Computed Tomography Simulation Using the Radon Transform

Initialization Code (optional)

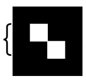
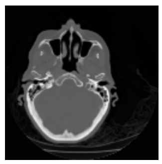
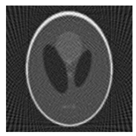
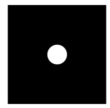
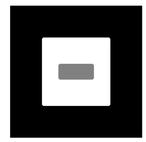
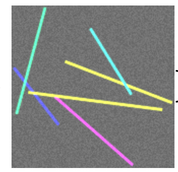
Manipulate

```
Manipulate[

 Row[{
   Dynamic[Refresh[

     If[originalImage === 0,
      {
       originalImage = getImage[imageName, imsize, images, keepColor];
       originalImageFFT2 = fft2[originalImage, fftOn3D,
         True, fftOn3DOriginalImageScale, fftOn3DInverseImageScale, imsizeForDisplay];
       nRowsForRadon = Round[Sqrt[(#[[1]]^2 + #[[2]]^2)]] & @ImageDimensions[originalImage]
      }];

     lastRadonImage = Radon[originalImage, {n, nRowsForRadon}, {-Pi/2, Pi/2}, Method → method];
     iradonImage =
      InverseRadon[lastRadonImage, "Filter" -> inverseMethod, "CutoffFrequency" → cutOffFrequency];
     displayResult[showSpectrum, iradonImage, originalImageFFT2, imsizeForDisplay, lastRadonImage,
      imsizeForDisplayOneImage, fftOn3D, fftOn3DOriginalImageScale, fftOn3DInverseImageScale, n],

     TrackedSymbols -> {event, method, n, cutOffFrequency,
       inverseMethod, showSpectrum, fftOn3DInverseImageScale}], SynchronousUpdating → False
   ],

   Dynamic[Refresh[event = Date[];
     originalImageFFT2 = fft2[originalImage, fftOn3D,
       True, fftOn3DOriginalImageScale, fftOn3DInverseImageScale, imsizeForDisplay];
     "", TrackedSymbols -> {fftOn3D, fftOn3DOriginalImageScale}]
   ],

   Dynamic[Refresh[
     event = Date[];
     originalImage = getImage[imageName, imsize, images, keepColor];
     originalImageFFT2 = fft2[originalImage, fftOn3D,
       True, fftOn3DOriginalImageScale, fftOn3DInverseImageScale, imsizeForDisplay];

     nRowsForRadon = Round[Sqrt[(#[[1]]^2 + #[[2]]^2)]] & @ImageDimensions[originalImage];
     lastRadonImage = Radon[originalImage, {n, nRowsForRadon}, {-Pi/2, Pi/2}, Method → method];

     "", TrackedSymbols -> {imageName, keepColor}], SynchronousUpdating → False]
  }],
 (*--------------------*)
 (* Start of controls  *)
 (*--------------------*)
 Grid[{
   {
    Grid[{
      {
```

```
     Control[{{n, 26, Style["n", 11]}, 1, maxNumberOfAngles, 1, ControlType → Trigger, DisplayAllSteps →
         True, ImageSize → Tiny, AnimationRate → Automatic, AppearanceElements → {"ProgressSlider",
          "ResetPlayButton", "PauseButton", "StepLeftButton", "StepRightButton", "ResetButton"}}]
    },

    {Control[{{cutOffFrequency, 0.50, Text@Row[{Style["f", Italic, 11]Style["c",Italic,11]}]},
        .01, 1, 0.01, ImageSize → Small, Appearance → "Labeled"}]},

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

    {Control[{ {inverseMethod, # &, 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}]}

   }, Alignment → Left, Spacings → {0, .4}
  ]
 },

 {
  Grid[{
    {

     Control[{ {imageName, "head CT", Style["image", 10]}, (# -> Style[#, 10]) & /@ {"head CT", "SheppLogan",
          "ellipse", "boxed", "Lena", "Mandrill", "ResolutionChart", "2 white spots", "Moon",
          "small dot", "Numbers", "Clock", "Bricks", "Bubbles2", "Wall", "vertical bar", "noisy image"},
        ControlType → PopupMenu, ImageSize → All}
     ],

     Control[{ {keepColor, False, Style["color", 10]}, {True, False}, ControlType → Checkbox }]
    },

    {
     Dynamic[Show[If[originalImage === 0,
        {
         originalImage = getImage[imageName, imsize, images, keepColor];
         nRowsForRadon = Round[Sqrt[(#[[1]]^2 + #[[2]]^2)]] & @ImageDimensions[originalImage];
         lastRadonImage = Radon[originalImage, {n, nRowsForRadon}, {-Pi / 2, Pi / 2}, Method → method];
         originalImageFFT2 = fft2[originalImage, fftOn3D, True,
           fftOn3DOriginalImageScale, fftOn3DInverseImageScale, imsizeForDisplay];
         originalImage
        },
        originalImage
       ], ImageSize → 180], SynchronousUpdating → False], SpanFromLeft
    }

   }, Alignment → Center, Spacings → {.5, .4}]
 },

 {
```

```
    Grid[{
       {
        Control[{ {showSpectrum, False, Style["show FFT", 10]}, {True, False}, ControlType → Checkbox }]
        ,
        Control[{ {fftOn3D, False, Style["3D view", 10]},
           {True, False}, ControlType → Checkbox , Enabled → Dynamic[showSpectrum == True]}]
       }
     }, Spacings → {4, 0}
    ]
   },

   {
    Grid[{
       {
        Style["original image spectrum y scale", 11]
       },
       {
        Control[{ {fftOn3DOriginalImageScale, .8, ""}, 0.01, 1, .01 ,
           ImageSize → Small, Enabled → Dynamic[showSpectrum == True], Appearance → "Labeled"}]
       }
     }, Spacings → .5, Alignment → Left
    ]
   },

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

 {{lastRadonImage, 0}, ControlType → None},
 {{originalImage, 0}, ControlType → None},
 {{event, "image"}, ControlType → None},
 {{originalImageFFT2, 0}, ControlType → None},
 {{maxNumberOfAngles, 128}, ControlType → None},
 {{imsize, {256, 256}}, ControlType → None},
 {{imsizeForDisplay, {200, 200}}, ControlType → None},
 {{imsizeForDisplayOneImage, {412, 450}}, ControlType → None},
 {{nRowsForRadon, 0}, ControlType → None},
 SynchronousUpdating → False,
 TrackedSymbols → {None},
 Alignment → Center,
 ImageMargins → 0,
 ControlPlacement → Left,
 AutorunSequencing → {{1, 30}},
 SynchronousInitialization → False,
 ImageMargins → 0,
 FrameMargins → 1,
 Initialization :→
```

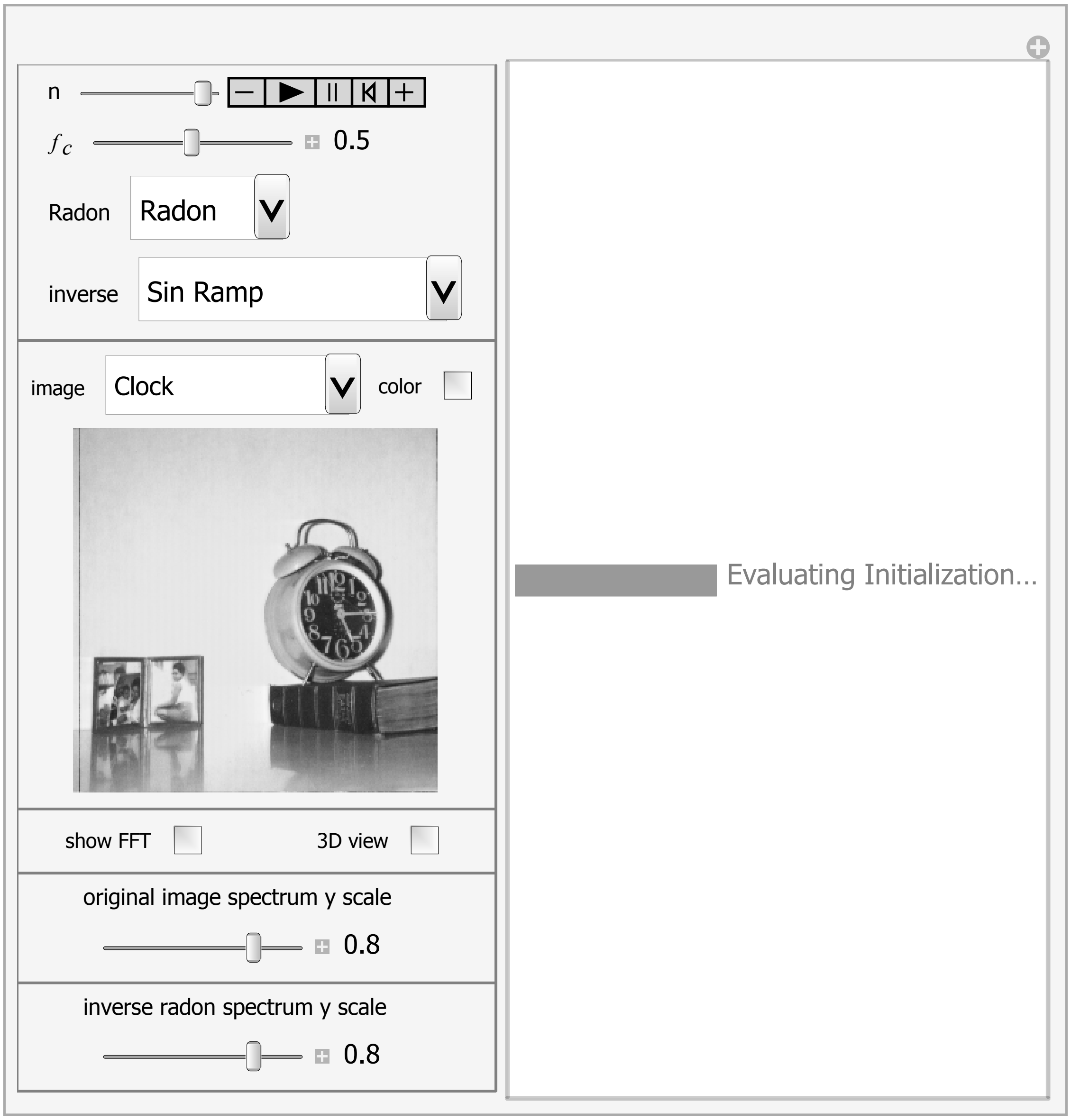
```
images =
```

{, , , , , , , };

```
(*--------------------*)
(*  read image        *)
(*--------------------*)
getImage[imageName_String, imsize_, images_, keepColor_] := Module[{returnedImage},

  Which[imageName == "2 white spots", returnedImage = images[[1]],
   imageName == "head CT", returnedImage = images[[2]],
   imageName == "SheppLogan", returnedImage = images[[3]],
   imageName == "ellipse", returnedImage = images[[4]],
   imageName == "small dot", returnedImage = images[[5]],
   imageName == "vertical bar", returnedImage = images[[6]],
   imageName == "boxed", returnedImage = images[[7]],
   imageName == "noisy image", returnedImage = images[[8]],

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

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

  (*Only use one channel to reduce memory on radon/inverse radon use later on*)
  (*unless user wants color*)
  If[ ImageChannels[returnedImage] == 3 && Not[keepColor],
   returnedImage = Image[ImageData[returnedImage][[All, All, 1]]]
  ];

  (*reduce the size to reduce memory need for future operations*)
  ImageResize[returnedImage, imsize]

 ];

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

  data =
   If[ImageChannels[theImage] == 3, ImageData[theImage][[All, All, 1]], ImageData[theImage][[All, All]]];

  (*center image before fft*)
  {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[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]]

  ]
 ];

(*--------------------*)
(* displayResult      *)
(*--------------------*)
displayResult[showSpectrum_, iradonImage_, originalImageFFT2_,
  imsizeForDisplay_, lastRadonImage_, imsizeForDisplayOneImage_, fftOn3D_,
  fftOn3DOriginalImageScale_, fftOn3DInverseImageScale_, n_] := Module[{},

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

         {
          Show[ImageAdjust[iradonImage], ImageSize → imsizeForDisplay],
          Show[ImageAdjust[ImageResize[lastRadonImage, 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[fft2[iradonImage, fftOn3D, False, fftOn3DOriginalImageScale,
             fftOn3DInverseImageScale, imsizeForDisplay], 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}
   ],
   Grid[{
     {
      Text@Row[{Style["number of projection angles = ", 12], Style[n, 12]}]
```

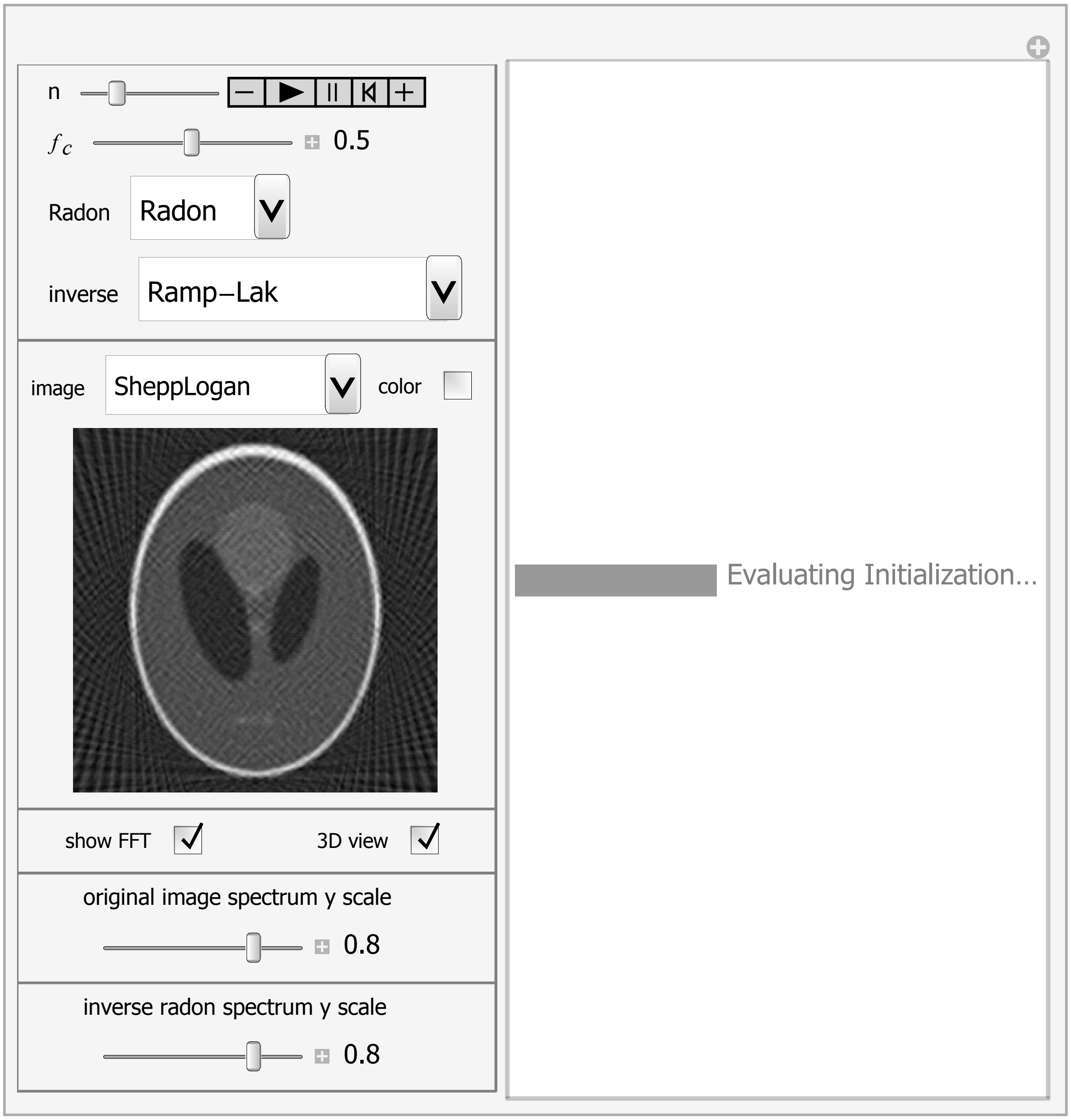
```
      },
      {
       Show[ImageAdjust[iradonImage], ImageSize → imsizeForDisplayOneImage]
      },
      {
       Text[Style["inverse Radon image", 12]]
      }
     }, Alignment → Center, Spacings → {0.6, 0.3}
    ]
   ]
  ]
```



Caption

This Demonstration illustrates Computed Tomography (CT). It applies the Radon transform on the selected image with a number of projection angles. Using the inverse Radon transform on the resulting Radon image shows that the reconstructed image starts to resemble the original image as the number of projection angles increases.

Snapshots



Details (optional)

The larger the number of projections applied on the original image, the more accurate the reconstructed image becomes. In computed tomography, many projections of the object are first generated from different angles. Then filtered back-projections are applied to reconstruct 2D image of the structure of a particular cross-section of the image. This is the basic idea used in X-ray medical imaging. The Radon transform and the inverse Radon transform (both added in *Mathematica* 8) are used to simulate this method. Up to 128 projections can be taken between $[-\pi/2, \pi/2]$. Then applying the inverse Radon transform on the resulting image gives the filtered back-projection image.

In this Demonstration, only one filtered backprojection is used per projection. The magnitude spectrum of the reconstructed image (the inverse Radon image) is updated as more backprojections are applied, showing that the spectrum approaches that of the original image. Ram–Lak and cosine ramp filters for inverse Radon transform generate the clearest reconstruction; however, streak lines appear across the reconstructed image; these do not appear in some of the other filters nor for the nonfiltered image.

The "n" slider represents the number of projections or angles to apply. The parameter f_c is *Mathematica*'s "`CutoffFrequency`" option for the inverse Radon. You can adjust the 2D frequency spectrum of the images for better viewing. A checkbox lets you change the view of the image magnitude spectrum from 2D to 3D. For original color images, a checkbox can be used to process the image in gray

only, as processing the image in color requires more time and memory.
References:

A. C. Kak and M. Slaney, Principles of Computerized Tomographic Imaging, New Brunswick, NJ: IEEE Press, 1988.

H. Murrell, Computer-Aided Tomography, *The Mathematica Journal*, 1996.

N. M. Abbasi, Report on Computed Tomography, school project, California State University, Fullerton, 2008.

Wikipedia, Window Functions

Control Suggestions (optional)

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

Search Terms (optional)

computed tomography

Radon

FFT

backprojection

Related Links (optional)

Radon transform

Authoring Information

Contributed by: Nasser M. Abbasi