Boston University Computer Science Department Technical Report

BUCS-TR-2009-007

Example-Based Image Registration via Boosted Classifiers

William Mullally, Stan Sclaroff, and Margrit Betke

Department of Computer Science
Boston University
Boston, MA 02215, USA
http://www.cs.bu.edu/groups/ivc
{ mullally, sclaroff, betke} @cs.bu.edu

Abstract. We propose a novel image registration framework which uses classifiers trained from examples of aligned images to achieve registration. Our approach is designed to register images of medical data where the physical condition of the patient has changed significantly and image intensities are drastically different. We use two boosted classifiers for each degree of freedom of image transformation. These two classifiers can both identify when two images are correctly aligned and provide an efficient means of moving towards correct registration for misaligned images. The classifiers capture local alignment information using multi-pixel comparisons and can therefore achieve correct alignments where approaches like correlation and mutual-information which rely on only pixel-to-pixel comparisons fail. We test our approach using images from CT scans acquired in a study of acute respiratory distress syndrome. We show significant increase in registration accuracy in comparison to an approach using mutual information.

1 Introduction

Registration problems can be viewed as optimization problems in which an objective function is minimized when images are correctly aligned. Solving an optimization problem requires knowing how to search for the optimal solution as well as when to terminate the optimization procedure. Typical solutions to these problems use the gradient of the objective function to search for the optimal solution which is identified by a local minimum of that function. We propose a semi-automatic solution to this problem which uses classifiers trained from examples of aligned images to both direct the search for the solution as well as identify when to terminate the optimization procedure.

Semi-automatic registration algorithms use information from images aligned by trained experts to align images which the experts have not seen. Typically, these approaches learn a measure of image similarity between pairs of example images and use this learned similarity measure to align new image pairs. There is a growing body of work [1–6], in which the similarity measure used to register images is dependent upon examples of correctly aligned images. Several approaches use a measure of joint intensity distribution [1–3]. Two new images can be aligned by warping one of the images so that their joint intensity distribution as closely as possible resembles the joint intensity distribution of the example images.

As in the method proposed by Babenko et al. [6] our method uses a boosting approach for local region matching that uses a boosting approach to build classifiers from rectangular image features. Unlike our approach, Babenko, et al.'s approach uses classifiers to identify invariant feature points in two images and the exhaustive comparison of all feature points in a source image to the target image.

The work most closely related to our proposed method is Zhou et al.'s approach [5] for estimating the movement of a contour between two images based on a boosted, discriminative similarity function. They showed results for tracking a contour in echocardiographic sequences and used a similarity function that was based on a boosted classifier trained with a set of weak classifiers on pairs of image patches. Positive examples were correctly aligned image patches and negative examples were incorrectly aligned image patches. The weak classifiers were piecewise constant functions that approximated the response of the comparison of local rectangular regions in the image. The motion vector of a point on the contour was determined by exhaustively comparing points in a source image to all points within a window in the target image and selecting the point which was most confidently classified as an aligned point.

Unlike Zhou et al.'s method [5], which was designed to non-rigidly align a contour in a pair of images, our approach aligns all points in the images for global transformations. For each degree of freedom in the transformation, we train two classifiers: one classifier to recognize when that parameter has reached correct alignment as well as another classifier to indicate the direction the parameter needs to move in order to become closer to correct alignment. Classifiers are trained on small image patch pairs to capture information about local joint image appearance. We use these classifiers in a voting scheme to correctly register images. For each transformation parameter, we use the alignment classifier to vote on whether or not the image is aligned. The direction classifier for that transformation parameter is used to take a vote on which direction the parameter should change. The transformation parameter is changed in correspondence with the majority vote. This process iteratively checks for alignment and updates the transformation parameter. The iteration terminates when a local maximum is attained in the number of locations that are reported "aligned" or the maximum number of iterations has been reached. Our algorithm decouples the registration of each degree of freedom of the transformation and allows for rapid on-line image registration.

2 Method

We will first describe our on-line registration algorithm which uses a collection of classifiers to align images. We will then describe how we create these classifiers. An overview of our algorithm is shown in Figure 1.

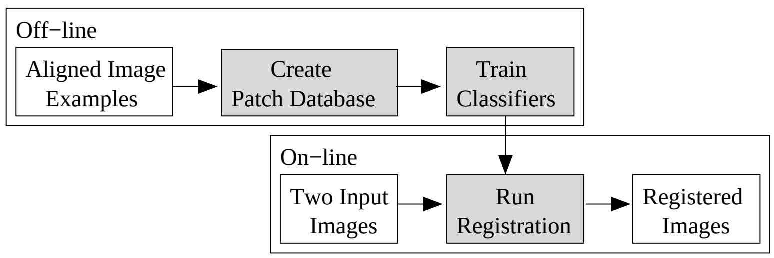


Fig. 1. An overview of our example-based approach to image registration. A collection of images and range of transformations is used to create set of labeled patch pairs. Alignment and Direction classifiers are trained on these patch pairs for each degree of freedom of the transformation parameters. The classifiers are then used to find correct registration of images. The training and registration portions of the algorithm can be run separately.

2.1 Image Registration

Let T be a function from $\Re^2$ to $\Re^2$ with N degrees of freedom parametrized by $\Theta = \theta_1 \ldots \theta_N$. Given two images I_1 and I_2, a registration between them is given by $T(\mathbf{x_1}, \mathbf{\Theta}) = \mathbf{x_2}$, where $\mathbf{x_1}$ is a point in I_1 and $\mathbf{x_2}$ is a point in I_2. Furthermore, let $p_{\Theta,\mathbf{x}}$ be a pair of image patches $i_{1,\mathbf{x}}$ and $i_{2,T(\Theta,\mathbf{x})}$ from locations $\mathbf{x}$ in image I_1 and $T(\Theta, \mathbf{x})$ in I_2. The goal is to find the parameter values of Θ that correctly align the two images.

Our method has an off-line component for training and an on-line component for registration as shown in Figure 1. In the off-line component of our framework, our algorithm creates a set of N binary classifiers, C_{Align} operating on p, to identify when correct alignment has been found. $C_{Align,n}$ indicates if parameter θ_n is aligned. In addition, the algorithm creates and uses another set of N binary classifiers, C_{Dir} operating on p, to provide an efficient means of searching the parameter space for a correct solution. The value of $C_{Dir,n}$ indicates if increasing or decreasing parameter θ_n will move images closer to a correct solution of the registration problem. The ideal responses of C_{Align} and C_{Dir} are a hat and a step function respectively. The on-line algorithm (detailed below) iteratively checks if a parameter value of Θ has produced a successful registration and updates those parameters that did not yield a correct registration. As Θ represents global transformation parameters, the majority vote of the classifiers over all corresponding points in the image is used to find updates for the image as a whole. This voting procedure allows our algorithm to tolerate up to 50% error in the classification of individual points at any iteration of the algorithm.

On-line Registration Algorithm
Input: $I_1, I_2, \Theta_{initial}$
for $n = 0$ to N,
  while$\left(\frac{1}{|I_{\Theta_n}|} \sum_{\mathbf{x}} C_{Align}(p_{\Theta,\mathbf{x}}) > \epsilon\right)$
    $dir = \sum_{\mathbf{x} \in \mathbf{I_\Theta}} C_{Dir,n}(p_{\Theta,\mathbf{x}})$
    $\theta_n = \theta_n + s_n \frac{dir}{|dir|}$
Output: Θ_{final}
where ϵ is a lower bound on the error, s_n is a step size set for each degree of freedom of T and I_{θ_n} is the portion of I_2 whose appearance can be changed by a change in θ_n.

2.2 Classifiers

We now describe the process of creating the classifiers C_{Align} and C_{Dir}. Given M image pairs $I_{m,1}$ and $I_{m,2}$, let correct alignment between each image pair be denoted by $T(\Theta^{(m)})$. Furthermore, given R transformations $T(\Theta^{(r)})$ which sample the transformation space in intervals equal to registration step size s_n for each degree of freedom θ_n, let $p_{m,\Theta^{(r)},\mathbf{x}}$ be a pair of image patches $i_{m,1,\mathbf{x}}$ and $i_{m,2,T(\mathbf{x},\mathbf{\Theta^{(m)}}+\mathbf{\Theta^{(r)}})}$ from locations $\mathbf{x}$ in images $I_{m,1}$ and $T(\mathbf{x}, \mathbf{\Theta^{(m)}} + \mathbf{\Theta^{(r)}})$ in $I_{m,2}$.

We denote set $\mathbf{D} = \{\mathbf{p_{m,\Theta^{(r)},x}}\}$ as the set of all such paired image patches used to train the classifiers. For each classifier, we divide the set $\mathbf{D}$ into two classes. The division of the data arises from the measured offset $\Theta^{(r)}$ of each patch pair from correct alignment. We use the labels $class_{-1}$ and $class_1$. Patch pairs are assigned to classes using the following functions:

$$l_{Align,n}(p_{m,\theta_n^{(r)},\mathbf{x}}) = \begin{cases} class_1 & \text{if } \theta_{\mathrm{n}}^{(\mathrm{r})} = 0, \\ class_{-1} & \text{otherwise}, \end{cases} \tag{1}$$

$$l_{Dir,n}(p_{m,\theta_n^{(r)},\mathbf{x}}) = \begin{cases} class_1 & \text{if } \theta_{\mathrm{n}}^{(\mathrm{r})} \geq 0, \\ class_{-1} & \text{otherwise}. \end{cases} \tag{2}$$

The function $l_{Align,n}$ is used for training $C_{Align,n}$, and $l_{Dir,n}$ is used for training $C_{Dir,n}$. It is important that the example set $\mathbf{D}$ varies over all transformation parameters and all examples are used to train each classifier. The differences in the examples used to train each classifier are represented by the class labels of the examples.

We create classifiers using Adaboost [7] on a collection of weak binary classifier based on Haar-like local rectangle features [8] which decide between $class_{-1}$ and $class_1$. The weak classifier compares rectangular regions belonging to two image patches. For each classifier, we define rectangular regions parametrized by (x, y, dx, dy, ρ) where (x, y) is the starting point of the box, (dx, dy) is the height and width of the box, and ρ is a weight. Let μ_a be the average intensity value and ρ_a be the weight of a box a belonging to the set of boxes A defined for image patch 1. Let μ_b be the average intensity value and ρ_b be the weight of a box b belonging to a set of boxes B defined for image patch 2. The classifier is created using the weighted sum of boxes A in comparison to the weighted sum of boxes B. We define the classifier as:

$$C(p) = \begin{cases} class_1 & \text{if } \sum_{\mathrm{a}\in\mathrm{A}} \rho_{\mathrm{a}}\mu_{\mathrm{a}} > \sum_{\mathrm{b}\in\mathrm{B}} \rho_{\mathrm{b}}\mu_{\mathrm{b}}, \\ class_{-1} & \text{otherwise.} \end{cases} \tag{3}$$

We have observed that we can achieve better accuracy if, instead of using a single boosted classifier, we partition the data and create a family of boosted classifiers from these partitions. We measure the average and standard deviation of intensity values from both source and target patches and partition the data in this four dimensional space. For any patch pair, let μ_1 and μ_2 be the average intensity for image patch i_1 and i_2 respectively, with standard deviations σ_1 and σ_2. The classification of this patch is then determined by the boosted classifier, $C(p) = C(\mu_1, \mu_2, \sigma_1, \sigma_2, p)$, trained on the partitioned data. We note that some partitions can be trained with examples from only one class. In these cases, the classifier will return that class label without computing a full chain of boosted classifiers.

3 Experiments and Results

We used 2D images from 5 volumetric CT scans of sheep before and after the infliction of acute respiratory distress syndrome (ARDS) [9]. This syndrome is characterized by the severe flooding and collapse of airways within the lung. This results in a dramatic, heterogeneous change in image appearance within regions of the lung (see sample images in Fig. 2). These changes are dramatic enough to introduce registration errors if standard similarity measures like correlation or mutual information are used. We used images sized 65 by 65 pixels and patches sized 5 by 5 pixels. We trained our classifiers from a set of 2,000 weak classifiers. Initial weights on training samples for boosting were proportionate to the total number of samples from each class.

We partitioned the training data along four dimensions using the average and standard deviation of intensity from both source and target patches. We created partitions of size 400 intensity units in μ_1 and μ_2 and 200 intensity units in σ_1 and σ_2. Classifiers are only created for partitions that have representatives in the training set, which in our experiments resulted in approximately 50 partitions.

As CT scans incorporate large areas of air surrounding a subject, we masked out areas of the images with intensity values lower than -900 HU. Training samples were only created when either the source patch, the target patch, or both contained some pixels that were not masked out. Masked regions were not included in the registration process.

We trained classifiers on the range of -3 to +3 pixels from correct alignment for translation in 1-pixel increments and -9^o to +9^o from correct alignment for

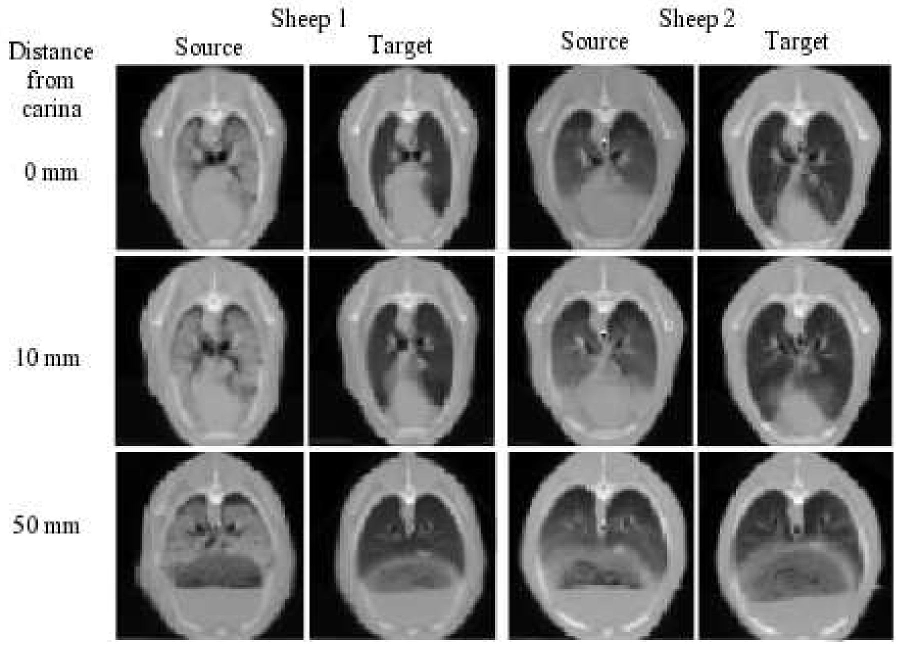


Fig. 2. Sample training and testing images from CT scans of two sheep. All training images included the carina. Images used in single scan experiments were offset in increments of 10 mm in the cranio-caudal direction from the training images.

rotation in 3^o increments (step size s_n). To test the accuracy of the registration, for each test pair we ran a registration with starting displacements in the range of -5 to +5 pixels from correct alignment for translation in 1-pixel increments and -15^o to +15^o from correct alignment for rotation in 3^o increments. From the possible 11^3 configurations of starting locations, we tested $11 \times 11 + 10 = 131$ configurations: all pairs of horizontal and vertical displacements with 0^o of rotational displacement and all rotation displacements with no translational displacement.

We tested our registration approach in two different ways. First, we trained classifiers on a single pair of images from one CT scan and used it to register 5 pairs of images of nearby slices from the same CT scan. Slices used were in 10 mm increments from the slice used to train the classifiers. We performed these single scan experiments on each of the 5 CT scans and report an average error of 0.23^o in rotation and 0.14 pixels in translation (Table 1). In comparison, using mutual information to achieve registration results in a much higher average error of 4.42^o in rotation and 5.73 pixels in translation.

We also performed a registration in which we trained classifiers using pairs of images from each of 4 different CT scans and tested registration on the fifth scan which was not part of the training. All image pairs were taken with the best view of the carina. We performed a round-robin experiment in which we tested all combinations of four training image pairs and one testing image pair. For these experiments, we report an average error of 4.29^o in rotation and 1.06 pixels in translation (Table 1). Using mutual information, the average error was of a similar level for rotations, 3.98^o, but higher for translation, 5.57 pixels.

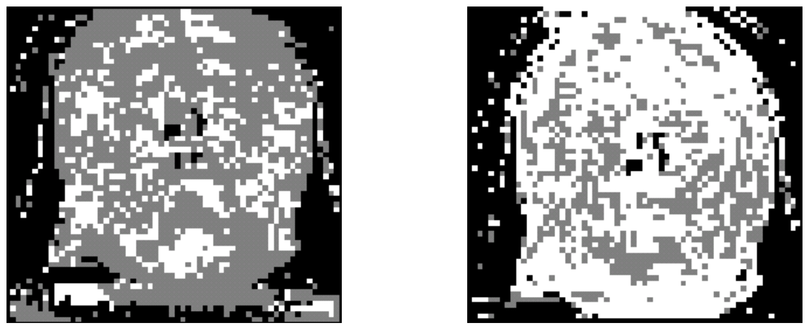
Table 1. Results. The last column refers to the number of experiments that resulted in an improved registration.

		Average Error	Std. Dev. Error	Number of Experiments	% Improved
Rotation Single Scan	Initial	8.18^o	5.04^o	$5 \times 5 \times 11=$ 275	
	Mutual Inform.	4.42^o	6.20^o		63%
	Our Approach	**0.23^o**	**0.67^o**		**90%**
Rotation Round Robin	Initial	8.18^o	5.04^o	$5 \times 11=$ 55	
	Mutual Inform.	3.98^o	6.58^o		62%
	Our Approach	**4.25^o**	**5.88^o**		**65%**
Translation Single Scan	Initial	4.19 pixels	1.56 pixels	$5^2 \times 11^2=$ 3,025	
	Mutual Inform.	5.73 pixels	2.10 pixels		9%
	Our Approach	**0.14 pixels**	**0.26 pixels**		**99%**
Translation Round Robin	Initial	4.19 pixels	1.56 pixels	$5 \times 11^2=$ 605	
	Mutual Inform.	5.57 pixels	2.02 pixels		9%
	Our Approach	**1.06 pixels**	**1.02 pixels**		**95%**

4 Discussion

For registering images from the same CT volume as the training images, our approach yields far lower error than a similar approach based on mutual information. In the round-robin experiment, our approach improves upon the initial misalignment of images. For rotation, it performs slightly less well than mutual information, while for translation it outperforms mutual information. In the round-robin experiment, the majority of large errors for our approach occurred in one test case. Without this test case, average error for our approaches shrinks to 2.18^o in rotation and 0.73 pixels in translation. Average errors for mutual information remain the same without this case.

As long as our approach can achieve classification rates lower than 50%, our solution is sufficient to find accurate registration of global registration parameters. Our training process can guarantee classification rates better than 50% on the training data. However, it does not guarantee that errors are not clustered. It is possible, for example, for C_{Align} to have an error rate of 1% on samples which are not correctly aligned but to have an error rate of 100% on samples which are correctly aligned. In such a case, the registration algorithm would never perfectly align images. In our experiments this has not occurred. Figures 3 shows how misclassifications tend to cluster regionally using our approach. It is part of our ongoing work to train classifiers in which errors do not cluster with respect to image locations and transformation parameters by using different error functions to train classifiers.



-5 pixels misalignment +5 pixels misalignment

Fig. 3. Classification of each pixel using a $C_{Dir,y}$. Locations where the classifier indicates alignment can be found by decreasing the y translation parameter are shown in white. Locations where the classifier indicates alignment can be found by increasing the y translation parameter are shown in gray. Regions in black are not part of the registration due to low attenuation values. Misclassifications are not evenly distributed in the image but are instead spatially clustered.

Our approach can be computationally expensive to train, but has faster online performance than that of the most similar approach [5]. Given the number of training images M, the size of the images I, the number of transformation parameters N, and the size of the set of transformations which sample the transformation space R, our algorithm has off-line complexity $O(MINR)$. Let $\hat{\theta}_n$ be the number of discrete samples of transformation parameter θ_n and $\max(\hat{\theta}_n)$ be the maximal number of samples of any of the parameters, then our on-line algorithm has complexity $O(IN \max(\hat{\theta}_n))$. In comparison, Zhou et al.'s boostmotion algorithm [5] has off-line complexity $O(MIR)$ and on-line complexity $O(IR)$. While our approach is computationally more expensive off-line, it is significantly faster on-line than Zhou et al.'s algorithm. This is because $R = \prod_N \hat{\theta}_n$ so typically $N \max(\hat{\theta}_n) \ll R$.

5 Conclusion

We have presented a novel framework for image registration based upon binary classifiers which estimate local alignment. Semi-automatic image registration can significantly reduce the amount of time medical practitioners need to analyze the imaging data they acquire. Our approach can be effectively used to register images in a volumetric CT scan given the alignment of a single slice in that scan.

In our experiments, our approach also successfully aligns images for one subject given examples of aligned scans from other subjects. Furthermore, at the expense of some additional off-line computation, we have shown a significant speed up in on-line performance in comparison to a similar approach. Our registration experiments suggest that boosted classifiers can be useful tools for aligning images in which disease and injury have dramatically changed image appearance.

Acknowledgment

This research was supported in part by NSF grants IIS-0705749 and IIS-0713229. We thank Kenneth Lutchen and Carissa Bellardine for providing the CT scans.

References

1. D. Cremers, C. Guetter, and C. Xu. Non-parametric priors on the space of joint intensity distributions for non-rigid multi-modal image registration. In *ICCV*. IEEE Computer Society, 2006.
2. C. Guetter, C. Xu, A. Saur, and J. Hornegger. Learning based non-rigid multi-modal image registration using Kullback-Leibler divergence. In *MICCAI'05*, pages 255–262, 2005.
3. M. E. Leventon and W. E. L. Grimson. Multi-modal volume registration using joint intensity distributions. In W. Wells, A. Colchester, and S. Delp, editors, *MICCAI'98*, pages 1057–1066. Springer Verlag, Berlin, 1998.
4. L. Zöllei, J. W. Fisher, and W.M.Wells. A unified statistical and information theoretic framework for multi-modal image registration. In *Information Processing in Medical Imaging (IPMI)*, pages 366–377, 2003.
5. S. K. Zhou, J. Shao, B. Georgescu, and D. Comaniciu. Boostmotion: Boosting a discriminative similarity function for motion estimation. In *CVPR'06*, 2006. 8 pp.
6. B. Babenko, P. Dollar, and S. Belongie. Task specific local region matching. In *ICCV'07*, 2007.
7. T. Hastie, J. Friedman, and R. Tibshirani. *The Elements of Statistical Learning*. Springer, 2001.
8. P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR'01*, pages 511–518, 2001.
9. W. Mullally, M. Betke, C. Bellardine, and K. Lutchen. Locally switching between cost functions in iterative non-rigid registration. In *Lecture Notes in Computer Science: Computer Vision For Biomedical Image Applications: First International Workshop, 2005*, volume 3765, pages 367–377, Beijing, China, October 2005. Springer Verlag.