COMP775 Final Exam

Derek Fulton

February 13, 2018

1 Image Acquisition

1.1 For each of the following imaging tasks what kind of imaging modality would you recommend and why?

1.1.1 Assessing if a bone in the leg is fractured or not

I would use a simple X-ray because a fracture will show up nicely with good contrast from the bone. Further, the image will be quite easy to compare with another image of a non-fractured bone.

1.1.2 Determining the location of a tumor in the brain

MRI because it is non-ionizing and good for showing contrast between soft tissues. The contrast between healthy brain and tumor will be evident, as the tumor is also composed of soft tissue.

1.1.3 Planning of radiation treatment for prostate cancer

I would use MRI because of the excellent soft-tissue contrast it provides relative to a CT. In this case, the contrast of soft tissue is important because it will allow us to differentiate between the normal prostate and the tumor. Therefore, we could specifically irradiate the tumor itself and not the entire prostate.

Remark That being said, I did shadow a radiation oncologist after high school and he almost exclusively used CT scans to plan treatment for his patients. I lack the domain knowledge to know exactly why that is, as the soft-tissue contrast from the MRI seems far more useful for the case of prostate cancer.

1.1.4 Surgical planning for a total knee replacement

I would use primarily a CT scan and potentially an MRI. The CT will allow for a fundamental spatial understanding of the bone structure, which is precisely what is being treated with a total knee replacement. The MRI would serve as a precautionary measure to observe the soft tissue surrounding these bones, which could provide valuable information for the surgeon.

1.2 Describe briefly the advantages and disadvantages (give one advantage and one disadvantage for each modality with respect to the other)

1.2.1 of magnetic resonance imaging (MRI) compared to computed tomography (CT)

- An advantage of MRI over CT is that it provides excellent soft-tissue contrast
- A disadvantage of MRI compared to CT is that it requires the patient to sit still for quite a long time, sometimes up to one hour.
- An advantage of CT over MRI is that it is faster and (usually) cheaper.
- A disadvantage of CT compared to MRI is that it is ionizing.

1.2.2 of projection x-ray imaging compared to computed tomography

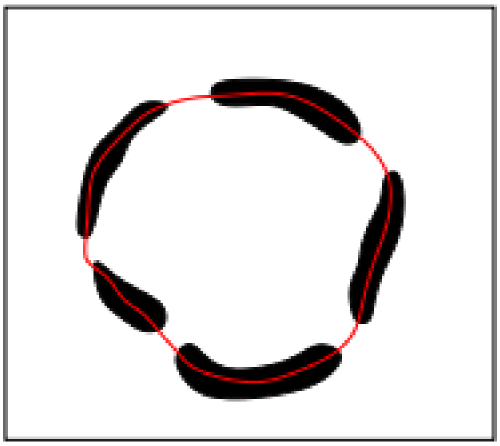
- An advantage of projection x-ray over CT is higher resolution
- A disadvantage of projection x-ray compared to CT is that it "flattens" three-dimensional data into a 2D projection
- An advantage of CT compared to projection x-ray is that is allows for a full 3D visualization from many angles
- A disadvantage of CT compared to projection x-ray is that it may deliver larger doses of radiation to the patient

1.2.3 of ultrasound compared to MRI

- An advantage of ultrasound compared to MRI is that it is instant and does not require a stationery patient
- A disadvantage of ultrasound compared to MRI is that is offers relatively poor resolution of soft tissue and is far noisier
- An advantage of MRI over ultrasound is that it preserves excellent resolution of soft tissues at greater depths from the skin
- A disadvantage of MRI compared to ultrasound is that it is orders of magnitude costlier and time-consuming

2 Segmentation

2.1 The red contour in the following image shows what should be considered a desirable segmentation result for the underlying image.



Assumptions

- The objects are of a constant intensity μ_i
- The background is also of a constant intensity μ_o
- The objects will form a reasonably clear and convex circular shape

2.1.1 Suggest two methods to accomplish such a segmentation result

Below I present two methods, a customized active contour method and a clever ellipse detection algorithm.

Customized Active Contour Method

An active contour method,

$$E(C) = \int_C g(C(s))^2 ds,$$

where $g(C(s))$ is defined such that it approaches 1 the more area of $C(s)$ exists over pixels of intensity μ_o, approaches 1 the length of $C(s)$ harshly if it is too far away from a pre-set paramater l and approaches 1 if the gradient is too large by a factor of $\gamma \|\nabla u(x)\|$. An example for a candidate $g(C(s))$ could be

$$\gamma \|\nabla C(s)\| + (C(s) - \mu_i)^2 + \beta(\int_0^1 ds - l),$$

where $\beta > 0$ and

$$l = \sum_1^{n_{components}} majorAxisLength(component).$$

Describe in detail what would need to be specified for an actual computer implementation. First we would need to quickly binarize the image and sum the major axis length of all the components. Next, we will minimize $E(C)$ with our parameter l. With this l and with $\beta = 1$ and $\gamma = 1$ we will minimize E(C), with an initialization of the largest circle possible that exists within the boundaries of the image. To do this discretely, it will be useful to employ some sort of indicator function. Further, as we have assumed the objects together make some sort of circular object, it might be convenient to start off simply shrinking the radius of the circle one step at a time until the energy decreases by more than the difference in arc length that results from shrinking the circle, or

$$|E_{n+1}(C) - E_n(C)| < 2|\pi r_n^2 - \pi r_{n+1}^2|.$$

Where the right-hand side of the inequality is multiplied by two because the arc length decreases and the number of background pixels also decreases. If this inequality is true, that means that our $C(s)$ has found some μ_i pixels. At this point, we could begin computationally optimization to minimize $E(C(s))$. One particular way to do this would be Matlab's *lsqcurvefit* function to minimize $E(C(s))$.

Discuss what challenges you expect to face when trying to achieve the desired segmentation result in practice. I suspect this method to be easier said than done. Further, finding the optimal values for β and γ would likely take some training by hand before this segmentation could be fully automated and robust.

Clever ellipse detection

A line Hough Transform uses two parameters: ρ and θ. An elliptical transform needs five, which can require a prohibitive amount of computational power. However, this has been accomplished through some clever assumptions so well that it has been implemented on mobile phones (Forniciari et al, 2014). It exploits the property of ellipses that lines orthogonal to parallel chords always pass through the center. This way there is no need to vote on every possible permutation of the five parameters, $x_0, y_0, R, rand\theta$ (the center, major axis, minor axis and tilt angle). I suspect it is possible to modify this line of reasoning to compare "parallel chords" between two objects at a time, do this for each radially adjacent pair of components and then use maximum likelihood estimation to decide on the most feasible ellipse. Note that though the desired segmentation result is curved, and an ellipse is not, there are benefits to parameterizing an image like this as an ellipse. It would allow for easy analysis of a population of these images, for example. It is less flexible in the segmentation phase but perhaps more flexible in the analysis phase.

2.1.2 Describe in detail what would need to be specified for an actual computer implementation.

First I will do another binarization of the image and label the components. To remove noise, it may be necessary to filter these components somehow (for example, a component of area 3 is likely noise) but this would require more assumptions. Next, for each component find the centroid (note if these components were not constant intensity, it may be useful to pick a weighted centroid). Pick any centroid at random and then search for the closest centroid belonging to another of the blobs. Then, do the same for the centroid that was closest to the first blob but is not the first blob's centroid. Continue this process until we end up where we started. We now have a general idea of the circular (soon-to-be elliptical) shape that is our desired segmentation.

Next, for each pair of adjacent centroids ($n_{centroids} = n_{adjacentpairs}$), take the longest diameter allowed in the original object that passes through this centroid and draw parallel chords between the two components. These parallel chords will give a rough approximation of where the center is, which will come in handy later.

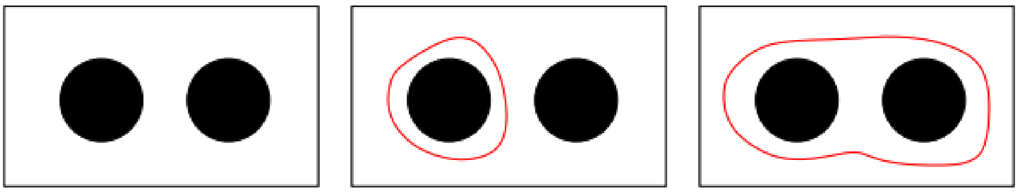
Finally we can use a function like Matlab's $lsqcurvefit$ to fit an ellipse to the points using maximum likelihood estimation (in Matlab, the function mle will accomplish this). Because we already have a good idea where the center of the ellipse will lie, we can vote at a high resolution within a small zone. This will allow the solution to be far less computationally expensive than a full 5-parameter elliptical hough transform, which would have a runtime complexity of $O(n^5)$ (which is a deal-breaker for many applications). Approximating the location of the center will decrease the runtime complexity considerably, allowing it to approach $O(n^3)$ if we know with high certainty where the center is.

After fitting the ellipse, it is theoretically possible to improve the fit to allow it to converge better toward the segmentation result mentioned in the problem. For example, we could combine the first method but with the initialization being this ellipse. If the active contour method ends up being especially expensive computationally, this might be a useful workaround.

2.1.3 Discuss what challenges you expect to face when trying to achieve the desired segmentation result in practice.

This method will fit an ellipse to the blobs in the image. However, this is also limiting because it may not necessarily be as robust as the first method, the active contour method. Another challenge is that this method rests heavily on the assumption that the object to be segmented is quite circular (obviously an ellipse-fit can still capture this if the major axis and minor axis are equal). The method will break down quite rapidly if the assumption of circularity breaks down.

2.2 The following images show an image (a) with two dark circles which should be segmented. You try a Chan-Vese with an indicator function and an active contour model.



Original Image (a) Initialization (b) Initialization (c)

2.2.1 What do you expect the resulting segmentation results for the Chan-Vese model to be for the two different initializations and why?

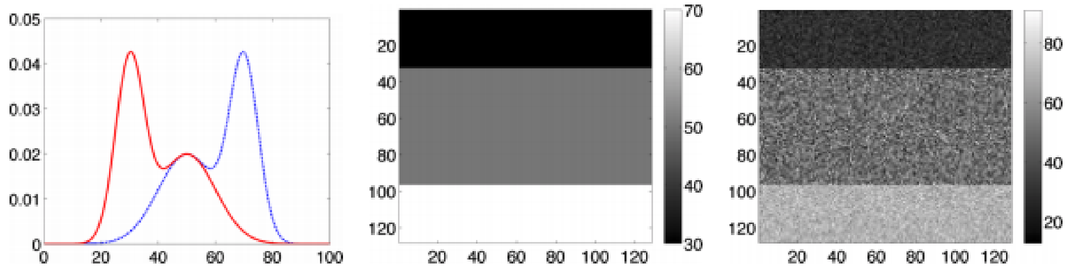
For both initializations, the Chan-Vese model will properly segment both circles even though the initialization (b) begins only around the first circle. That is because segmenting both circles will result in a lower overall energy. This is because the last term in the integral, $-(I(x) - \mu_o)^2 u$ is always negative either negative or zero. Therefore to minimize the energy we will want u to be 1 inside the circle. As the second term, $(I(x) - \mu_i)^2 u$ is always zero inside the circle and is either zero (if $u = 0$) or positive (if $u > 0$) outside the circle, we want u to be 0 outside the circle.

2.2.2 What do you expect the resulting segmentation results for the active contour model to be for the two different initializations and why?

For initialization (b) I expect the active contour model will close in on the first circle only. That is because of the nature of $g(C(s))$. As this function will approach zero as it gets closer to an edge, and it is initialized relatively close to the edge of the first circle (closer than to the edge of the second circle at all points in initialization (b)), it will be minimized when hugging the first circle.

For initialization (c) I expect that the active contour model will shrink the initialization (c) until it surrounds both circles. In the area between the circle, the contour will continue to shrink until it is a line spanning the edges of the two circles that overlaps with the imaginary line connecting the two centers of the circles.

2.3 The following images show a clean image (b) and a noisy image (c). The intensity distributions for what should be regarded foreground and background respectively are given in (a).



Distribution (a) Clean Image (b) Noisy Image (c)

2.3.1 Where do you expect the optimal segmentation boundary to be located for the noise-free case (b)? How do you expect the results to change for the noisy image (c) and what influence will the constant γ have on the solution?

Before answering this question I will make a couple of observations/assumptions. From inspecting the clean image (b) and the legend on its immediate right, it appears the bottom stripe is white (corresponding to an intensity value of 70) and the top stripe is black (corresponding to an intensity value of 30). It is certain that the middle stripe has a grayscale intensity between that of the top and bottom stripes, and it appears this intensity is roughly 50.

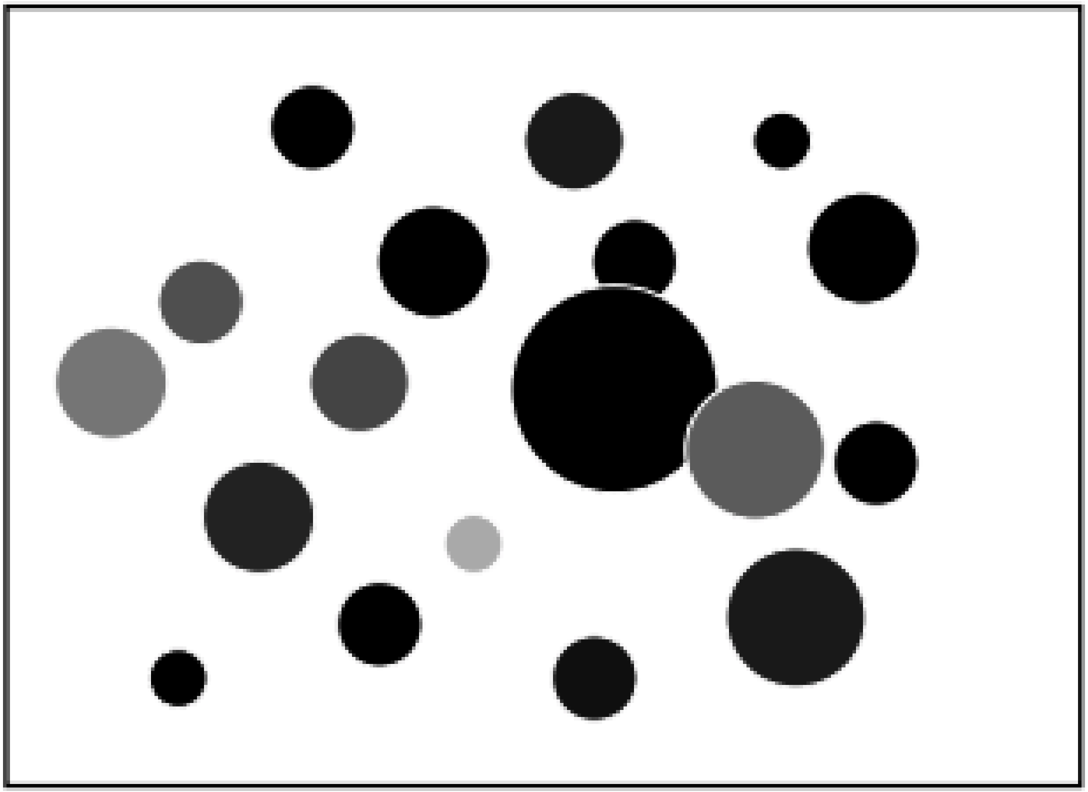
For the noise-free case (b) I suspect the segmentation boundary to exist anywhere on the bottom white stripe. This is because of the second term, $log\frac{p_{bg}(I(x))}{p_{fg}(I(x))}$. This term is greater than zero when the probability of a given pixel belonging to the background is greater than the probability of that pixel belonging to the foreground (or wherever the solid red line is above the solid blue line on distribution (a)). Correspondingly, it is less than zero when the foreground probability is greater than the background probability. It is equal to zero when the probabilities are the same, which can be seen graphically at the intersection of the red curve and the dotted blue curve.

Therefore to minimize the energy, we would like $u(x)$ to equal 1 when the foreground probability is highest. Looking at image (b) this is likely going to be on the brightest pixels, the foreground pixels with a value of 70. Turning again to distribution (a) we see this is roughly where the peak of the Gaussian probability distribution exists.

For the noisy case (c) the same analysis applies, but depending on the strength of the noise, some special cases may occur. The only way one of our high $u(x)$ values could exist on the middle gray stripe is if the noise was such that there was a cluster of pixels as bright or brighter than the foreground on the line between the gray stripe and the white stripe, and adjacent to those but on the foreground side there were pixels that were as dark or darker than the average gray stripe value. However, due to the first term $\gamma\|\nabla u(x)\|$, this is unlikely because there incurs a penalty for steep gradients.

The constant γ is a coefficient to the gradient of $u(x)$. The constant γ allows us to fine-tune the penalty associated with a steep gradient or a "bendy" line. This promotes smoother segmentations. In the noise-free case, the segmentation will be a straight line because all of the white foreground pixels on the edge between the white and the gray have a high likelihood of belonging to the foreground, and it is possible to for the segmentation boundary to be completely straight. If the image is noisy, the segmentation boundary may avoid darker pixels in the bottom stripe by "steering" away from them. The decision to steer around a darker pixel or plow through it will depend on γ. High gamma will promote the segmentation boundary to plow through the darker pixel and low gamma will not penalize steering at all.

2.4 You are given the following image of circular objects of different sizes. Your objective is to extract the center of the circles and their respective radii for subsequent statistical analysis.



2.4.1 Propose a possible approach to obtain these measures

This problem could be solved with a number of approaches. However, because the image is so simple and the edges are so well-defined, I believe edge detection followed by a circular Hough Transform could provide the best estimation of the centers and radii of the circles. Assuming noise will be overlaid onto the original image, this approach should be robust to noise up until an extreme level of noise.

- Run Canny edge detection with $\sigma = 3$.
- For each pixel (x, y), for each radius, check how many edge pixels exist
- Because there is nothing in this image besides circular objects, use the resultant accumulator array from the Hough Transform to estimate the centers and radii of the circles.

2.4.2 Discuss how you expect noise to affect the result of your proposed algorithm.

The algorithm should be reasonably resistant to noise. One possible issue from noise is less precise peaks in the accumulator array. For example, the vote score for a certain circle at (x_0, y_o) of radius r_0 may be reasonably high over a small range of radii and center locations. In this case, it might be necessary to perform some sort of maximum likelihood estimation on the region in order to find the true values (x_0, y_o) and r_0. This will incur some uncertainty that will carry over into any subsequent statistical analysis and the more noise there is, the more uncertainty there will be.

That being said, because we know as a ground-truth that the image consists solely of circular objects of different sizes, the Hough transform method should be quite effective until there is an extreme level of noise. And if there is an extreme level of noise, no method may be able to get the job done.

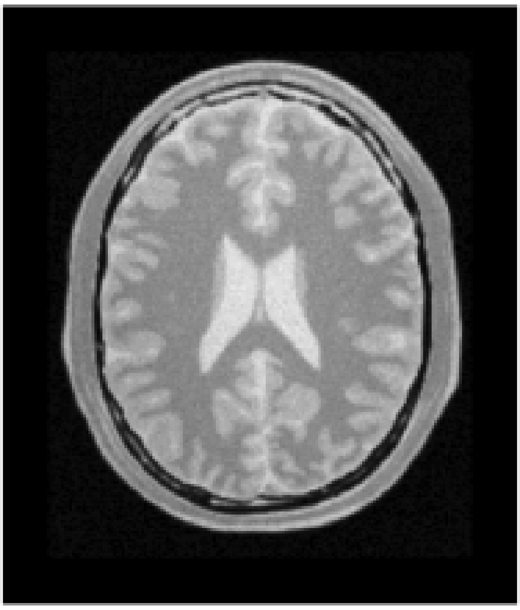
2.4.3 Discuss how you expect object overlaps to affect the result of your proposed algorithm.

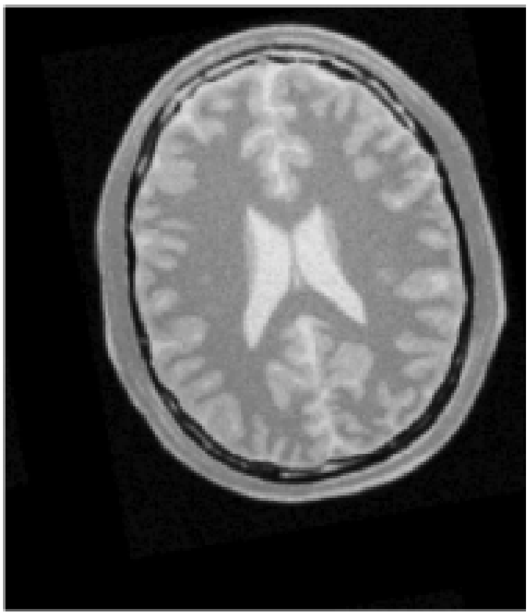
The object overlaps in this particular image are not too bad. By a quick inspection, only two circles are obscured in any way. And the proportion of these circles which is covered up is far less than half of the circle's arc. As the arc is what will be detected by the edge detector, and later the Hough Transform, the Hough Transform should be quite robust in the face of overlaps. But naturally, the bigger the overlap and the more overlaps present, the more susceptible the algorithm will be to noise.

3 Registration

3.1 Standard registration problems

3.1.1 What transformation model and what similarity measure do you suggest for image registration? Why?



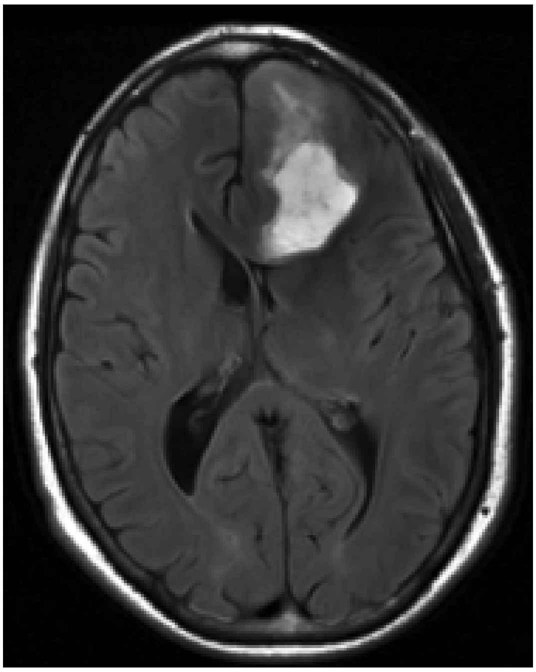


If we are investigating the change in brain morphology over time in the same patient, it would be best to compare the same axial slice. If this were the case, an alignment transformation would be quite useful and then the two images could be compared on a pixel-by-pixel basis (with methods like SSD and Mutual Information).

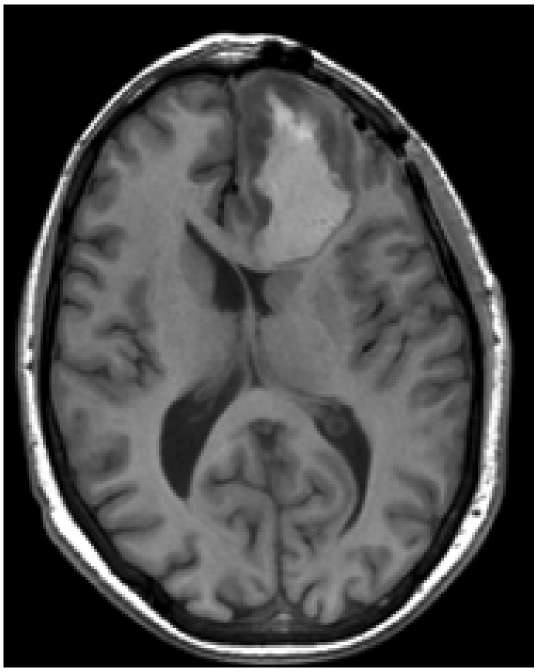
However although these slices look quite similar, we can not assume it is the same slice. In which case, any sort of alignment and similarity measure would tell us how the brain morphology differs over one day and over a different slice. Therefore, because we can not assume the image was taken of the same axial depth there we are not looking at the *exact same thing.*

Practically speaking, however, these two specific images do look quite similar. I suspect a rigid transformation including rotation and translation would be able to align the outlines of the skulls quite closely. In this case, I suggest a simple SSD similarity measure.

3.1.2 What transformation model and what similarity measure do you suggest for image registration? Why? (Ignore the influence of the tumor present in these scans for your answer)



Flair



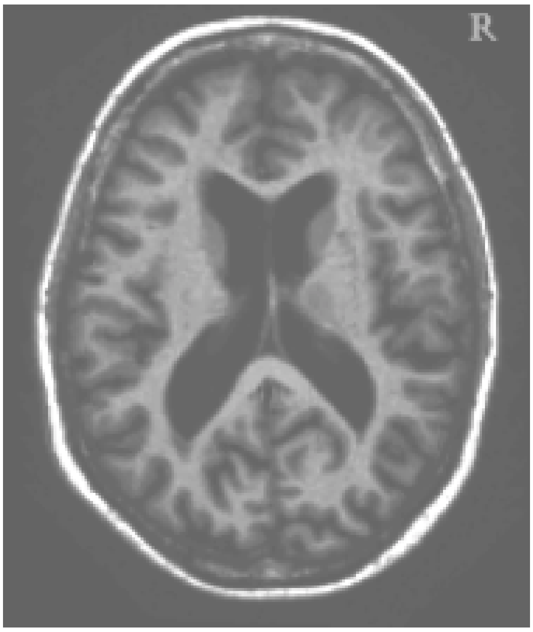
T1

Similarly to the previous question **3.1**, it is not assumed that the slices come from the exact same axial depth. Differently from the previous question, the two images were taken using slightly different variations of MRI.

Although we have not assumed the same exact axial depth, again these images look quite similar and I suspect they are very close to the same axial depth. I suggest a simple rigid transformation involving translation and rotation but no scaling. Note that even though the dimensions of the skulls in the image appear to be nearly identical, they are still not from the same exact slice and thus should not be scaled.

Normalized Cross-Correlation would be inappropriate because the intensity distributions will be different as a result of the different MR acquisition sequences). Similarly, SSD would not be much better. A better option is, if we decide the slices are close enough to faithfully align, Mutual Information.

3.1.3 What transformation model and what similarity measure do you suggest for image registration? Why?



Because the variance in human head circumference is not negligible, it is appropriate to use an affine transformation here which includes rotation, translation and scaling. Note that this assumes that the relative position of features of brain anatomy will scale linearly with brain circumference. In actuality I am sure the development and structure of the brain has been studied extensively and very carefully modeled, but for now I will assume that the scaling can be conducted as a simple linear transformation.

Because the acquisition sequence is the same in both pictures and therefore the intensity distributions should be similar, I believe it appropriate to use Normalized Cross-Correlation. This will highlight the main difference in this image, which is that there exists a large X-shaped gap in the center of one brain and a much smaller X-shaped gap in the center of the other.

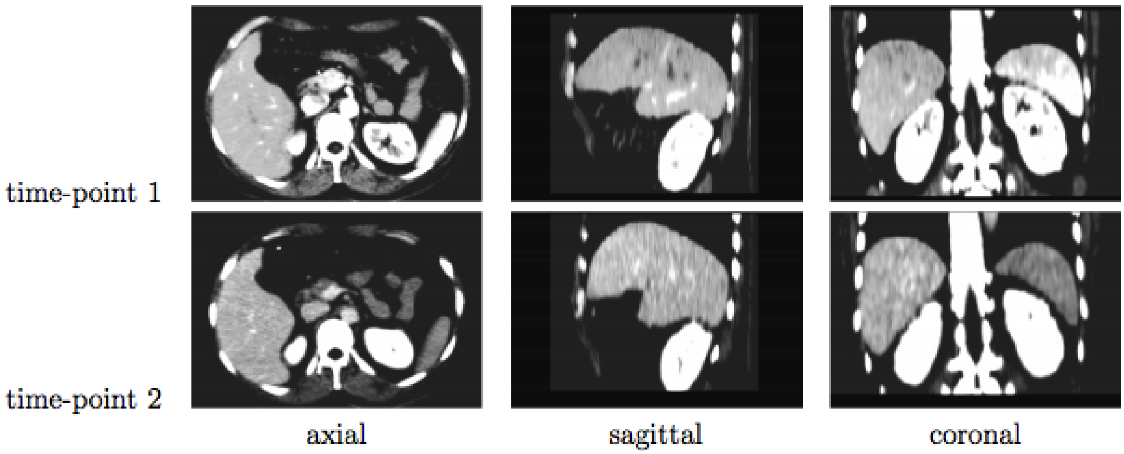
3.2 Specialized registration problems

3.2.1 Registration of MR knee images of the same patient with different levels of articulation. Registration approach: Diffusion registration with SSD.


Diffusion registration is inappropriate here. It is most appropriate in brain imaging where the flow of specific molecular tags is mapped out in 3D by calculating the eigenvectors of 3x3 diffusion tensors. This is the key assumption of diffusion registration, and this assumption is broken by attempting to register an MRI of a knee.

If some sort of clever nonlinear transformation was applied to produce a perfect curvature registration faithful to the differing degrees of articulation and the complicated biomechanics of the knee, SSD could be useful method. Otherwise, the key assumption of SSD that the images are perfectly aligned and shall be compared on a pixel-per-pixel basis is broken here. However, I believe that to be a difficult task and NCC might be able to prove useful without having to conduct a tricky elastic registration.

3.2.2 Registration of torso images (including the lung and the ribcage) at different points of the breathing cycle. Registration approach: Curvature Registration with SSD.

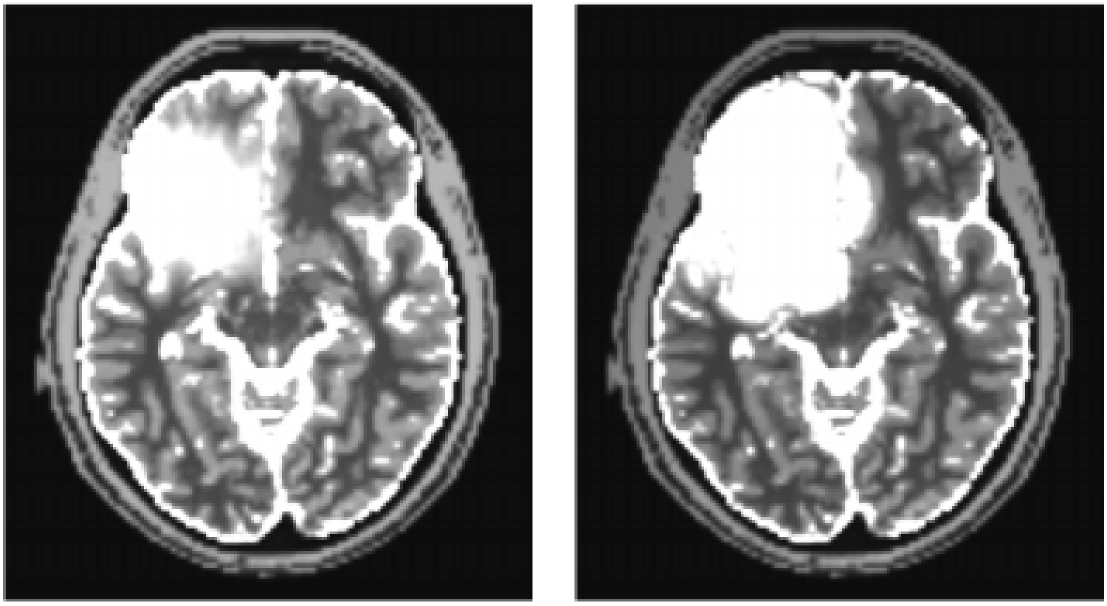


I assume the goal of these images is to investigate something related to the dynamic motion of the breathing cycle. In that case, we should be most interested in difference in position and morphology of these components (such as lungs, ribs, diaphragm, etc).

This goal cannot be robustly achieved with curvature registration. Curvature registration would allow us to nonlinearly align each of these components to each other, making them appear as if they are in the same place. This key assumption of curvature registration is therefore broken.

SSD could potentially be useful, as there will be considerable differences between the positions and shapes of the structures that move during breathing. However, SSD does tells us nothing about *how* the morphology of the structures change and *where* they move during breathing. A better approach would be a segmentation, tracking and morphological analysis of the various components of the image. These data could be tracked throughout the breathing cycle and then analyzed to product valuable insights.

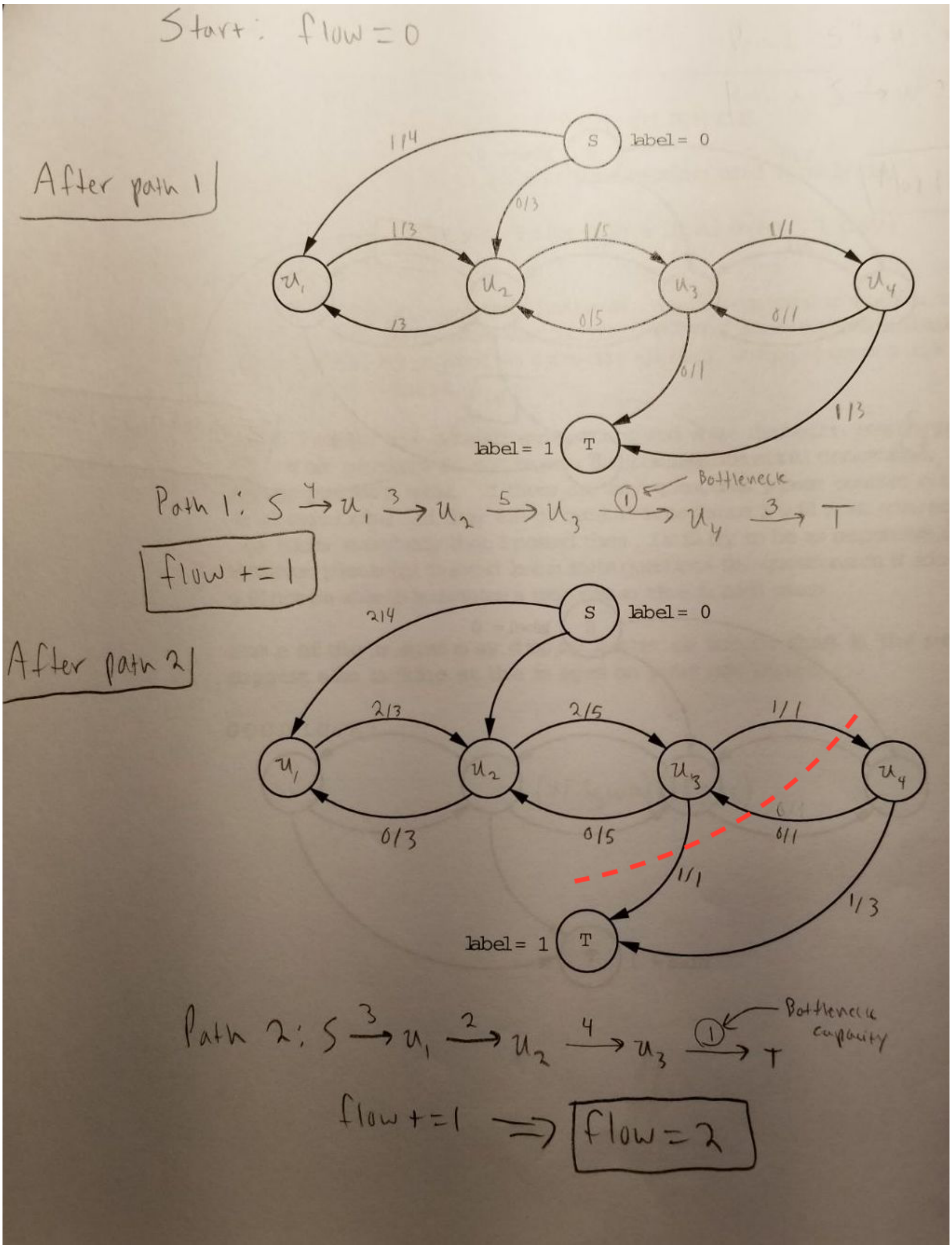
3.2.3 Registration of brain images of the same patient at different stages of brain tumor growth. Registration approach: Elastic registration with SSD



A key assumption in elastic registration is that the images need to be deformed "elastically" (nonlinearly) to register. For example, elastic registration could scale an image more in one region than another. Similarly to question (3.1.1), this key assumption is not necessary in this case because of the fixed nature in which the brain sits in the skull. Assuming it is an image of similar depth axially, it would be much more useful to use a simple rigid transformation without scaling.

After the images are aligned, however, assuming all other variables are constant, SSD could be quite useful. This confirms a simple visual inspection, that the tumor has grown considerably in the second image. However, if the goal is to track the progress of the tumor then SSD may not be the most ideal because it is responsive to changes in other regions of the brain as well, and not just the tumor. To focus on the tumor, some sort of thresholding technique and morphological analysis could be used after alignment as opposed to a simple SSD calculation.

4 Minimum cut/maximal flow



(a) **What is the maximal flow supported by this network?**
The maximum flow supported by this network is 2.

(b) **Determine the segmentation result corresponding to the minimum cut** The minimum cut is the red line. Note that the sum of the saturated edges cut equals two, even though three edges are cut (one edge is not saturated). This is equal to the maximum flow, as it should be.

(c) **Compute the solution (minimizer) of energy 4 not using the Ford-Fulkerson algorithm (by a method of your choise). Compare the result to the one obtained**

in b) The method of my choice is brute-force. Similar to our example in class of an image with three pixels, this one has four. Therefore there are 16 possible combinations of unique ways to label four pixels, and Python's *itertools* makes this easy. Here is the pseudo code:

- Set ENERGY = inf
- For each item in the list itertools.product([0, 1], repeat=4) compute $E(u)$ using equation (4).
- If the result is less than ENERGY, overwrite ENERGY with this result

The result I got for the minimum cut was two and the result I got for the corresponding energy was -2.5. The constant term in the derivation of $E(u)$,

$$\sum_{p \in P} -max(0, -r_p),$$

is not included in the graph analysis but does factor into the energy. It is equal to -4. The difference between this and the energy value is $\frac{3}{2}$, which is different than my maximum flow/minimum cut value of 2.