The Fanbeam Projection of a Point

Consider the (directional) projection of a point (x_0, y_0) along the vector $(x_0 - x_\mathrm{s}, y_0 - y_\mathrm{s})$ onto a horizontal detector plane with center $(x_\mathrm{d}, y_\mathrm{d})$ (cf. the illustration in Figure 1). The projection

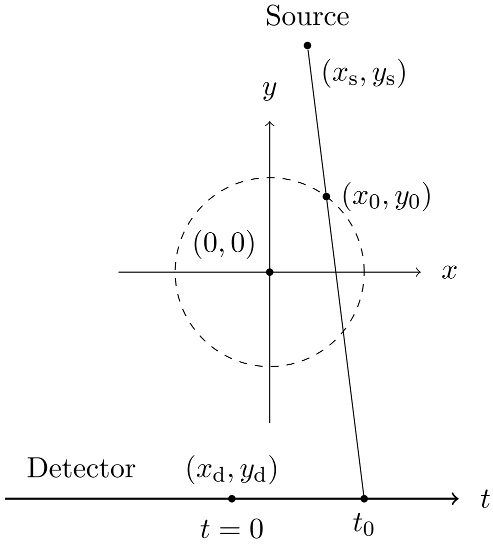

Figure 1: Fanbeam geometry: directional projection of a point.

$(x_\mathrm{d} + t_0, y_\mathrm{d})$ can be expressed as the intersection of the detector plane and the line that passes through $(x_\mathrm{s}, y_\mathrm{s})$ and (x_0, y_0), i.e., there exists a scalar $\alpha \neq 0$ such that

$$\begin{bmatrix} x_\mathrm{d} + t_0 \\ y_\mathrm{d} \end{bmatrix} = \alpha \begin{bmatrix} x_\mathrm{s} \\ y_\mathrm{s} \end{bmatrix} + (1 - \alpha) \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}.$$

The second of these two equations implies that

$$\alpha = \frac{y_\mathrm{d} - y_0}{y_\mathrm{s} - y_0},$$

and from the first equation we can obtain the position t_0 on the detector plane

$$t_0 = \alpha(x_\mathrm{s} - x_0) + x_0 - x_\mathrm{d} = \frac{y_\mathrm{d} - y_0}{y_\mathrm{s} - y_0}(x_\mathrm{s} - x_0) + x_0 - x_\mathrm{d}.$$

Rewriting this expression yields

$$\begin{aligned} t_0 &= \frac{y_\mathrm{d} - y_\mathrm{s} + y_\mathrm{s} - y_0}{y_\mathrm{s} - y_0}(x_\mathrm{s} - x_0) + x_0 - x_\mathrm{d} \\ &= \left(\frac{y_\mathrm{d} - y_\mathrm{s}}{y_\mathrm{s} - y_0} + 1 \right) (x_\mathrm{s} - x_0) + x_0 - x_\mathrm{d} \\ &= (x_\mathrm{s} - x_\mathrm{d}) - (y_\mathrm{s} - y_\mathrm{d}) \frac{x_\mathrm{s} - x_0}{y_\mathrm{s} - y_0}. \end{aligned}$$

Now if we rotate (x_0, y_0) by an angle of $-\theta$ around origo (equivalently, rotate the source and the detector counterclockwise around the origo by θ), we obtain the following expression for the position on the detector plane

$$t_0(\theta; x_\mathrm{s}, y_\mathrm{s}, x_\mathrm{d}, y_\mathrm{d}, x_0, y_0) = (x_\mathrm{s} - x_\mathrm{d}) - (y_\mathrm{s} - y_\mathrm{d}) \frac{x_\mathrm{s} - x_0 \cos(\theta) - y_0 \sin(\theta)}{y_\mathrm{s} + x_0 \sin(\theta) - y_0 \cos(\theta)} \tag{1}$$

as a function of the source position $(x_\mathrm{s}, y_\mathrm{s})$, the detector center $(x_\mathrm{d}, y_\mathrm{d})$, and the point (x_0, y_0).

Model Calibration

The expression for the position of the projection of a point as a function of the rotation angle θ can be used to calibrate our forward model. To this end, we will place a pin (or something that resembles a "point") in our X-ray CT scanner and collect a number of projections from, say, m different angles. If we let $\hat{p}_i(t)$ denote the projection at angle θ_i, we can estimate the projection centroid as

$$\hat{\rho}_i = \frac{\displaystyle\int_{-w/2}^{w/2} t\,\hat{p}_i(t)\,dt}{\displaystyle\int_{-w/2}^{w/2} \hat{p}_i(t)\,dt} \approx \frac{\sum_{j=1}^{r} t_j b_{ij}}{\sum_{j=1}^{r} b_{ij}} \tag{2}$$

where w is the width of the detector, $t_1, \ldots, t_r$ are the detector pixel positions (the pixel centers), and b_{ij} is the sinogram pixel corresponding to detector pixel i and projection angle θ_j. This is illustrated in Figure 2.

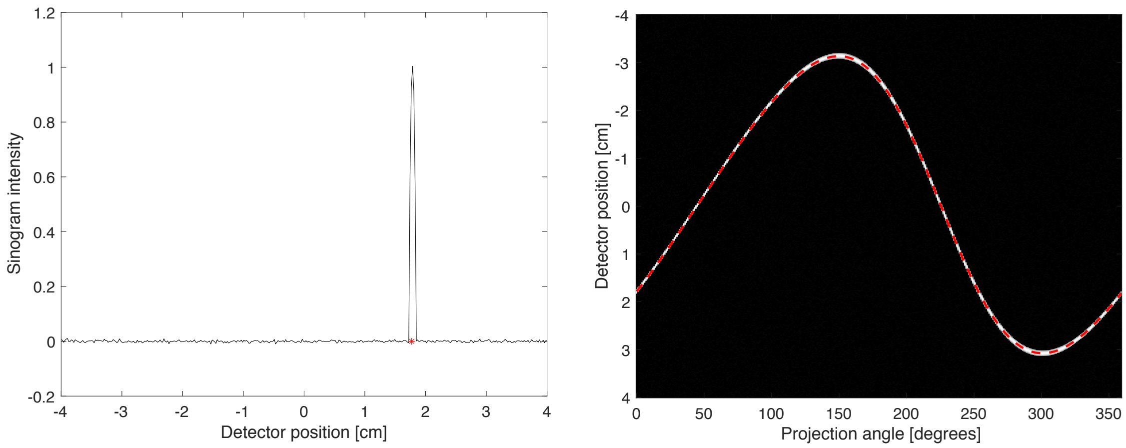

Figure 2: Centroid estimation from pin projections: (left) a single projection and the estimated centroid marked with a red asterisk; (right) sinogram with centroid estimates overlaid as a dashed red curve.

To calibrate the geometry, we will formulate a nonlinear least-squares problem

$$\text{minimize} \quad (1/2)\sum_{i=1}^{m}(t_0(\theta_i;\beta) - \hat{\rho}_i)^2$$

with variables $\beta = (x_{\mathrm{s}}, y_{\mathrm{s}}, x_{\mathrm{d}}, y_{\mathrm{d}}, x_0, y_0)$. In other words, our model $t_0(\theta_i;\beta)$ should (approximately) aggree with the centroid estimate $\hat{\rho}_i$. We will use the so-called Levenberg–Marquardt (LM) method to "solve" (i.e., find a local minimum) our nonlinear least-squares problem. The LM method is an iterative method, and each iteration requires the solution of a regularized (linear) least-squares problem

$$\begin{aligned} \beta^{(k+1)} &= \operatorname*{argmin}_{\beta} \left\{ (1/2)\|J_{t_0}(\beta^{(k)})(\beta - \beta^{(k)}) - r(\beta^{(k)})\|_2^2 + (\gamma/2)\|\beta - \beta^{(k)}\|_2^2 \right\} \\ &= \beta^{(k)} + \operatorname*{argmin}_{u} \left\{ (1/2)\|J_{t_0}(\beta^{(k)})u - r(\beta^{(k)})\|_2^2 + (\gamma/2)\|u\|_2^2 \right\} \end{aligned} \tag{3}$$

where $r^{(k)}$ is a residual vector with elements

$$r_i^{(k)} = \hat{\rho}_i - t_0(\theta_i;\beta^{(k)}), \quad i = 1, \ldots, m$$

and $J_{t_0}(\beta^{(k)})$ is the $m \times 6$ Jacobian matrix with rows $\nabla_\beta t_0(\theta_i;\beta^{(k)})^T$.

Exercises

1. Verify that the gradient of $t_0(\theta;\beta)$ (with respect to β) can be expressed as

$$\nabla_\beta t_0(\theta;\beta) = \begin{bmatrix} \frac{y_\mathrm{d}-y_0^\theta}{y_\mathrm{s}-y_0^\theta} \\ -(x_\mathrm{s}-x_0^\theta)\frac{y_\mathrm{d}-y_0^\theta}{(y_\mathrm{s}-y_0^\theta)^2} \\ -1 \\ \frac{x_\mathrm{s}-x_0^\theta}{y_\mathrm{s}-y_0^\theta} \\ -(y_\mathrm{s}-y_\mathrm{d})\frac{y_0-x_\mathrm{s}\sin(\theta)-y_\mathrm{s}\cos(\theta)}{(y_\mathrm{s}-y_0^\theta)^2} \\ (y_\mathrm{s}-y_\mathrm{d})\frac{x_0-x_\mathrm{s}\cos(\theta)+y_\mathrm{s}\sin(\theta)}{(y_\mathrm{s}-y_0^\theta)^2} \end{bmatrix}$$

where $x_0^\theta = x_0\cos(\theta) + y_0\sin(\theta)$ and $y_0^\theta = -x_0\sin(\theta) + y_0\cos(\theta)$. The MATLAB function below computes the function values $t_0(\theta_i;\beta)$ and the Jacobian matrix:

```
function [t0,J] = t0eval(theta,beta)
% Computes function values and Jacobian matrix
% for fanbeam calibration problem. The input beta
% is the vector (xs,ys,xd,yd,x0,y0).
assert(iscolumn(theta),'theta must be a column vector');
assert(length(beta)==6,'beta must be a vector of length 6');
x0t =  beta(5)*cosd(theta)+beta(6)*sind(theta);
y0t = -beta(5)*sind(theta)+beta(6)*cosd(theta);
g = beta(1)-x0t;
h = beta(2)-y0t;
t0 = beta(1)-beta(3)-(beta(2)-beta(4)).*g./h;
J = zeros(length(theta),6);
J(:,1) = (beta(4)-y0t)./h;
J(:,2) = -(beta(1)-x0t).*(beta(4)-y0t)./h.^2;
J(:,3) = -1;
J(:,4) = g./h;
J(:,5) = -(beta(2)-beta(4)).*...
          (beta(6)-beta(1)*sind(theta)-beta(2)*cosd(theta))./h.^2;
J(:,6) = (beta(2)-beta(4)).*...
          (beta(5)-beta(1)*cosd(theta)+beta(2)*sind(theta))./h.^2;
end
```

2. Download the zip-file `ExWeek3Day4_data.zip` from the course website and load one of the three mat-files. Each of the mat-files contains a regular sinogram, a pin calibration sinogram, the nominal geometry, and the projection angles. Compute a reconstruction with the nominal geometry and inspect both the reconstruction (x) and the residual image ($Ax - b$). Does the residual image tell you anything useful about your reconstruction model?

3. Estimate the geometry (β) using the LM method. Use the nominal geometry (included in the mat-file) as an initial guess.

4. Check that your estimate geometry is consistent with the calibration sinogram, i.e., plot $t_0(\theta;\hat{\beta})$ where $\hat{\beta}$ denotes your estimated parameters and check that it is consistent with the calibration sinogram.

5. Use `fanbeamtomolinearmod.m` (included in the zip-file) or ASTRA to compute a new reconstruction with your refined geometry. With `fanbeamtomolinearmod.m` (a modified version of `fanbeamtomolinear.m`) you can input the source and detector positions as follows:

```
gs = ys/(ys-yd)*dw;    % grid size in cm
Ac = fanbeamtomolinearmod(N,theta,p,[ys,xs]/gs,[dw,xd]/gs,(ys-yd)/gs);
Ac = (gs/N)*Ac;
```

   Here `R=ys` is the vertical distrance from origo to the source and `sd=ys-yd` is the vertical distance from the source to the detector.

6. Inspect both the reconstruction (x) and the residual image ($Ax - b$) obtained with the refined geometry. Compare with the reconstruction and residual image obtained with the nominal geometry. Is there a noticeable difference?

7. *(Optional)* Analyze the expression $t_0(\theta; \beta)$. Does the parameter vector β uniquely determine the geometry? In other words, if β and β' are two parameter vectors and $t_0(\theta; \beta) = t_0(\theta; \beta')$, does that imply that $\beta = \beta'$?