Ensemble Registration: Aligning Many Multi-Sensor Images Simultaneously

Jeff Orchard[a], Laure Jonchery[b]

[a]David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Canada;
[b]École Nationale Supérieure d'Informatique et de Mathématiques Appliquées de Grenoble, Institut National de Grenoble, Grenoble, France

ABSTRACT

To register three or more images together, current approaches involve registering them two at a time. This pairwise approach can lead to registration inconsistencies. It can also result in diminished accuracy because only a fraction of the total data is being used at any given time. We propose a registration method that simultaneously registers the entire ensemble of images. This ensemble registration of multi-sensor datasets is done using clustering in the joint intensity space. Experiments demonstrate that the ensemble registration method overcomes serious issues that hinder pairwise multi-sensor registration methods.

Keywords: registration, multi-sensor, least-squares, clustering, generalized Euclidean distance

1. INTRODUCTION

Registration is the process of bringing images or volumes of the same object into alignment. It is a common image processing problem that crops up in many applications such as medical imaging, remote sensing, and computational vision.

One is often tasked with registering more than two images. For example, object tracking in video sequences, or registration of many medical images requires that many images of the same object be registered into a common reference frame. We refer to these image sets as *image ensembles*.

Currently, the vast majority of registration methods register only two images at a time. If more than two images are available, registering only two at a time can lead to problems. First of all, which pairs of images should be registered to each other? With four images, there are six different pairs of images that can be chosen and registered. Figure 1 shows that some pairs of images might be more suitable for registration than others. The figure shows a contrived example of multi-sensor imaging. The "true" object being imaged, shown on the left in Fig. 1, consists of a large circle encapsulating four ellipses. However, only two of the four ellipses are visible in each image, simulating the phenomenon of multi-sensor image acquisitions (eg. magnetic resonance imaging (MRI) vs. computed tomography (CT), or different frequency bands for remote sensing). Moreover, a different pair of ellipses is visible in each image, causing some pairs of images to contain disjoint content. Once the images were formed, we blurred the images slightly using a Gaussian kernel ($\sigma = 1$ pixel), and then added a small amount of Gaussian white noise ($\sigma = 5$). The figure shows that image pairs diagonally across from each other are more difficult to register.

Hence, the accuracy of the registration solution depends on which pairs of images are registered. In addition, once a set of image pairs has been registered, there is typically no way to guarantee that any redundancy is internally consistent. That is, the transform to register B-to-A is likely different from the composite transform of B-to-C followed by C-to-A. Finally, if one has multiple images at their disposal, it seems statistically sub-optimal to use only a subset of the information at a time.

Our hypothesis is that registering all available images simultaneously will allow us to use more information and guide us to a registration solution that is both more accurate, and internally consistent. We call this approach *ensemble registration*. Our goal is to implement such a registration method for ensembles of multi-sensor images.

Further author information: (Send correspondence to J.O.)
J.O.: E-mail: jorchard@uwaterloo.ca, Telephone: 1 519 888 4567 ext 35037
L.J.: E-mail: laure.jonchery@gmail.com, Telephone: +33 (0)4 76 82 72 24

To be presented at *SPIE Electronic Imaging*, Palo Alto, Jan. 2009.

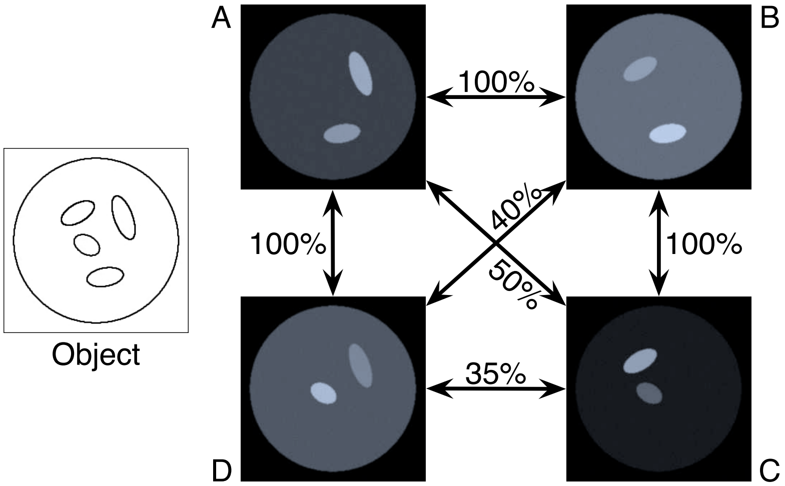


Figure 1. Success rate for pairwise registration of the multi-sensor phantom dataset. The edge labels give the percentage of trials for which FLIRT's 64-bin NMI registration had an average pixel displacement of less than 3 pixels.

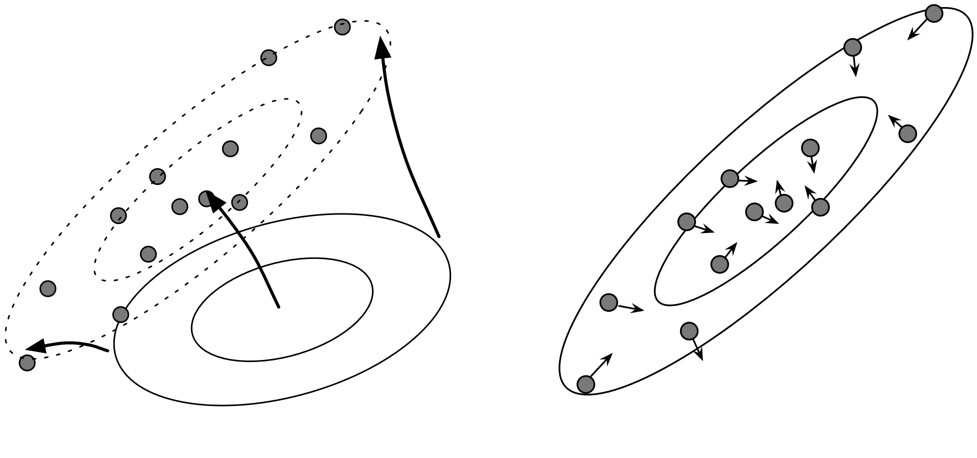
2. BACKGROUND

Extensive work has been done in the field of pairwise image registration. In the realm of multi-sensor image registration, popular methods include mutual information (MI)[1, 2] and normalized mutual information (NMI).[3] These methods are based on quantifying the dispersion in the joint intensity space. Consider a single pixel selected from two overlaid images. The intensity of that pixel in the two different images can be represented as a 2-tuple, and plotted as a single point in the joint intensity space, where each of the two axes represent the intensity in one of the images. The methods of MI and NMI measure the dispersion in the joint intensity space by forming a joint histogram and computing is entropy. While these methods could, in principle, be used to register three or more images simultaneously, computer memory requirements become an issue because the joint histogram must have an additional dimension for each image involved. For example, to register four images at the same time using MI requires a 4-dimensional joint histogram. If each dimension is broken into 256 bins, the resulting joint histogram will have over 4 billion bins! Thus, these histogram-based methods (including Kullback-Leibler distance methods[4, 5]) are not appropriate for simultaneously registering multiple images.

The first demonstration of ensemble registration that we are aware of was published in 1998 by Woods et al.[6] Given a set of images, Woods constructs his cost function by adding together the sum of squared differences (SSD) between all possible pairs of images. Minimizing this cost function yields what Woods calls the "reconciled mean transformations". However, since his method uses the SSD cost function, it is only suitable for mono-sensor image registration.

Some other ensemble registration methods have recently emerged in the literature.[7, 8] However, these methods have not been demonstrated on multi-sensor image ensembles, but rather focus on the problem of registering a set of images from the same modality to form a template (or so-called *atlas*). A different domain-specific method was designed to simultaneously register sets of brain MR images, but relies on the use of a human brain atlas to perform tissue classification, and then aligns the tissue-classification images.[9] Hence, these methods are not suitable for general-purpose multi-sensor ensemble registration. In another registration method, the points in the joint intensity scatter plot are treated as nodes in a graph, and the registration quality is measured by the length of the minimum-length spanning tree.[10]

We propose a general-purpose ensemble registration method that, like other multi-sensor registration methods, endeavors to reduce the dispersion of points in the joint intensity space. However, our method models the joint distribution directly using clustering, and therefor avoids the need to form the memory-intense joint histogram.



(a) Modeling

(b) Unscattering

Figure 2. The two main processes in ensemble registration

3. METHOD

3.1. Overview of Registration by Clustering

The general idea behind our registration approach is to reduce the dispersion in the joint intensity scatter plot by drawing the scatter points toward a small set of regressors planted in the joint intensity space. As registration proceeds, not only will the scatter points form tight clusters around the regressors, but the regressors will move to better accommodate the locations of these clusters. In essence, we want to move the regressors and images to minimize the sum of squared distances from our scatter points to these regressors.

We iteratively apply two steps, called *modeling* and *unscattering*. The modeling step holds the scatter points fixed, and adjusts the regressors. A regressor is a description of the position, orientation and elongation of a cluster of points. This process is depicted in Fig. 2(a). Conversely, the unscattering step holds the regressors fixed, and adjusts the scatter points (by moving the images) so that they move closer (on average) to the regressor centres. This process is depicted in Fig. 2(b). More details about these two steps follow.

Ultimately, our ensemble's joint intensity scatter plot will be represented by a set of regressors, each occupying a different region of the joint intensity space. After each modeling and unscattering step, the scatter points are assigned to their closest regressor, and that assignment is used in the next step. We call this process *assignment*.

3.2. Modeling

The clusters of points in the joint intensity space are often elongated. Why? As an illustration, consider white matter and gray matter in a human brain. The two tissues give a slightly different MR signal, so show up as slightly different pixel intensities in MR images. In fact, the distribution of intensities between the two tissues overlap considerably in many MR images. Moreover, many pixels in a brain image contain both white matter and gray matter, so we can expect a continuum of intensities between that of white matter and gray matter. Thus, when including these images in an ensemble of images, the white/gray matter pixels will form an elongated cluster in the joint intensity space.

We can model a cluster of points in the joint space as a focal point with a distribution around it that is not necessarily isotropic. The variance of a cluster of points can be encoded in a covariance matrix, denoted Σ. If μ is the cluster's centroid, the covariance can be computed using

$$\Sigma = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu)(x_n - \mu)^{\mathrm{T}} \ . \tag{1}$$

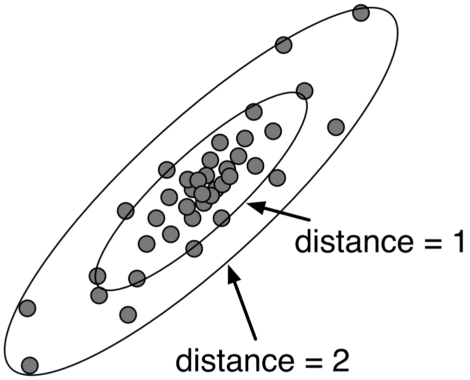


Figure 3. Cluster with two distance level-curves

Our desire is to incorporate this covariance matrix into our computation of dispersion. That is, we are interested in tabulating the degree to which points in the scatter plot stray from the clusters, but allowing for the fact that clusters can have different shapes. For example, if the cluster forms a cloud that is elongated and tilted on an angle, we want our cost function to penalize for points that do not conform to that distribution. This is what the Generalized Euclidean Distance (GED) does. In essence, it warps the space around a cluster centroid so that distance is measured relative to the cluster distribution. Figure 3 shows a cluster with unit-distance level-curves.

Once the covariance matrix is known, the GED between a point x and the cluster centroid μ can be computed using

$$d^2_{\mathrm{GED}}(x;\mu,\Sigma) = (x-\mu)^{\mathrm{T}}\Sigma^{-1}(x-\mu) \ . \tag{2}$$

It is worth mentioning that Σ is a symmetric positive semidefinite matrix, and can be thought of as a scaling operation in a rotated reference frame. One can see this by computing the eigenvector decomposition of Σ,

$$\Sigma = \mathbf{V}^{\mathrm{T}}\Lambda\mathbf{V} \ ,$$

where $\mathbf{V}$ is a unitary (rotation) matrix containing the eigenvectors, and Λ is a diagonal matrix with the (non-negative) eigenvalues on its diagonal. It can be shown that the square roots of the eigenvalues give the semi-axes of the unit-distance hyper-ellipses. Hence, the square roots of the eigenvalues are the scaling factors, and $\mathbf{V}$ is the change-of-basis matrix operator that rotates the frame of reference.

Substituting $\left(\mathbf{V}\Lambda^{-1}\mathbf{V}^{\mathrm{T}}\right)$ in for Σ^{-1} in (2) gives

$$\begin{aligned} d^2_{\mathrm{GED}}(x;\mu,\Sigma) &= (x-\mu)^{\mathrm{T}}\Sigma^{-1}(x-\mu) \\ &= (x-\mu)^{\mathrm{T}}\mathbf{V}\Lambda^{-1}\mathbf{V}^{\mathrm{T}}(x-\mu) \\ &= \left(\Lambda^{-1/2}\mathbf{V}^{\mathrm{T}}(x-\mu)\right)^{\mathrm{T}}\left(\Lambda^{-1/2}\mathbf{V}^{\mathrm{T}}(x-\mu)\right) \\ &= \|\Lambda^{-1/2}\mathbf{V}^{\mathrm{T}}(x-\mu)\|_2^2. \end{aligned} \tag{3}$$

Hence, a cluster in the joint space is modeled by a regressor at μ with covariance Σ, and the cost function for this cluster is

$$C = \sum_{n=1}^{N}(x_n-\mu)^{\mathrm{T}}\Sigma^{-1}(x_n-\mu) \ , \tag{4}$$

where the sum over n includes all pixels that are assigned to the regressor in question.

Notice that one way to minimize our cost is to allow our regressor to have very large (or infinite) covariance. Then, the GED would essentially be very small (or zero). To avoid this degenerate solution, we need to add a term to our cost function that penalizes for large covariances. Here is how. If we represent Σ as $\alpha\overline{\Sigma}$, where $\det\left(\overline{\Sigma}\right) = 1$, then we can write our cost as

$$C = \frac{1}{\alpha}\sum_{n=1}^{N}(x_n - \mu)^{\mathrm{T}}\overline{\Sigma}^{-1}(x_n - \mu) \ . \tag{5}$$

We add to our cost function a penalty term that is a multiple of the determinant of the covariance matrix. Our cost function then becomes

$$C = \gamma\alpha^{D} + \frac{1}{\alpha}\sum_{n=1}^{N}(x_n - \mu)^{\mathrm{T}}\overline{\Sigma}^{-1}(x_n - \mu) \ , \tag{6}$$

where D is the number of images in our ensemble, and γ is a chosen constant used to adjust the affect of the penalty term. It is simple to find the optimal α value by taking the derivative of (6) with respect to α and setting it equal to zero. The result is

$$\hat{\alpha} = \left(\frac{\sum_{i=1}^{N}(x_i - \mu)^{\mathrm{T}}\overline{\Sigma}^{-1}(x_i - \mu)}{D\gamma}\right)^{\frac{1}{D+1}} \ .$$

We then redefine Σ using $\hat{\alpha}$ so that $\Sigma = \hat{\alpha}\overline{\Sigma}$. This new, rescaled version of Σ is used as the regressor covariance.

Using GED, we can evaluate the total distance of all the scatter points from their nearest regressors. Thus, our total cost is

$$C = \sum_{n=1}^{N} d^2_{\mathrm{GED}}(x_m; \mu_{k_n}, \Sigma_{k_n}) \ , \tag{7}$$

where μ_{k_n} and Σ_{k_n} are the parameters that define the regressor closest to x_n.

Now that we know how to optimally model our clusters using GED, we turn our attention to the problem of finding the best set of motion increments ("nudges") that move the scatter points closer (on average) to these new regressors.

3.3. Unscattering

For simplicity, we describe the method using only three images, each with three motion parameters (rigid-body motion in 2D), even though the method works for any number of images and any number of motion parameters.

Consider the task of registering three images denoted $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$. When $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$ are overlaid, each pixel corresponds to one point in a 3D scatter plot, which can be written $\mathbf{s}_n = [\mathbf{a}_n \ \ \mathbf{b}_n \ \ \mathbf{c}_n]^{\mathrm{T}}$ (for notational brevity, we use n to index pixels, voxels, etc.). Our goal is to find $\mathbf{m_a}$, $\mathbf{m_b}$, and $\mathbf{m_c}$, the motion parameters that encode the relative spatial relationships among the three images.

Applying a motion to the images changes the intensities of their pixels. Hence, each scatter point in the joint intensity space moves as a function of the motion parameters $\mathbf{m} = [\mathbf{m_a}^{\mathrm{T}} \ \mathbf{m_b}^{\mathrm{T}} \ \mathbf{m_c}^{\mathrm{T}}]^{\mathrm{T}}$, since $\mathbf{s}_n(\mathbf{m}) = [\mathbf{a}_n(\mathbf{m_a}) \ \ \mathbf{b}_n(\mathbf{m_b}) \ \ \mathbf{c}_n(\mathbf{m_c})]^{\mathrm{T}}$. Hence, our cost function can be written as a function of our motion parameters,

$$C(\mathbf{m}) = \sum_{n} \|\Lambda_{k_n}^{-1/2}\mathbf{V}_{k_n}^{\mathrm{T}}(\mathbf{s}_n(\mathbf{m}) - \mu_{k_n})\|_2^2 \ . \tag{8}$$

Note that we are assuming that the motion increments are small, and that the identity of the nearest regressor does not change.

In the unscattering phase, the objective is to reduce the value of the cost function by moving the scatter points closer to their regressors (while holding the regressors fixed). This can only be accomplished by moving the images. Clearly, how the images are moved determines how the cost function changes value.

Suppose that $\mathbf{m_a} = [x_\mathbf{a}\ y_\mathbf{a}\ \theta_\mathbf{a}]^\mathrm{T}$ is the 3×1 vector of motion parameters that are applied to image $\mathbf{a}$. Then we can linearly approximate the n-th pixel intensity of $\mathbf{a}$ as,

$$\mathbf{a}_n(\mathbf{m_a}) \approx \mathbf{a}_n + \frac{\partial \mathbf{a}_n}{\partial x_\mathbf{a}} x_\mathbf{a} + \frac{\partial \mathbf{a}_n}{\partial y_\mathbf{a}} y_\mathbf{a} + \frac{\partial \mathbf{a}_n}{\partial \theta_\mathbf{a}} \theta_\mathbf{a} \ .$$

Linear approximations for $\mathbf{b}_n$ and $\mathbf{c}_n$ can be written similarly using motion parameters $\mathbf{m_b}$ and $\mathbf{m_c}$. Based on those linear approximations, we can represent $\mathbf{s}_n(\mathbf{m})$ as

$$\mathbf{s}_n(\mathbf{m}) = \begin{bmatrix} \mathbf{a}_n \\ \mathbf{b}_n \\ \mathbf{c}_n \end{bmatrix} + \begin{bmatrix} \frac{\partial \mathbf{a}_n}{\partial x_\mathbf{a}} & \frac{\partial \mathbf{a}_n}{\partial y_\mathbf{a}} & \frac{\partial \mathbf{a}_n}{\partial \theta_\mathbf{a}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\partial \mathbf{b}_n}{\partial x_\mathbf{b}} & \frac{\partial \mathbf{b}_n}{\partial y_\mathbf{b}} & \frac{\partial \mathbf{b}_n}{\partial \theta_\mathbf{b}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial \mathbf{c}_n}{\partial x_\mathbf{c}} & \frac{\partial \mathbf{c}_n}{\partial y_\mathbf{c}} & \frac{\partial \mathbf{c}_n}{\partial \theta_\mathbf{c}} \end{bmatrix} \begin{bmatrix} \mathbf{m_a} \\ \mathbf{m_b} \\ \mathbf{m_c} \end{bmatrix} . \qquad (9)$$

We can rewrite (9) using the more compact notation $\mathbf{s}_n(\mathbf{m}) = \mathbf{s}_n + \mathbf{L}_n\mathbf{m}$. Plugging that into (8) yields a cost function that is quadratic in $\mathbf{m}$,

$$\sum_n \left\| \Lambda_{k_n}^{-1/2} \mathbf{V}_{k_n}^\mathrm{T} (\mathbf{s}_n + \mathbf{L}_n\mathbf{m} - \mu_{k_n}) \right\|_2^2 . \qquad (10)$$

The entity inside the $\|\cdot\|_2$ delimiters is a vector. Notice that rather than using a summation of the square of the length of those vectors, we can rewrite our cost function in (10) by simply stacking the vectors into one giant vector, and taking the square of its length. Doing so gives us

$$\left\| \begin{bmatrix} \Lambda_{k_n}^{-1/2} \mathbf{V}_{k_n}^\mathrm{T} \mathbf{L}_1 \\ \vdots \\ \Lambda_{k_n}^{-1/2} \mathbf{V}_{k_n}^\mathrm{T} \mathbf{L}_N \end{bmatrix} \mathbf{m} + \begin{bmatrix} \Lambda_{k_n}^{-1/2} \mathbf{V}_{k_n}^\mathrm{T} (\mu_{k_1} - \mathbf{s}_1) \\ \vdots \\ \Lambda_{k_n}^{-1/2} \mathbf{V}_{k_n}^\mathrm{T} (\mu_{k_N} - \mathbf{s}_N) \end{bmatrix} \right\|_2^2 . \qquad (11)$$

If we define $\overline{\mathbf{L}}$ as the matrix containing the stack of $(\Lambda_{k_n}^{-1/2} \mathbf{V}_{k_n}^\mathrm{T} \mathbf{L}_n)$ matrices, and $\overline{\mathbf{d}}$ as the column-vector containing the stack of $\left(\Lambda_{k_n}^{-1/2} \mathbf{V}_{k_n}^\mathrm{T} (\mu_{k_n} - \mathbf{s}_n)\right)$ vectors, our registration problem becomes

$$\min_\mathbf{m} \left\| \overline{\mathbf{L}}\mathbf{m} + \overline{\mathbf{d}} \right\|_2^2 , \qquad (12)$$

which yields a solution by solving the normal equations,[11]

$$\overline{\mathbf{L}}^\mathrm{T} \left(\overline{\mathbf{L}}\mathbf{m} + \overline{\mathbf{d}} \right) = 0 \quad \Longrightarrow \quad \mathbf{m} = \left(\overline{\mathbf{L}}^\mathrm{T} \overline{\mathbf{L}} \right)^{-1} \overline{\mathbf{L}}^\mathrm{T} \overline{\mathbf{d}} \ . \qquad (13)$$

This square system of equations is relatively small in size, with one unknown for each motion parameter. For example, with 10 images, each with 6 motion parameters, the system has 60 unknowns. Solving such a linear system is trivial using today's computers.

Since the spatial transformation is linearly approximated, the linear least-squares problem above is embedded in a fixed-point iterative framework. After each iteration, the computed nudge, $\mathbf{m}$, is added to the current best-guess set of motion parameters. Note that even though our example has nine motion parameters, three of those can be chosen arbitrarily because we are ultimately only concerned with registering the images *with respect to each other*. Hence, we have adopted the custom of subtracting $\mathbf{m_a}$ from all the sets of motion parameters after each iteration so that, in essence, $\mathbf{a}$ remains fixed and the other images are registered to it. It is important to note, however, that the ensemble registration method still performs registration of all the images simultaneously.

We still need to address the issue of how one finds an initial set of regressors. Let us assume that the images are nearly registered. We do not see this as a flaw, since almost all registration methods make the same assumption – globally optimal registration methods only exist for very specific registration scenarios.[12, 13] The initial set of regressors is determined by automatic clustering of the joint intensity scatter plot. We use the method of normalized cuts[14] to find the most significant clusters, and place a point regressor at the centroid of each cluster. The number of initial regressors is specified by the user.

4. EXPERIMENTS

To test the efficacy of our multi-sensor ensemble registration method, we performed registration experiments on a number of different image ensembles. We implemented our registration method in Matlab (MathWorks Inc., Natick, Massachusetts). The method, which we will refer to as the "Full Ensemble" method, uses our GED-based clustering in the joint space to simultaneously registers all the images of an ensemble. For comparison, we also use two other registration methods. The first of those methods, called the "Pairwise Ensemble" method, also uses the ensemble registration technique, but only to register two images at a time. In a sense, this pairwise method acts as an experimental control to distinguish the effect of using the joint intensity regressors from that of performing full ensemble registration. The last method is another pairwise registration method that uses FLIRT's implementations of least-squares, correlation ratio, and normalized mutual information (with 64 histogram bins).[15] A custom "schedule" file was developed to guide FLIRT to search for the appropriate transformations with rotations and translations in the range of $[-10, 10]$ degrees or pixels (see[16] for a discussion of the optimization strategies used by FLIRT). We denote these methods by the cost function being used, eg. "NMI", "CR", and "LS".

The registration methods were then used to estimate the transformation parameters for a variety of test ensembles. Five different image ensembles were used for testing. The images within each ensemble were already registered so a gold standard registration was known. From each ensemble of images, test trials were generated by randomly displacing the images. Then, the registration methods were used to estimate the transformations among the images within each ensemble trial. For each ensemble trial, the pairwise registration methods registered each image to each other image (including symmetric pairs, so A was registered to B *and* B was registered to A in two separate runs). Hence, for a dataset of N images, the pairwise methods each performed a total of $N(N-1)$ registration runs. The full ensemble method was run once on each test ensemble to register all images simultaneously.

The performance of each registration method was gauged by comparing the estimated transformations to the gold standard transformations. This difference was quantified using the average pixel displacement – how far each pixel was from its true, registered position, averaged over all pixels used in the registration. Hence, an average pixel displacement of 0 indicates perfect registration, and a large average pixel displacement means poor registration.

4.1. Mono-Sensor Time Series

The first dataset was a time series of T2*-weighted MR images, typical of those acquired in a functional MRI experiment.[17] Starting with an 80 × 80 T2*-weighted MR image, the time-series consists of 56 copies of the original with simulated motion, activation and noise. Artificial activation at a level of 4% of the mean brain intensity was added to the occipital region (8 frames on, 8 frames off). Random rigid-body displacements were generated, uniformly choosing translations and a rotation from the range $[-10,\ 10]$ pixels or degrees. After applying the motion, we added Gaussian-distributed noise ($\sigma = 5$) to each of the 56 images. Because of the time it takes to complete the 3080 pairwise registration runs, we produced just one of these test ensembles. The full ensemble and the pairwise ensemble methods were each run with only one regressor. The FLIRT method was run using the least-squares cost function.

4.2. Variable Illumination

Figure 4 shows an image ensemble of the same face (640 × 480) with five very different light positions, ranging from far left to far right. Registration was performed using only the pixels in the region of interest, outlined in F1 of Fig. 4. This set of images poses a very difficult registration scenario, since image F1 and F5 have very little content in common. Ten trial ensembles were generated by applying randomly-generated rigid-body displacements, chosen uniformly from the range $[-10,\ 10]$ pixels or degrees.

The full ensemble image was run using six regressors, while the pairwise ensemble method was run using three regressors. The FLIRT method was run using the NMI cost function with 64 bins.

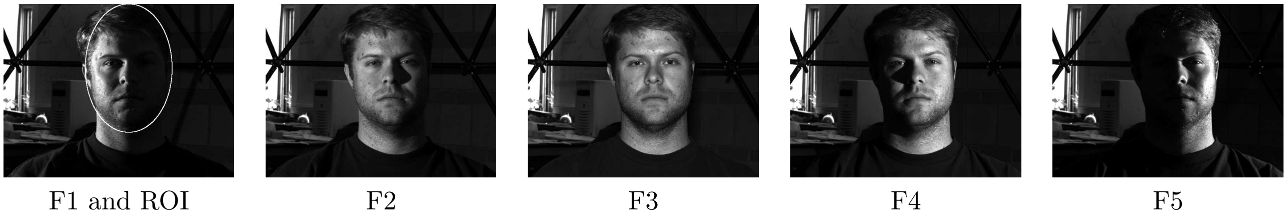


Figure 4. Face images used to test the effect of variable illumination. The ROI is outlined in image F1.

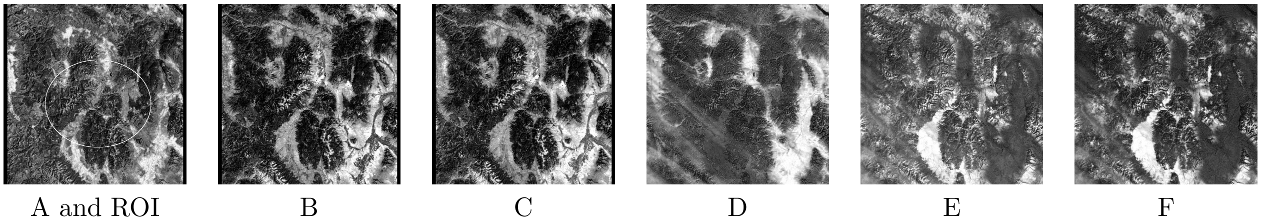


Figure 5. Satellite images used to test the affine registration. The ROI is outlined in image A.

4.3. Affine Registration

We also used six Landsat satellite images, each acquired with a different sensor. Figure 5 shows the images, with the region of interest used for the registration outlined in A. These six images (761 × 748 pixels in size) were used to generate ten trial ensembles, each with a different set of randomly-generated 6-degree-of-freedom affine displacements (two scales, one shear, one rotation, and two translations). The motion parameters were chosen uniformly from the following ranges: [0.95, 1.05] for scales, [-0.2, 0.2] for the shear, [-5, 5] degrees for the rotation, and [-5, 5] pixels for the translations. The full ensemble method used six regressors, while the pairwise ensemble method used 3 regressors. The FLIRT method was run using the correlation ratio as the cost function.

4.4. Three-Dimensional

We used the training dataset from the Retrospective Image Registration Evaluation (RIRE) project as a test case. This particular dataset consists of pre-registered PET, CT, and T1/T2/PD-weighted MR images, for a total of five registered volumes. We scaled down the volumes so that they were 80 × 80 × 26 voxels (isotropic) in size. Figure 6 shows the same slice from each of the volumes. Ten trial ensembles were generated by copying the volumes and applying a different set of randomly-generated rigid-body (6 degree-of-freedom) displacements (chosen uniformly from [-5, 5] voxels or degrees). The full ensemble method was run with six regressors, while the pairwise ensemble method was run with four. The FLIRT method was run using correlation ratio as a cost function.

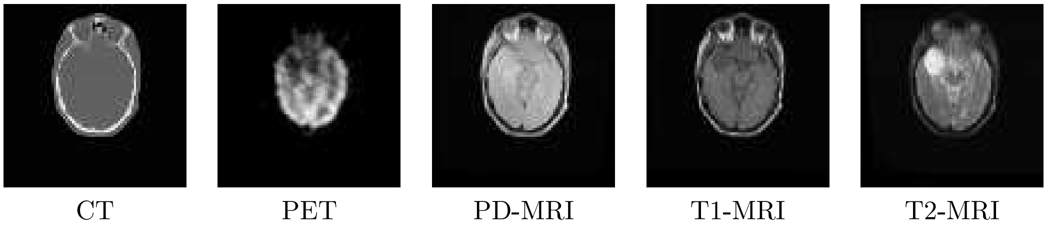


Figure 6. A slice from the RIRE ensemble used to test 3D registration.

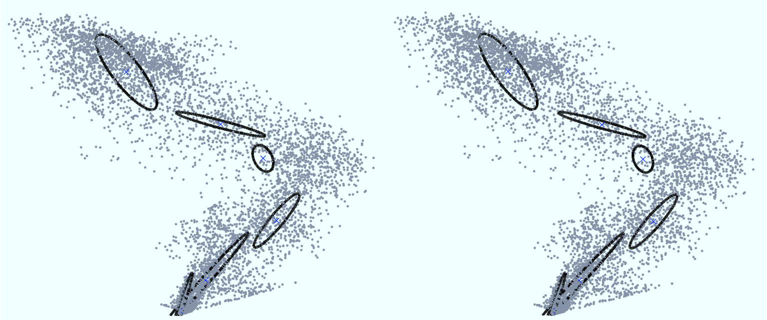


Figure 7. 3D joint intensity scatter plot (stereo pair) and corresponding regressors for the RIRE ensemble

Table 1. Average pixel displacement for face ensemble

Method	F1-F2	F1-F3	F1-F4	F1-F5	F2-F3	F2-F4	F2-F5	F3-F4	F3-F5	F4-F5	Mean
NMI (pairwise)	33	96	10	41	64	89	34	49	99	15	54.6
Ensemble (pairwise)	18	21	12	15	20	21	17	17	25	14	18.0
Ensemble (full)	0.46	0.97	1.8	1.8	1.1	2.1	2.0	1.7	2.1	0.44	1.45

4.5. Disjoint Content

Finally, ten trial ensembles of the multi-sensor phantom (shown in Fig. 1) were generated using randomly-generated rigid-body displacements chosen uniformly from the range $[-10,\ 10]$ pixels or degrees. The full ensemble method and the pairwise NMI method were run on the ten trial ensembles, and their average pixel displacements recorded. The full ensemble method used five regressors.

5. RESULTS

Figure 7 is a stereo-pair showing the 3D scatter plot for the RIRE test ensemble. Note that the ensemble has five images, so two dimensions of the five-dimensional joint intensity space are not represented. The ellipses depict the two largest axes of the hyper-ellipsoids, drawn in 3D. Observing the unit-ellipsoids among the scatter points shows that the regressors faithfully represent the location and conformation of the clusters in the joint intensity space.

Next, we study the quantitative results comparing the registration methods on the different test ensembles. For brevity, pairwise trials involving different permutations of a pair of images were averaged (i.e. trials registering A-to-B and B-to-A were combined into a single average).

5.1. Mono-Sensor Time Series

Among the 56 images in the simulated functional MRI time series, the pairwise methods performed 3080 pairwise registrations. All three methods correctly registered all the images. There were differences, however, in the accuracy. The mean average pixel displacement for the full ensemble method was 0.0234 pixels. The pairwise ensemble method was only slightly less accurate, with a mean average pixel displacement of 0.0266 pixels. The least-squares implementation of FLIRT yielded a mean average pixel displacement of 0.0872, over 3.5 times larger than the error displayed by the full ensemble method.

5.2. Variable Illumination

The pairwise methods had great difficulty in registering the face images with variable illumination. The average pixel displacements, averaged over all ten trials, are listed in Table 1.

Table 2. Average pixel displacement for 3D RIRE ensemble

Method	CT-PET	CT-PD	CT-T1	CT-T2	PET-PD	PET-T1	PET-T2	PD-T1	PD-T2	T1-T2	Mean
NMI (pairwise)	6.1	7.7	2.6	8.2	7.6	5.9	6.7	6.0	4.9	7.7	6.34
Ensemble (pairwise)	1.9	0.80	1.1	1.2	2.6	2.9	2.6	0.93	1.4	2.0	1.73
Ensemble (full)	1.9	0.74	0.87	0.72	1.8	1.7	1.4	0.50	0.73	0.92	1.13

Table 3. Average pixel displacement for multimodal phantom

Method	A-B	A-C	A-D	B-C	B-D	C-D	Mean
NMI (pairwise)	0.59	4.1	0.31	0.83	7.2	4.2	2.87
Ensemble	0.54	0.67	0.32	0.31	0.53	0.64	0.50

5.3. Affine Registration

Out of the 300 pairwise registrations of the 10 trials of the six satellite images, the full ensemble method failed on 20 of them. The pairwise ensemble method failed on 177 pairs, and FLIRT's correlation ratio method failed on 42 of the pairs. It is worth noting that the 20 misregistration cases for the full ensemble method were the result of only two registration failures. In each case, one of the six images failed to converge to the others, and was thus recorded as 10 misregistered image pairs.

5.4. Three-Dimensional

The results for the 3D registration test, using the RIRE ensemble, are shown in Table 2. Both pairwise methods were less accurate than the full ensemble method. The full ensemble method successfully registered the entire set of images, with the exception of what appears to be a slight misregistration of the PET image.

5.5. Disjoint Content

As Table 3 shows, a number of image pairs in the multi-sensor phantom proved to be very difficult for the pairwise methods to register accurately. The most notable case is the pairwise registration of A-to-C, and B-to-D. In those image pairs, none of the smaller ellipses are common between the two images. That is, neither of the two ellipses in A are visible in C (and *vice versa*). It should come as no surprise, then, that the pairwise methods exhibited much larger errors for those image pairs. However, the full ensemble method was able to register those images without issue.

No specific timing tests were performed, but it is worth noting that the speed of the full ensemble method was comparable to – and often faster than – the pairwise registration methods.

6. CONCLUSIONS

The full ensemble registration method outlined in this paper shows improved accuracy compared to pairwise registration methods. Part of its improved accuracy comes from a reduction in the number of registration failures, where the pairwise methods diverge from the true solution.

A major component of the full ensemble method's success is the fact that it is less prone to spurious data such as noise. Since it uses all the images simultaneously, the effect of noise is spread out. Using only two images at a time puts more weight on the noise that happens to be present in those two images.

The full ensemble method can even register images that have no content in common because it implicitly couples all the images together. The disjoint multi-sensor phantom experiment demonstrates this concept nicely. Think of each image as a node in a graph, and shared content between two images as an edge in that graph. In principle, the entire ensemble of images can be registered so long as the graph is connected. Moreover, the more edges in the graph, the more statistically robust the registration will be.

The full ensemble method used in our experiments is a prototype, and more investigations are needed to establish its limitations and potential. One area of possible improvement is in finding an initial set of regressors, since the placement of these regressors influences how the solution evolves. Despite our success using a simple normalized min-cut method,[14] we plan to investigate better ways to find a dependable initial clustering.

A characteristic of our method is the assignment of scatter points to regressors. In minimizing our cost function, a scatter point is pulled toward its nearest regressor with a "force" proportional to the square of its distance from that regressor. However, if a scatter point moves slightly and happens to land closer to a different regressor, then the force suddenly chances, pulling the point toward the new regressor. How this sudden discontinuity affects our solutions is not clear. We plan to investigate this issue further, as well as implement a similar method with "soft assignment", where each scatter point is assigned a membership vector indicating how much it belongs to each regressor.

ACKNOWLEDGMENTS

The authors thank the following organizations for the use of their data: the Retrospective Image Registration Evaluation project, Landsat, and the Yale Face Database. The authors are also thankful to the Natural Science and Engineering Research Council of Canada and the Canada Foundation for Innovation for funding of this project. J. O. is also grateful to Alexander Wong and Prof. Richard Mann for stimulating and helpful conversations.

REFERENCES

1. A. Collignon, F. Maes, D. Delaere, D. Vandermeulen, P. Suetens, and G. Marchal, "Automated multi-modality image registration based on information theory," in *Proceedings of Info Proc Med Imaging*, Y. Bizais, C. Barillot, and R. Di Paola, eds., pp. 263–274, 1995.
2. W. M. Wells III, P. Viola, H. Atsumi, S. Nakajima, and R. Kikinis, "Multi-modal volume registration by maximization of mutual information," *Med Image Anal* **1**(1), pp. 35–51, 1996.
3. C. Studholme, D. L. G. Hill, and D. L. Hawkes, "An overlap invariant entropy measure of 3D medical image alignment," *Pattern Recognition* **32**, pp. 71–86, 1999.
4. A. C. Chung, W. M. Wells III, A. Norbash, and W. E. L. Grimson, "Multi-modal image registration by minimising Kullback-Leibler distance," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI'02)*, T. Dohi and R. Kikinis, eds., **LNCS 2489 (part II)**, pp. 525–532, 2002.
5. C. Guetter, C. Xu, F. Sauer, and J. Hornegger, "Learning based non-rigid multi-modal image registration using Kullback-Leibler divergence," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI'05)*, J. Duncan and G. Gerig, eds., **LNCS 3750**, pp. 255–262, 2005.
6. R. P. Woods, S. T. Grafton, C. J. Holmes, S. R. Cherry, and J. C. Mazziotta, "Automated image registration: I. General methods and intrasubject, intramodality validation," *J Comput Assist Tomogr* **22**, pp. 139–152, 1998.
7. K. K. Bhatia, J. V. Hajnal, B. K. Puri, A. D. Edwards, and D. Rueckert, "Consistent groupwise non-rigid registration for atlas construction," in *Proc.of the IEEE International Symposium of Biomedical Imaging (ISBI'04)*, **1**, pp. 908–911, April 2004.
8. L. Z ollei, E. Learned-Miller, E. Grimson, and W. Wells, "Efficient population registration of 3D data," in *First International Workshop on Computer Vision for Biomedical Image Applications*, **LNCS 3765**, pp. 291–301, October 2005.
9. P. Lorenzen, M. Prastawa, B. Davis, G. Gerig, E. Bullitt, and S. Joshi, "Multi-modal image set registration and atlas formation," *MIA* **10**, pp. 440–451, 2006.
10. H. Neemuchwala, A. Hero, and P. Carson, "Image matching using alpha-entropy measures and entropic graphs," *Signal Processing* **85**(2), pp. 277–296, 2005.
11. G. H. Golub and C. F. Van Loan, *Matrix Computations*, Third Edition, Johns Hopkins University Press, Baltimore, MD, 1996.
12. B. S. Reddy and B. N. Chatterji, "An FFT-based technique for translation, rotation, and scale-invariant image registration," *IEEE Trans Image Proc* **5**(8), pp. 1266–1271, 1996.

13. G. Wolberg and S. Zokai, “Robust image registration using log-polar transform,” in *Proc. Int. Conf. on Image Processing*, **1**, pp. 493–496, 2000.
14. J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Trans Pattern Analysis and Machine Intelligence* **22**, pp. 888–905, August 2000.
15. M. Jenkinson and S. Smith, “A global optimisation method for robust affine registration of brain images,” *Med Image Anal* **5**(2), pp. 143–156, 2001.
16. M. Jenkinson, P. Bannister, M. Brady, and S. Smith, “Improved optimization for robust and accurate linear registration and motion correction of brain images,” *NeuroImage* **17**(2), pp. 825–841, 2002.
17. P. A. Bandettini, E. C. Wong, R. S. Hinks, R. S. Tikofsky, and J. S. Hyde, “Time course EPI of human brain function during task activation,” *Magn Reson Med* **25**, pp. 390–397, 1992.