Reconstruction From Ideal Projections Using a Computer Algebra System

Joseph M. Winograd and James H. McClellan

Corresponding Author:
Joseph M. Winograd
ECS Dept.
Boston University
Boston, MA, 02215
(617) 353-3559

James H. McClellan
Dept. of Electrical Engineering
Georgia Institute of Technology
Atlanta, GA, 30332

Abstract

Computer algebra systems can be used to perform analyses requiring extensive algebraic manipulation and provide a means of performing precise mathematical explorations that are impractical if done by hand. The ability to graphically display results makes these systems useful also as educational aids and for demonstrating non-intuitive mathematical relationships in particular. In this paper, we use the Georgia Tech Signal Processing Packages for *Mathematica* to illustrate the problem of reconstruction from parallel-line projections. Derivations of two general solutions of the Radon inversion are presented, demonstrating the utility of computer algebra systems for communicating advanced mathematical concepts.

1 Introduction

Computers have long been used by scientists, engineers, and mathematicians for the solution of numerical problems requiring more calculations than can be reasonably performed by hand. Computer algebra systems allow them to also be used to solve problems involving extensive manipulation of symbolic expressions, thereby facilitating a class of investigations that were previously impractical. Symbolic analysis offers a level of precision not possible with numerical computation because mathematical expressions are stored and manipulated in an unevaluated symbolic form, thereby maintaining infinite precision throughout processing.

Not only are computer algebra systems useful for problem solving, but they have also been recognized as important tools for education.[1] One important aspect of their educational value is the facilitation of exploring and demonstrating non-intuitive mathematical relationships. One such non-intuitive, yet extraordinarily useful, mathematical relationship is the Radon inversion formula for reconstruction from parallel projections. Simply put, Radon inversion is the mathematical process by which a multidimensional function is found exactly when only straight line integrals across it are known. The physical analogue of this process, tomography, is likewise the process of determining the internal structure of an object by the analysis of radiation that has passed through it. Many computed tomography (CT) systems based upon a discretization of this reconstruction problem have been built and widely used with great success. However, fundamental constraints on realizable systems require that the conditions for perfect reconstruction be compromised. Consequently, much research effort has been expended in attempts to bring the performance of these systems as close as possible to the mathematical ideal.

Numerical simulations of the techniques available for approximating ideal reconstruction have been presented in the literature, but analytic derivations have generally been omitted due to their algebraic complexity. Computer algebra systems present an environment in which such analytic formulations can be generated, allowing ideal mathematical reconstructions to be compared with proposed implementations. Further, when a computer algebra system is used the results at each step of the reconstruction may be inspected and displayed graphically, so that the effects of the various operations (such as projection or discretization) can be readily demonstrated for educational purposes.

This article presents the application of Georgia Tech Signal Processing Packages for *Mathematica*[2] to the reconstruction of functions from parallel-line projections and demonstrates its utility, and that of computer algebra systems in general, for the exploration and demonstration of advanced mathematical relations.

2 *Mathematica* and the SPP

Mathematica is a commercially available computer algebra system that offers a wide range of mathematical functions as well as a powerful programming language.[3] The programming language provides support for several high-level programming paradigms including procedural, functional, object-oriented, and rule-based constructs. The Georgia Tech Signal Processing Packages for *Mathematica* (SPP)[4,5] were developed using this language and provide tools for the representation and analysis of both continuous and discrete multidimensional signals and systems. While its development still continues, the SPP currently offers 16 signal functions and 30 operators. The analyses presented in this paper utilize new capabilities of the SPP to perform forward and inverse continuous-time multidimensional Fourier transforms, as well as continuous-time symbolic convolution.[6]

A public domain version of the SPP is available via anonymous Internet ftp[7] from `gauss.eedsp.gatech.edu` (Internet Protocol address 130.207.224.26). They reside in the `Mathematica/` directory in compressed UNIX `tar` format (`SigProc2.0.tar.Z` at the time of publication), zip format for the IBM PC running Windows (`SigProc2_IBM_PC.zip`), and binhexed self-extracting Macintosh archive format (`SigProc2.0.mac.sea.hqx`). Downloading and installation instructions may be found in the `README` file in the same directory.

This paper presents some expressions that were used as input to *Mathematica* to perform the mathematical operations discussed in the text. These are shown in `Courier` type and were evaluated on a NeXT cube using *Mathematica* version 2.0 with the SPP version 2.9. The figures accompanying this paper were generated by *Mathematica*.

3 Reconstruction from Parallel-Line Projections

In a variety of scientific situations we wish to determine the internal structure of an object but are unable to examine its interior directly. Such situations arise in the fields of astronomy, medicine, geophysics, microscopy, manufacturing, and others, and can often be solved using tomography. Tomography is a process by which the internal structure of the object is determined indirectly by examination of the residual intensity of a radiation that has passed through it. Successful application of tomography requires that this radiation be attenuated in a manner proportional to the variations of interest within the structure.

The practice of tomography has its theoretical basis in the mathematical problem of reconstructing a n-dimensional function when only its $(n-1)$-dimensional projections are known. Projections are defined as line integrals across the extent of the function. Several projection geometries have been studied, but we concern ourselves here with projections taken along parallel lines, as illustrated in Fig. 1. The problem of inverting the projection process to reconstruct the original function was first solved by J. Radon[8] in 1917 as a purely theoretical problem. It was later solved independently by R. Bracewell in 1956 with application to radio astronomy[9] and A. Cormack in 1963 with application to medical imaging.[10] In this section, we present their methods of performing reconstructions from parallel projections and then show how they can be illustrated using the SPP.

3.1 Convolution Back-Projection Method of Reconstruction

Consider parallel projections of the two-dimensional function $f(x_1, x_2)$ taken at various rotational orientations. The orientation of each projection can take the form of a change of variables from (x_1, x_2) into (u_1, u_2) according to

$$u_1 = x_1 \cos\theta + x_2 \sin\theta \quad (1)$$

$$u_2 = -x_1 \sin\theta + x_2 \cos\theta \quad (2)$$

A projection (or *Radon transform*) of $f(x_1, x_2)$ at orientation θ is then

$$p_\theta(u_1) = \int_{-\infty}^{\infty} f(u_1, u_2) du_2 \quad (3)$$

Using the original Radon inversion,[8] when its projections are known continuously for $0 \leq \theta < \pi$, the function $f(x_1, x_2)$ can be exactly reconstructed by

$$f(x_1, x_2) = \frac{1}{2\pi} \int_0^\pi g_\theta(x_1 \cos\theta + x_2 \sin\theta) d\theta \quad (4)$$

where

$$g_\theta(t) = -\frac{1}{\pi}\frac{d}{dt}\int_{-\infty}^{\infty}\frac{p_\theta(\tau)}{t-\tau}d\tau \tag{5}$$

Equation (5) defines the filtered projection function $g_\theta(t)$. It is the derivative of the Hilbert transform of the projection at angle θ. $f(x_1, x_2)$ is reconstructed from $g_\theta(t)$ by rotating it to the angle θ and back-projecting it across the (x_1, x_2) plane. The process described by equations (4) and (5) is called *convolution back-projection* and is illustrated in the section entitled "Convolution Back-Projection Using the SPP."

While the Radon inversion reconstructs the original function exactly, it is not directly implementable because it requires that projections be taken across a continuous range of angles and that a convolution of infinite extent be performed. Due to the derivative operation, it is also highly sensitive to noise. Thus, alternate solutions have been developed, one of which we introduce next.

3.2 Filtered Back-Projection Method of Reconstruction

An alternate approach to solving the reconstruction problem was derived by R. Bracewell who developed a Fourier space formulation of the Radon inversion.[9] The Fourier transform of the Radon transform of $f(x_1, x_2)$ taken at $\theta = 0$ is

$$\begin{aligned} P_0(\omega_1) &= \int_{-\infty}^{\infty} p_0(x_1)e^{-j\omega_1 x_1}dx_1 \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x_1, x_2)e^{-j\omega_1 x_1}dx_1 dx_2 \end{aligned} \tag{6}$$

Comparing this to the Fourier transform of $f(x_1, x_2)$

$$F(\omega_1, \omega_2) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x_1, x_2)e^{-j(\omega_1 x_1 + \omega_2 x_2)}dx_1 dx_2 \tag{7}$$

we see that

$$P_0(\omega_1) = F(\omega_1, \omega_2)|_{\omega_2=0} \tag{8}$$

An orthogonal change of the variables of a function produces an identical change of variables in the Fourier transform of that function. Since rotation is an orthogonal transformation, we can generalize this result as

$$P_\theta(\kappa_1) = F(\kappa_1, \kappa_2)|_{\kappa_2=0} \tag{9}$$

where

$$\kappa_1 = \omega_1 \cos\theta + \omega_2 \sin\theta \tag{10}$$

$$\kappa_2 = -\omega_1 \sin\theta + \omega_2 \cos\theta \tag{11}$$

Equation (9) is called the *projection-slice theorem* and shows that the Fourier transform of each projection of a function is identical to an axial slice of the multidimensional Fourier transform of the function taken through the origin at the angle of projection.

The function $f(x_1, x_2)$ can be reconstructed in Fourier space by converting the convolution back-projection to a Fourier space operation. The Fourier space equivalent of the derivative of the Hilbert transform of the projection is

$$G_\theta(\kappa_1) = |\kappa_1| P_\theta(\kappa_1) \tag{12}$$

$|\kappa_1|$ is typically called the *Radon filter*. Using equations (9) and (12), the reconstruction formulae (4) and (5) can be written in the Fourier domain as

$$\begin{aligned} f(x_1.x_2) &= \frac{1}{4\pi^2} \int_0^\pi \int_{-\infty}^{\infty} G_\theta(\kappa_1) e^{j\kappa_1(x_1\cos\theta + x_2\sin\theta)} d\kappa_1 d\theta \\ &= \frac{1}{4\pi^2} \int_0^\pi \int_{-\infty}^{\infty} |\kappa_1| P_\theta(\kappa_1) e^{j\kappa_1(x_1\cos\theta + x_2\sin\theta)} d\kappa_1 d\theta \end{aligned} \tag{13}$$

which is the Fourier space inversion formula for exact reconstruction.

3.3 Convolution Back-Projection Using the SPP

The SPP operator `CPulse[ 1, x1^2+x2^2 ]` defines a circle at the origin of "density" and radius 1. By successively applying the `ScaleAxis`, `RotateAxes`, and `Shift` operators, we can transform this circle into an ellipse of arbitrary size, position, and orientation.

```
ellipse = Shift[ {x1o, x2o}, {x1,x2} ][
    RotateAxes[ -thetao, {x1, x2} ][
        ScaleAxis[ {1/x1s, 1/x2s}, {x1, x2} ][
            CPulse[ 1, x1^2 + x2^2 ] ] ] ]
```

Fig. 2 shows a plot of this function with center at $(2, 3)$, major axis length 2, minor axis length 1, and orientation of $\frac{\pi}{3}$ radians about its center. We will now illustrate the convolution back-projection method of reconstructing this function from its projections.

The projection function $p_\theta(u_1)$ (which we shall call `pellipse`) of the generalized `ellipse` function can be found using the projection-slice theorem. It is the inverse Fourier transform of an axial slice of the two-dimensional Fourier transform of `ellipse`. This is derived in the SPP with

```
pellipse =
    SPSimplify[
    InvCTFTransform[
        (TheFunction[ RotateAxes[theta,{k1,k2}][
            TheFunction[ CTFTransform[ ellipse,
                {x1,x2}, {k1,k2} ] ] ] ] /.  k2->0) //.
        (* Apply a few non-standard simplifications.  *)
        {Sqrt[a_^2 b_] :> Abs[a] Sqrt[b],
        1/Sqrt[a_^2 b_] :> 1/(Abs[a] Sqrt[b]),
        (a_ t_+b_ t_) :> (a+b) t},
    k1, u1 ], Variables->u1 ]
```

Fig. 3 is a plot of `pellipse` versus `theta`. It depicts the variation in the projection function as projections are taken at different angles. It is interesting to note that the path taken by the projection of the ellipse is sinusoidal, as we would expect from the projection of a rotating object onto a fixed axis.

Reconstruction of the original ellipse by convolution back-projection requires two distinct steps. First, the back-projection function $g_\theta(t)$ (which we'll call `gellipse`) is derived from $p_\theta(u_1)$. Following equation (5), $g_\theta(t)$ is the derivative of the Hilbert transform of the projection function.

```
gellipse = D[ CTPiecewiseConvolution[ pellipse /.  u1->t,
                                        1/t, t ], t ]
```

Fig. 4 shows the filtered projection function `gellipse`. The filtered projection function varies with `theta` in a manner similar to the projection function.

The original ellipse may now be reconstructed from `gellipse` using the back-projection integral of equation (4). This operation is best explained graphically. Each angle of θ contributes one back-projection to the total function, where each back-projection consists of aligning the one-dimensional function $g_\theta(t)$ in the (x_1, x_2) plane along the x_1 axis, "projecting" it in both directions along the x_2 axis, and rotating both axes about the origin by the angle θ. Fig. 5 shows the back-projection of $g_\theta(t)$ at $\theta = 0$ (note that this back-projection actually extends to both plus and minus infinity along the x_2 axis). The reconstruction consists of the continuous summation (integration) of all of the back-projections of $g_\theta(t)$ for $0 \leq \theta < \pi$. This is difficult to evaluate analytically because of the rotation of the coordinate system across back-projections. It is also an operation that cannot be precisely implemented because only a finite set of projections may be taken of a physical object. When the back-projection function is only known for a discrete set of angles, the reconstruction integral must be approximated by weighting the measurements with the difference between the angles θ_i for which projections have been taken.[11] Equations (14)-(16) define this reconstruction.

$$\Delta\theta_0 \triangleq \theta_0 - \theta_{N-1} + \pi \tag{14}$$

$$\Delta\theta_i \triangleq \theta_i - \theta_{i-1}, \qquad 1 \leq i \leq N-1 \tag{15}$$

$$f(x_1, x_2) = \frac{1}{2\pi} \sum_{i=0}^{N-1} \Delta\theta_i g_\theta(x_1 \cos\theta_i + x_2 \sin\theta_i) \tag{16}$$

The function `recon[ g, N ]`, shown below, implements the summation of equation (16) for `N` discrete back-projections of the function `g` equally spaced between 0 and π.

```
recon[ g_, N_ ] := Sum[ RotateAxes[ -theta, {x1,x2} ][ g ],
    { theta, 0, Pi - Pi/N, Pi/N } ] / (2 N)
```

We can use this function to perform approximate reconstructions of the ellipse (Fig. 2) by using the back-projection function `gellipse` as parameter `g`. Fig. 6 shows such reconstructions using 2, 4, 8, and 16 projections.

3.4 Filtered Back-Projection Using the SPP

We now present a second example of reconstruction from ideal projections, this time using the filtered back-projection method of reconstruction. We begin by defining an arbitrary Gaussian function, `gauss`, in much the same manner as we defined an arbitrary ellipse to illustrate the convolution back-projection technique.

```
gauss =
    Shift[ {x1o, x2o}, {x1, x2} ][
        RotateAxes[ -thetao, {x1, x2} ][
            ScaleAxis[ {1/x1s, 1/x2s}, {x1, x2} ][
                Exp[ -(x1^2 + x2^2) ] ] ] ]
```

Fig. 7 shows an instance of this arbitrary Gaussian function with scaling of 2 in the x_1 dimension, rotation of $\frac{\pi}{3}$ radians about the origin, and translation so its center is at $(2, 3)$.

As before, we will reconstruct the original function from its projections. The filtered back-projection method operates in the Fourier domain, so we first calculate the Fourier transform of the projection function. The projection-slice theorem tells us that it is equal to an axial slice of the Fourier transform of the original signal. This slice $P_\theta(\kappa_1)$ (which we'll call `Pgauss`) is derived in the SPP using

```
Pgauss =
    Simplify[
        TheFunction[
            RotateAxes[ theta, {k1, k2} ][
                TheFunction[
                    CTFTransform[ gauss, {x1, x2}, {w1, w2} ]
                ] /.  {w1->k1,w2->k2}
            ]
        ] /.  k2->0
    (* Perform a non-standard simplification.  *)
    ] //.  (a_ t_ + b_ t_) :> (a + b) t
```

Fig. 8 shows `Pgauss` for the Gaussian function of Fig. 7 and is plotted as the angle of slicing, `theta`, varies from 0 to π. The Fourier transform of a shifted Gaussian is a modulated (and scaled) Gaussian, so we see only slight variations in the frequency slices as we vary the angle.

To reconstruct from this, we first apply the Radon filter in Fourier space and take the inverse Fourier transform to generate the back-projection function `ggauss` by

```
ggauss = InvCTFTransform[ Abs[k1] Pgauss, k1, u1,
            Simplify->False
         ] /.  u1->x1
```

This operation corresponds to the inner integral of equation (13). Note that `ggauss` corresponds directly to the filtered back-projection function $g_\theta(t)$ of the convolution back-projection derivation. `ggauss` is shown (versus `theta`) in Fig. 9.

The outer integral of equation (13) performs a back-projection analogous to equation (5) in the convolution back-projection derivation; that is, it integrates the rotated back-projections to reconstruct the original function. We can again use the `recon` function to approximation this ideal reconstruction and generate reconstructions from `ggauss` using only a finite number of projections. Fig. 10 shows reconstructions of the Gaussian function of Fig. 7 using 2, 4, 8, and 16 projections.

4 Noise in Reconstructions

Comparing Fig. 6 with Fig. 10, we see that for a given number of projections the quality of our reconstructions varies greatly. Convolution back-projection and filtered back-projection are mathematically equivalent processes and the operations performed in our derivations here are mathematically ideal, so this variation cannot be attributed to the methods of analysis.

The difference in reconstruction quality is due to the frequency composition of the two functions to use in our examples. The elliptically shaped function contains an appreciable amount of high-frequency energy, due to its discontinuous nature, and the Gaussian function contains negligible high-frequency components. The reconstruction processes described here perform either a derivative operation or frequency-domain Radon filtering, both of which amplify high-frequency signals enormously. When we reconstruct using only a finite number of projections, we introduce error in our function at all frequencies present in the signal. In the case of the Gaussian this error occurs in the relatively harmless low-frequency region, but in the ellipse it introduces a great deal of high-frequency noise.

CT implementations process sampled, thus band-limited, projections, and therefore do not introduce high frequency noise in this way. Nevertheless, they have noise problems of their own due to aliasing of high-frequencies in the sampled projections and noise sensitivity introduced during the physical projection and sensing processes. Mathematical treatments of these noise effects in reconstruction can be found in the literature.[12]

5 Conclusions

The primary goal of this paper is to show the capabilities of computer algebra systems for demonstrating abstruse mathematical relationships. We have used a computer algebra system to present two generalized analytic solutions to the problem of reconstruction from parallel projections. The mathematical expressions derived in these two examples are too long and unwieldy to immediately apprehend—or publish—so we have relied on the graphical powers of *Mathematica* for their illustration.

We have also demonstrated how the ability to manipulate extremely large expressions, coupled with the maintenance of infinite mathematical precision, sets symbolic computation apart from more traditional methods of computer-based mathematical analysis. The *Mathematica* computer algebra system, supplemented with the Georgia Tech Signal Processing Packages, provides a rich set of tools for performing symbolic signal analysis and further investigations into multidimensional problems such as reconstruction from projections.

Our examples of reconstructing mathematical models of simple geometric figures can be extended to more complex scenarios. One such figure that we have analyzed is the Shepp-Logan head phantom (Fig. 11) which approximates the structure of the human head (including several simulated tumors) with a set of ellipses.[13] Other avenues of study facilitated by this system include the development of alternate reconstruction filters for improving reconstruction quality when a finite number of projections are used and the reconstruction of three-dimensional objects from two-dimensional projections. Symbolic analysis allows the performance of alternate filters to be analytically compared with that of the mathematical ideal and also enables the direct extension of formulae into higher dimensions. The SPP provides ample capabilities for the investigation of these topics.

Of practical importance to users of this system is its interactive performance. The example derivations presented in this paper were performed on an original NeXT cube with a Motorola 68030 CPU and 68882 FPU running at 25 MHz and 16 MB of RAM. The symbolic result from each example takes between a few seconds and a few minutes to produce. Generating plots of the reconstruction formulae (Figs. 6 and 10) takes longer, with each requiring about an hour to render. This is because the approximate reconstruction formula, equation (16), produces a large symbolic expression which must be evaluated numerically at many points to generate the figures. Plots of this kind may be generated once and saved on a mass-storage device to be reloaded later for quick display.

The primary limitation imposed on the use of this system is the same as that which is often faced in traditional mathematical analysis; the intractability of many integral equations. Although this system can integrate impressive expressions, there remain many that it cannot. Success in the evaluation of any particular integral transform will depend on the creativity of the researcher in formulating the expression, and the quality of the integration knowledge-base of the symbolic mathematics system in use. The SPP extends *Mathematica*'s knowledge-base with integration rules for the signal processing operators that it defines and allows new rules to be added by the user.

6 Acknowledgments

This research was supported in part by the Joint Services Electronics Program contract #DAAH-04-93-G-0027.

7 References

1. See, for instance, B. L. Evans *et al*, IEEE Trans. on Educ. **36**, 1, 72 (1993); D. M. Cook, Computers in Physics **4**, 2, 197 (1990); D. M. Cook, Computers in Physics **4**, 3, 309 (1990); M. Horbatsch, Computers in Physics **4**, 6, 656 (1990).

2. *Mathematica* is a trademark of and available from Wolfram Research, Inc., 100 Trade Center Drive, Champaign, Ill., 61820-7237.

5. S. Wolfram, *Mathematica*, (Addison-Wesley, 1988).

4. B. L. Evans *et al.*, IEEE Trans. on Educ. **36**, 1, 72 (1993).

5. B. L. Evans and J. H. McClellan in *Symbolic and Knowledge-Based Signal Processing*, eds. A. V. Oppenheim and S. H. Nawab, (Prentice-Hall, 1992).

6. K. West and J. H. McClellan, IEEE Trans. on Educ. **36**, 4, 386 (1993).

7. A brief introduction to Internet ftp can be found in G. Ricast, Computers in Physics **8**, 1, 20 (1994).

8. J. Radon, Berichte Saechsische Akad. Wissenschaft (Leipzig), Math. Phys. Klass 69, 262 (1917).

9. R. N. Bracewell, Aust. J. Phys. **9**, 198 (1956).

10. A. M. Cormack, J. Appl. Phys. **34**, 2722 (1963).

11. R. M. Mersereau and A. Oppenheim, Proc. IEEE **62**, 10, 1319 (1974).

12. For an overview see A. C. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging*, (IEEE Press, 1988).

13. L. A. Shepp and B. F. Logan, IEEE Trans. Nucl. Sci. **21**, 21 (1974).

Figure Captions

Figure 1: Parallel projections of an object are taken by integrating along parallel lines as shown.

Figure 2: Plot of `ellipse` with center at $(2, 3)$, major axis length 2, minor axis length 1, and rotated about its center by $\frac{\pi}{3}$ radians.

Figure 3: Projection of the ellipse of Fig. 2 for $0 \leq \theta < \pi$.

Figure 4: Projections of Fig. 3 after Radon filtering.

Figure 5: Filtered projection at $\theta = 0$ back-projected across the (x_1, x_2) plane.

Figure 6: Reconstruction of the ellipse in Fig. 2 using 2, 4, 8, and 16 back-projections.

Figure 7: Plot of `gauss` scaled by 2 in the x_1 dimension and rotated by $\frac{\pi}{3}$ radians about its center, which is translated to $(2, 3)$.

Figure 8: Slices of the Fourier transform of the Gaussian function of Fig. 7 taken at for $0 \leq \theta < \pi$.

Figure 9: Projections of Fig. 8 after Radon filtering.

Figure 10: Reconstruction of the Gaussian function in Fig. 7 using 2, 4, 8, and 16 projections.

Figure 11: A commonly used model of the interior of the human head that can be analyzed using the system described in this paper.