projection-based scheme to reduce the memory requirements of the transform-based approach and is described in Section 4. We verify our results by performing some experiments on the aforementioned hollow spheres.

1.1 Shape representation using generalized functions

Before proceeding, we introduce the following notation:

\vec{x}	The spatial coordinates
$I(\vec{x})$	The D -dimensional image containing the N -dimensional shapes
\vec{p}	The vector containing the parameters of the curve. Often a subset

of the parameters specifies the location of the shape. It is, therefore, sometimes convenient to write $\vec{p} = \{\vec{q}, \vec{x}_o\}$, with \vec{x}_o the location of the shape (the center of the sphere), and \vec{q} the remainder of the parameters (the radius of the sphere).

$c(\vec{p})$	A member of a class of shapes described by the parameter vector \vec{p} .
$\vec{c}(\vec{s}; \vec{p})$	The coordinates of a point belonging to the shape $c(\vec{p})$. The coordinates \vec{s} allow us to specify a specific point on the shape.
$\mathcal{C}(\vec{x}; \vec{p})$	A set of constraint functions that together define the shape. The number of constraint functions depends on the dimensionality of the shape: $D-N$ constraints are necessary to describe a N -dimensional shape. For a point that lies on the shape, all the constraint functions evaluate to zero: $\mathcal{C}_i(\vec{x}; \vec{p}) = 0$ for all i .
$C(\vec{p}, \vec{x})$	A kernel, also called template, that represents the shape given by \vec{p} as an image with spatial coordinates \vec{x} . We can model the image I as a sum of several of these templates.

Shapes can be described in different ways. The notation $\vec{c}(\vec{s}; \vec{p})$ represents the shape. For a circle in 2D centered at \vec{x}_o and with radius r this becomes

$$\vec{c}(\phi; \{r, \vec{x}_o\}) = \vec{x}_o + r \begin{pmatrix} \cos \phi \\ \sin \phi \end{pmatrix}, \quad (1)$$

with ϕ a free coordinate letting us specify an arbitrary point on the circle. Alternatively, a shape can be defined through the specification of a constraint; this is known as the implicit representation. In the case of a circle:

$$\mathcal{C}(\vec{x}; \{r, \vec{x}_o\}) = 0 \quad \text{with} \quad \mathcal{C}(\vec{x}; \{r, \vec{x}_o\}) = \|\vec{x} - \vec{x}_o\| - r. \quad (2)$$

Now recall that the shapes we are looking for are embedded in an image and not directly available as a set of points. This means that standard results from differential geometry, such as the expression for the curvature of a plane curve [7]

$$\kappa = \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{(\dot{x}^2 + \dot{y}^2)^{\frac{3}{2}}} \quad (3)$$

are not directly applicable. In this example, the curvature of a curve embedded in an image in the form of an isophote can still be obtained through the well-known result for the isophote curvature [8–10].

In general, however, it may be beneficial to make the embedding of the shapes explicit. The basic ingredients for such a description are the constraint-based description and the Dirac delta function. The theoretical basis for this description can be found in Gel'fand et al. [11, Chapter III, Section 1, “Generalized Functions Concentrated on Smooth Manifolds of Lower Dimension”], who give

a very lucid account of this subject matter. It is not our intention to give a complete exposition of this material; we will merely touch upon the essentials.

Consider an N -dimensional shape in D -dimensional space. At any point on the shape we can span a local coordinate system. We will denote the local coordinate vector by \vec{u} . The first N coordinates $u_1 \dots u_N$ span the subspace in which the shape lies and the $D - N$ remaining coordinates $u_{(N+1)} \dots u_D$ span the subspace normal to the shape. In fact, these last coordinates act as constraint functions: we can set $\mathcal{C}_i = u_{i+N}$. We can describe the infinitesimal neighborhood I^n by

$$I^n(\vec{u}) = \delta(u_{(N+1)}, \dots, u_D). \quad (4)$$

If we choose an orthogonal coordinate system, then this reduces to

$$I^n(\vec{u}) = \prod_{i=N+1}^D \delta(u_i). \quad (5)$$

The scaling of the u_i should be such that they correspond to the Euclidean distance to the shape. This ensures that a) the points on the shape contribute equally in an integral and b) the overall scale is such that the D -dimensional volume integral over the image (assuming there is only the one shape) yields the N -dimensional hyper-volume of the shape. If the constraint functions are chosen according to these principles, we write

$$I(\vec{x}) = \delta(\mathcal{C}(\vec{x}; \vec{p})). \quad (6)$$

Let us examine two simple examples in three-dimensional space. The $x - y$ plane is described by the constraint $z = 0$. Therefore,

$$I_{[x-y \text{ plane}]}(x, y, z) = \delta(z), \quad (7)$$

represents a plane in the $x - y$ plane. A line along the x -axis is described by the constraint $y = 0$ and $z = 0$:

$$I_{[\text{line along } x]}(x, y, z) = \delta(y)\delta(z). \quad (8)$$

1.2 The Radon transform

The Radon transform is named after J. Radon who showed how to describe a function in terms of its (integral) projections [1]. The mapping from the function onto the projections is the Radon transform. The inverse Radon transform corresponds to the reconstruction of the function from the projections. The original formulation of the Radon transform is as follows:

$$\mathcal{R}\{I\}(d, \phi) = \int_{\mathbb{R}} I(d \cos \phi - s \sin \phi, d \sin \phi + s \cos \phi) ds, \quad (9)$$

with the projection along the lines $c_l(d, \phi)$ with the parameterization as given in Figure 1. Within the realm of image analysis, the Radon transform is mostly known for its role in computed tomography. It is used to model the process of acquiring projections of the original object using X-rays. Given the projection data, the inverse Radon transform, in whatever form (e.g. back-projection), can be applied to reconstruct the original object.

The Radon transform can also be used for shape detection. We reformulate the Radon transform:

$$\mathcal{R}\{I\}(d, \phi) = \int_{(x,y) \text{ on } c_l(d,\phi)} I(x, y) dx dy = \int_{\mathbb{R}^R} I(x, y) \delta(x \cos \phi + y \sin \phi - d) dx dy. \quad (10)$$

It is now trivial to generalize the Radon transform to arbitrary shapes $c(\vec{p})$. We give three equivalent formulations, leaving it to the reader to decide which is the easiest to interpret:

$$\mathcal{R}_{c(\vec{p})}\{I\}(\vec{p}) = \int_{\vec{x} \text{ on } c(\vec{p})} I(\vec{x}) d\vec{x} = \int_{\mathbb{R}^N} I(\vec{c}(\vec{s}; \vec{p})) \left\| \frac{\partial \vec{c}}{\partial \vec{s}} \right\| d\vec{s} = \int_{\mathbb{R}^D} I(\vec{x}) \delta(\mathcal{C}(\vec{x}; \vec{p})) d\vec{x}. \quad (11)$$

For our study of the discretization of the transform, the third formulation is the most convenient. The mathematical properties of this generalized form of the Radon transform have been extensively studied in [12].

Now imagine that there is a shape in the image with parameter set \vec{a} . When $\vec{p} \neq \vec{a}$, the Radon transform will evaluate to some finite number which is proportional to the number of intersections between the shapes $c(\vec{p})$ and $c(\vec{a})$, as illustrated in Figure 1. However, when $\vec{p} = \vec{a}$, the Radon transform yields a large response (a peak in the parameter space). This response is proportional to the N -dimensional hyper-volume of the shape. We can now interpret the Radon transform as follows: it provides a mapping from image space to a *parameter space* spanned by the parameters \vec{p} . The function created in this parameter space, $P(\vec{p})$, contains peaks for those \vec{p} for which the corresponding shape $c(\vec{p})$ is present in the image. Shape detection is reduced to the simpler problem of peak detection.

The third formulation of the Radon transform in equation (11) demonstrates an important reason for using generalized functions. In this notation, we can recognize the form of a linear integral operator³ \mathcal{L}_C with kernel C :

$$(\mathcal{L}_C I)(\vec{p}) = \int_{\mathbb{R}^D} C(\vec{p}, \vec{x}) I(\vec{x}) d\vec{x}. \quad (12)$$

Therefore, if we allow the kernel C to be a generalized function, then we can express the Radon transform in this format, which is particularly convenient

³ This is known as a Fredholm operator.

to study its discretization. In case of a Radon transform, the kernel C is of the form: $C(\vec{p}, \vec{x}) = \delta(\mathcal{C}(\vec{x}; \vec{p}))$. In terms of shape detection, the role of the operator \mathcal{L}_C is to compute the match (the inner product) between the image and a template C for a given parameter set \vec{p} . Here we see the connection between the Radon transform and template matching.

Often, the parameters \vec{p} consist of the position of the shape \vec{x}_o and the actual shape parameters \vec{q} . In this case the kernel has a special (shift-invariant) structure:

$$C(\{\vec{q}, \vec{x}_o\}, \vec{x}) = C(\{\vec{q}, \vec{x}_o + \vec{d}\}, \vec{x} + \vec{d}) \quad \text{for any } \vec{d}. \quad (13)$$

The operator \mathcal{L}_C now reduces to a set of convolutions:

$$(\mathcal{L}_C I)(\vec{q}, \vec{x}_o) = (K_C(\vec{q}) *_{\vec{x}} I)(\vec{x}_o) \quad \text{with} \quad K_C(\vec{q}, \vec{x}) = C(\{\vec{q}, \vec{x}\}, \vec{0}). \quad (14)$$

This implies a large speed-up: using the convolution property of the Fourier transform, each convolution reduces to a multiplication in the Fourier domain.

Use of the Radon transform for shape detection dates back to 1965 [13]. The technique these authors describe is essentially a Radon transform. Rosenfeld [14] describes this technique (for straight lines) in Section 8.4.e “Coordinate Conversion”. Neither [13] nor [14] identify this technique as the Radon transform. Our study of the discretization of the Radon transform in the next section also applies to the popular Hough transform which is often used for shape detection. The Hough transform itself can be considered a particular discretization of the Radon transform [15,3–5].

2 Sampling the Radon transform

The errors produced by discrete versions of these transforms have been investigated by several authors, most notably in the context of the Hough transform. Here, we will investigate the effects of sampling on the continuous formulation of the Radon/Hough transform and whether these may be avoided.

It is important to discriminate between two models for the input data. In the first model [16,6] the data consists of a set of points on a continuous input space:

$$I(\vec{x}) = \sum_i \delta(\vec{x} - \vec{x}_i). \quad (15)$$

In this model, noise corresponds to perturbations in the position of the points. Given this model, the conditions under which $P(\vec{p})$ can be sampled were given by Kiryati and Bruckstein [16].

In our work we consider a different model. In most image analysis problems we start out with a continuous gray value image, which is subsequently sampled

for computer processing. Assuming the image is band-limited and properly sampled according to the Nyquist criterion [17], the sampled image represents the continuous function perfectly: the original can be reconstructed from the sampled image. The sampled image still contains all the subtleties of the continuous original, including sub-pixel information. Curves embedded in the image are faithfully represented. Sethian [18] has used this property to implement curve evolution schemes on a fixed grid, embedding the curves as zero-crossings in a sampled gray value image.

A large class of images is band-limited by virtue of the image acquisition process, for example those obtained using a diffraction-limited optical system. If, however, the image is not band-limited, such as input data of the form given in (15), the image must be low-pass filtered to make it band-limited, implying a certain loss of resolution.

We now arrive at one of the key points of the paper: it is not sufficient that the sampled input image is a faithful representation. The discrete processing and the result thereof must be faithful to their original continuous counterparts as well: the continuous operation and the chain of operations consisting of sampling, a discrete operation, and reconstruction must be identical. An operation for which this is possible, is said to be *sampling invariant*, a concept introduced by Verbeek [9] and van Vliet [19]. They have considered convolutions and multiplicative combinations of convolutions.

Here we follow van Ginkel [20] and study the conditions under which equation (12) is sampling-invariant. There are two aspects. Keeping \vec{p} fixed, we will first consider under which conditions we may replace I and C by sampled (along \vec{x}) versions and the integral by a summation. If these conditions are satisfied, we may compute $P(\vec{p})$ for an arbitrarily chosen \vec{p} . We must then show that it is possible to sample the parameter space $P(\vec{p})$, so that we only need to evaluate $P(\vec{p})$ on a discrete set of points. For simplicity we restrict ourselves to a one-dimensional example: $\vec{p} \rightarrow p$ and $\vec{x} \rightarrow x$. The Fourier axes corresponding to p and x are denoted by \tilde{p} and \tilde{x} respectively. The sampling distance along x is Δx . The discrete coordinate corresponding to x is n , i.e. the sampled version of $I(x)$ is $I(n \Delta x)$. We first investigate under which conditions the following is true:

$$P(p) = \int_{\mathbb{R}} C(p, x) I(x) dx = \Delta x \sum_{n \in \mathbb{Z}} C(p, n \Delta x) I(n \Delta x) . \quad (16)$$

We denote the band-limit (along x) of the product $C(p, x) I(x)$ by $b_x\{CI\}$. With p fixed, the sampling criterion for the computation of this integral is a relaxed version of the Nyquist criterion [9],

$$\tilde{x} > b_x\{CI\} . \quad (17)$$

Also the band-limit of CI can be expressed in that of C and I ,

$$b_x\{CI\} \leq b_x\{C\} + b_x\{I\} . \quad (18)$$

It follows that both the kernel C and the image I must be band-limited to allow discretization. Proper sampling of the image I is a prerequisite for any image analysis and therefore introduces no new restrictions. This is not true for the kernel C , which in general is not band-limited. We must impose a bandwidth limitation on C . This clearly leads to a different Radon transform, but reflects a conscious choice with well-understood consequences. These will be discussed in Section 2.1. The alternative, sampling C without imposing a band-limit first, leads to aliasing effects.

We can compute $P(p)$ for an arbitrary value of p . If $P(p)$ is band-limited, it can be safely sampled, provided the correct (Nyquist) rate is used. We determine whether $P(p)$ is band-limited by computing its Fourier transform:

$$\mathcal{F}\{P\}(\tilde{p}) = \mathcal{F}\left\{\int_{\mathbb{R}} C(p, x)I(x)dx\right\}(\tilde{p}) = \int_{\mathbb{R}} \mathcal{F}_p\{C(p, x)\}(\tilde{p}, x)I(x)dx . \quad (19)$$

If C is band-limited along the p -axis with band-limit $b_p\{C\}$, then the integral above evaluates to zero for $\tilde{p} > b_p\{C\}$, which means that $P(p)$ is band-limited as well.

The discussion above also holds for the complete multi-dimensional operation: our argument holds for each spatial dimension x_i separately and for each parameter dimension p_j as well. The same ideas also extend trivially to other sampling schemes, such as the hexagonal grid.

2.1 Band-limiting the kernel $C(\vec{p}, \vec{x})$

The Gaussian filter is approximately band-limited with critical sample spacing σ [19] and corresponding band-limit $b = \frac{1}{2}\sigma^{-1}$. Its properties, in particular good simultaneous frequency and spatial localization [21], and not introducing new structure [22], make it a good choice for band-limiting $C(\vec{p}, \vec{x})$. We obtain C_b , a band-limited version of C , as follows:

$$C_b(\vec{p}, \vec{x}) = C(\vec{p}, \vec{x}) * G(\vec{p}, \vec{x}; \Sigma) . \quad (20)$$

The diagonal covariance matrix Σ reflects that we impose band-limitation along each dimension separately.

By its nature the function $C(\vec{p}, \vec{x}) = \delta(\mathcal{C}(\vec{x}; \vec{p}))$ is in general very sparse: for any given \vec{p} , the points \vec{x} which belong to the shape span some curve or manifold in $C(\vec{p}, \vec{x})$. The Radon transform for hyper-spheres provides a convenient example

to investigate the structure of $C(\vec{p}, \vec{x})$ and the effects of band-limitation. The parameter vector \vec{p} consists of the center \vec{x}_o of the D -dimensional sphere and its radius r : $\vec{p} = (x_1, \dots, x_D, r)$. The kernel C becomes

$$C(\{r, \vec{x}_o\}, \vec{x}) = K(r, \vec{x}_o - \vec{x}) \quad \text{with} \quad K(r, \vec{\xi}) = \delta\left(\frac{1}{2}\sqrt{2}(\|\vec{\xi}\| - r)\right). \quad (21)$$

The function K represents a cone. Each point on the surface of this cone should have equal weight as discussed in Section 1.1. The cone has a single coordinate u normal to the cone, see Figure 2. This coordinate is chosen such that it corresponds to the Euclidean distance to the surface: $u = \frac{1}{2}\sqrt{2}(\|\vec{\xi}\| - r)$. In Section 3 we discuss some issues that lead to a different choice for the normalization.

What is the effect on K of the Gaussian smoothing applied to C ? Let us first consider the effect of the smoothing applied along the \vec{p} -axes. All parameters share the same units and it is therefore logical to use the same σ_K along each dimension. The effect on a local surface patch, if it can be considered planar locally ($\sigma_K \ll r$), is that the Dirac profile is substituted by a Gaussian profile

$$K_b(r, \vec{x}_o - \vec{x}; \sigma_K) = K(r, \vec{x}_o - \vec{x}) *_{\vec{x}_o, r} G(\vec{x}_o, r; \sigma_K) \approx G\left(\frac{1}{2}\sqrt{2}(\|\vec{x}_o - \vec{x}\| - r); \sigma_K\right) = \sqrt{2} G(\|\vec{x}_o - \vec{x}\| - r; \sqrt{2} \sigma_K). \quad (22)$$

The next step is to apply a Gaussian along the \vec{x} -axes. Because the structure of K along axes \vec{x}_o and \vec{x} is not independent, the Gaussian smoothing along \vec{x}_o implies a Gaussian smoothing along \vec{x} . This is evident from (22). Therefore it is unnecessary to apply the Gaussian smoothing along the \vec{x} -axes, unless the required smoothing σ_s along the \vec{x} -axes is larger than that required along the \vec{p} -axes. A consequence is that applying the regularization in parameter space only, as proposed in [16], is in this case sufficient to avoid discretization errors.

On a rectangular grid, the maximum frequency allowed depends on the orientation of the signal component with respect to the Cartesian axes (in 2D up to $\sqrt{2}$ times the highest frequency along the Cartesian axes). Given that the cone has a 45° angle with respect to the Cartesian axes in $\|\vec{x}\| - r$ space, we can relax the Gaussian regularization reducing its size by a factor of $\sqrt{2}$:

$$K_b(r, \vec{x}; \sigma_K) = \sqrt{2} G(\|\vec{x}\| - r; \sigma_K). \quad (23)$$

The consequences of the imposed band-limit are as follows: as long as the Gaussian is small with respect to the curvature of C , the effects of the Gaussian are negligible. In fact, it is possible to interpolate in \vec{p} -space and obtain sub-pixel accuracy. High-curvature structures of C correspond either to highly curved shapes or to shapes which vary rapidly as a function of the parameters with respect to the sampling rate chosen for these parameters. In neither case is it reasonable to expect good results anyway. The band-limitation does lead

to a bias in the estimated radius. This effect and how it can be compensated for using a normalization term is the topic of Section 3.

2.2 A perfect discretization?

Under some circumstances it is possible to avoid discretization errors altogether. We first consider the convolution case [23, Section 8.4, “Discrete-time processing of continuous-time signals”]. Let $l(x)$ be an ideal low-pass filter with a cut-off frequency that corresponds to the band-limit of the image $I(x)$, and $f(x)$ the filter to be sampled. Applying $l(x)$ to $I(x)$ has no effect: $I(x) * l(x) \equiv I(x)$. This leads to:

$$I(x) * f(x) = [I(x) * l(x)] * f(x) = I(x) * [l(x) * f(x)], \quad (24)$$

both $I(x)$ and the term between the brackets are band-limited and can be sampled.

This principle is also applicable to equation (12), but only along the \vec{x} dimensions. Only in special cases, such that of the hyper-spheres, is this sufficient. In the general case it remains necessary to impose a band-limit along the parameter axes.

Ideal low-pass filters have some undesirable properties; ringing artifacts being the most important. In practice, the effects of using a Gaussian filter, optionally oversampling first, instead of an ideal low-pass filter should be limited.

3 Accuracy of the parameter space

Due to the definition of the function $K_b(r, \vec{x}; \sigma_K)$ in (22), large hyper-spheres will produce a higher value (higher confidence) than small ones in the parameter space. The consequence is two-fold:

- a few disconnected sections in the input will be selected as a large hyper-sphere with a higher confidence than a smaller but complete hyper-sphere, and
- the radii of hyper-spheres with thick walls will be over-estimated.

To avoid this, the spheres that compose $K_b(r, \vec{x}; \sigma_K)$ should be normalized. That is, the integral of their gray-values should remain constant for any r . This results in

$$K_b(r, \vec{x}; \sigma_K) = \frac{1}{S_D(r)} G(\|\vec{x}\| - r; \sigma_K), \quad (25)$$

where $S_D(r) = Qr^{-(D-1)}$ is the surface area of a D -dimensional hyper-sphere of radius r , and Q is a constant that depends on D .

This normalization causes a distortion of the shape of the cone, which in turn leads to an underestimation of the radius. We showed in [24] that the error in the position of a maximum in $P(\vec{p})$ along the r -axis, assuming $r \gg \sigma_K$, is dominated by the position of the maximum along the r -axis of $K_b(r, \vec{x}; \sigma_K)$. To correctly estimate the radius of hyper-spheres, this maximum should lie at $\|\vec{x}\|$, but is shifted to $\|\vec{x}\| + \epsilon(\|\vec{x}\|)$ by the normalization (see Figure 3). We find the position of the sifted maximum by equating the derivative of Equation (25) to zero,

$$\frac{\partial K_b}{\partial r} = Q \left[\frac{-(D-1)}{r^D} + \frac{\|\vec{x}\| - r}{\sigma_K^2 r^{D-1}} \right] G(\|\vec{x}\| - r; \sigma_K) = 0, \quad (26)$$

$$-(D-1)\sigma_K^2 - (\|\vec{x}\| - r)r = 0, \quad (27)$$

and solving for $\epsilon = r - \|\vec{x}\|$, which yields

$$\epsilon(\|\vec{x}\|) \approx -\frac{(D-1)\sigma_K^2}{\|\vec{x}\|} - \frac{(D-1)^2\sigma_K^4}{\|\vec{x}\|^3} + O(\|\vec{x}\|^{-5}). \quad (28)$$

We will be using only the first term of this equation. As shown in [24], when taking into account the shapes in the image a new term of $O(\|\vec{x}\|^{-3})$ should be added. However, it is not possible to correct for it since no assumptions about the image can be made.

Thus, the measured radius, R_m , is given by $R_m = \|\vec{x}\| + \epsilon(\|\vec{x}\|)$. We substitute $\|\vec{x}\|$ by R_c , which we then can change such that R_m becomes equal to $\|\vec{x}\|$. R_c is the radius we use for the creation of the convolution kernel $K_b(r, \vec{x}; \sigma_K)$ at r .

$$\|\vec{x}\| = R_c - \epsilon(R_c) \implies R_c = \frac{1}{2}\|\vec{x}\| + \sqrt{\frac{1}{4}\|\vec{x}\|^2 - (D-1)\sigma_K^2}. \quad (29)$$

$$K_b(r, \vec{x}; \sigma_K) = \frac{1}{S_D(r)} G(\|\vec{x}\| - R_c(r); \sigma_K). \quad (30)$$

4 Reducing memory requirements

The parameter space for the Radon transform typically has more dimensions than the input image. This implies that these parameter spaces might not fit into the available computer memory. This constraint has traditionally prevented wide-spread use of these transforms for 3D images.

Many authors have tackled this problem in a variety of manners. Most notably, Ballard and Sabbah [25] propose to partition the parameter space into two or more spaces with independent parameters, which can be computed sequentially. For example, it is possible to first locate all the object centers, and in a second stage determine the sizes. For (hollow) spheres, this requires the use of gradient information to locate centers [26], and is only practical for very few objects due to the cost of the second stage. Atherton and Kerbyson [27] encode the radius as the phase in the complex parameter space, thereby simplifying this approach. Hsu and Huang [28] also use a dimensionality reduction to detect 3D ellipsoids (with 6 parameters, the axes are supposed to lie on the grid). They split the 6D parameter space into two 4D parameter spaces, which have to be combined to find the objects.

Another method often employed involves splitting the parameter space into overlapping regions, from which the maxima are extracted. This does not involve a reduction of dimensionality, but incurs a penalty in computational cost because of the overlap. In the case of a sphere, it is natural to split the parameter space along the r -axis, since a slice $P(r_i, \vec{x}_o)$ is computed by a single convolution. We will call this method the Sliding Window method (SW).

We propose a different approach to reduce the memory requirements. Spheres can be detected very efficiently by storing only the maximum projection along the r -axis of P , together with the location of these values on the r -axis (if one is prepared to ignore concentric spheres). That is, we keep

$$S(\vec{x}_o) = \max_r \{P(r, \vec{x}_o)\} \quad (31)$$

and

$$R(\vec{x}_o) = \arg \max_r \{P(r, \vec{x}_o)\} . \quad (32)$$

The local maxima in $S(\vec{x}_o)$ indicate the location of the center of the spheres, and $R(\vec{x}_o)$ gives the corresponding radii. Both of these can be computed by a small modification of the Radon algorithm. Instead of storing all the $P(r_i, \vec{x}_o)$ slices, we propose to take the point-wise maximum of each slice with the previously computed intermediate result. This does not add any computational cost to the algorithm, since finding the local maxima needs to be done anyway. This maximum projection even simplifies this task. We call this method the Maximum Projection method (MP), and is both faster and much less memory-hungry than the SW method.

The resulting parameter space $S(\vec{x}_o)$ is not band-limited. But, if the spheres are clearly identifiable and well separated, it turns out to have nicely-shaped peaks (i.e. the neighborhoods of the local maxima are band-limited or nearly so). Thus, it is still possible to obtain the center of the sphere with sub-pixel accuracy. However, the r -axis at each position has been discretized to sampling locations. The accuracy to which r can be estimated depends on the number

of samples taken, not the band-limit of $K_b(r, \vec{x}_o; \sigma_K)$ along the r -axis.

It is possible to implement such a Radon transform for other shapes as well, in which case the maximum projection can be taken over more than one dimension. That is, only the spatial dimensions need to be kept, all other dimensions can be collapsed into one maximum image and one or more maximum position images, of which there are as many as parameter dimensions are reduced.

5 Results

5.1 Evaluation

To demonstrate the claims made in the previous sections, we computed the Radon transform of 25 synthetically generated, 3D test images, 128^3 pixels in size, each containing 20 spheres of different radii (between 6 and 18 pixels) at random, sub-pixel locations. Some of the spheres were touching, but none were overlapping. These spheres had a Gaussian profile (with $\sigma_i = 1$), thereby approximating band-limitness. We computed the Radon transform with the two methods explained above (SW and MP), using the normalized kernel of Equation (25) and the corrected kernel of Equation (30) (setting $\sigma_K = 1$). To determine the position of the peaks with sub-pixel accuracy, we used a cubic spline interpolator to resample the immediate vicinity of the maximum with a 100-fold pixel density along each of the axes. The SW method uses a window of 7 slices in the radius direction, from which 2 slices overlap the previous region. It required five times as much memory, and took about twice as much time to finish, as compared to the MP method. Apparently the algorithm we used to find the local maxima is relatively expensive compared to the convolutions themselves.

We evaluated both methods by computing the differences between the true parameters of the spheres and the estimated ones. Table 1 compares the errors in the location for the different methods. We found that both SW and MP found the location of the spheres with the same accuracy (for any given sphere the parameters found by the two methods are almost identical; differences are in the order of 2%). The bias is very small, not significant in relation to the standard deviation. Figure 4 shows the errors in the estimated radii for the SW method with and without correcting the kernel for bias. The MP method found the rounded values of the radii found by the SW method.

To examine the influence of noise, we added Gaussian noise to the images used above and repeated the experiments; the results are also shown in Table 1 and Figure 4. The input images had a SNR of 2 (with $\text{SNR} = \frac{\max I(\vec{x}) - \min I(\vec{x})}{\sigma_N}$, $I(\vec{x})$

the uncorrupted image and σ_N the standard deviation of the noise probability density distribution). The standard deviation in the errors do not increase much for this noise level. This shows that the projection method is a good approximation with or without noise, and shows that the Radon transform itself is robust with respect to noise.

The graphs in Figure 5 are computed without the use of interpolation (by densely sampling along the r -axis), and show the attainable accuracy in the estimation of the radius with and without normalization and/or correction. They also show the effect of the width σ_K of the probes and the width σ_i of the spheres in the image. The non-normalized method is heavily influenced by σ_i , but not at all by σ_K . Normalizing the kernel makes the transform almost independent of σ_i , but it becomes dependent on σ_K . The bias correction proposed in Equation (30) removes this dependency, but the assumption made ($r \gg \sigma$) breaks down for smaller radii and larger σ .

5.2 Ballotini

As a demonstration application, we used a rather poor-quality X-ray micro-CT image of ballotini (small, hollow glass beads, see Figures 6a and 7a). Some of the glass walls give a very wide response in the imager (probably caused by refraction or reflection). In one such region many small spheres can be fitted. To avoid this, we replaced the kernel K_b by a kernel K'_b that penalizes for high gray-values inside the sphere:

$$K'_b(r, \vec{x}_o; \sigma_K) = K_b(r, \vec{x}_o; \sigma_K) - K_b(r - d, \vec{x}_o; \sigma_K) , \quad (33)$$

with K_b the corrected kernel as given in Equation (30) and d the difference in radius. By requiring that the inner part of the sphere be empty, the discriminating abilities of the transform (for these images) are greatly enhanced (compare image b to e in Figures 6 and 7). The computational complexity remains the same. We set $d = 4\sigma_K$, such that the sphere $K_b(r, \vec{x}_o; \sigma_K)$ is not affected too much (see Figure 8), since that would cause a large underestimation of the radius. In the synthetic test images of the evaluation we performed earlier, this setting leads to a slightly larger bias: an underestimation of about 0.1 pixel for large spheres, see Figure 9. This bias might be corrected for in the same manner as before.

To find the spheres in the parameter space $S(\vec{x}_o)$, a threshold is used to decide which local maxima are important enough to represent a sphere in the input images. More complex decision rules could be used, but are outside the scope of this paper. Figures 6 and 7 show the results for two different slices of the 3D image.

6 Conclusions

We have given the conditions under which the Radon transform can be computed free of discretization errors. Assuming that the input image is correctly sampled, these conditions are met by imposing a band-limit on the operator function C . This has no consequences for sufficiently smooth shapes. The parameter space that results is band-limited, which allows sampling and interpolation, and thus sub-pixel accuracy in the estimated parameters.

To avoid a larger weight being assigned to larger shapes, the convolution kernel should be normalized. We studied the effect of this normalization in the case of the Radon transform for spheres, and propose a way of correcting for the bias this introduces in the estimated radius.

The Radon transform reduces to a convolution for position-type parameters, yielding a large speed-up. We propose a memory-efficient implementation, computing (through convolution) a single r -slice of $P(r, \vec{x}_o)$ at a time. We keep track of the maximum projection and the argument-maximum projection along the r -axis as we compute the slices. We argue that this approach can be used for other shapes as well.

We have applied this modified Radon transform to a 3D image of glass hollow beads. To compute its parameter space we have employed a convolution kernel that contains not only a sphere, but also a second, smaller, concentric sphere with negative gray-values. The resulting parameter space has a much higher discriminating ability than that which would result from the same transform with a single sphere.

Acknowledgements

The authors wish to thank Scott Singleton and Dave Rowlands at Unilever Research and Development Colworth House (Bedfordshire, UK), for their hospitality and permission to use the ballotini image in this paper. This work was partially supported by the Dutch Ministry of Economic Affairs through their IOP program and by Unilever Research and Development Vlaardingen (The Netherlands).

References

- [1] J. Radon. Über die Bestimmung von Funktionen durch ihre Integralwerte längs gewisser Mannigfaltigkeiten. *Berichte Sächsische Akademie der*

- Wissenschaften, Leipzig, Mathematisch-Physikalische Klasse*, 69:262–277, 1917.
- [2] P.V.C. Hough. Method and means for recognizing complex patterns. US patent nr. 3069654, 1962.
 - [3] G.C. Stockman and A.K. Agrawala. Equivalence of Hough curve detection to template matching. *Communications of the ACM*, 20(11):820–822, 1977.
 - [4] J. Sklansky. On the Hough technique for curve detection. *IEEE Transactions on Computers*, 27(10):923–926, October 1978.
 - [5] S.R. Deans. Hough transform from the Radon transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(2):185–188, March 1981.
 - [6] J. Princen, J. Illingworth, and J. Kittler. A formal definition of the Hough transform: Properties and relationships. *Journal of Mathematical Imaging and Vision*, 1:153–168, 1992.
 - [7] J.J. Stoker. *Differential Geometry*. John Wiley and Sons, Inc, 1969.
 - [8] L. Kitchen and A. Rosenfeld. Gray-level corner detection. *Pattern Recognition Letters*, 1(2):95–102, 1982.
 - [9] P.W. Verbeek. A class of sampling-error free measures in oversampled band-limited images. *Pattern Recognition Letters*, 3:287–292, 1985.
 - [10] J.J. Koenderink and W. Richards. Two-dimensional curvature operators. *Journal of the Optical Society of America A*, 5(7):1136–1141, July 1988.
 - [11] I.M. Gel’fand and G.E. Shilov. *Generalized Functions. Volume 1, Properties and Operations*. Academic Press, 1964.
 - [12] I.M. Gel’fand, M.I. Graev, and N.Ya. Vilenkin. *Generalized Functions. Volume 5, Integral Geometry and Representation Theory*. Academic Press, 1966.
 - [13] M.J. Bazin and J.W. Benoit. Off-line global approach to pattern recognition for bubble chamber pictures. *IEEE Transactions on Nuclear Science*, 12:291–295, August 1965.
 - [14] A. Rosenfeld. *Picture Processing by Computer*. Academic Press, 1969.
 - [15] M. van Ginkel, C.L. Luengo Hendriks, and L.J. van Vliet. An short introduction to the radon and hough transforms and how they relate to each other. Technical report, Pattern Recognition Group, Delft University of Technology, December 2003. Technical Report PH-2003-03.
 - [16] N. Kiryati and A.M. Bruckstein. Antialiasing the Hough transform. *CVGIP: Graphical Models and Image Processing*, 53(3):213–222, May 1991.
 - [17] R.N. Bracewell. *The Fourier Transform and its Applications*. McGraw-Hill, third edition, 2000,1965.
 - [18] J.A. Sethian. Curvature and the evolution of fronts. *Communications in Mathematical Physics*, 101:487–499, 1985.

- [19] L.J. van Vliet. *Grey-Scale Measurements in Multi-Dimensional Digitized Images*. PhD thesis, Delft University of Technology, Delft, The Netherlands, October 1993.
- [20] M. van Ginkel. *Image Analysis using Orientation Space based on Steerable Filters*. PhD thesis, Delft University of Technology, Delft, The Netherlands, 2002. digital version available:
http://www.ph.tn.tudelft.nl/PHDTheses/MvGinkel/thesis_vanginkel.html.
- [21] A. Papoulis. *Signal Analysis*. McGraw-Hill, 1977.
- [22] J.J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–370, 1984.
- [23] A.V. Oppenheim, A.S. Willsky, and I.T. Young. *Signals and Systems*. Prentice-Hall International, London, 1983.
- [24] C.L. Luengo Hendriks, M. van Ginkel, and L.J. van Vliet. Underestimation of the radius in the radon transform for circles and spheres. Technical report, Pattern Recognition Group, Delft University of Technology, May 2003. Technical Report PH-2003-02.
- [25] D.H. Ballard and D. Sabbah. Viewer independent shape recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(6):653–660, 1983.
- [26] M. van der Glas, F.M. Vos, and A.M. Vossepoel. Assessment of center of rotation of the glenohumeral joint by a modified Hough transform. In W.J. Niessen and M.A. Viergever, editors, *4th International Conference on Medical Image Computing and Computer-Assisted Intervention - MICCAI 2001, Utrecht, The Netherlands*, volume 2208 of *Lecture Notes on Computer Science*, 2001.
- [27] T.J. Atherton and D.J. Kerbyson. Using phase to represent radius in the coherent circle Hough transform. IEE Colloquium on Hough Transforms, 1993.
- [28] C.-C. Hsu and J.S. Huang. Partitioned Hough transform for ellipsoid detection. *Pattern Recognition*, 23:275–282, 1990.

About the Author—CRIS LUENGO received his M.Sc. in 1998 and his Ph.D. in 2004, both from the Department of Applied Physics of the Delft University of Technology, the Netherlands. He is currently a visiting research fellow at Lawrence Berkeley National Laboratories (Berkeley, California), where he is working on a method to quantitatively analyze animal gene expression in 3D. His research interests within image analysis include feature detection, measurement, image enhancement, mathematical morphology, linear methods, and the link between the continuous and discrete domains.

About the Author—MICHAEL VAN GINKEL obtained his M.Sc. in 1995 and his Ph.D. in 2002, both in Applied Physics, at Delft University of Technology. His PhD thesis received the 2004 E.S. Gelsema award for the best thesis/paper in Pattern Recognition and Image Analysis in the Netherlands

over the period 2002-2003. His research focuses on the measurement and characterization of objects and structures in images with the goal of relating these to the physical properties of said structures. He is currently a Madame Curie fellow at Unilever Research Colworth.

About the Author—PIET VERBEEK studied physics at Leyden University, the Netherlands. His Ph.D. thesis (1973) was on quantum statistics and systems theory of magnetic relaxation. In 1973 he joined the Pattern Recognition Group of the Department of Applied Physics at the Delft University of Technology. Since 1974 he has worked on image processing. Some topics: 3D skeletonization (1978), cell nucleus texture analysis (1979), texture segmentation (1980), alpha-hull (1981), video speed range sensor system (since 1985), max-min filtering (1988), distance transform and robot collision avoidance (1986-91), measurement in 2D and 3D grey images (since 1985), texture analysis in 2D and 3D grey images (since 1994).

About the Author—LUCAS VAN VLIET was born in 1965. He received the M.Sc. degree in applied physics in 1988 and the Ph.D. degree (cum laude) in 1993. His dissertation entitled “Grey-scale measurements in multi-dimensional digitized images” presents novel methods for sampling-error free measurements of geometric object features. He is a Full Professor in multidimensional data analysis with the Faculty of Applied Sciences, Delft University of Technology, Delft, The Netherlands. He has worked on various sensor, restoration, and measurement problems in quantitative microscopy. His current research interests include segmentation and analysis of objects, textures, and structures in multidimensional digitized images from a variety of imaging modalities. Dr. van Vliet was awarded a fellowship from the Royal Netherlands Academy of Arts and Sciences (KNAW) in 1996.

Figures

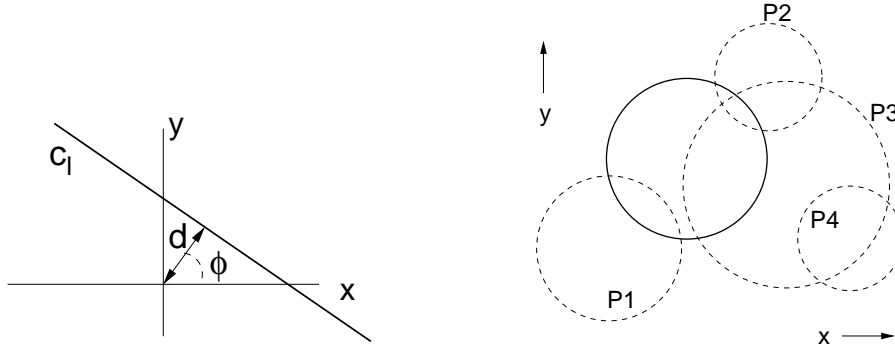


Fig. 1. Illustrating the generalized Radon transform. Left: the *normal* parameterization of a line. The parameters are the distance d from the line to the origin, through the normal of the line that intersects the origin, and the angle ϕ between that same normal and the x -axis, as indicated in the diagram. Right: curve detection using the generalized Radon transform. Integrating the intensity values along each of the candidate curves P1–P4 yields small numbers. Only if a candidate curve happens to fully coincide with a curve in the image (the solid black circle), will the integral yield a large response.

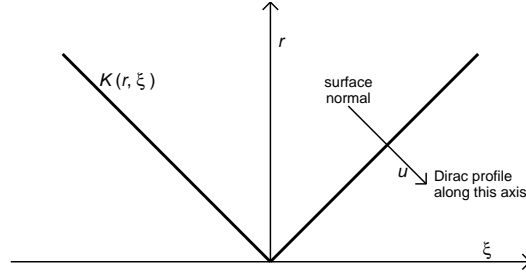


Fig. 2. The function $K(r, \vec{x})$ is a cone. For correct normalization, it is required that this function behave like the Dirac delta function along the normal to the surface.

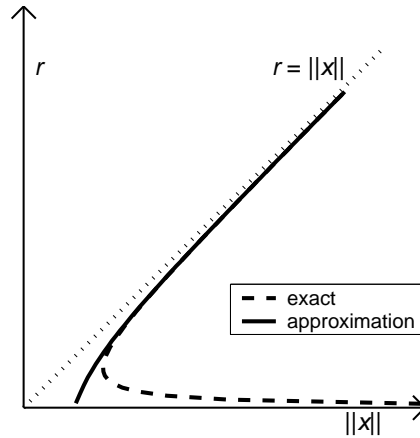
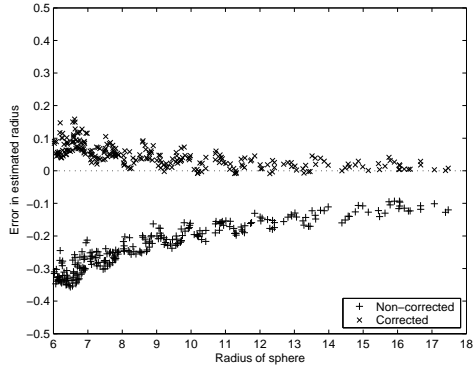
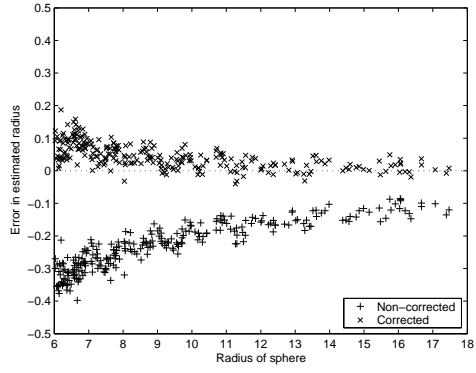


Fig. 3. The normalized function $K_b(r, \vec{x}, \sigma_K)$ from Equation (25) is distorted according to Equation (27). The correction we use is an approximation valid for larger r , and given by Equation (28).

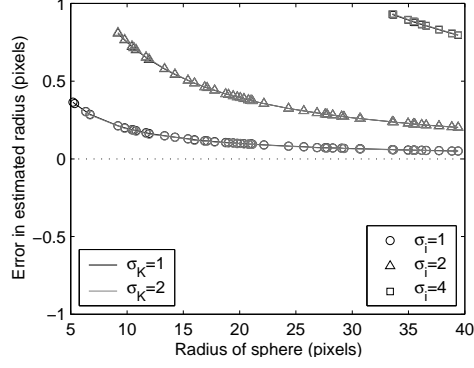


(a) $\text{SNR} = \infty$

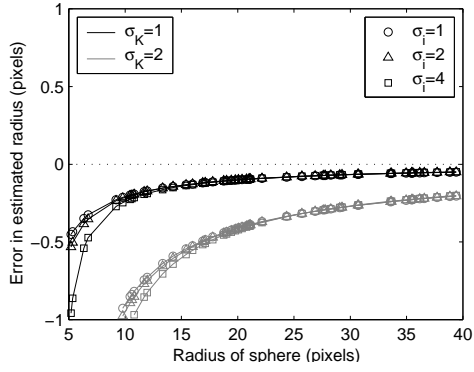


(b) $\text{SNR} = 2$

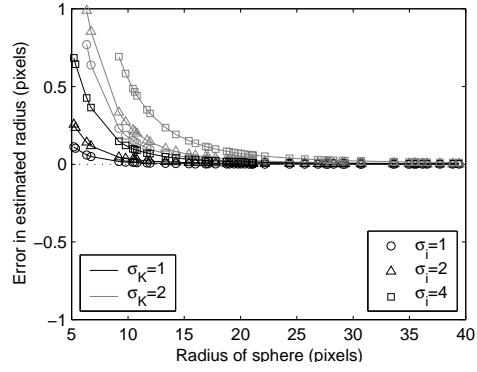
Fig. 4. Error made when estimating the radius of spheres in synthetic 3D images. The bias of the corrected method is due to the approximations used when computing Equation (28). The oscillatory nature of the line is due to the interpolation along the r -axis: the error is smaller for values near sample points.



(a) Non-normalized kernel



(b) Normalized kernel



(c) Corrected kernel

Fig. 5. Minimal error in estimation of the radius of spheres in synthetic 3D images, obtained without interpolation. These graphs show the effectiveness of the normalization and correction of the kernel $K_b(r, \vec{x}; \sigma_K)$, as well as the effect of σ_K and the σ_i used to create the input images.

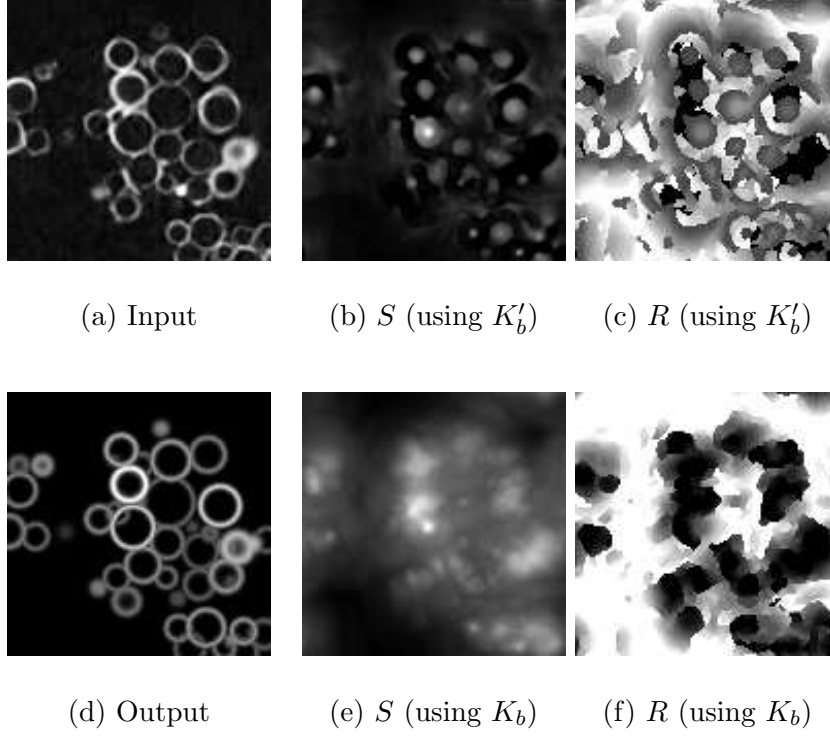


Fig. 6. One slice of the 3D ballotini image and the corresponding slices of the results of the Radon transform. a: Slice of the input image. b: $S(\vec{x}_o)$ and c: $R(\vec{x}_o)$ obtained using the modified $K'_b(r, \vec{x}_o; \sigma_K)$ from (33). d: Image generated with the found parameters. e: $S(\vec{x}_o)$ and f: $R(\vec{x}_o)$ obtained using the regular $K_b(r, \vec{x}_o; \sigma_K)$ from (30).

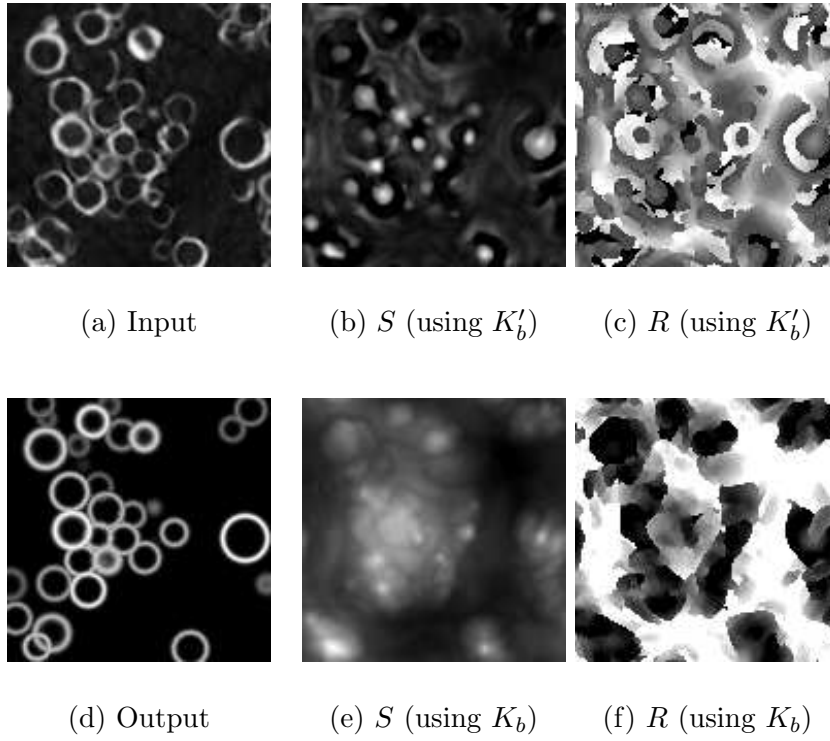


Fig. 7. One slice of the 3D ballotini image and the corresponding slices of the results of the Radon transform. a: Slice of the input image. b: $S(\vec{x}_o)$ and c: $R(\vec{x}_o)$ obtained using the modified $K'_b(r, \vec{x}_o; \sigma_K)$ from (33). d: Image generated with the found parameters. e: $S(\vec{x}_o)$ and f: $R(\vec{x}_o)$ obtained using the regular $K_b(r, \vec{x}_o; \sigma_K)$ from (30).

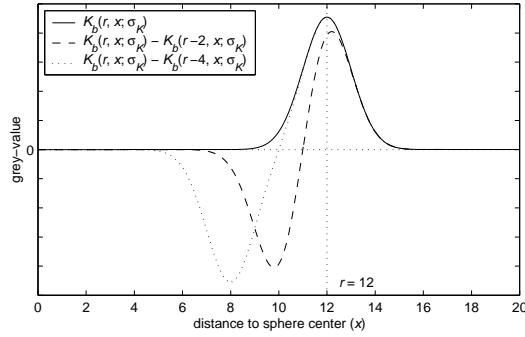


Fig. 8. Effect of the negative inner sphere: if it is too close to the outer sphere, the profiles will overlap too much, causing an outwards shift of the outer sphere. A distance of 4σ introduces only a small error. (In this graph, $\sigma = 1$.)

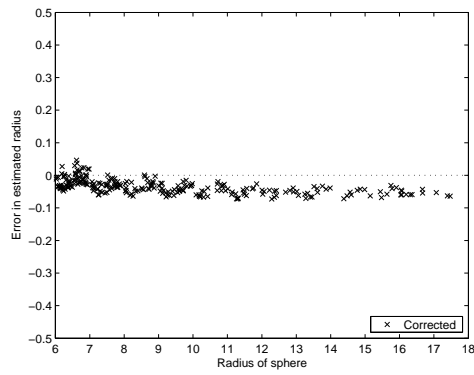


Fig. 9. Error made when estimating the radius of spheres in synthetic 3D images, using a negative inner sphere in the convolution kernel. Compare to the results for the corrected kernel in the left graph of Figure 4, which is computed on the same images. The underestimation here is about 0.1 pixel for large spheres.

Tables

Table 1

Error made when estimating the position of spheres in synthetic 3D images. The numbers shown are standard deviations of the error (in pixels) in the x , y and z -directions for the various methods (estimated from 500 spheres of radius between 6 and 18 pixels).

		$std(\delta x)$	$std(\delta y)$	$std(\delta z)$
noiseless image	non-corrected SW	0.0387	0.0360	0.0383
	corrected SW	0.0406	0.0381	0.0405
	corrected PM	0.0432	0.0445	0.0462
noisy image	non-corrected SW	0.0503	0.0487	0.0507
	corrected SW	0.0520	0.0498	0.0525
	corrected PM	0.0578	0.0579	0.0579