Analogy Based Segmentation of Volumetric Data

Daniel James Hood

A paper submitted to the
Computer Science Electrical Engineering Department
in partial fulfillment of the requirements for the M.S. degree at
the University of Maryland Baltimore County

January 4, 2004

CMSC-698 Advisory Committee:

Marc Olano (Advisor), Assistant Professor of Computer Science
Penny Rheingans (Reader), Assistant Professor of Computer Science

Certified by _____ _____
Marc Olano Date

Analogy Based Segmentation of Volumetric Data

Daniel James Hood
Computer Science Electrical Enginerring Department
University of Maryland Baltimore County
Baltimore, Maryland 21250

January 4, 2004

Abstract

This paper describes a method for automatically segmenting volumes by example using an analogy based approach. This approach consists of two stages. In the first, design phase, a pair of volumes is presented, where the first volume is the raw MRI or CT data, and the second volume represents the segmented version of the first, this combined pair makes up the learning data. In the second, an application phase, the learned segmentation is applied to some new volume in order to create an "analogous" or learned segmented result.

We show that this technique proves to be an accurate way to automatically segment a volume. High quality segmentations can be accomplished, even when given fairly small sample or sub-sample sizes. By down-sampling or sub-sampling the training data we are able to accomplish these learned segmentations in a reasonable time frame with around nintey-five percent accurracy.

We believe that a tool that can learn how to segment volumes by example will provide the medical community with another valuable means of managing medical data. Such a utility would surely have applications in visualization, medical examinations, and virtual surgeries.

Keywords. volumetric, analogy, segmentation, classification, medical imaging, magnetic resonance imaging (MRI).

Contents

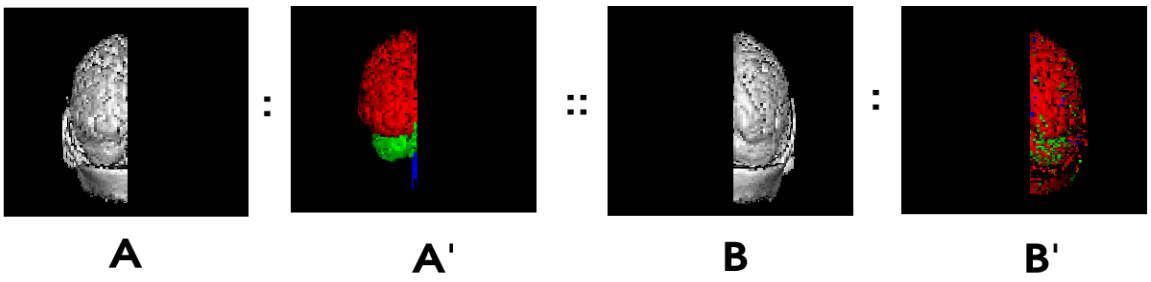

Figure 1: A volume analogy. New "analogous" volume B' relates to B in the same way that A' relates to A. Here A, A', B are all inputs to the algorithm and B' is the output.

1 Introduction

Analogy is a reasoning process that people use quite often to solve problems, provide explanations and make predictions. This paper explores the use of analogies as a means of automatically segmenting complex medical volumetric data. In particular, we attempt to solve the following problem:

Problem "Volume Analogy" Given a pair of volumes A and A' (the raw MRI/CT data and the segmented volumes, respectively), along with some additional unsegmented raw MRI/CT data volume B, synthesize a new segmented volume B' such that:

$$A : A' :: B : B'$$

Essentially we want to find an "analogous" volume B' such that it relates to B in the same way as A' relates to A. We describe an approach that produces results with a high level of accuracy. This approach is depicted in (Figure 1).

Due to the shear sizes of such volumes (typically on the order of $2^7 \times 2^8 \times 2^8 \approx 8.4$ million voxels), it is favorable to have semi or fully automated techniques to segment such data. It is not practical for a human reviewer to manually segment every voxel in a volume due to the vast size. Thus, it is desirable to have a method of accurately and automatically segmenting volumetric data, given some well know segmented training pairs.

This paper starts out with a brief introduction of the analogy based framework in which new segmented volumes are constructed. This is followed by a brief summary of existing synthesis and segmentation techniques and then discusses how parts of each of these two disciplines are pulled together for the learned volume segmentation framework. Data structures and algorithms that are at the heart of the analogy framework are discussed. Following these details are the promising statistical and rendered results of the output of this algorithm. Lastly this paper discusses some of the future opportunities for this framework. Optimizations and tuning of the existing algorithm are outlined, as well as possible future extensions and applications of this framework.

2 Related Work

Our approach pulls fundamentals from work in both image generation and volume segmentation. The following is a survey of some of the closest related work.

Image generation builds off of many distinct areas including machine learning, texture synthesis, and image based rendering.

It has been a goal since the creation of artificial intelligence to build computer systems that are capable of reasoning and solving problems by way of analogies. Some of the first successful attempts at analogy based techniques were Winston's work on reasoning by analogy [17] and Evans' work on geometric analogy intelligence [6]. More recently this notion of machine learning has been applied to various areas of computer graphics including style machines [2] and video texture [14].

Texture synthesis deals with the generation of new images that mimic the texture of a given sample image. Since its introduction almost a decade ago [8], much work and refinement has gone into this area. Several nearest neighbor approaches [3] and [5] have provided high-quality synthesized images. Neighborhoods of pixels consisting of multi-scale representations on the images [16] have also proven to generate high quality images. Multi-scale neighborhoods have also proven to generate high quality images [1], where patches of texture are expanded out of the target image space, rather than searching the sample texture in its entirety.

Segmentation of volumetric data is another area of active research. Traditional methods of segmenting data involve either statistical analysis of the data, or rely on trained individuals to examine all of the voxels in the volume and painstakingly segment and label each area accordingly. Early methods of segmenting data [12], although relatively effective, have been replaced by more sophisticated means of segmenting voxels. One such model [4] accounts for the fact that a given voxel may contain various materials and not just a single matter. In [7] and [10] the authors use stochastic methods to generate transfer functions. This model is fundamentally different than the previous two in that it tends to work back-wards through the rendering pipeline by first analyzing and computing characteristics about the final rendered image.

Lastly, this paper most strongly derives its work from the Image Analogies framework specified by [9]. The Image Analogies framework is a technique for synthesizing or generating new images based on the concept of an analogy. Hertzman et. al. used this analogy framework to perform many diverse tasks in areas such as: traditional image filters, improved texture synthesis, super-resolution, texture transfer, artistic filters, and texture-by-number.

The last area, texture-by-number, involved a training pair where the A image was a numbered representation of the A' image. This numbering scheme was used to represent the various features of the image. The example images of terrain presented, involved coloring the B image green where vegetation was to be rendered in B', blue where water was to be synthesized, black for roadways, and so on. These numbers represented the partitioned or segmented data. This partition data was used to synthesize various two-dimensional examples.

We expand upon the Image Analogies framework and tailor our model to segment volumetric data by learning the relationship in a training pair and applying that knowledge to derive a new segmented volume. Essentially, we will be attempting to figure out these numbers (the segmentation labels) from a volume, rather than using the labels to generate an image.

3 Volume Analogy

Definitions and Data Structures

As input, our algorithm takes a set of three volumes, the unsegmented source raw MRI/CT data volume A, the source segmented volume A' and the unsegmented target raw MRI/CT data volume B. The framework produces the segmented target volume B' as its output as illustrated in Figure 1.

Our approach assumes that the two source volumes are registered, meaning that the raw MRI/CT data at any given voxel at index p in A corresponds to the segmented data at the voxel located at index p in A'. This also

holds true for the B and B' pair. For clarity, index p will be used for source volumes A and A', while index q will specify a voxel in the target B and B' pair.

Since this paper deals with segmentation of volumetric data, the A' and B' volumes both contain segmentation labels. However with this model, the associated data in A' and B' is not limited to segmentation labels, but rather can be any arbitrary data. Other possibilities for such data are discussed later in Section 6.

In summary, our algorithm maintains the following data structures. Raw MRI/CT data volumes consisting of densities measurements of $A(p)$ and $B(q)$ are inputs. A segmented volume comprised of segmentation labels of $A'(p)$ are also input. Lastly, the segmented volume $B'(q)$ is the output:

$$\begin{array}{ll} A(p) & \text{array } p \in \textit{SourcePoint} \text{ of } \textit{Feature} \\ A'(p) & \text{array } p \in \textit{SourcePoint} \text{ of } \textit{Feature} \\ B(q) & \text{array } q \in \textit{TargetPoint} \text{ of } \textit{Feature} \\ B'(q) & \text{array } q \in \textit{TargetPoint} \text{ of } \textit{Feature} \end{array}$$

where *SourcePoint* and *TargetPoint* are three dimensional voxel locations in the source and target pairs, respectively.

The Algorithm

Given this notation, the volume analogies algorithm is presented below. For every target voxel q in B' that is being computed, we take the corresponding voxel neighborhood centered about q in B and find the closest matching voxel neighborhood centered about p in A. Once the closest matching neighborhood centered over p is found in A, the corresponding segmentation label in $A'[p]$ is assigned into $B[q]$.

This algorithm can more precisely be described in the following pseudo-code:

VOLUME-ANALOGY(A, A', B, B')
1 **for each** $q \in B$
2 **do** $p \leftarrow$ FIND-BEST-MATCH(A, B, q)
3 $B'[q] \leftarrow A'[p]$

The FIND-BEST-MATCH function takes two MRI/CT data volumes A and B, as well as the current q for which the best match is to be found.

This function maintains several local variables. A temporary scalar variable $difference_{curr}$ is used to store the current difference as returned by the CALC-DIFFERENCE function (outlined below). The scalar $difference_{best}$ maintains the best difference seen thus far as returned by CALC-DIFFERENCE. Initially, this variable is set to positive infinity and is continually refined. Also maintained is $closestMatch$ which is the index corresponding center of the best matching neighborhood.

Basically, this function searched through the entire volume A and calculates the difference between neighboring voxels around $A(p)$ and the voxels around $B(q)$. If the current difference between these two sets of neighboring voxels is better (less than) what has been seen thus far, then this current difference is recorded as the new best difference, and the corresponding voxel is also recorded. Otherwise, there is already a better match. In either case, processing continues with the next voxel in A.

More precise outline of this algorithm is as follows:

FIND-BEST-MATCH(A, B, q)
1 $difference_{best} \leftarrow \infty$
2 **for each** $p \in A$
3 **do** $difference_{current} \leftarrow$ CALC-DIFFERENCE(A, B, p, q)

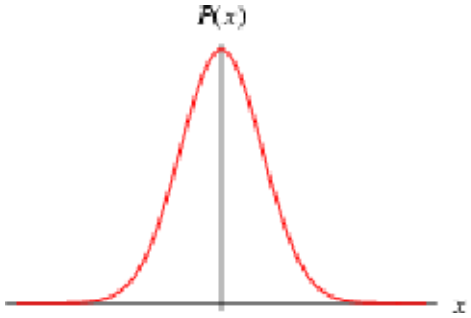

Figure 2: A sample Gaussian filter kernel over one dimensional space.

4 **if** $difference_{current} < difference_{best}$
5 **then** $difference_{best} \leftarrow difference_{current}$
6 $closestMatch \leftarrow p$
7 **return** $closestMatch$

Thus, the core of the algorithm lies in the CALC-DIFFERENCE function. This function takes two MRI/CT data volumes A and B, as well as the current indices p and q for which the difference is to be calculated.

In order to find the closest match for any given neighborhood of voxels, the overall difference between any two sets of voxels needs to be computed. A three dimensional filter kernel is applied to both neighborhoods to weighting each surrounding voxel based on its distance and importance. For this paper, we chose to apply a three dimensional Gaussian pyramid over both sets of voxels.

The overall difference is the difference between each corresponding set of voxels (in A and B) in each neighborhood and scaled by the associated filter kernel weight. The summation of all of the weighted differences under the kernel is returned as the overall difference between both neighborhoods of voxels.

Again, more precisely the pseudo-code would be as follows:

CALC-DIFFERENCE(A, B, p, q)
1 $TotalDiff \leftarrow 0$
2 **for each** $voxel \in neighborhood$
3 **do** $TotalDiff \leftarrow TotalDiff + |B - A| \times filterKernel[voxel]$
4 **return** $TotalDiff$

Filter Kernel

A Gaussian filter kernel was used to weight the cells that are in the voxel neighborhood during difference computations. A Gaussian kernel was chosen as it has proven to work well in other various other reconstruction works. A one-dimension version is shown in Figure 2. This kernel allows the cells that are closest to the center to have the most weight in determining which voxel neighborhood is the closest match. As cells get farther from the center, they hold less and less weight. In order to obtain a reasonable runtime, this kernel was reduced to consider a given voxel and its immediate neighbors only.

A summary of the full algorithm is shown in Figure 3. It takes in two RAW MRI/CT data volumes A and B, and a segmented version of A called A'. A newly segmented volume B' is generated in scan line order,

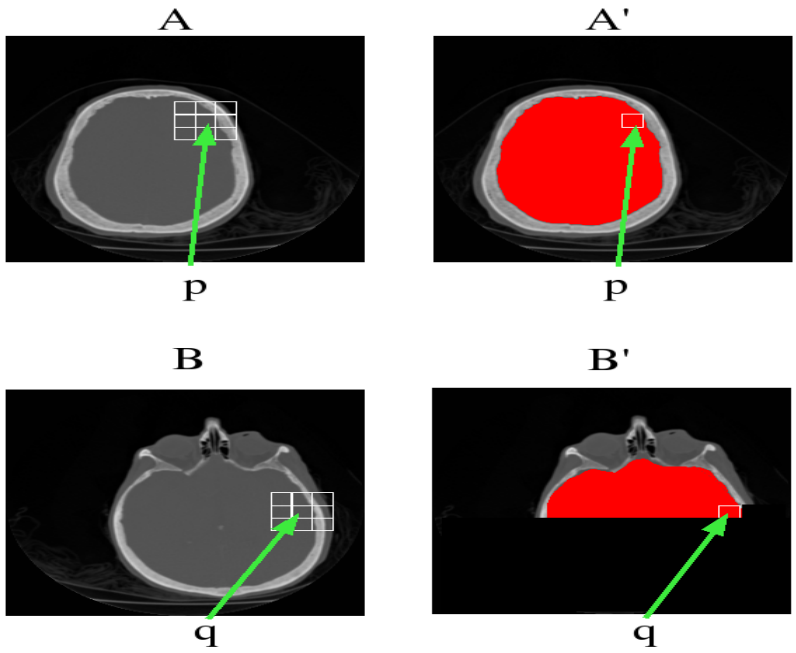

Figure 3: Here the neighborhood around $B[q]$ most closely matches the neighborhood around $A[p]$ and thus $A'[p]$ is recorded into $B'[q]$ as the best match. In this diagram the Cerebral Cortex labeled information is shown in red in the A' and B' set.

by looking at the best set of matching voxels centered about p in A for each q in B, and carrying over the segmented data that is located in A'. In this example, the best matching set of voxels for the area around $B(q)$ is located at the highlighted voxels shown at $A(p)$. The associated label at $A'(p)$, in this case the cerebral cortex, is then carried over into $B(q)$.

4 Source Data sets

The datasets that are used in this paper are derived from the Stanford volume data archive [15]. To limit run time, both datasets have been down-sampled from their original size.

The MRbrain dataset has been down-sampled from 109 slices of 256×256 pixels to 54 slices at 128×128 pixels. The segmented data used in the MRbrain set was courtesy of [13]. The CThead dataset has been down-sampled from 113 slices of 256×256 pixels to 56 slices at 128×128 pixels.

The MRbrain dataset is segmented into four different categories:

- Cerebral Cortex
- Cerebellum
- Spinal Cord
- and "other" which is the catch-all for everything that is not part of the central nervous system.

n	% of original volume sampled	% of correct voxels
2	12.500%	95.8116%
4	1.505%	94.8346%
8	0.1736%	94.1811%

Figure 4: MRbrain data results

An example of this segmentation can be seen in Figure 5A, where the various parts of the brain are rendered in different colors, and any voxels labeled "other" are rendered transparent.

5 Results

There are several ways to measure the results of this analogy based segmentation technique. This paper will present both quantitative results as well as figures of rendered synthesized segmented volumes.

5.1 MR Brain

Since both raw MRI data and the segmented data were available for the MRbrain set, this set provides a means to check the algorithm.

A smaller sub-portion of the original MRbrain data set for both the MRI and segmented data is used as the A and A' volumes respectively. Then, using the original MRbrain dataset as the B volume, we attempt to generate volume B'. Since the correct solution to B' is already known (the original segmented data), it is possible to determine how accurate of a reconstruction was accomplished.

The overall correctness can be computed by counting the number of matches where $B'[q] = O_{seg}[q]$,and dividing the the number of calls in the volume, where $O_{seg}[q]$ is the original segmented version of the MRbrain dataset.

Figure 4 shows the resulting percentage of voxels that were correctly reconstructed given the specified sub-sample of the original MRI and segmentation volumes that made up A and A'. The percentage of the volume sampled came from considering the Gaussian pyramid centered at every n^{th} slice, at every n^{th} row, at every n^{th} column. As the figure illustrates, fairly accurate reconstruction occurs even when sampling fairly small sub-sections of the original MRbrain data.

When using a large volume as the training pair, this algorithm can be fairly slow. For the down-sampled datasets, where $n = 2$, it takes approximately eight hours to generate a new segmented volume. However, if you increase n, the algorithm speeds up dramatically. It takes about one and a half hours when $n = 4$ and only about fifteen minutes when $n = 8$. It is worthy to note that these speedups are not directly proportional to the quality. A forty fold increase in speed only results in about a one and a half percent drop in accuracy. These runtimes were obtained on a 600MHz Intel Pentium III.

The original segmented MRbrain data, as well as the resulting volumes for each of the aforementioned analogies is shown rendered in Figure 5. The original MRbrain segmented data is shown in Figure 5A, and Figures 5B-D are the rendered images from $n = 2$ down to $n = 8$. In these figures, the cerebral cortex is rendered red, the cerebellum is rendered green and the spinal cord is rendered blue.

In these rendered images, the areas of the brain are still relatively segmented and colored appropriately even as the portion of the original volume greatly decreases to under one percent. Take note of the artifacts around

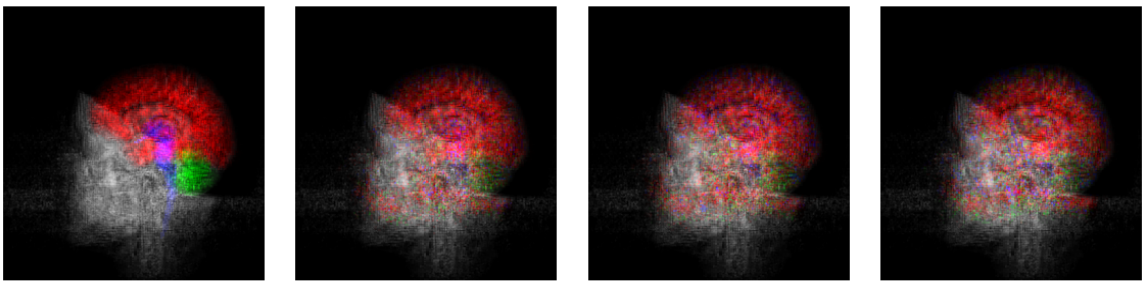

Figure 5: A) The original segmented data, B-D) are rendered images from the MRbrain analogies in descending sub-sample size.

		Original Labels			
		Cortex	Cerebellum	Spinal Cord	Other
Labels Computed by Analogy	Cortex	**43,135**	2,092	1,243	12,847
	Cerebellum	2,541	**2,805**	138	2,547
	Spinal Cord	946	151	**653**	727
	Other	10,886	2,374	564	**801,087**

Figure 6: Matrix showing detailed analysis of learned segmentation labels (taken from learned segmented volume where $n = 2$).

the rest of the skull and facial tissue. There are a number of cells that have been mistakenly categorized as parts of the central nervous system. We believe that these false identifications can be further reduced or even eliminated with a larger filter kernel (as discussed in Section 6).

Figure 6 shows, in more detail, the results of a learned volume segmentation. The columns represent the actual data taken from the original MRbrain segmentation, and the rows in each column represent the number of occurrences that particular tissue was classified with the associated label. The bold-faced entries on the diagonal represent the number of voxels with correctly learned segmentation labels. For example, the first column would read: for the voxels that make up the Cortex, 43,125 of them were correctly classified as Cortex, with 2,541 incorrectly marked as Cerebellum, 946 incorrectly marked as Spinal Cord, and 10,886 incorrectly marked as Other. The percentage of correct voxels presented in Figure 4 is actually calculated by taking the sum of these bold face entries and dividing by the number of voxels in the volume.

These findings support what is seen in the rendered volumes (Figure 5). For the Cerebral Cortex, the numbers indicate that most of the Cortex tissue has been correctly identified. However, there is also a significant portion of the Cerebral Cortex that was classified as Other. The lighter colored red Cortex is consistent with portions being segmented as air, and thus subtracting from the brightness. The Cerebellum was also mostly correct, however, significant portions were mistakenly classified as Cortex and Other. This is consistent with the lighter green Cerebellum caused by the additional air, as well as a hint of red from the incorrect Cortex labellings. The Spinal Cord was the least correctly guessed of the four classifications. The high amount of incorrect Cortex and Other guesses, serves to both lighten the blue from the Brain Stem, and also adds more red into the region. Finally, the Other category was the most correctly generated label. A small portion of the Other category was incorrectly labeled as Cortex, which explains the additional red at the bottom of the head.

There are two observations that we would like to make regarding this data and the rendered volumes. First, we believe that part of the high level of inaccuracy for the Brain Stem is caused by poor sampling. The

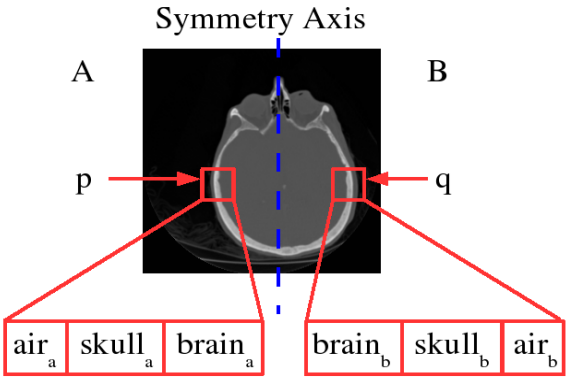

Figure 7: Illustration of symmetry and two voxel neighborhoods with example data labels.

Brain Stem is the smallest tissue in the original volume, and sampling one out of every eight voxels only considers a very small sub-section. Second, regarding the portion of Other that was incorrectly classified as Cortex, we believe that this is partially due to inconsistencies in the volume. Looking closely at Figure 5, all of the renderings have a slightly lighter color (especially noticeable on the black background) on the lower third of the image. At this point in the volume, the average MRI/CT data becomes slightly more dense. The amount of redness below this line significantly increases, as a result of the incorrect Cortex labeling due to the inconsistency.

Consider now that the segmentation data provided as the learning pair is not distributed evenly across the entire volume as in the left portion of the head that is given in A and A' as shown in Figure 1. The human brain automatically sees that the volume represented in the source pair is the other half of the head that is the target volume B. However, this requires special consideration when generating new volumes.

For example, Figure 7 illustrates a cross section of this problem. Assume that everything on the left side of the symmetry axis is given as a source volume A and everything on the right side is given as the target volume B'. So let us consider the case where the voxel neighborhood on the right side around q is being matched, and CALC-DIFFERENCE is being called to compare the neighborhoods around $A(p)$ and $B(q)$.

If this symmetry is not considered, then the difference will be computed as,

$$|brain_b - air_a| \times c_1 + |skull_b - skull_a| \times c_2 + |air_b - brain_a| \times c_3,$$

where c_n is the appropriate scalar constant. In this case, the first and third differences will be rather high since both of these are taking the difference between two different materials (brain and air). This results in a high difference between these 2 neighborhoods. Whereas, if this symmetry is considered when examining the neighborhood in A around p, the neighborhood can be mirrored to mimic the missing right side of the head. Thus the difference will be more closely calculated as,

$$|brain_b - brain_a| \times c_1 + |skull_b - skull_a| \times c_2 + |air_b - air_a| \times c_3,$$

again where c_n is the appropriate scalar constant. The result of this second computation will yield a much closer match, as all of the individual differences are between like materials. Obviously, this assumption is not appropriate for all data, such as the human midsection where the organ structure is asymmetric, however, it can improve synthesis of symmetric organs such as the head and extremities.

Since the human body is symmetrical, for the most part, this fact can be exploited to our advantage. If we further take into account this fact, we essentially obtain extra data out of the A and A' data sets, by also

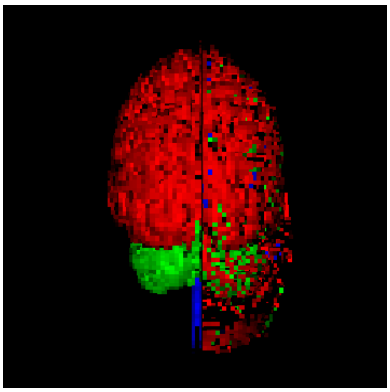

Figure 8: An example rendering where the known symmetry of the data is also considered.

taking the mirror of these datasets about the major axis. A composited image of such a generated volume is illustrated in Figure 8.

5.2 CT Head

One of our original goals was to automatically segment the CThead dataset from the example learned from the MRbrain dataset, which had been manually segmented by hand. Attempts to segment the CThead dataset were not met with the same success as the MRbrain dataset. We attribute this lack of success to the failure to produce a high-quality A' for the learning set. We performed three various methods of running the CThead data through our analogies model:

Initial dataset. The initial segmentation attempt for the CThead dataset failed completely with no apparent connection between head or brain structure. Since these volumes were acquired from different recording devices, the range of data for the two volumes was not consistent. Since the difference between similar cells could be very high and the difference between distinct cells could be low, the resulting segmented data was not correct.

Simple scaled dataset. The next step was to scale the B CT data into the same range as the A MRI data. This also produced poor results. Objects in the volume such as metal rods, fillings, etc. introduce outliers that throw off the analogy and could skew the entire set of results.

Statistically adjusted dataset. Finally, falling back on the statistics, it makes the most sense to look at the histograms that represent the distribution of voxels over the range of possible values. Overall, if considering data of the same nature, it makes sense that these histograms should look as similar as possible. However, when examining the quantity and distributions of the peaks in both the MRbrain and the CThead histograms, there seemed to be many more inconsistencies between the two human heads than expected.

Scaling, shifting and attempts at various other techniques to best align the data all failed to produce segmented volumes that were believable. Without well aligned data we observed that the volume analogy failed to produce reasonably segmented volumes. We believe that such a task, if possible, is well beyond the scope of this paper. Another possible cause of this disproportionate data could be due to the partially removed skull in one of the datasets. Since a substantial portion of the skull is gone, this has the effect of replacing bone with air in one of the volumes. This is definitely a contributing factor in the large difference between the two histograms.

5.3 Challenges and Considerations

Here we will outline some of the challenges and special considerations that arose during the creation of our volume analogies framework.

As mentioned, the lack of properly aligned and distributed volumes proved to be a hurdle that we were unable to easily overcome. We suspect that volumes coming off of the same capture device would have less of a problem with the alignment and distribution of data.

In order to bring the runtimes down to a reasonable level, the Gaussian kernel had to be significantly scaled back. We had initially tried a kernel that considered at least a $5 \times 5 \times 5$ neighborhood area. This results in 125 voxels, each of which would have to have its difference computed to its corresponding voxel in the comparison dataset. This amount of computation for each cell in our volume datasets did not allow for reasonable runtimes, thus this was scaled back to only consider immediate neighbors.

In order to allow the analogy to run in an acceptable amount of time, a multi-scaled refinement to segment the data was also set aside. In the current implementation, only the finest level is being generated.

6 Discussion and Future Work

In this paper, we have discussed a framework for segmenting a volume by example using an analogy. We have shown that this method proves that analogy based segmentation of volumetric data is possible. The results presented for this model are reasonable, and are even more promising, in that future refinement and tuning of this approach may yield even better constructed segmentations, or even other applications.

There is still much work that needs to be done, both in improving the approach described here, as well as experimenting and investigating other approaches to synthesizing volumetric data. Here is a brief list of some of the areas of future work that we wish to pursue:

Speeding up the algorithm. The performance of this algorithm is fairly slow. The time analysis of the current implementation is best described as the product of the training pair A, A' size and the target pair B, B' size. Using the current implementation of this algorithm on these down-sampled volumes takes a number of hours to generate the segmented data. Introducing an algorithm such as an approximate-nearest-neighbor search (ANN) could reduce half of the problem, (searching through A for the best match) down to a logarithmic order and would surely result in a significant speedup. Another technique to speedup the match finding is to use a Fast Fourier Transform (FFT) based method. Recent work by [11] shows speedups of around 120 fold for finding matches in a $150 \times 100 \times 30$ video sequence, reducing the match finding time from 10 minutes down to 5 seconds. Finding the best match for a particular cell in a three dimensional volume and finding the best match for a given pixel in a video is a very similar problem. We anticipate that such gains would be reproducible for our problem.

Tuning and improving the algorithm. Since the algorithm was working on a rather large dataset, several desirable features had to be removed or cut back in order to complete this project in a timely matter. For example, the Gaussian kernel that was used in the difference computations had to be scaled back from its original size and complexity. We believe that this is one of the reasons that outliers are seen in the rendered images. Such a scaled back kernel filter results in false guesses that cause these outliers to be segmented incorrectly. A larger filter size would in fact be able to better capture the larger scale structure, and cut back such inconsistencies. We also believe that successful integration of a multi-scaled approach would help capture the high level structure as well.

Generation of other forms of data. This paper presented an approach for analogous based segmentation of volumetric data where the contents of A' were essentially labels that represented the segmented data from A.

Another logical choice to segmentation labels would be a color model. It would be fairly straight-forward to adapt this algorithm to work for an RGB[A] color scheme. The Visible Human Project at the National Institute of Health makes available very high resolution MRI/CT and RGB data for both male and female cadavers. This RGB data was obtained using destructive techniques (slicing and photographing the cadaver) and thus could not be replicated for a living patient. However, non destructive data gathering such as MRI or CT could be used with such an approach to visualize the inside of a patient, and could be of use for applications such as surgical visualization.

Generation of higher order data sets. Generating volumetric data was a logical expansion upon previous ideas of image synthesis and generation. By taking the dimensionality a step further to four dimensions, some other applications present themselves. Datasets that study fluid dynamics such as turbine air flow or the flow of liquid through objects could be used to create new analogies. For example, suppose for a given stationary object (A), there exists data about how some fluid flows around this object A'. Then, given another object (B), would it be possible to take the A and A' pair and construct what the flow of some fluid would look like (B') over this new object? There has also been recent work done in animation of the aforementioned Visible Human datasets, where again there is a stationary object, in this case the MRI data (A). Then, given the three dimensional stream that is the animated data (A') and a new MRI model (B), would it be possible to generate a new animation stream (B') for the new model?

Automatic abnormality detection. In this current approach where the best match is found for a given voxel neighborhood and then synthesized, the strength of the match is not used. In other words, even though the score is temporarily recorded while finding the best match, the result of that score does not propagate to the output volume, and is never seen by the end-user. This data could be used beneficially. What if, rather than simply throwing away the strength of the match, this information was recorded as just that, the strength of the match. In other words, how sure is the algorithm that this particular data shows up in the original set? We propose that an acceptable maximum difference could be set such that any data that does not meet a minimum match score is flagged, perhaps for human review. Consider the following analogy: Assume that the algorithm is given MRI or CT data of normal healthy tissue (A) and the segmented version of that volume (A'), where all of the data is accounted for (labeled as normal healthy tissue). Then, imagine that the algorithm is given another volume (B), where the tissue contains abnormalities, such as tumorous material. When the algorithm encounters the tumorous material it will struggle to find a string match in the learning pair. This area could then perhaps be segmented into another label dedicated for suspicious tissue that should be examined more closely by a medical professional. We envision such a derivation that is capable of such a procedure.

References

[1] Michaeil Ashikmin. Synthesizing natural textures. In *2001 ACM Symposium on Interactive 3D Graphics*, 2001.

[2] Matthew Brand and Aaron Hertzmann. Style machines. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 183–192. ACM Press/Addison-Wesley Publishing Co., 2000.

[3] Jeremy S. De Bonet. Multiresolution sampling procedure for analysis and synthesis of texture images. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 361–368. ACM Press/Addison-Wesley Publishing Co., 1997.

[4] R. A. Drebin, L. Carpenter, and P. Hanrahan. Volume rendering. In *Computer Graphics*, pages 65–74, 1988.

[5] Alexexi Efros and Thomas Leung. Texture synthesis by non-parametric sampling. In *7th IEEE Internations Conference on Computer Vision*, 1999.

[6] T. G. Evans. A program for the solution of geometric analogy intelligence test questions. 1968.

[7] T. He, L. Hong, A. Kaufman, and H. Pfister. Generation of transfer functions with stochastic search technologies. In *Proceedings IEEE Visualization*, 1993.

[8] David J. Heeger and James R. Bergen. Pyramid-based texture analysis/synthesis. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 229–238. ACM Press, 1995.

[9] Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David H. Salesin. Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 327–340. ACM Press, 2001.

[10] A. Kaufman, R. Bakalash, D. Cohen, and R. Yagel. A surevy of architectures for volume rendering. In *IEEE Engineering in Medicine and Biology*, 1990.

[11] Vivek Kwatra, Arno SchÖdl, Irfan Essa, Greg Turk, and Aaron Bobick. Graphcut textures: image and video synthesis using graph cuts. *ACM Trans. Graph.*, 22(3):277–286, 2003.

[12] Mark Levoy. Display of surfaces rom volume data. In *IEEE Computer Graphics and applications*, pages 29–37, 1988.

[13] Barthold Lichtenbelt, Randy Crane, and Shaz Naqvi. Introduction to volume rendering. Hewlett Packard Professional Books, 1998.

[14] Arno Schödl, Richard Szeliski, David H. Salesin, and Irfan Essa. Video textures. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 489–498. ACM Press/Addison-Wesley Publishing Co., 2000.

[15] Stanford Data Archives. The stanford volume data archive. In *http://graphics.stanford.edu/data/*, 1989.

[16] Li-Yi Wei and Marc Levoy. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 479–488. ACM Press/Addison-Wesley Publishing Co., 2000.

[17] P. H. Winston. Learning and resoning by analogy. *Communications of the ACM*, 23, 1980.

Marc Olano, CMSC-698 Advisor