Fuzzy segmentation of X-ray fluoroscopy images

Daniel B. Russakoff, Torsten Rohlfing, and Calvin R. Maurer, Jr.

Image Guidance Laboratories, Department of Neurosurgery, Stanford University, Stanford, CA

ABSTRACT

Segmentation of fluoroscopy images is useful for fluoroscopy-to-CT image registration. However, it is impossible to assign a unique tissue type to each pixel. Rather each pixel corresponds to an entire path of tissue types encountered along a ray from the X-ray source to the detector plate. Furthermore, there is an inherent many-to-one mapping between paths and pixel values. We address these issues by assigning to each pixel not a scalar value but a fuzzy vector of tissue probabilities. We perform this segmentation in a probabilistic way by first learning typical distributions of bone, air, and soft tissue that correspond to certain fluoroscopy image values and then assigning each value to a probability distribution over its most likely generating paths. We then evaluate this segmentation on ground truth patient data.

Keywords: fuzzy segmentation, fluoroscopy, probabilistic DRR, image registration, CT.

1. INTRODUCTION

X-ray fluoroscopes are common in today's operating room. They are used to provide real-time intraoperative data to surgeons during an operation. Unfortunately, these images can be very difficult for a computer to interpret as each pixel represents the entire path of air, tissue, and bone travelled by the X-ray from the source to the detector plate. In addition, there are theoretically a very large number of paths that can account for a single pixel value. This many-to-one mapping between paths and pixels leads to an inherent ambiguity in the reverse mapping from pixel to path. In current (e.g., fluoroscopy to CT) algorithms for registration of three-dimensional (3-D) images and two-dimensional (2-D) projections, these ambiguities are ignored since these methods operate entirely on the images themselves.

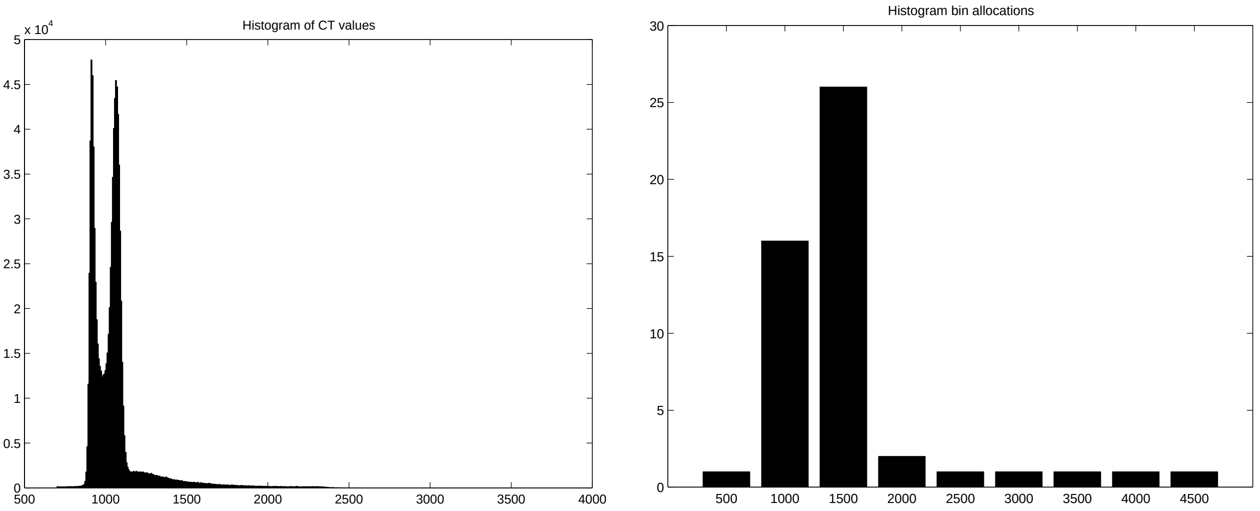
We would like to make it possible for the 2-D to 3-D registration to take place not in the image space, but in the path space. Most existing algorithms proceed by computing a Digitally Reconstructed Radiograph (DRR) from a pre-operative CT and trying to match it up with an intra-operative image. In the process of creating the DRRs, the generating paths for each pixel are calculated and then thrown away. Using our segmentation scheme, one would be able to map the pixel values from the fluoroscopic image into their corresponding generating paths and perform the registration in path space preserving the information lost by the many-to-one mapping.

The segmentation algorithm runs in two phases, the *learning phase* and the *calculation phase*. The learning is done on a patient-specific basis, and is restricted to a certain viewpoint, either anterior-posterior (AP) or lateral. It begins by projecting a large number of synthetic test rays through a patient CT volume from a certain viewpoint. Those rays are stored along with their corresponding calculated fluoroscopic values. The fluoroscopy values are then quantized into bins along with their corresponding paths. These bins now represent integer pixel values as opposed to the floating point intensities coming from the DRR calculation. Then, for each bin, we perform a clustering of all of the paths that map to the fluoroscopy value represented by that bin. The *clustroids* of this clustering are the distinct generating paths that map to a given pixel value. The clustroid of a data set refers to the point in space which minimizes the total distance (in a given metric) between it and all of the other points in the cluster.[1] We will refer to the set of clustroids and the number of paths in each cluster as the *signature* of a pixel value.

Given a signature, there are a number of ways to perform the calculation phase. That is, given an input fluoroscopic image, there are many ways to assign each pixel a probability distribution over the paths in its signature. This is where we actually put numbers into our fuzzy segmentation. Two such methods are proposed and evaluated on patient data.

Further author information: (Send correspondence to D.B.R.)
D.B.R., T.R., and C.R.M.: Image Guidance Laboratories, Department of Neurosurgery, Stanford University, 300 Pasteur Drive, MC 5327, Room S-012, Stanford, CA 94305-5327, phone: +1 (650) 498-7958, fax: +1 (650) 724-4846; D.B.R.: dbrussak@stanford.edu; T.R.: rohlfing@stanford.edu; C.R.M.: calvin.maurer@igl.stanford.edu.

Medical Imaging 2002: Image Processing, Milan Sonka, J. Michael Fitzpatrick,
Editors, Proceedings of SPIE Vol. 4684 (2002) © 2002 SPIE · 1605-7422/02/$15.00



(a)

(b)

Figure 1. (a) Histogram of values in entire CT volume (n.b. values are plotted from 0-4095 instead of starting at -1024) (b) number of bins allocated for each set of 500 values based on the CT histogram in (a).

2. METHODS

2.1. Learning

The learning begins with a pre-operative CT scan of the patient and a viewpoint (AP or lateral). We then proceed by randomly shooting rays through the CT volume and using a simple physical model of the X-ray process. Three-dimensional CT and 2-D X-ray projection imaging are based on the very same physical processes.[2–4] After traversing a homogeneous piece of matter with attenuation coefficient μ and thickness d, an X-ray with initial intensity J_0 is detected with a remaining intensity

$$J_d = J_0 e^{-\mu d}. \tag{1}$$

For a continuous path L (i.e., ray) in space through a heterogeneously attenuating object, the total attenuation μ_L along L is equal to the integral

$$\mu_L = \int\limits_{-\infty}^{\infty} \mu(L(t)). \tag{2}$$

For paths of finite length, the integral in the above equation can be discretely approximated by a finite sum if the attenuation coefficient is only sampled at N discrete locations $\mathbf{x}_i$ with a uniform spacing δ. The effective attenuation can then be written as a discrete sum

$$\mu_L = \delta \sum_{i=0}^{N-1} \mu(L(\mathbf{x}_i)). \tag{3}$$

By calculating the effective attenuation for each path, we can predict the fluoroscopy value that we would expect to see. As mentioned, these floating point DRR values are discretized into bins which represent integer pixels.

2.1.1. Path representation

To smooth out the data and make it easier to store, each path is represented by a histogram of the CT values (-1000 to 3000) it encounters along the way. We set the number of histogram bins to be 50, or roughly 10% of the number of voxels through which a typical path would travel. This, however, does not account for the fact that, for any given CT, different values may predominate. To account for this, we used variable-sized histogram bins. The width of the bins were chosen in the following way. We calculate the histogram of CT values over the entire CT volume using 9 bins, one for each 500 values (e.g. $\text{bin}_1 = (-\infty, -500], \text{bin}_2 = (-500, 0] ..., \text{bin}_9 = (3000, \infty)$) and assign bin widths for our path based on the relative sizes of this histogram. For example, if, for a given CT volume, a majority of the

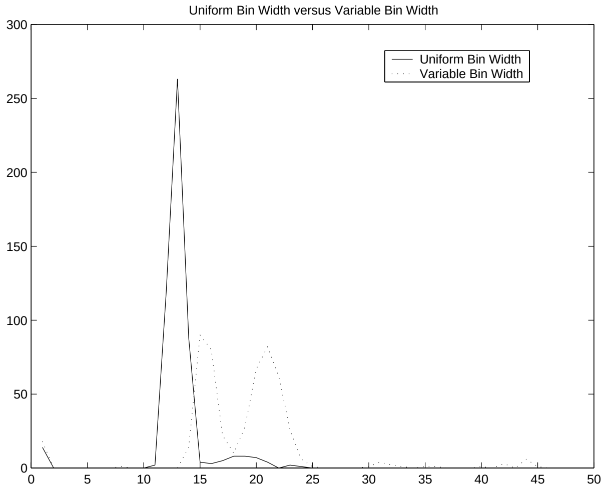


Figure 2. Two depictions of the same path histogram. The solid line shows the path with uniform bin widths, while the dotted line is the same path using the bin widths derived from Figure 1b.

values fall into bin_4, we would assign the most bins in our path representation to bin_4's range $((500, 1000])$. As a result, the path histogram's bins in that range would be narrower.

This scheme allows us to use narrower bins where the bulk of the values in a path would fall, allowing more power in discriminating between two paths. See Figure 1 for an example. Using variable-sized bins instead of uniform ones, lets us make better use of the expressive power of the bins in the areas where we are likely to need it the most (Figure 2).

Another issue we need to address with respect to the path representation is that of the weighting of each path. Because of the projection involved in the imaging process, some paths will travel through more voxels than others. As we do not want to weight those paths more than paths that go through fewer voxels, we add air voxels to each of the shorter paths until all paths have the same weight. Adding air allows us to compare path histograms more accurately and, as air theoretically does not attenuate X-rays, it does not change the physical properties of the path.

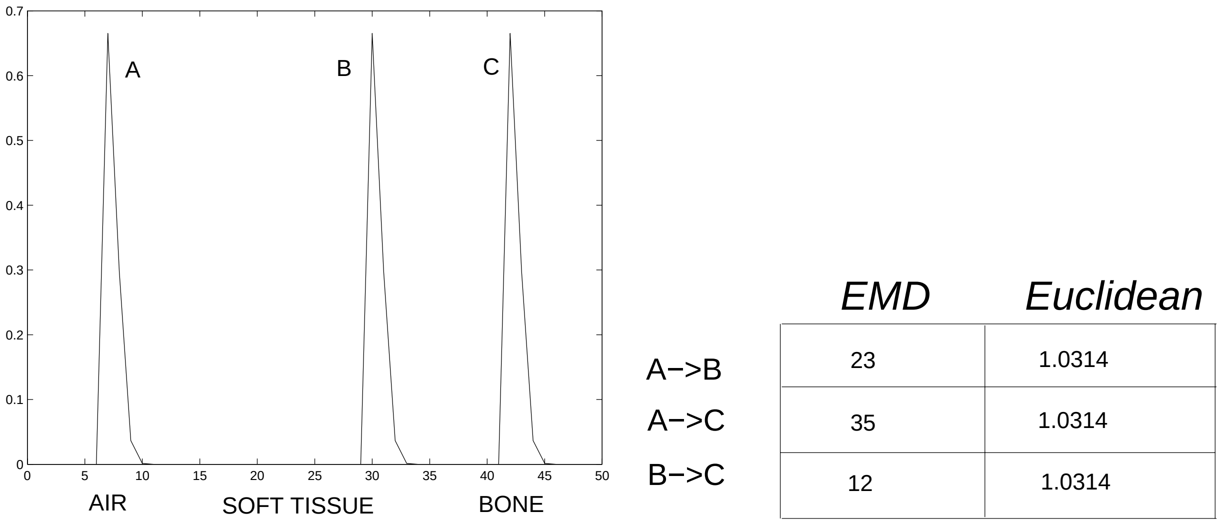
2.1.2. Earth Mover's Distance

Now that we have a path representation, we need to define a metric space on which to compare different paths. For our purposes, the commonly used Euclidean distance won't quite work as it does not model our data correctly. For example, consider an extreme case of three paths,

1. one that goes through x voxels of air and nothing else
2. one that goes through x voxels of bone and nothing else
3. one that goes through x voxels of slightly denser bone and nothing else.

Treating each path as a vector, and using the Euclidean metric, the distance between any two of these vectors is the same, namely $\sqrt{x^2 + x^2}$. This does not accurately model the fact that paths 2 and 3 should be closer than paths 1 and 2 or 1 and 3. For this behavior, we turn to the *Earth Mover's Distance* (EMD).[5]

Under certain assumptions, EMD is a metric over probability distributions that calculates the minimum amount of work to transform one distribution into the other. Intuitively, one can think of the first distribution as mounds of dirt and the second as holes in the otherwise flat ground. The EMD calculates the minimum amount of work needed to fill in the holes of the second distribution with the dirt from the first. Applied to our situation, the EMD models exactly what we need. Now paths 2 and 3 will be considered more similar since we don't have to move the dirt as far. Figure 3 illustrates these concepts in action. The EMD constitutes a metric space in one dimension (1-D) as long as the distributions are equal weight, or sum to the same value. Since we can always add air to any path without changing its properties it is not difficult to meet this assumption.



(a)

	EMD	*Euclidean*
A−>B	23	1.0314
A−>C	35	1.0314
B−>C	12	1.0314

(b)

Figure 3. Illustrative example of EMD vs. Euclidean distance between paths. *Left:* Plot of histograms for three different paths, *right:* corresponding values for EMD and Euclidean distance. An ideal metric will report that the distance between B and C is smaller than between A and B which, in turn, is smaller than the distance between A and C.

2.1.3. Clustering using the k-means algorithm

With a metric in place for comparing paths, we now turn our attention to determining which paths are responsible for a certain pixel value. Though theoretically there are a large number of paths that could produce a given pixel value, by restricting the viewpoint to AP or lateral, we are hopefully restricting this number to a more manageable size, one that can be calculated by unsupervised learning techniques.

Recall that all DRR values from the training paths have been discretized into bins representing their corresponding pixel values. Hence, for each bin, we have all of the paths that map to it. It is from this set that we will attempt to model the many-to-one mapping. Taking each bin separately, and treating each path histogram associated with that bin as a vector, we perform a standard vector quantization using the popular k-means algorithm[6] with an initial point refinement scheme.[7] The k-means method is a very powerful unsupervised clustering algorithm provided that your data is roughly uniformly distributed and that you know k, the number of clusters, in advance. If you do not know k, there are a number of ways to divine it.

For our purposes, we chose a hybrid method to infer k. We perform k-means clusterings on the data with successive k in the range from 2 to 8. We then evaluate each clustering using the measure:

$$IC = \frac{\frac{1}{N}\sum_{i=1}^{k}\sum_{x \epsilon C_i} d(x - z_i)^2}{\min_{i \neq j}(d(z_i - z_j)^2)}. \tag{4}$$

where z_i is the clustroid of cluster C_i and $d(x, y)$ is a distance metric of our choosing.[8] The numerator of this measure is the average intracluster distance and the denominator is the minimum intercluster distance. A good clustering will, at the same time, minimize the former and maximize the latter. Hence, the k that yields the lowest value IC for its clustering will be the winner.

This gets us most of the way but we're still left to wonder in the case where $k = 2$. Should this clustering remain as is, or should it be merged into one big cluster? To answer this question, we use a measure of the clustering's isolation:

$$I = \frac{1}{N}\sum_{i=1}^{N} \nu_\beta(x_i). \tag{5}$$

where $\nu_\beta(x)$ is the fraction of the β nearest neighbors of x which have the same label as x.[9] This value I will be lowest when a clustering artificially breaks apart a tight group of data as the values of $\nu_\beta(x)$ will be small along the 'fault line' between the $k = 2$ clusters. On the other hand, if the two clusters are well separated in space, I will be close to 1. Based on empirical observation, we choose the point at which we decide to keep $k = 2$ to be $I > 0.75$.

We need to take care of one more small detail to use k-means with our EMD metric space. Standard k-means in a metric space requires the definition of a clustroid. In a Euclidean space the clustroid is defined as the center of mass (centroid) of the points. The clustroid does not seem like it would be a well-defined concept in the EMD sense. Fortunately, EMD has a particularly useful property in this respect . When applied in 1-D to equal weight distributions, EMD is simply the Manhattan distance (L_1 norm) between the cumulative distribution functions of the distributions in question. Furthermore, in the metric defined by the Manhattan distance, the analog to the centroid is well defined as the median vector of the data. So, if instead of treating the path data as is, we consider the cumulative distributions of the path vectors, we can do our clustering in the metric space defined by the Manhattan distance where the clustroid is well-defined.

2.1.4. Signatures

We proceed as above and perform the k-means clustering with our calculated optimal k in the EMD metric space for the paths corresponding to each pixel. The result of this clustering is a set of k clusters. We use this information to define a *signature* for each pixel v where signature s_v consists of the $\mathbf{m}_1, .., \mathbf{m}_k$ clustroids and the $w_1, ..., w_k$ numbers of elements belonging to each cluster. Having created a signature for each pixel value, we have learned the 'many' aspect of the many-to-one mapping. Now, we turn our attention to calculating a likely probability distribution over the representative clustroids given an entirely new fluoroscopic image to classify.

2.2. Calculation

Starting with a new fluoroscopic image, we would like to be able to use the signatures to say something about the paths used to create them. This is the final step and the one that actually assigns the fuzzy segmentation. Probabilistically speaking, we would like to recover the distribution over the paths $\mathbf{m}_1, .., \mathbf{m}_k$ of v's signature S_v conditioned on the pixel value v, or $P(\mathbf{m}_i|v)$. Since the pixel value alone is not enough information to help us, we can also throw in other information $I(v)$ such as v's neighbors and its position in the image to arrive at $P(\mathbf{m}_i|v, I(v))$ While this is something we cannot calculate directly, we can use Bayes' rule to represent this distribution as

$$P(\mathbf{m}_i|v, I(v)) = \frac{P(v, I(v)|\mathbf{m}_i)P(\mathbf{m}_i)}{P(v, I(v))}. \tag{6}$$

Since $P(\mathbf{m}_i)$ is simply related to the relative sizes of the clusters in S_v $(w_1, .., w_k)$ and $P(v, I(v))$ is the same for each i, all that's left is to calculate $P(v, I(v)|\mathbf{m}_i)$. We suggest a couple of ways to approximate these values and assign a distribution over the pixels and evaluate them versus a gold standard.

2.2.1. Maximum likelihood

The first, and most obvious way to approximate this distribution is to ignore all neighbor information and simply assign probabilities based on the relative likelihoods of each of the $\mathbf{m}_i$'s in the training set. This is the same as setting $P(v, I(v)|\mathbf{m}_i) = P(v|\mathbf{m}_i)$, which we know by construction to be equal to 1.

2.2.2. Neighbor information

Another way to approximate $P(v, I(v)|\mathbf{m}_i)$ is to consider the neighborhood information contained in $I(v)$. To do this, we take each cluster from each signature and, at several resolutions, calculate the median path up ($\mathbf{m}_i^u$), down ($\mathbf{m}_i^d$), left ($\mathbf{m}_i^l$), and right ($\mathbf{m}_i^r$) in the training data. Now, given a new test image, we look at pixel value v and its neighbors at the same resolutions. Each neighbor itself corresponds to a signature (S_u, S_d, S_l, S_r) and we assign $P(v, I(v)|\mathbf{m}_i)$ to be:

$$P(v, I(v)|\mathbf{m}_i) = \text{argmin}(j_1, j_2, j_3, j_4) \sum_{\alpha=1}^{L} [\text{EMD}(S_v^\alpha.\mathbf{m}_i^u, S_u^\alpha.\mathbf{m}_{j_1}) + \text{EMD}(S_v^\alpha.\mathbf{m}_i^d, S_d^\alpha.\mathbf{m}_{j_2}) + \text{EMD}(S_v^\alpha.\mathbf{m}_i^l, S_l^\alpha.\mathbf{m}_{j_3}) + \text{EMD}(S_v^\alpha.\mathbf{m}_i^r, S_r^\alpha.\mathbf{m}_{j_4})]. \tag{7}$$

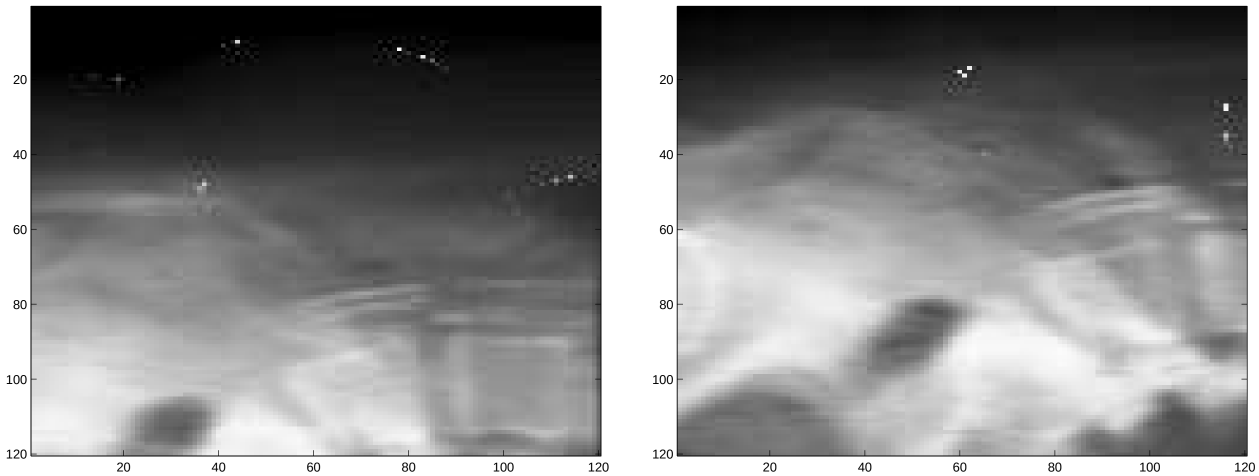


Figure 4. Two example training DRR images. The training set consisted of 80 images (120x120) of the lower lumbar vertebra taken from a lateral viewpoint. For each of the images, the focal point moved around in a sphere of a diameter of around 5 cm.

where, for example, $S_u^\alpha.\mathbf{m}_j$ is the j-th clustroid of signature S_u associated with the pixel above v at resolution level α, and L is the total number of resolution levels. In other words, given the neighborhood of v in the test image, we approximate $P(v, I(v)|\mathbf{m}_i)$ as the minimum EMD between $\mathbf{m}_i$ and its most likely configuration of neighboring paths. With this approximation in hand, all that we need to do is multiply by $P(\mathbf{m}_i)$ and renormalize to arrive at our final, fuzzy distribution.

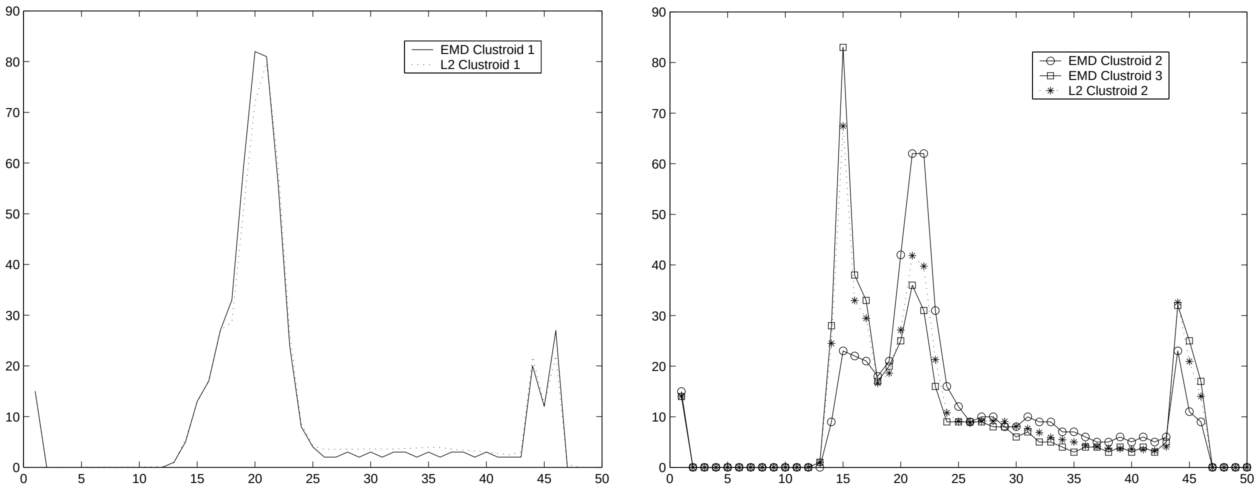
3. RESULTS

We tested our algorithm using the lateral viewpoint of a CT dataset of lower lumbar vertebra. We took 80 training images (resolution 120 × 120) (Figure 4) moving the focal point around an area with a diameter of roughly 5 cm and altering image plane orientation by angles on the order of 5 degrees. This gave us 1,152,000 paths with which to perform the clustering. We discretized the DRR values into 500 bins which each corresponded to its own unique signature.

We first set out to validate our use of the Earth Mover's Distance over the more standard L_2 norm. Figure 5 illustrates this point quite well. Using the same paths and clustering algorithm, we created signatures for the same pixel using the two different metrics. As you can see on the left, EMD and L_2 both agree on one of the clustroids. However, on the right we see that L_2 settled on just one additional clustroid, while EMD found 2 more, each of which looks quite different. This is the general trend we see in the data with the clustering based on EMD returning almost 100 more clustroids than that based on L_2.

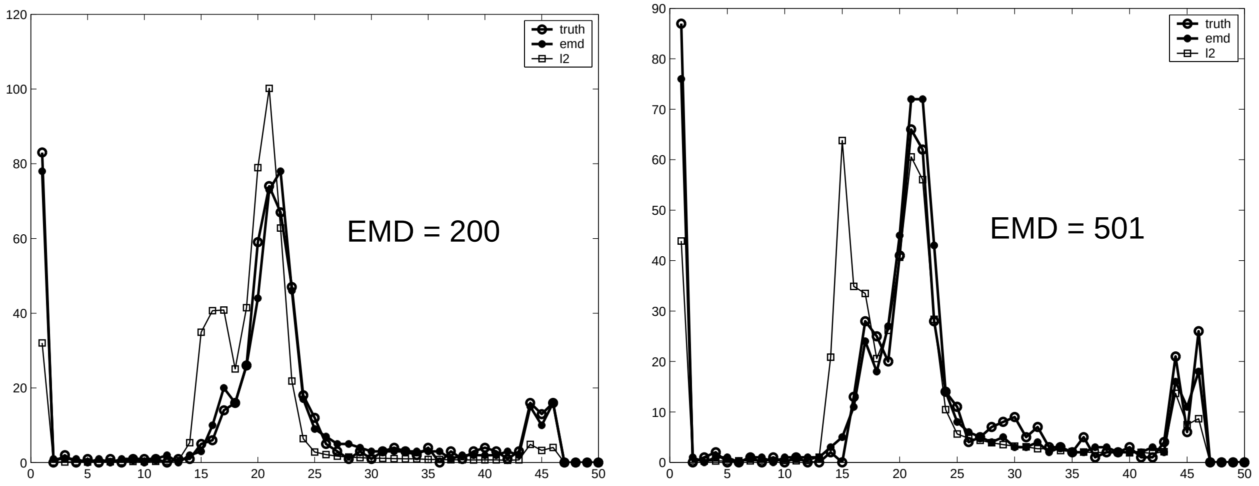
Our algorithm was tested on new images taken from the area around, but not in, our training set. That is, we calculate an entirely new DRR and treat it as a new fluoroscope image. By doing this, we have the actual paths used to calculate the DRR as a gold standard with which to compare our results. One way of validating our segmentation is to see if the clustroids we have learned in the signature actually accurately model the data. To test this, first we mapped each new pixel v to its signature S_v and then found the clustroid which minimized the EMD between it and the ground truth. If we accurately model the data, we should expect to see low value of the average EMD between a signatures best clustroid and the actual data. Figure 6 shows these results.

The average EMD over 10 test images taken within the range of the test set (144,000 paths) was 305 for that signatures learned with the EMD metric while the average for the L_2 metric signatures was 377. This experiment is particularly important as it indicates that even if we do not have the correct distribution over the clustroids in the signature for each pixel, as long as one of the clustroids is correct, a registration algorithm can use it to lock onto a solution in path space.



(a) (b)

Figure 5. Both figures represent two signatures for the same pixel value. One was derived from the vector quantization using the EMD metric space, the other using L_2. In (a) we see that the first clustroids agree. In (b) the EMD signature has two quite distinct clustroids while the L_2 signature has only one. The L_2 metric wasn't able to distinguish between these two, clearly different paths.



(a) (b)

Figure 6. Plots of the best EMD metric-based clustroid, the best L_2 metric-based clustroid, and the ground truth. The EMD is clearly better and differs from the truth by 200 in (a) and 501 in (b). Over all 10 test images (144,000) paths, the average EMD between the best EMD metric-based clustroid and the truth is 305.

Finally, we attempted to test our probability distribution approximation's ability to actually assign distributions to the clustroids of each signature. We used 10 images within the range of the test set and 10 test images in a range twice as large as that of the test set. The results are depicted in Table 1. While certainly demonstrating that use of neighbor information is better than a naive, maximum likelihood approach, they also suggest why a fuzzy segmentation is better than a real one. If you have a many-to-one mapping and use a hard segmentation, you will frequently throw out the correct solution. By keeping all of the solutions around as long as possible, if the right one is among them, you will be able to use it.

attempt	close neighbors	close mle	far neighbors	far mle
1st try	62%	42%	53%	41%
2nd try	91%	81%	90%	81%

Table 1. Validation of fuzzy segmentation against ground truth. This table summarizes the results of treating our fuzzy segmentation scheme as a hard segmentation and selecting the clustroid with the highest probability. The first row represents the percentage for which the clustroid with the highest probability is also the one closest to the ground truth. The second is the same but for the top two mostly likely clustroids.

4. DISCUSSION

This paper has added an important apparatus to a new technique for performing 2-D to 3-D fluoroscopy-to-CT registration. Currently, registration algorithms calculate DRRs and then try and match them up with fluoroscopic images using image-to-image similarity measures. This is a fundamentally ambiguous operation as there is more than one path through the CT that can create a given pixel value. Since we already go through the trouble of calculating each of the paths in creating the DRR, it would be nice to try and recover the 'many' of the many-to-one mapping when given only a fluoroscopy pixel value. This would allow us to perform the registration in path space without the ambiguities.

Towards this end, we've introduced a metric space from the image database literature that we demonstrate to be particularly good for comparing histograms of X-ray paths. This metric, the Earth Mover's Distance, when used in concert with techniques from unsupervised learning allows us to find the signature of a pixel, or the small number of representative paths that map to that pixel value in fluoroscopic images from a reasonably restricted viewpoint. All of this occurs offline. When given a new fluoroscopy image, we also introduce online algorithms for approximating the distribution over a pixel's signature. This distribution is a fuzzy segmentation of the fluoroscopy image into the potential path histograms that make up its elements.

It should again be noted that because we are using a fuzzy segmentation, we do not have to be perfect with these distributions. For the registration problem, what we need to insure is that, in general, one of the clustroids in a signature matches up correctly with the ground truth. As long as this is the case, a similarity metric based on the this data[4] should work.

As this research progresses, several avenues will need to be explored. First and foremost, we will need a better model of the X-ray process so that our DRRs match the actual fluoroscopy values as closely as possible. This would allow us to test our segmentation on actual fluoroscopic image data. Once this is done, we can proceed to test path space image registration and see how well it compares to image-to-image registration techniques.

ACKNOWLEDGMENTS

TR was supported by the National Science Foundation under Grant No. EIA-0104114. The authors acknowledge support for this research provided by CBYON, Inc., Mountain View, CA.

REFERENCES

1. V. Ganti, R. Ramakrishnan, J. Gehrke, A. Powell, and J. French, "Clustering large datasets in arbitrary metric spaces," in *Proceedings of the Fifteenth International Conference on Data Engineering*, pp. 502–511, 1999.
2. G. T. Herman, *Image Reconstruction from Projections*, Academic Press, 1980.
3. Z.-H. Cho, J. P. Jones, and M. Singh, *Foundations of Medical Imaging*, John Wiley & Sons, New York, NY, 1993.
4. T. Rohlfing, D. Russakoff, M. Murphy, and C. Maurer Jr., "An intensity-based registration algorithm for probabilistic images and its application for 2-D to 3-D image registration," in *Proceeding of SPIE Medical Imaging: Image Processing*, vol. 4684, 2002.
5. Y. Rubner, C. Tomasi, and L. J. Guibas, "A metric for distributions with applications to image databases," in *Proceedings of the 1998 IEEE International Conference on Computer Vision*, pp. 59–66, 1998.
6. R. Gray, *Entropy and Information Theory*, Springer-Verlag, 1990.
7. P. Bradley and U. Fayyad, "Refining initial points for *k*-means clustering," in *Proceedings of the 15th International Conference on Machine Learning*, pp. 91–99, 1998.

8. S. Ray and R. Turi, "Determination of number of clusters in k-means clustering and application in colour image segmentation," in *Proceedings of the 4th International Conference on Advances in Pattern Recognition and Digital Techniques*, pp. 137–143, 1999.
9. R. Pauwels and G. Frederix, "Non-parametric clustering for image segmentation and grouping," *Journal of Computer Vision and and Image Understanding* **75**, pp. 73–85, May 1999.