X-Ray Function Program Information

1. Contributors and history.

In May, 2004, Richard J. Gardner and Markus Kiderlen of the University of Aarhus, Denmark, conceived the algorithm described below for reconstructing planar convex bodies from noisy measurements of their X-rays. This work was eventually published in [3].

In September, 2005, Western Washington University (WWU) undergraduate Mark Lockwood began work on implementing the algorithm. Mark was the principal developer of the basic program, making contributions until around 2009, including the incorporation of simulated annealing.

WWU undergraduate Kyle Rader brought the program to its present form by significantly extending its capabilities, redesigning the GUI, and configuring it for use on the Geometric Tomography website. It was subsequently modified by WWU undergraduate students Ian Fisk and Elliott Skomski.

Both the mathematics and implementation were supported by National Science Foundation grants DMS-0203527, DMS-0603307, DMS-1103612, and DMS-1402929.

2. How the program works.

The following description is a slightly edited version of the one in [1], where other algorithms for reconstruction from X-rays are also presented.

The Gardner-Kiderlen X-ray algorithm [3] arose from theoretical work [4] in which it was shown that there are certain sets of four directions in $\mathbb{R}^2$ such that the exact X-rays of a planar convex body in these directions determine it uniquely among all planar convex bodies. For example, directions specified by the four vectors $(0, 1)$, $(1, 0)$, $(2, 1)$, and $(-1, 2)$ constitute such a set [2]. The GKXR algorithm is based on the simple observation that given a sufficiently dense set of lines meeting a convex body K, the convex hull of all the points at which the lines intersect the boundary of K will form a convex polygon that approximates K well. The algorithm attempts to find this polygon for the set of X-ray measurement lines.

Fig. 1 shows a schematic diagram of the basis of the algorithm. The unknown object is the oval K, assumed to lie inside the circle. For clarity, only a single X-ray, taken in the direction u, is considered in Fig. 1, although in practice X-rays in several different directions are used. We shall assume here that four X-rays are taken, the default setting in the program. For each X-ray direction u, detector pixels are located at the equally spaced points $t_1, \dots, t_k$ on the axis in the orthogonal direction v, where k is the number of beams selected in the program. The dotted lines through these points represent beams along which measurements are taken. A pair of points (in Fig. 1, one red and one blue, shown in purple if they coincide) is placed randomly on each of the $4k$ beams. Since the geometry of the beams is known, the position of each point can be described by a single real variable giving the location of the point on the beam. Therefore the position of all of the points can be described by a single vector variable z with $8k$ real components.

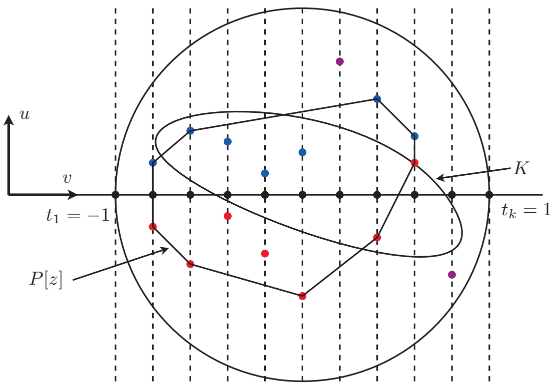

FIGURE 1. Illustration showing the basic principles behind the GKXR algorithm.

An initial guess for K is obtained by forming the convex hull of all $8k$ points, except those for which a pair coincides, i.e., the purple points. This is the convex polygon labeled $P[z]$ in Fig. 1. The convex hull is computed using a standard algorithm as a subroutine. The reason for ignoring the purple points in taking the convex hull is that if a beam does not meet K, there must be some mechanism to eliminate the pair of points that lies on that line. In practice, a threshold ("point elimination" in the program) is set so that a pair of points is eliminated if they become too close in the iterative optimization procedure to be described next.

In order to improve the initial guess, the positions of the pairs of points on the beams must be adjusted. This is effected by computing the sum, over all beams, of the squares of the differences between the measured X-ray value for K and the corresponding X-ray value of $P[z]$. This least squares sum is the objective function in an optimization problem with $8k$ real variables and an optimization routine is used to drive the value of the objective function down to a minimum. The output of the algorithm is the convex polygon $P_k = P[z]$ corresponding to the optimal vector z of these real variables.

In [3] it is shown that for any finite set of directions for which the corresponding exact X-rays determine a convex object uniquely, the output P_k converges to K as $k \to \infty$, even when the measurements are affected by Gaussian noise of fixed variance. Moreover, this remains true even if the optimization problem is not solved exactly, but only within an error $\varepsilon_k > 0$, provided $\varepsilon_k \to 0$ as $k \to \infty$; see [3, p. 337]. In practice, the optimization problem involved is heavily non-linear. The `fmincon` function from Matlab's Optimization Toolbox is used, along with simulated annealing to improve performance.

References

[1] A. Alpers, R. J. Gardner, S. König, R. S. Pennington, C. B. Boothroyd, L. Houben, R. Dunin-Borkowski, and K. J. Batenburg, Geometric reconstruction methods in electron tomography, *Ultramicroscopy* **128** (2013), 42–54.

[2] R. J. Gardner and P. Gritzmann, Discrete tomography: Determination of finite sets by X-rays, *Trans. Amer. Math. Soc.* **349** (1997), 2271–2295.

[3] R. J. Gardner and Markus Kiderlen, A solution to Hammer's X-ray reconstruction problem, *Adv. Math.* **214** (2007), 323–343.

[4] R. J. Gardner and P. McMullen, On Hammer's X-ray problem, *J. London Math. Soc.* (2) **21** (1980), 171–175.