Applications of Computed Tomography Techniques to Radiation Therapy – An Investigation

Evan Knouse

Physics 491

Final Report

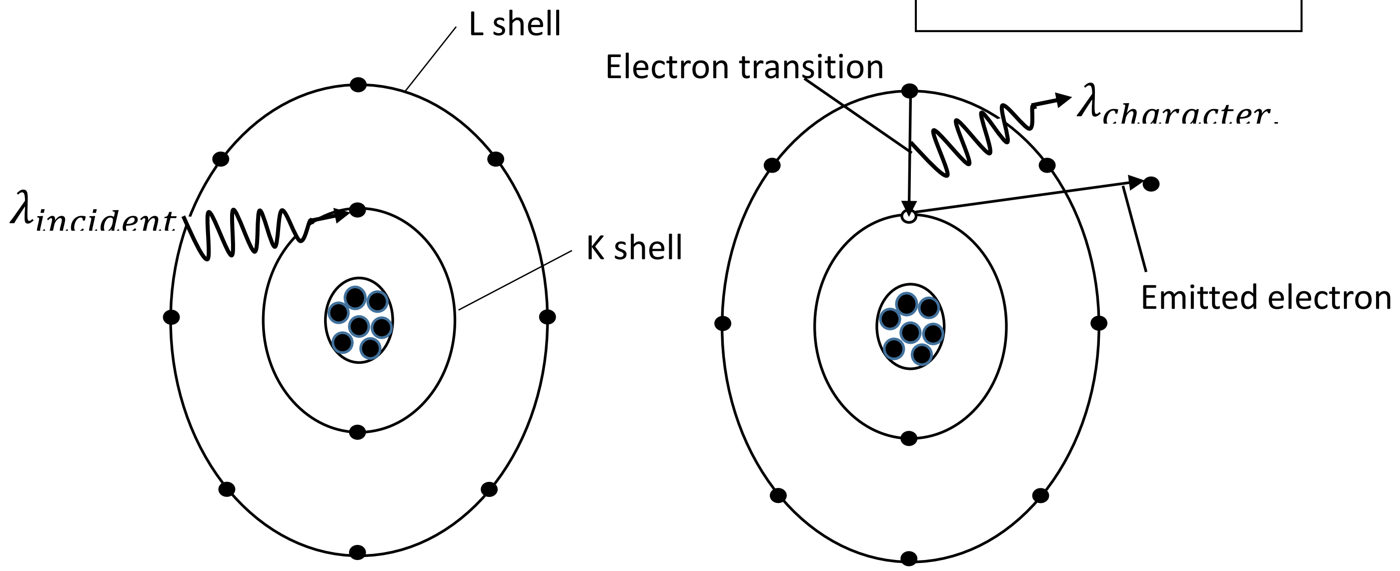
The utility of x-ray radiation was observed almost immediately upon discovery by Wilhelm C. Röntgen, when the first x-ray image was taken of his wife's hand on December 22nd, 1895 [1]. The discovery of the x-ray's ability to illuminate the interior of human physiology prompted the mathematical and conceptual development of computed tomography (CT). The invention of CT by Alann M. Cormack and Godfrey N. Hounsfield, both of whom shared the 1979 Nobel Prize in physiology or medicine for their work, contributed immensely to the fields of diagnostic and interventional medicine [1]. The therapeutic use of high energy electromagnetic and particulate radiation in the treatment of cancer, commonly referred to as radiation therapy or radiotherapy, benefits greatly from applications of computed tomography. The primary advantage of computed tomography's use in radiotherapy is the precision and accuracy afforded in locating and characterizing anomalies in the physiology to be treated. Perhaps more subtly, the innovations required to beneficially manipulate images from computed tomography data, such as those innovations in signal processing which pertain to image filtering, may also provide further enhancement of radiotherapy procedures. Application of the signal processing techniques from CT to radiotherapy requires a discussion of the mathematics of convolution, the radon transform, and x-ray attenuation. Naturally, the physical properties necessary to provide radiotherapy, or perform CT, are fundamentally a result of the character of the interactions between particulate or electromagnetic radiation and matter. By understanding the attenuation characteristics of materials and high energy radiations, we are allowed a method of inferring the structure of visible-light-impermeable objects. It is the same attenuation dynamics of materials and radiations which provide us with a method of dose deposition in radiotherapy.

The problem of image quality is of paramount importance to CT. Because of the various sources of photons reaching the detector, the image quality is dependent on many factors. When an x-ray beam is directed through a sample, many of the incident photons are either scattered, absorbed, or pass right through [1]. The photons that are scattered or pass right through are detected by the detector array.

However, those photons which have been scattered out of the incident beam will not contribute positively to the resulting image. The reason for the image degrading effect of scattered photons is that many of these will eventually reach the detector, but at random angles and with random intensity [1]. This has the effect of creating noisy, blurry, images, which reduce structural contrast. More work must be done to improve the diagnostic quality of the raw image. To improve the image, CT reconstruction algorithms generally employ some form of signal filter. The specific form of the filter affects its functionality. Development of the signal filters used to improve CT images requires the understanding of the mathematics of convolution. We will defer a thorough discussion of convolutions and their usage to a subsequent section of this paper. For now, we have arrived at the intent of this investigation. The intent of this paper is to provide one example of application of the signal processing tools employed in CT scanning procedures to radiation therapy. The problem of blur in the CT image reconstruction process also arises in the treatment of tumors, as it is inherently a result of the physical properties of the interactions of high energy radiations with matter. The dose diffusion around the target dose deposition point can, theoretically, be remedied with a few tricks borrowed from CT signal filtering. To explore the concept, we will first thoroughly discuss the mechanisms of x-ray attenuation in matter. Specifically, the energy range of interest for this study is from 10 to 100 kiloelectronvolts (keV). Next, we will discuss the mathematics of the radon transform, and convolution, as they apply to computed tomography. This will lead us naturally to a discussion of the methods and mathematics of computed tomography. Finally, with all concepts laid out explicitly, we can examine the problem of dosage diffusion, or blur, in radiation therapy. As well, we will provide a cost/benefit analysis of the use of dosage filtering. To tackle this problem, we must first put all the concepts into place.

The mechanisms of x-ray attenuation in matter provide the essential physics necessary to our discussion. The primary attenuation processes for the 10 – 100 keV x-ray energy range, or (roughly) the diagnostic energy range, are the Photoelectric Effect, Coherent scattering, and Compton scattering [1]. A generalized view of the Photoelectric Effect is shown in Figure 1:

Figure 1: Photoelectric Effect



In the Photoelectric Effect, a photon imparts all its kinetic energy to a target electron. The electron will be emitted from its host atom with energy equal to that of the incident photon, minus the binding energy of the electron to the atom. For the photoelectric effect to occur, the incident photon must be of greater energy than the binding energy of the electron to its host atom [1]. The binding energy is dependent on the atomic number, Z, of the host atom, as well as which electron orbital shell the photoelectron was emitted from [1]. Higher atomic number elements will tend to hold onto inner shell electrons much more closely than their outer shell electrons, due to the stronger coulombic attraction of the nucleus. This causes binding energy to increase with shell proximity to the nucleus [1]. The ejected electron ionizes the host atom, leaving a vacancy in its place. Electrons from higher orbital shells will transition to fill the vacancy, emitting a photon in the process. The photon which is emitted in the electron transition will have an energy equivalent to the difference between the binding energies of the orbital shells. Since these energy differences are characteristic of each atomic species, the emitted photon, known as a characteristic x-ray, serves to identify the elemental composition of target samples.

The photons in the incident x-ray beam which fulfill the criteria for the photoelectric effect to take place will effectively be removed from the beam, and will not be detected. However, the characteristic x-rays emitted by the electronic transitions undergone by target atoms have the potential to reach the detector, but will likely result in further photoabsorption, or Coherent and Compton scattering (to be discussed) events. The probability of the Photoelectric Effect occurring is approximately proportional to $\frac{Z^3}{E^3}$, where Z is the atomic number of the target material, and E is the energy of the incident photon [1]. Thus, materials of higher atomic number Z will result in an increased number of photoabsorption events. This is particularly useful in CT, where contrast is concerned, because the photoelectric effect has an increased probability of occurring in bone, for example, as well as showing higher contrast between materials of different atomic number Z [1]. In Coherent scattering, an incident photon will cause the excitation of the entire electron cloud surrounding a target atom, rather than individual electrons [1]. Once the incident photon dissipates all its energy, the target atom will radiate the excess energy as a photon. In general, the radiated photon will be of the same energy as the photon which caused the excitation. However, the radiated photon is usually emitted at a slightly different angle than the angle of the incident photon [1]. The reemitted photon will continue to have further interactions as it traverses the material, just as in the Photoelectric Effect. The radiated photon, if detected, will not contribute positively to the image, and will occur as noise. In the diagnostic energy range, the process of Coherent scattering will only account for at most 12% of interactions [1]. The most dominant interaction mechanism for x-rays in the diagnostic energy range is Compton scattering. A schematic of the Compton scattering process is shown in Figure 2. In Compton scattering, incident photons interact primarily with outer shell electrons of target atoms [1]. When the incident photon strikes the target electron, the kinetic energy imparted to the electron will cause it to be ejected from its orbital, ionizing the atom. The incident photon will scatter off in a different direction, with a slightly depleted energy. Due to conservation of energy, the energy of the incident photon is divided between the ejected electron and

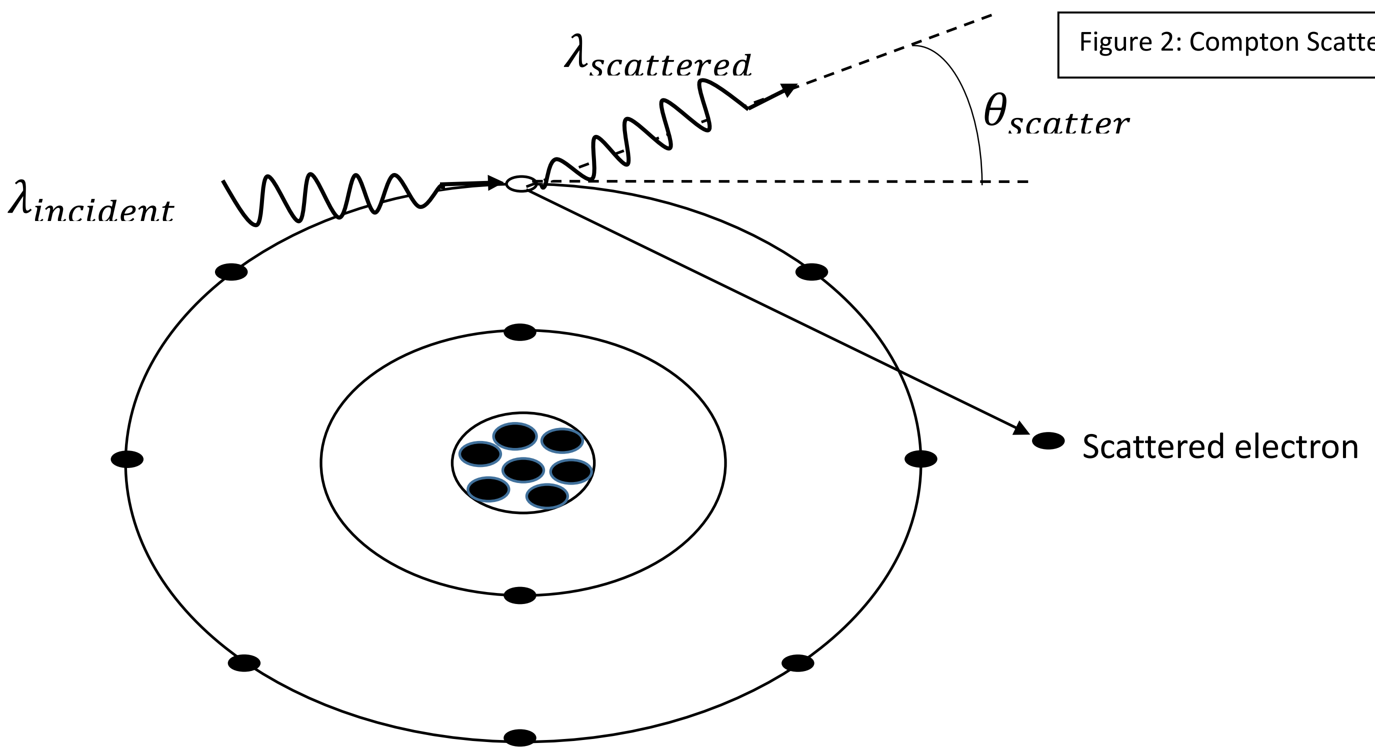
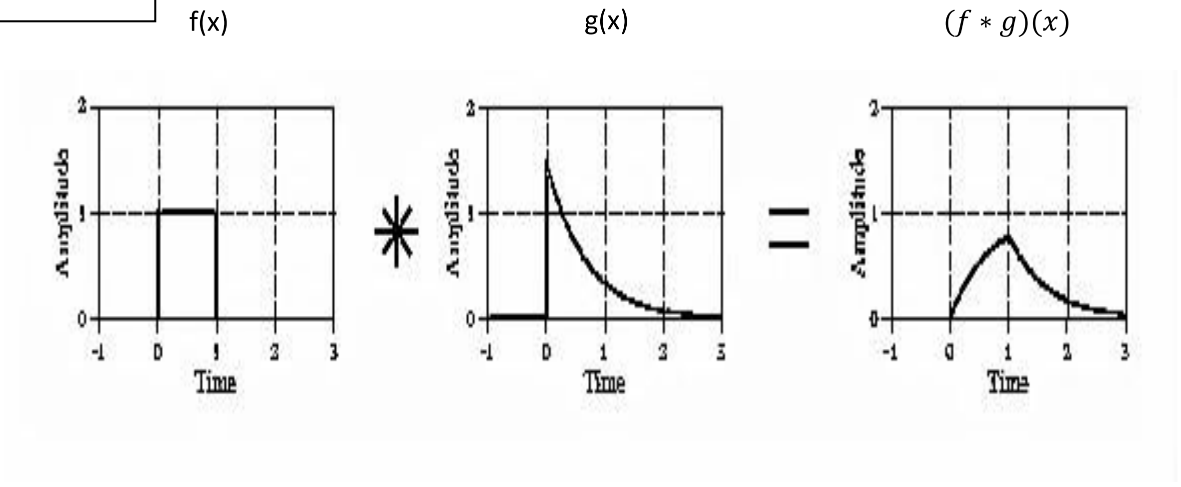


Figure 2: Compton Scattering

scattered photon, while the binding energy of the electron to the atom is negligible compared to the interaction energy [1]. After scattering, the photon will continue through the sample, causing Photoelectric effect, Coherent scattering, or further Compton scattering events. For increasing incident photon energies, the angle that the photon scatters through with respect to the angle of incidence will tend towards the forward direction [1]. The photons scattered in this way are more likely to reach the detector, which has a negative effect on the contrasting abilities of the primary photon [1]. The probability of a Compton scattering event to occur is, in effect, proportional to the electron density (electrons/gram x material density) of the material, with an increased likelihood of occurrence in hydrogenous materials [1]. All the interaction processes described previously culminate in a heuristic description of x-ray attenuation, described by the following equation: $I_{out} = I_{in}e^{-\mu x}$, where μ is the average attenuation coefficient of the material along the path of the x-ray, and x is the distance travelled by the x-ray through the material. The equation equates the transmitted intensity of x-ray radiation, I_{out}, with the incident x-ray beam intensity, I_{in} [1]. Using this equation, i.e. measuring the incident and transmitted intensity of the x-ray(s), the attenuation along the path of the x-ray(s) can be determined. The process of inferring the x-ray- beam-attenuating effects of different materials along the path of x-rays through tissue and bone, for example, forms the basis of computed tomography. This process will be explained in a succeeding section of this paper, now that we are equipped with an understanding of the mechanisms and character of x-ray attenuation. It is now time to understand the mathematics of the convolutions and the Radon transform.

To begin this portion of our discussion, we will examine the mathematical technique of convolution. A convolution is a mathematical object of the form: $(f * g)(x) = \int_{-\infty}^{\infty} f(p)g(x-p)dp$, where the functions $f(x)$ and $g(x)$ are functions for which the Fourier transform exists [2]. The requirement that the Fourier transform exists is necessary due to the relationship between convolution and the Fourier transform: $\mathfrak{F}(f * g) = \sqrt{2\pi}\mathfrak{F}(f)\mathfrak{F}(g)$, where $\mathfrak{F}(f)$ and $\mathfrak{F}(g)$ are the Fourier transforms of the functions f and g, respectively [2]. The details of the Fourier transform are not the intent of this discussion, and as such will intentionally be downplayed. The property we will take note of, however, is the Fourier transforms ability to transform a function from the spatial (or time) domain to the frequency domain. Truthfully, the only reason we require this property is that many computations which are inhibitive in their difficulty in the spatial domain become rudimentary in the frequency domain. The convolution is often one such computation which is easier to perform in the frequency domain, as it is reduced to a multiplication of Fourier transforms. Once the Fourier transforms of f and g have been performed, one simply needs to perform the inverse Fourier transform of their product to obtain the convolution. To understand what happens when we perform a convolution, a visual depiction of the process is shown in Figure 3 [3]. As we can see in the figure, the convolution of two functions is a type of weighted sum of the functions, resulting in a new function which is a combination of the

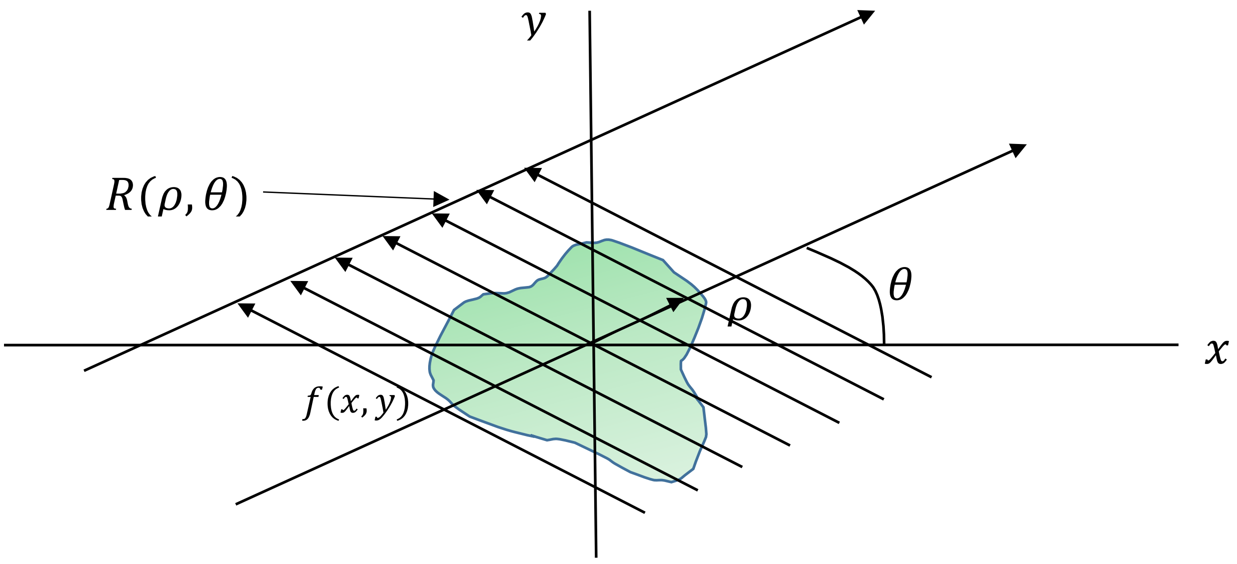
Figure 3: Convolution



characteristics of the two functions. In CT, convolutions provide a means of filtering the images formed for various purposes, such as edge enhancement, smoothing, noise reduction, or blur removal. Being able to manipulate CT images allows us to produce diagnostically useful images from somewhat "messy" data. The theoretical motivation of the investigation at hand is now evident. Where the filtering of images is done digitally for CT, it is hopeful that we could design a functional absorbing filter in physical space which would manipulate the dosage distribution in radiation therapy procedures, using the same techniques of signal filtering. As we will see shortly, the other important mathematics we require for CT is the Radon transform. The mathematics of the Radon transform was developed by its namesake inventor, Johann Radon, in 1917 [2]. Radon proposed that by taking parallel ray views of an unknown object, from multiple angles, and performing the line integral along each of the rays, the profile of the unknown object could be determined. The mathematics which describe this procedure are now known as the Radon transform, and one form of the transform is as follows [4]:

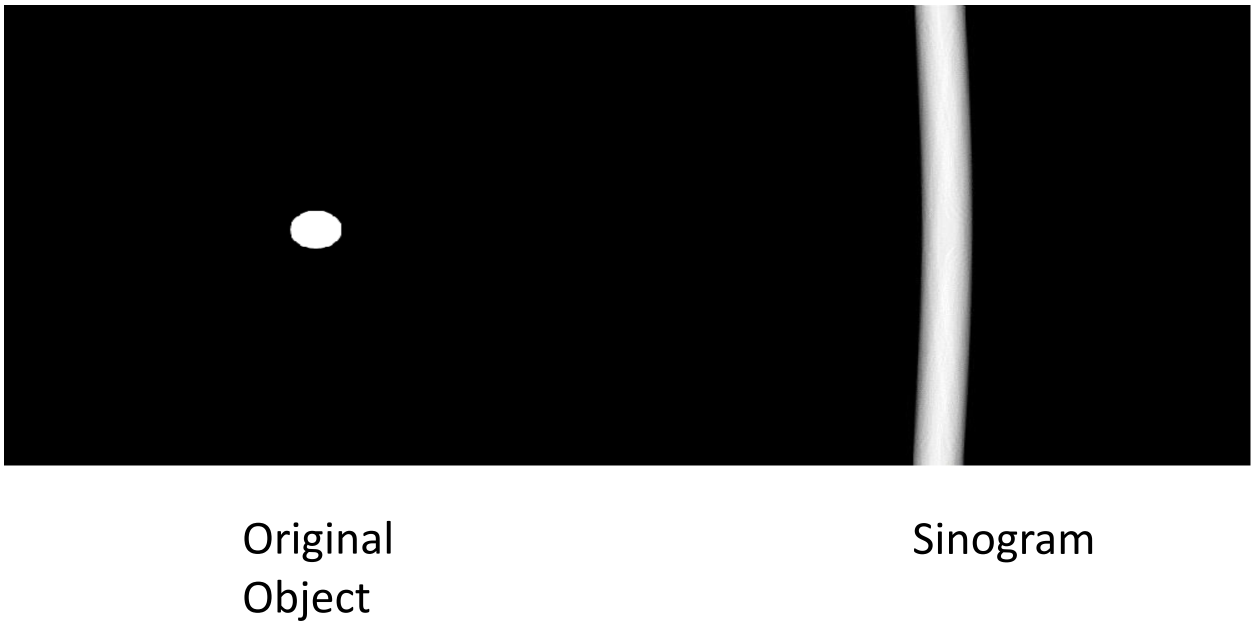
$R(\rho,\theta) = \iint_{-\infty}^{\infty} f(x,y)\delta(\rho - x\cos\theta - y\sin\theta)dxdy,$ where ρ is the distance of individual projections to the origin, θ is the angle at which the projections are offset from the origin, and $f(x,y)$ is the twodimensional function describing the profile of the object. An image of the Radon transform geometry is shown in Figure 5. Using x-rays to acquire the image, the projections through the image will contain

Figure 5: Radon Transform Geometry



the attenuation data obtained as the radiation passes through the object. In this way, the Radon transform is representative of the projection data from CT. A plot of the Radon transform, or in CT, the projection data, is commonly called a sinogram. An image of a sinogram is shown in Figure 4.

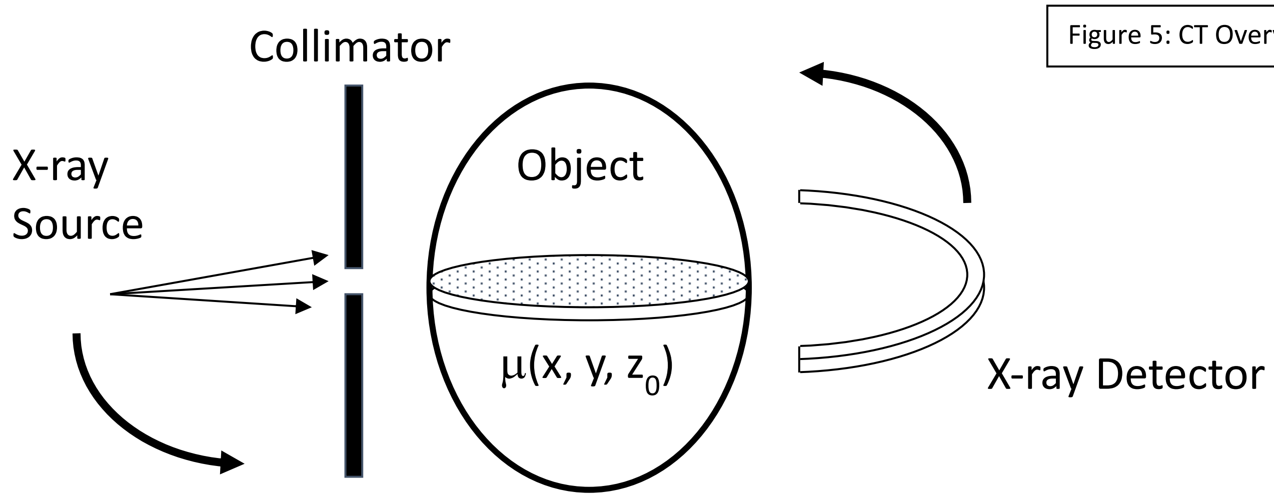
Figure 4: Sinogram



The prevalence of sinograms in CT will become evident soon. With the mathematics of convolution and the Radon transform in place, we have arrived at our discussion of Computed Tomography.

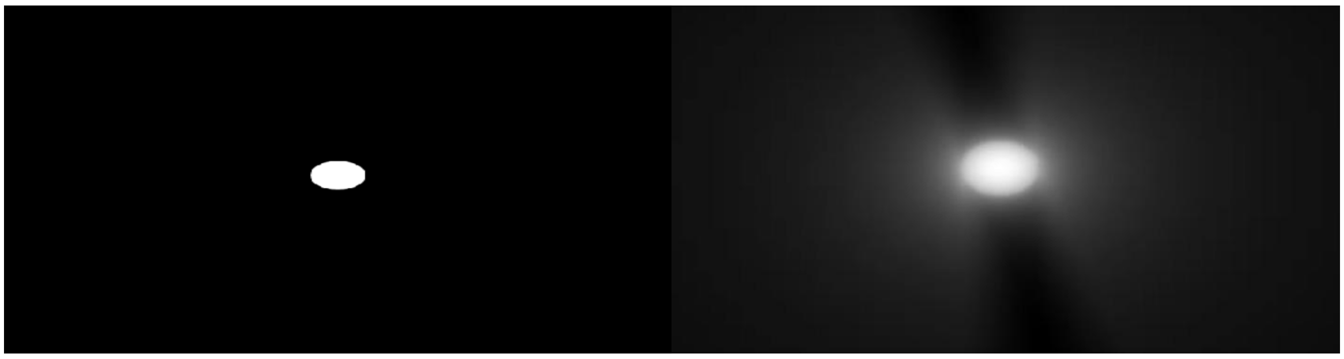
A basic schematic of a typical CT set up is shown in Figure 5. The x-ray source is positioned and collimated to direct photons through the target in slices of thickness z_0. The target slice can be described as a distribution of materials with attenuation properties at each point, $\mu(x, y, z_0)$. As the x-rays traverse the slice, the materials present will attenuate them, and they will eventually be transmitted and detected on the other side of the target. A limited number of detectors are used to detect all the transmitted x-rays, and the value detected corresponds to what is called a "ray" [1]. This is a way of organizing where the x-rays are coming from. The intensity of the transmitted x-ray beam is

Figure 5: CT Overview



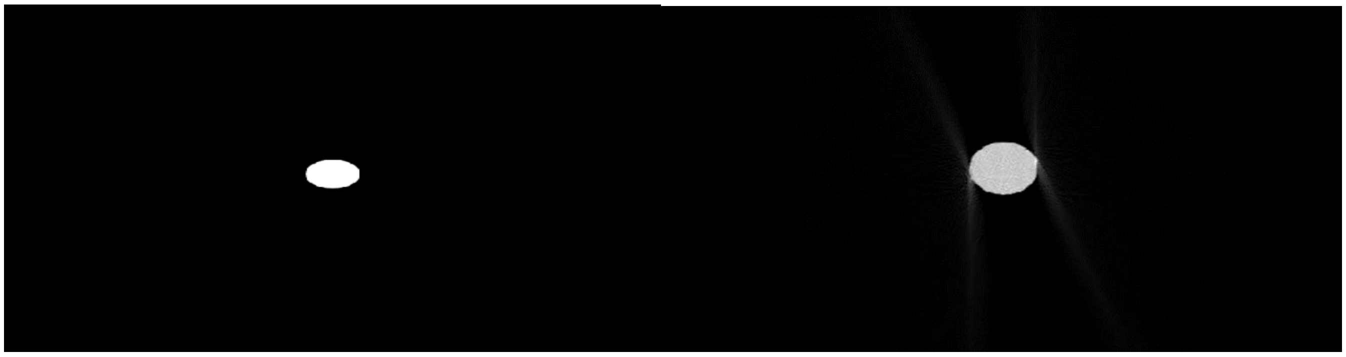
measured across the detector array, and the data is stored digitally on a computer. Next, the entire transmission process is repeated for a slightly different angle of incidence through the sample, by rotating the x-ray source and detector apparatus. As we might guess, this is where the Radon transform is performed in CT. The raw data obtained from all the projections through the sample can be plotted as a sinogram, as in Figure 4. After data acquisition, the computer determines the attenuation along each ray in the slice by using the equation $I_{out} = I_{in}e^{-\mu x}$, manipulated for μ. After the computer has determined the attenuation along each ray, a reconstruction algorithm is employed to obtain an image of the slice. Although there are numerous approaches to reconstruction, the most commonly employed reconstruction algorithm is called filtered back-projection [1]. The process of reconstructing an image is, upon first guess, merely as simple as performing the acquisition process in reverse [1]. That is, once the attenuation value of each ray is computed, the value is “smeared” back across the image plane, creating lines in which the pixel value (in greyscale) corresponds to the attenuation coefficient [1]. Once this is performed for each angle at which projections were taken, the image we are seeking should be formed. However, this process results in the adverse, diffuse-blur property, as shown in Figure 6. To counteract this blur, a convolution filter is used (discussed previously). Once the blur has been “filtered” out, back-

Figure 6: Blurred Projection Image



projection can be performed. The areas of higher attenuation tend to reinforce, and similarly for the areas of lower attenuation [1]. The effects can be seen in Figure 7 (compare to Figure 6).

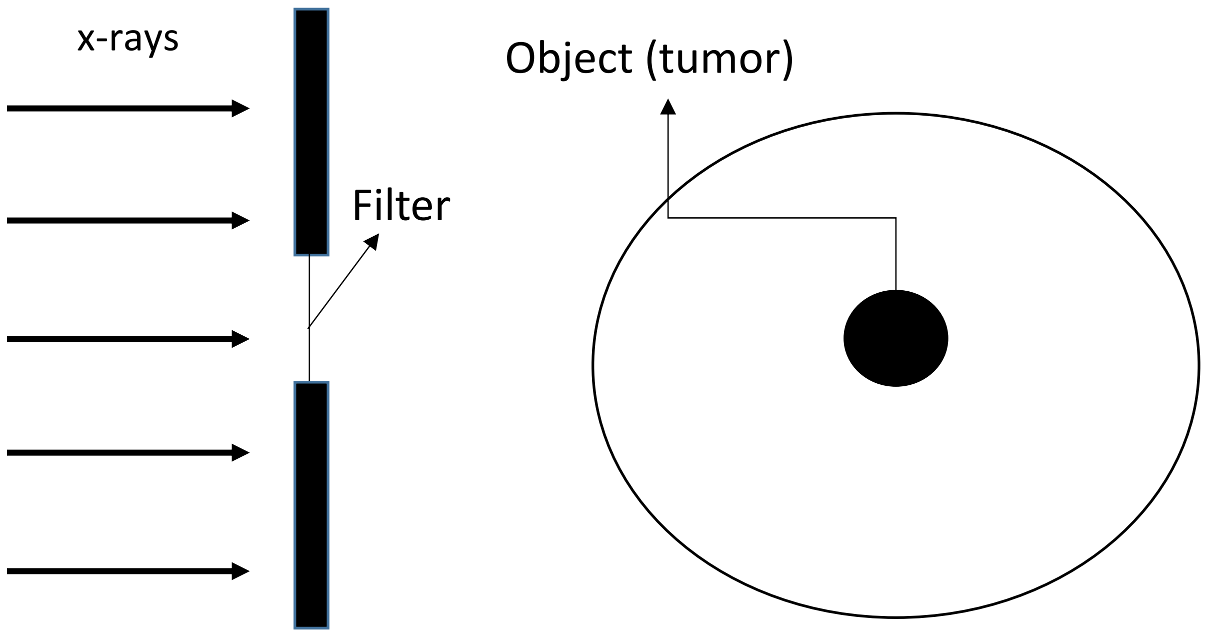
Figure 7: Filtered Projection Image



The improvement in image quality produced by applying filter functions, via convolutions, is invaluable to diagnostic radiology, as we can see. If we can imagine that the image formed by radiation projections in CT could be correlated with radiation dosage in radiotherapy, we would desire the same method of improving the "sharpness" of the dosage distribution. Finally, we have arrived at the research done to investigate this possibility.

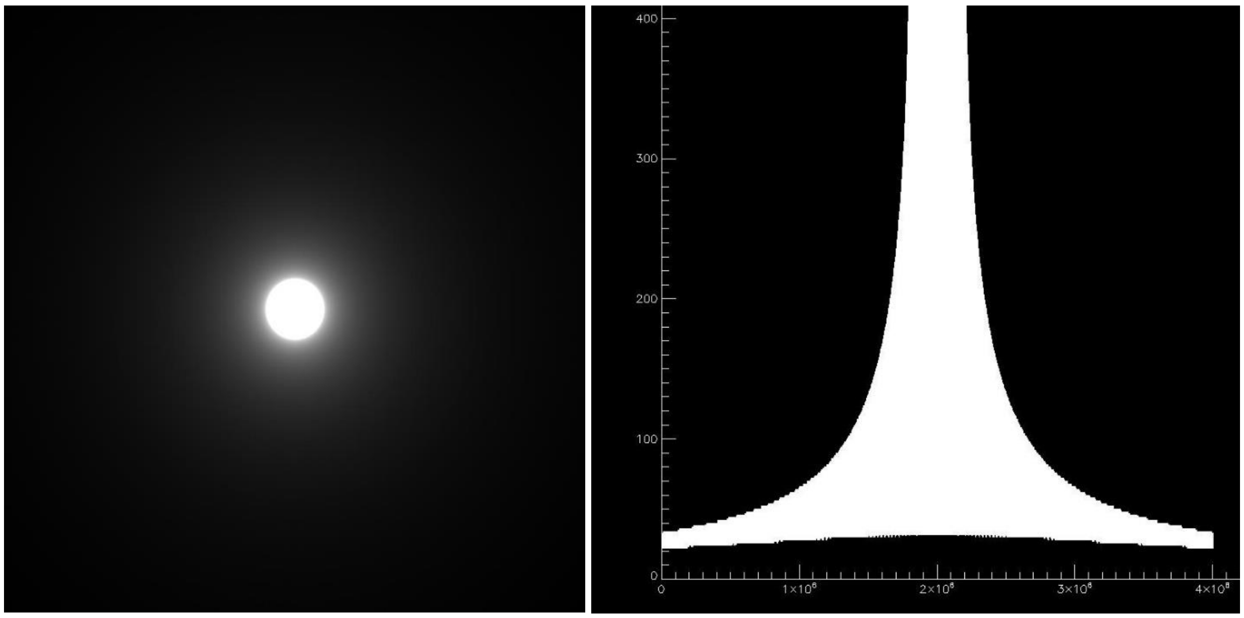
Borrowing our ideas from CT, we sought, theoretically, the design of a physical absorbing object which, when placed in front of a beam of radiation, would improve the dosage distribution deposited into a sample. To investigate this, several different pieces of code were developed and used to simulate a model situation. The code used is included as an appendix to this paper. The procedure goes as follows. Using the IDL developing platform, code was created to emulate dose deposition to a small tumor. The model situation is shown in Figure 8.

Figure 8: Simulation Visualization



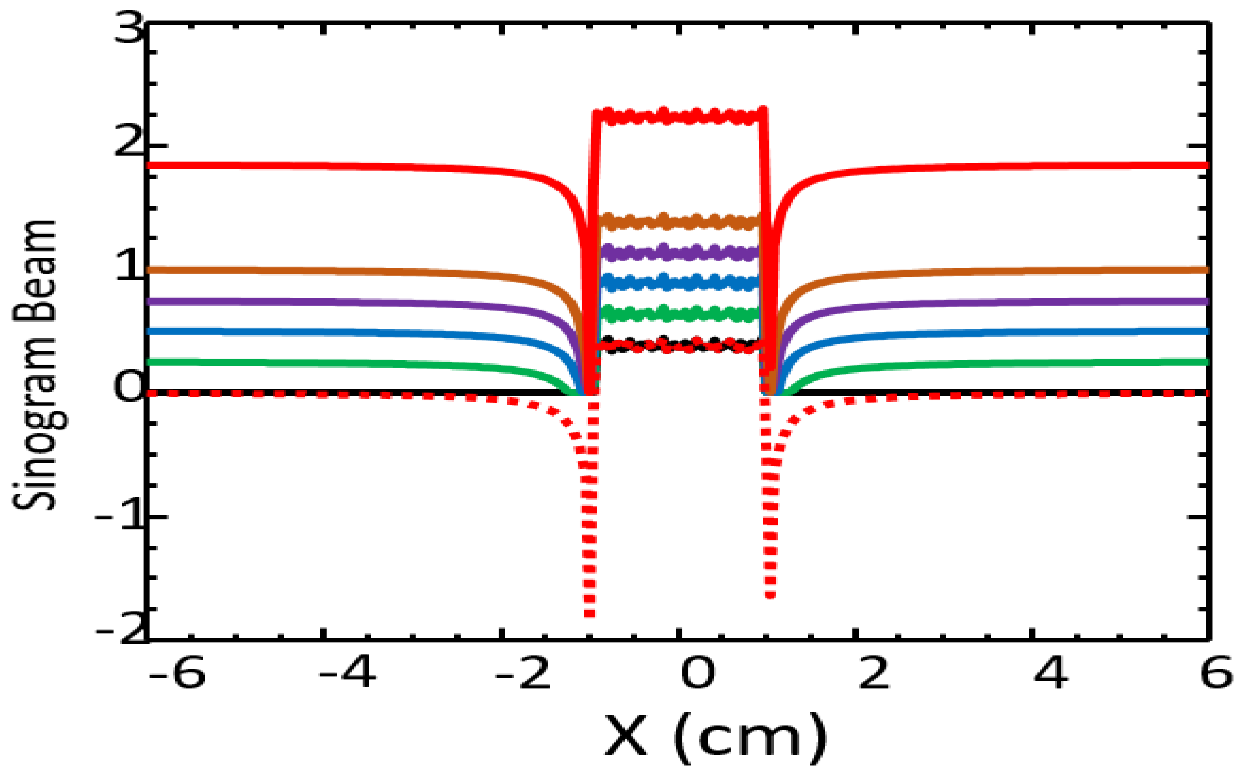
In the simulation, no notion of dimensions is necessary to properly interpret the results. However, conceptually, it helps to think of the tumor model as a circular anomaly of approximately 2 cm in diameter. The tumor model used was perfectly circular, for simplicity. As well, the x-ray source was simulated to produce a parallel beam geometry, for simplicity, with a size of 20 cm across, to give a sense of scale. The x-rays were collimated into a beam of 2 cm across, to match the diameter of the tumor. Then, the code was used to simulate the effect of rotating the x-ray source around the target, as would be the case in a realistic radiotherapy procedure. The resulting simulation is shown in Figure 9.

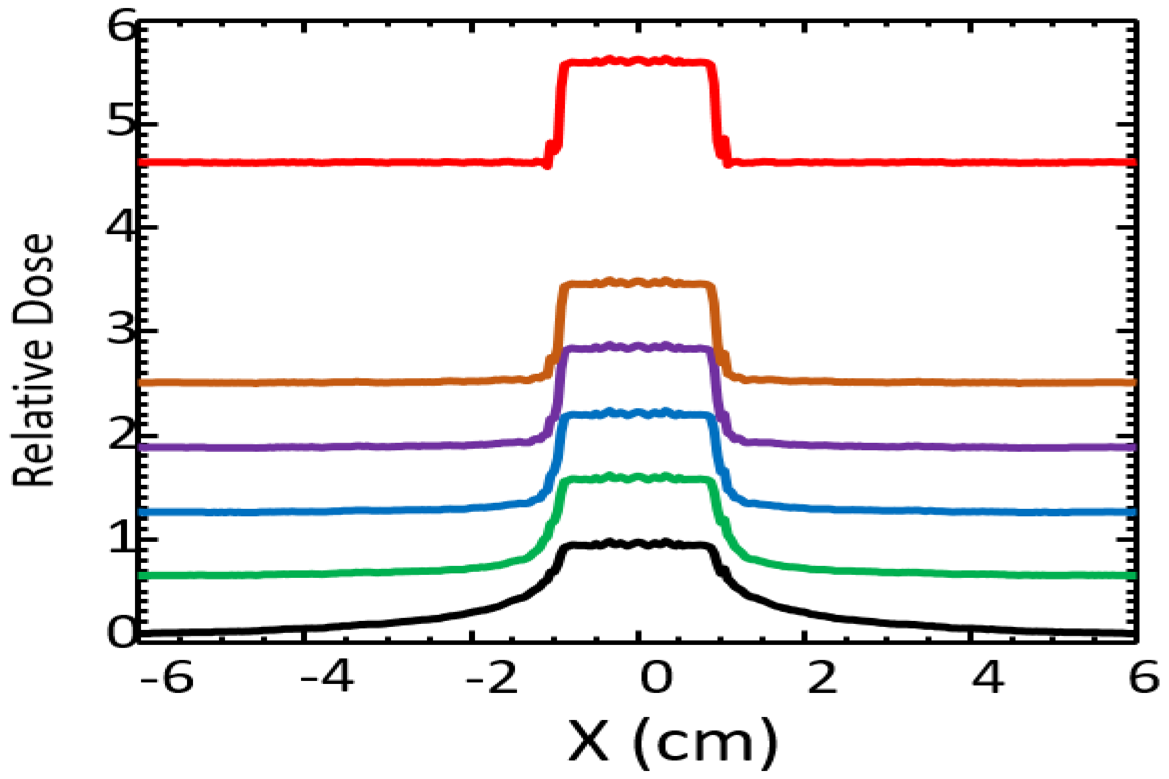
Figure 9: Model Dose



The plot in the above-right figure shows the diffuse dosage blur characteristic of unfiltered projections of radiation through an object, in this case our model tumor. This is the exact problem we are seeking to correct. In a real-life radiotherapy situation, if the central peak of this plot was the target dose deposition, i.e. the bulk of a tumor, much of the outlying area would presumably be healthy tissue. To deliver maximum dosage to the tumor, while sparing healthy surrounding tissue, we would try our best to mitigate the blur, or diffusion, of radiation deposition around our target. Using the filtering techniques described previously, an attempt was made to do just that. The height of the distribution is our only reference to scale, as we equate this approximately with number of x-rays deposited. In this way, subsequent plots can be interpreted as showing relative dose. A slight tangent to the discussion is necessary here, to explain the type of filter being used. The filter used to manipulate the dosage distribution is a sinc(x) function (sin(x)/x). The filter function must be truncated in such a way as to prevent negative values, unlike the natural sinc(x) function. This is because the negative values correspond to negative x-ray flux values, which is not realistic. Back to the matter at hand, by applying the ideal ramp filter – a sinc(x) function – whose Fourier transform is the rectangular function, we arrived at the following dose profiles in Figure 10.

Figure 10: Filtered Beam – Relative Dose





The sinogram which produces the plot above cannot have negative values, otherwise the situation would become unphysical. Because of this property, we can see that the filtering applied to sharpen the dosage distribution requires drastic increases in the relative dose deposition. This is likely not desirable in most radiation therapy situations. As such, we are forced to accept that although we can sharpen the dosage distribution, its utility in real-life radiation therapy is detrimental to patient health, and should be abandoned.

In summary, despite the promise of this technique for manipulating dose distributions, the results are not feasible. In a real-life radiotherapy situation, the dose deposited in a patient must be maintained at safe levels before all else. Using this technique to sharpen a dosage deposition profile was primarily intended to preserve healthy tissue, or provide some advantage where tissues with higher radiosensitivity might surround a target. However, as we have shown, the benefit of sharpening the dose distribution is outweighed heavily by the required increase in relative dose throughout the patient.

This conclues our investigation.

Bibliography

[1] – “The Essential Physics of Medical Imaging”, Jerrold T. Bushberg, et. al, 2nd ed., Lipincott,Williams, and Wilkins, 2002

[2] – “Advanced Engineering Mathematics”, Erwin Kreyszig, 10th ed., John Wiley & Sons, Inc., 2011

[3] – “The Scientist and Engineer’s Guide to Digital Signal Processing”, Steven W. Smith, California Technical Publishing, 2011

[4] – *"The Radon Transform - Theory and Implementation"*, Peter Toft, Ph.D. thesis, Department of Mathematical Modelling, Technical University of Denmark, June 1996. http://petertoft.dk/PhD/

Appendix – code routines

- Create field of view tumor irradiation model

```
; field of view is 20 cm
; tumor is 2cm in the middle


  ; Create 2D mesh for domain

  x    = mesh(-10, 10, 2001) & ux = fltarr(2001) + 1.0
  y    = mesh(-10, 10, 2001) & uy = fltarr(2001) + 1.0

  box  = fltarr(2001,2001)

  r2   = (x#uy)^2+(ux#y)^2

  iins = where(sqrt(r2) le 10.0)

  angles = mesh(0, 2.0*!pi, 501)

  beam = fltarr(2001)
  ibeam = where(abs(y) le 1.0)

  beam(ibeam) = 1.0

  sbar = ux#beam

  for ia=0,499 do box = box + rot(sbar,angles(ia)/!dtor)

  tvs, box
```

- Perform CT reconstruction, to create sinogram and filter

```
pro make_ct_reconstruction_from_24bit_bmp, bmp_24bit, nxy=nxy,
no_filter=no_filter
if n_elements(nxy) eq 0 then nxy=501 ;size of the "squared" imput image,
sinogram & recons
loadct, 0
loop:
if n_elements(bmp_24bit) eq 0 then bmp_24bit =
dialog_pickfile(path='C:\IDL\IDL Data\Fun\',filter='*.bmp')
if n_elements(bmp_24bit) eq 0 then goto, loop
pth   = file_dirname(bmp_24bit,/mark)
bf    = file_basename(bmp_24bit)
name  = file_basename(bf,'.bmp')
print, ' Making movie directories...'
ot1   = pth + 'Movie\Sinogram\'  & print, '     Made Sinogram Directory:',ot1
ot4   = pth + 'Movie\FBP2\'      & print, '     Made FBP Recon
Directory:',ot4
ot2   = pth + 'Movie\Before-Sino-After\'
;ot3   = pth + 'Movie\FBP\'

file_mkdir, ot1, ot2, ot4
logo  = read_bmp(pth+bf)
blogo = congrid(float(reform(logo(0,*,*))),nxy,nxy)
glogo = congrid(float(reform(logo(1,*,*))),nxy,nxy)
rlogo = congrid(float(reform(logo(2,*,*))),nxy,nxy)
phi   = mesh(0.0,180.0,nxy)+180.0
ux    = fltarr(nxy)+1.0 & uy = ux & x = findgen(nxy) & y = x
rcr   = sqrt(((x-nxy/2)#uy)^2 + (ux#(y-nxy/2))^2)
iz    = where(rcr ge (nxy/2-1))
rlogo(iz) = 0.0 & glogo(iz) = 0.0 & blogo(iz) = 0.0
rimg  = rlogo ;congrid(rlogo,360,360)
gimg  = glogo ;congrid(glogo,360,360)
bimg  = blogo ;congrid(blogo,360,360)
orig  = bytarr(3,nxy,nxy)
window, /free, xsize=nxy, ysize=nxy
tvscl, rimg & tr=tvrd() & orig(2,*,*) = tr
tvscl, gimg & tg=tvrd() & orig(1,*,*) = tg
tvscl, bimg & tb=tvrd() & orig(0,*,*) = tb
wdelete
write_bmp, ot2+'Original.bmp', orig

sz   = size(rimg) & nx = sz(1) & ny = sz(2)
ths   = mesh(0,180.0,(1.0*nx)+1)
nth   = n_elements(ths)
; now make the sinogram
```

```
srim  = fltarr(nx,nth)
sgim  = fltarr(nx,nth)
sbim  = fltarr(nx,nth)
wx = 2*nx & wy = max([ny,nth])
; Make the Sinogram
aVid = IDLffVideoWrite(ot1+name+'_Sinogram.avi')
mVid = IDLffVideoWrite(ot1+name+'_Sinogram.mp4')
avidStream = aVid.AddVideoStream(wx, wy, 20)
mvidStream = mVid.AddVideoStream(wx, wy, 20)

for ith = 0, nth-1 do begin
  rrimg = rot(rimg,ths(ith)) & rgimg = rot(gimg,ths(ith)) & rbimg =
rot(bimg,ths(ith))
  window, /free, xsize = wx, ysize = wy
; RED
  timg = rrimg & tvscl, timg, 0
  xyouts, 20, 20, '!7h!3 =
'+strcompress(string(ths(ith),format='(F6.1)'),/remove_all)+'!uo!n', /dev,
charsize=1.5
  line = total(timg,2) & srim(*,nth-ith-1) = line & tvscl, srim, 1
  plots, (wx/2-1)+[0,0], [0,wy-1], /dev & plots, (wx/2  )+[0,0], [0,wy-1],
/dev & plots, (wx/2+1)+[0,0], [0,wy-1], /dev
  r = tvrd()
; GREEN
  timg = rgimg & tvscl, timg, 0
  xyouts, 20, 20, '!7h!3 =
'+strcompress(string(ths(ith),format='(F6.1)'),/remove_all)+'!uo!n', /dev,
charsize=1.5
  line = total(timg,2) & sgim(*,nth-ith-1) = line & tvscl, sgim, 1
  plots, (wx/2-1)+[0,0], [0,wy-1], /dev & plots, (wx/2  )+[0,0], [0,wy-1],
/dev & plots, (wx/2+1)+[0,0], [0,wy-1], /dev
  g = tvrd()
; BLUE
  timg = rbimg & tvscl, timg, 0
  xyouts, 20, 20, '!7h!3 =
'+strcompress(string(ths(ith),format='(F6.1)'),/remove_all)+'!uo!n', /dev,
charsize=1.5
  line = total(timg,2) & sbim(*,nth-ith-1) = line & tvscl, sbim, 1
  plots, (wx/2-1)+[0,0], [0,wy-1], /dev & plots, (wx/2  )+[0,0], [0,wy-1],
/dev & plots, (wx/2+1)+[0,0], [0,wy-1], /dev
  b  = tvrd()
  t  = bytarr(3,wx,wy) &  t(0,*,*) = b &  t(1,*,*) = g &  t(2,*,*) = r
  tw = bytarr(3,wx,wy) & tw(0,*,*) = r & tw(1,*,*) = g & tw(2,*,*) = b
  atime = aVid.Put(avidStream, tw)
  mtime = mVid.Put(mvidStream, tw)
;  write_bmp, ot1+'SINO_'+string(ith,format='(I05)')+'.bmp', t
  wdelete
endfor
for i=0,19 do atime = aVid.Put(avidStream, tw)
for i=0,19 do mtime = mVid.Put(mvidStream, tw)
uy  = fltarr(nx)+1
pln = fltarr(nx,nx)
aVid.Cleanup
mVid.Cleanup
;******
;FUNCTION Filtered_Back_ProjectionZ, SINO, $
;  X0=X0, DX=DX, PHI0=PHI0, width=width, rings=rings
```

```
  if n_elements(width) eq 0 then width = 9
  if n_elements(rings) eq 0 then rings = 1
  sizs = size(srim)
  if sizs(0) ne 2 then stop,"Sinogram should be 2d: x-angle."
  n = sizs(1)
  n_phi = sizs(2)


  n_half_rot = 1
  dx = 0
  x0 = (nx-1.0)/2.0   ; center of image
phi0 = !pi ;0.0
  y0 = x0
start_angle=phi0
  rx0 = findgen(nx)-(nx-1.0)/2.0 & rx = findgen(nx)-x0 & ux = fltarr(nx)+1.0
& ry = findgen(nx)-y0 & uy = ux
  r2  = sqrt((rx#uy)^2 + (ux#rx)^2) & iz = where(r2 gt x0)
  x_array = rx#uy & y_array = ux#ry
  phis  = ths*!dtor + phi0
  ua = fltarr(n_phi)+1.0
  srsino = fltarr(n,n_phi) &  sgsino = fltarr(n,n_phi) &  sbsino =
fltarr(n,n_phi)
  for ip=0,n_phi-1 do srsino(*,ip) = interpol(reform(srim(*,ip)),rx0,rx)
  for ip=0,n_phi-1 do sgsino(*,ip) = interpol(reform(sgim(*,ip)),rx0,rx)
  for ip=0,n_phi-1 do sbsino(*,ip) = interpol(reform(sbim(*,ip)),rx0,rx)
  sinog = bytarr(3,nxy,n_phi)
  window, /free, xsize=nxy, ysize=n_phi
  tvscl, srsino & tr = tvrd() & sinog(2,*,*)=tr
  tvscl, sgsino & tg = tvrd() & sinog(1,*,*)=tg
  tvscl, sbsino & tb = tvrd() & sinog(0,*,*)=tb
  write_bmp, ot2+'Sinogram.bmp', sinog
;  if keyword_set(rings) then begin
;    sav = total(ssino,2)/float(n_phi)
;    dif = sav - smooth(sav,width,/edge_truncate)
;    for ip=0,n_phi-1 do ssino(*,ip) = ssino(*,ip)-dif
;  endif
  offset = nx         ; pad by n on each end
  zero_value = !pi/4
  vector = fltarr(nx)
  odd_ones = 2*lindgen(nx/2)
  vector(odd_ones) = -1.0/(!pi*(1+odd_ones)^2)
  big_vector = [ reverse(vector), zero_value, vector ]
  ramp_matrix = fltarr(nx,nx)
  for y=0,nx-1 do ramp_matrix(0,y) = big_vector((nx-y):(nx-y)+(nx-1))
  if keyword_set(no_filter) then begin
    ramp_matrix = fltarr(nx,nx)
    for y=0,nx-1 do ramp_matrix(y,y)=1.0
  endif
  rrecon = fltarr(nx,nx) & grecon = fltarr(nx,nx) & brecon = fltarr(nx,nx)

  a1Vid = IDLffVideoWrite(ot4+name+'.avi')
  a2Vid = IDLffVideoWrite(ot4+name+'_Full.avi')
  m1Vid = IDLffVideoWrite(ot4+name+'.mp4')
  m2Vid = IDLffVideoWrite(ot4+name+'_Full.mp4')
  a1vidStream = a1Vid.AddVideoStream(nx, ny, 20)
  a2vidStream = a2Vid.AddVideoStream(wx, wy, 20)
  m1vidStream = m1Vid.AddVideoStream(nx, ny, 20)
  m2vidStream = m2Vid.AddVideoStream(wx, wy, 20)
  for i_phi=0,n_phi-1 do begin
    rprojf = ramp_matrix#srsino(*,i_phi) & gprojf =
ramp_matrix#sgsino(*,i_phi) & bprojf = ramp_matrix#sbsino(*,i_phi)
    rprofj = interpol(rprojf,rx0,rx)      & gprofj = interpol(gprojf,rx0,rx)
& bprofj = interpol(bprojf,rx0,rx)
    ; filtration
    rrecon = rrecon + rot(rprojf#uy,phis(i_phi)/!dtor) & rrecon(iz) = 0.0
    grecon = grecon + rot(gprojf#uy,phis(i_phi)/!dtor) & grecon(iz) = 0.0
    brecon = brecon + rot(bprojf#uy,phis(i_phi)/!dtor) & brecon(iz) = 0.0
    window, /free, xsize=wx, ysize = wy
    ;RED
    sim = srim & recon = rrecon
    tvscl, sim, 0 & plots, [0,wx/2-1], ny-1-i_phi, /dev, col=0 & tvscl,
recon, 1
    plots, (wx/2-1)+[0,0], [0,wy-1], /dev & plots, (wx/2  )+[0,0], [0,wy-1],
/dev & plots, (wx/2+1)+[0,0], [0,wy-1], /dev
    xyouts, 20, 20, '!7h!3 =
'+strcompress(string(ths(i_phi),format='(F6.1)'),/remove_all)+'!uo!n', /dev,
charsize=1.5
    r = tvrd()
    ;GREEN
    sim = sgim & recon = grecon
    tvscl, sim, 0 & plots, [0,wx/2-1], ny-1-i_phi, /dev, col=0 & tvscl,
recon, 1
    plots, (wx/2-1)+[0,0], [0,wy-1], /dev & plots, (wx/2  )+[0,0], [0,wy-1],
/dev & plots, (wx/2+1)+[0,0], [0,wy-1], /dev
    xyouts, 20, 20, '!7h!3 =
'+strcompress(string(ths(i_phi),format='(F6.1)'),/remove_all)+'!uo!n', /dev,
charsize=1.5
    g = tvrd()
    ;BLUE
    sim = sbim & recon = brecon
    tvscl, sim, 0 & plots, [0,wx/2-1], ny-1-i_phi, /dev, col=0 & tvscl,
recon, 1
    plots, (wx/2-1)+[0,0], [0,wy-1], /dev & plots, (wx/2  )+[0,0], [0,wy-1],
/dev & plots, (wx/2+1)+[0,0], [0,wy-1], /dev
    xyouts, 20, 20, '!7h!3 =
'+strcompress(string(ths(i_phi),format='(F6.1)'),/remove_all)+'!uo!n', /dev,
charsize=1.5
    b = tvrd()
    tw = bytarr(3,wx,wy) & tw(0,*,*) = r & tw(1,*,*) = g & tw(2,*,*) = b
;    write_bmp, ot3+'FBP_'+string(i_phi,format='(I05)')+'.bmp', t
    a2time = a2Vid.Put(a2vidStream,tw)
    m2time = m2Vid.Put(m2vidStream,tw)
    wdelete
    window, /free, xsize=nx, ysize=ny
    r = bytscl(rrecon,min=0.0,max=max(rrecon)) & g =
bytscl(grecon,min=0.0,max=max(grecon)) & b =
bytscl(brecon,min=0.0,max=max(brecon))
    r(iz) = 255b & g(iz) = 255b & b(iz) = 255b

    tn = bytarr(3,nx,ny) & tn(0,*,*)=r & tn(1,*,*)=g & tn(2,*,*)=b
    tv, tn, true=1
;    write_bmp, ot4+'FBP_'+string(i_phi,format='(I05)')+'.bmp', tw
    wdelete
    a1time = a1Vid.Put(a1vidStream, tn)
    m1time = m1Vid.Put(m1vidStream, tn)

  endfor
  included = float(shift(dist(nx),nx/2,nx/2) lt nx/2)
  r = bytscl(rimg,min=0.0,max=255.0) & g = bytscl(gimg,min=0.0,max=255.0) & b
= bytscl(bimg,min=0.0,max=255.0)
  r(iz) = 255b & g(iz) = 255b & b(iz) = 255b
  tn = bytarr(3,nx,ny) & tn(0,*,*)=r & tn(1,*,*)=g & tn(2,*,*)=b
;  write_bmp, ot4+'FBP_'+string(n_phi,format='(I05)')+'.bmp', t
  for i=0,19 do a1time = a1Vid.Put(a1vidStream, tn)
  for i=0,19 do a2time = a2Vid.Put(a2vidStream, tw)
  for i=0,19 do m1time = m1Vid.Put(m1vidStream, tn)
  for i=0,19 do m2time = m2Vid.Put(m2vidStream, tw)
;******
a1Vid.Cleanup
a2Vid.Cleanup
m1Vid.Cleanup
m2Vid.Cleanup
tmp = tn
tmp(0,*,*) = reform(tn(2,*,*)) & tmp(2,*,*) = reform(tn(0,*,*))
write_bmp, ot2+'Recon.bmp', tmp
loadct, 0
return
end
```