Computed Tomography simulation using Radon Transform (incremental angles version)

Initialization Code (optional)

Manipulate

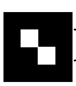
```
Manipulate[
 Row[{

   (*--------------------*)
   (* reset event        *)
   (*--------------------*)
   Dynamic[Refresh[
     If[Not[event == "reset"],
      {

       If[originalImage == 0,
        {
         originalImage = getImage[imageName, imsize];
         originalImageFFT2 = fft2[originalImage, fftOn3D,
           True, fftOn3DOriginalImageScale, fftOn3DInverseImageScale, imsizeForDisplay]
        }];

       If[isSweep,
        lastRadonImage = Radon[originalImage,
          {n, ImageDimensions[originalImage][[2]]}, {-Pi / 2, -Pi / 2 + (n) * (Pi / 128)}, Method → method],
        lastRadonImage = Radon[originalImage, {n, ImageDimensions[originalImage][[2]]},
          {-Pi / 2, Pi / 2}, Method → method]
       ];

       iradonImage =
        InverseRadon[lastRadonImage, "Filter" -> inverseMethod, "CutoffFrequency" → cutOffFrequency];
       result = ImageAdjust[iradonImage];
       iradonFFT2 = fft2[result, fftOn3D, False,
         fftOn3DOriginalImageScale, fftOn3DInverseImageScale, imsizeForDisplay];

       event = "reset"
      }
     ];

     Grid[{
       {
        Text@Row[{Style["number of projection angles = ", 12], Style[n, 12]}], SpanFromLeft
       },
       {
        Grid[{

          {
           Show[result, ImageSize → imsizeForDisplay],
           Show[ImageAdjust@ImageResize[lastRadonImage, imsize], ImageSize → imsizeForDisplay]},
```

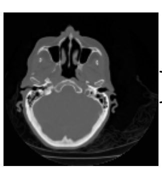
```
        {Text[Style["Inverse Radon image", 12]], Text[Style["Radon image", 12]]}
       }, Frame → None,
       FrameStyle -> Directive[Thickness[.001], Gray], Alignment → Center, Spacings → {.6, .3}]
     }
     ,
     {
      Grid[{
        {Show[iradonFFT2, ImageSize -> imsizeForDisplay],
         Show[originalImageFFT2, ImageSize -> imsizeForDisplay]},
        {Text[Style["Inverse Radon magnitude spectrum", 12]],
         Text[Style["Original image magnitude spectrum", 12]]}
       }, Frame → None, FrameStyle -> Directive[Thickness[.001], Gray],

       Alignment → Center, Spacings → {.6, .3}
      ]
     }

   }, Alignment → Center, Spacings → {0, .5}
  ],
  TrackedSymbols → {event}]
],

(*--------------------*)
(* method event       *)
(*--------------------*)
Dynamic[Refresh[event = "method";
  If[originalImage == 0,
   {
    originalImage = getImage[imageName, imsize];
    originalImageFFT2 = fft2[originalImage, fftOn3D,
      True, fftOn3DOriginalImageScale, fftOn3DInverseImageScale, imsizeForDisplay]
   }];

  lastRadonImage =
   Radon[originalImage, {n, ImageDimensions[originalImage][[2]]}, {-Pi / 2, Pi / 2}, Method → method];

  "", TrackedSymbols → {method, imageName}]
],

(*--------------------*)
(* trigger event      *)
(*--------------------*)
Dynamic[Refresh[event = "n"; "", TrackedSymbols → {n}]],

(*--------------------*)
(* fftOn3D event      *)
(*--------------------*)
Dynamic[Refresh[event = "fftOn3D";
  If[originalImage == 0,
   {
    originalImage = getImage[imageName, imsize];
    lastRadonImage =
     Radon[originalImage, {n, ImageDimensions[originalImage][[2]]}, {-Pi / 2, Pi / 2}, Method → method]

   }];
  originalImageFFT2 = fft2[originalImage, fftOn3D,
    True, fftOn3DOriginalImageScale, fftOn3DInverseImageScale, imsizeForDisplay];

  "", TrackedSymbols → {fftOn3D}]],

(*-------------------------*)
(* cutOffFrequency event   *)
(*-------------------------*)
Dynamic[Refresh[event = "cutOffFrequency"; "", TrackedSymbols → {cutOffFrequency}]],
```

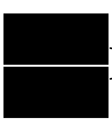
```
  (*-------------------------*)
  (* isSweep event           *)
  (*-------------------------*)
  Dynamic[Refresh[event = "isSweep"; "", TrackedSymbols → {isSweep}]],

  (*-------------------------*)
  (* inverseMethod event     *)
  (*-------------------------*)
  Dynamic[Refresh[event = "inverseMethod"; "", TrackedSymbols → {inverseMethod}]],

  (*-------------------------------------*)
  (* fftOn3DOriginalImageScale event     *)
  (*-------------------------------------*)
  Dynamic[Refresh[event = "fftOn3DOriginalImageScale";
    If[originalImage == 0,
     {
      originalImage = getImage[imageName, imsize];
      lastRadonImage =
       Radon[originalImage, {n, ImageDimensions[originalImage][[2]]}, {-Pi / 2, Pi / 2}, Method → method]
     }];

    originalImageFFT2 = fft2[originalImage, fftOn3D,
      True, fftOn3DOriginalImageScale, fftOn3DInverseImageScale, imsizeForDisplay];

    "", TrackedSymbols → {fftOn3DOriginalImageScale}]
  ],
  (*-------------------------------------*)
  (* fftOn3DInverseImageScale event      *)
  (*-------------------------------------*)
  Dynamic[Refresh[event = "fftOn3DInverseImageScale"; "", TrackedSymbols → {fftOn3DInverseImageScale}]],

  (*------------------*)
  (* image event      *)
  (*------------------*)
  Dynamic[Refresh[
    event = "image";

    originalImage = getImage[imageName, imsize];
    originalImageFFT2 = fft2[originalImage, fftOn3D,
      True, fftOn3DOriginalImageScale, fftOn3DInverseImageScale, imsizeForDisplay];

    lastRadonImage =
     Radon[originalImage, {n, ImageDimensions[originalImage][[2]]}, {-Pi / 2, Pi / 2}, Method → method];

    "", TrackedSymbols → {imageName}]]
 }],

(*--------------------*)
(* Start of controls  *)
(*--------------------*)

Grid[{

  {
   Framed[Grid[{

     {
      Control[{{n, 1, Style["N", 11]}, 1, maxNumberOfAngles, 1, ControlType → Trigger,
        DisplayAllSteps → True, ImageSize → Tiny, AppearanceElements → {"ProgressSlider",
          "ResetPlayButton", "PauseButton", "StepLeftButton", "StepRightButton", "ResetButton"}}]
     },
```

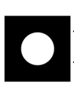
```
        {
         Control[{{cutOffFrequency, 1, Text@Row[{Style["f", Italic, 11]Style["c",Italic,11]}]},
            .01, 1, 0.01, ImageSize → Small, Appearance → "Labeled"}]
        }

      }, Spacings → {.5, .3}, Alignment → Left
    ], FrameStyle -> Directive[Thickness[.001], Gray]
   ]
  },

  {
   Framed[Grid[{

       {
        Control[{ {method, "Radon", Style["Radon", 11]},
           (# -> Style[#, 11]) & /@ {"Radon", "Hough"}, ControlType → PopupMenu, ImageSize → All}],

        Control[{ {isSweep, False, Style["incremental", 11]},
           {True, False}, ControlType → Checkbox, ImageSize → All}]

       },

       {
        Control[{ {inverseMethod, # Cos[# Pi] &, Style["Inverse", 11]},
           {(1 + Cos[# Pi]) / 2 & → Style["Hann", 11],
            1 & → Style["Rectangular", 11],
            # & → Style["Ramp-Lak", 11],
            # Sin[# 2 Pi ] & → Style["Sin Ramp", 11],
            # Cos[# Pi ] & → Style["Cosine Ramp", 11],
            ((1 - 0.16) / 2 - (1 / 2) Cos[# Pi] + 0.08 Cos[# 2 Pi]) & → Style["Blackman", 11],
            (0.355768 - 0.487396 Cos[# Pi] + 0.144232 Cos[# 2 Pi]) - 0.012604 Cos[# 3 Pi] & →
             Style["Nuttal window", 11],
            Sinc[#] & → Style["Shepp-Logan", 11],
            (.54 + .46 Cos[# Pi]) & → Style["Hamming", 11],
            Sqrt[1 / (1 + #^ (2))] & → Style["Butterworth order 1", 11],
            Sqrt[1 / (1 + #^ (4))] & → Style["Butterworth order 2", 11],
            Sqrt[1 / (1 + #^ (6))] & → Style["Butterworth order 3", 11],
            None → Style["No filter", 11]
           }, ControlType → PopupMenu, ImageSize → All}], SpanFromLeft
       }
      }, Spacings → {.5, .3}, Alignment → Left], FrameStyle -> Directive[Thickness[.001], Gray]
   ]
  },

  {
   Framed[Grid[{
       {Control[{ {imageName, "head CT", Style["image", 11]},
          (# -> Style[#, 11]) & /@ {"Lena", "Mandrill", "ResolutionChart", "2 white spots", "4 white spots",
             "Moon", "head CT", "horizontal line", "large dot", "small dot", "Numbers", "Clock",
             "Bricks", "Bubbles2", "Wall", "SheppLogan", "ellipse", "square", "vertical bar"},
          ControlType → PopupMenu, ImageSize → All}]},
       {Dynamic[Show[originalImage, ImageSize → 180]]}
      }, Alignment → Center, Spacings → {0, .5}], FrameStyle -> Directive[Thickness[.001], Gray]
   ]
  },

  {

   Control[{ {fftOn3D, False, Style["show fft in 3D", 11]}, {True, False}, ControlType → Checkbox }]

  },

  {
```

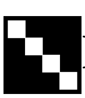
```
  Grid[{
     {
      Style["original image spectrum y-scale", 11]
     },
     {
      Control[
        { {fftOn3DOriginalImageScale, .8, ""}, 0.01, 1, .01 , ImageSize → Small, Appearance → "Labeled"}]
     }
    }, Spacings → .5, Alignment → Left, Frame → All, FrameStyle -> Directive[Thickness[.001], Gray]
  ]
 },

 {
  Grid[{
     {
      Style["inverse radon spectrum y-scale", 11]
     },
     {
      Control[
        { {fftOn3DInverseImageScale, .8, ""}, 0.01, 1, .01 , ImageSize → Small, Appearance → "Labeled"}]
     }
    }, Spacings → .5, Alignment → Left, Frame → All, FrameStyle -> Directive[Thickness[.001], Gray]
  ]
 }
}],

{{lastRadonImage, 0}, ControlType → None},
{{originalImage, 0}, ControlType → None},
{{event, "reset"}, ControlType → None},
{{result, 0}, ControlType → None},
{{originalImageFFT2, 0}, ControlType → None},
{{iradonFFT2, 0}, ControlType → None},
{{maxNumberOfAngles, 128}, ControlType → None},
{{imsize, {256, 256}}, ControlType → None},
{{imsizeForDisplay, {200, 200}}, ControlType → None},
{{iradonImage, 0}, ControlType → None},
```

{{im1, }, ControlType → None},

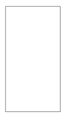
{{im2, }, ControlType → None},

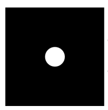
{{im4, }, ControlType → None},

{{im5, }, ControlType → None},

{{im6, }, ControlType → None},

{{im7, }, ControlType → None}, {{im8,  }, ControlType → None},

{{im9,  }, ControlType → None},

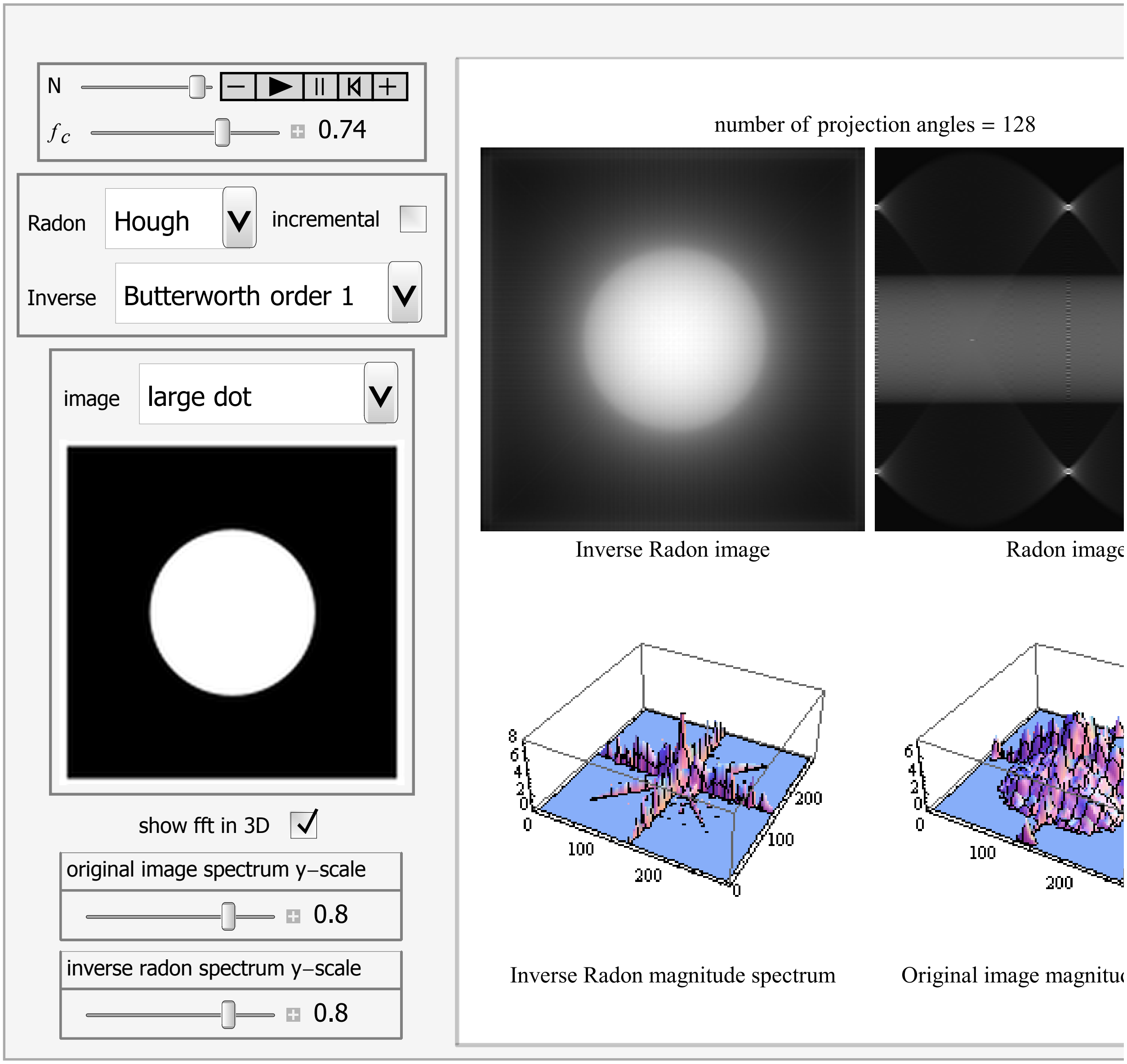
{{im10,  }, ControlType → None},

{{im11,  }, ControlType → None},

```
ContinuousAction → False,
SynchronousUpdating → True,
TrackedSymbols → {None},
Alignment → Center,
ImageMargins → 0,
ControlPlacement → Left,
Initialization :→
 (
  (*--------------------*)
  (*  read image        *)
  (*--------------------*)
  getImage[imageName_String, imsize_] := Module[{returnedImage},

    Which[imageName == "2 white spots", returnedImage = im1,
     imageName == "4 white spots", returnedImage = im6,
     imageName == "head CT", returnedImage = im2,
     imageName == "horizontal line", returnedImage = im4,
     imageName == "large dot", returnedImage = im5,
     imageName == "SheppLogan", returnedImage = im7,
     imageName == "ellipse", returnedImage = im8,
     imageName == "square", returnedImage = im9,
     imageName == "small dot", returnedImage = im10,
     imageName == "vertical bar", returnedImage = im11,

     imageName == "Bricks", returnedImage = ExampleData[{"Texture", imageName}],
     imageName == "Bubbles2", returnedImage = ExampleData[{"Texture", imageName}],
     imageName == "Wall", returnedImage = ExampleData[{"Texture", imageName}],

     True, returnedImage = ExampleData[{"TestImage", imageName}]
    ];

    If[ ImageChannels[returnedImage] == 3, returnedImage = Image[ImageData[returnedImage][[All, All, 1]]]];
```



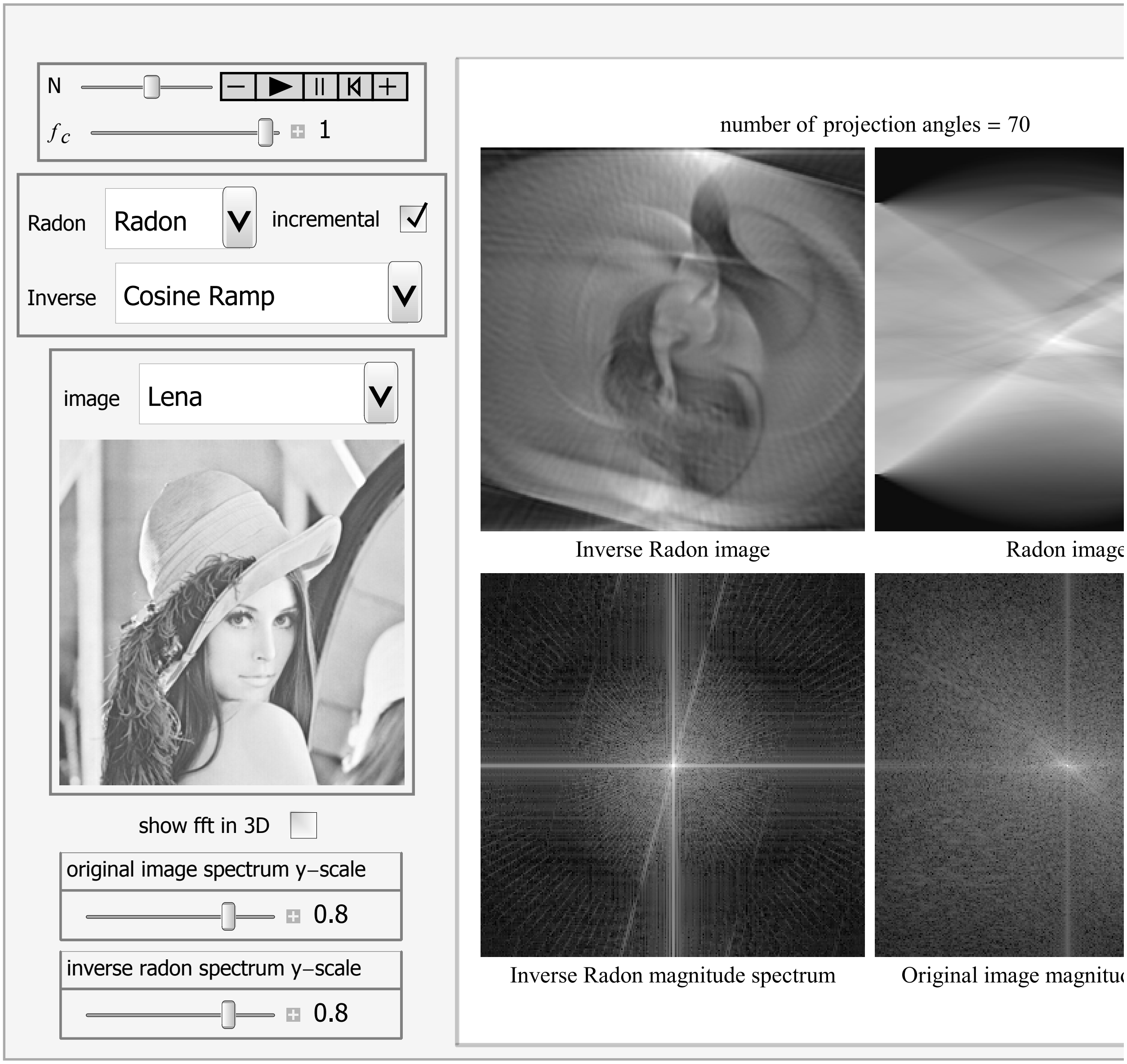
```
      ImageResize[returnedImage, imsize]
     ];

    (*--------------------*)
    (* find FFT2          *)
    (*--------------------*)
    fft2[theImage_, fftOn3D_, isOriginal_, yForOriginal_, yForRadon_, imsizeForDisplay_] :=
     Module[{data, i, j, nRow, nCol, fw, tb, scale},

      data = ImageData[theImage][[All, All]];
      {nRow, nCol} = Dimensions[data];
      tb = N@Table[i + j, {i, nRow}, {j, nCol}];
      data = data * Power[-1, tb];

      fw = Fourier[data, FourierParameters → {1, 1}];

      scale = If[fftOn3D,
        1,
        If[isOriginal, 100 * yForOriginal, 10 * yForRadon ]
       ];

      abs = scale * Log[0.1 + Abs@fw];

      If[fftOn3D,

       Image@ListPlot3D[abs, MaxPlotPoints → 60, ImageSize → imsizeForDisplay,
         PlotRange → {All, All, {0, If[isOriginal, yForOriginal Max[abs], yForRadon Max[abs]]}},
         ImageMargins → 2, ImagePadding → 20],

       ImageAdjust[Image[abs]]

      ]
     ]
   )

 ]
```

Caption

Simulation of Computed Tomography (CT) is illustrated by applying Radon transform on the selected image with an increasing number of projection angles and using the inverse Radon transform on the resulting Radon image to show that the reconstructed image starts to resemble the original image as the number of projection angles increases. This version has incremental angle option.

Snapshots



Details (optional)

The larger the number of backprojections used, the more accurate the reconstructed image will be. In CT, many projections are first generated of the object at different angles before applying backprojection in order to reconstruct an image of the 2D structure of an internal cross section of the image.

In this simulation, the Radon transform, added in *Mathematica* 8, is used to simulate this process, by taking up to 128 projections between $[-\pi/2, \pi/2]$ and then applying inverse Radon transform on the resulting images.

Only one backprojection is used for each angle. The magnitude spectrum of the reconstructed image (the inverse Radon image) is updated as more backprojections are made, showing that the spectrum approaches that of the original image. Cosine ramp filter for inverse Radon transform found to generate the clearest reconstruction, however, streak lines appear across the reconstructed image which are not there in the non filtered image.

The parameter N in the UI represents the number of projections or angles to use. The parameter f_c is the CutoffFrequency option for the inverse Radon. The scale slides can be used to adjust the 2D frequency spectrum of the images for better viewing. A checkbox is available to change the view of the image magnitude spectrum from 2D to 3D.

Reference: Kak, A.C., and M. Slaney, Principles of computerized tomographic imaging. IEEE press. 1988.

Reference: Hugh Murrell, Computer-Aided Tomography, the *Mathematica* Journal, 1996.

Reference: Nasser M. Abbasi, report on computed tomography, school project, California State University, Fullerton, 2008

Reference: Wikiperdia window functions

Control Suggestions (optional)

- ☐ Slider Zoom
- ☐ Drag Locators
- ☐ Rotate and Zoom in 3D
- ☐ Automatic Animation
- ☐ Gamepad Controls
- ☐ Resize Images
- ☐ Bookmark Animation

Search Terms (optional)

computed tomography

Radon

FFT

backprojection

Related Links (optional)

Radon transform

Authoring Information

Contributed by: Nasser M. Abbasi