A parallel implementation of 2-D/3-D image registration for computer-assisted surgery

Fumihiko Ino*, Yasuhiro Kawasaki and Takahito Tashiro

Graduate School of Information Science and Technology,
Osaka University, 1-3 Machikaneyama,
Toyonaka, Osaka 560-8531, Japan
E-mail: ino@ist.osaka-u.ac.jp E-mail: y-kawask@ist.osaka-u.ac.jp
E-mail: tashiro@ist.osaka-u.ac.jp
*Corresponding author

Yoshikazu Nakajima

Intelligent Modeling Laboratory,
The University of Tokyo, 2-11-1 Yayoi,
Bunkyo-ku, Tokyo 113-8656, Japan
E-mail: nakajima@iml.u-tokyo.ac.jp

Yoshinobu Sato and Shinichi Tamura

Graduate School of Medicine,
Osaka University, 2-2 Yamadaoka,
Suita, Osaka 565-0871, Japan
E-mail: yoshi@image.med.osaka-u.ac.jp
E-mail: tamuras@image.med.osaka-u.ac.jp

Kenichi Hagihara

Graduate School of Information Science and Technology,
Osaka University, 1-3 Machikaneyama,
Toyonaka, Osaka 560-8531, Japan
E-mail: hagihara@ist.osaka-u.ac.jp

Abstract: Image registration is a technique usually used for aligning two different images taken at different times and/or from different viewing points. A key challenge for medical image registration is to minimise computation time with a small alignment error in order to realise computer-assisted surgery. In this paper, we present the design and implementation of a parallel two-dimensional/three-dimensional (2-D/3-D) image registration method for computer-assisted surgery. Our method exploits data parallelism and speculative parallelism, aiming at making computation time short enough to carry out registration tasks during surgery. Our experiments show that exploiting both parallelisms reduces computation time on a cluster of 64 PCs from a few tens of minutes to less than a few tens of seconds, a clinically compatible time.

Keywords: image registration; medical image processing; high performance computing; MPI; performance evaluation; optimisation; bioinformatics research and applications; computer-assisted surgery.

Reference to this paper should be made as follows: Ino, F., Kawasaki, Y., Tashiro, T., Nakajima, Y., Sato, Y., Tamura, S. and Hagihara, K. (2006) 'A parallel implementation of 2-D/3-D image registration for computer-assisted surgery', *Int. J. Bioinformatics Research and Applications*, Vol. 2, No. 4, pp.341–358.

Biographical notes: Fumihiko Ino received his BE, ME and PhD Degrees in Information and Computer Sciences from Osaka University, Osaka, Japan, in 1998, 2000 and 2004, respectively. He is currently an Assistant Professor in the Graduate School of Information Science and Technology at Osaka University. His research interests include parallel and distributed systems, software development tools, and performance evaluation. He received the Best Paper Award at the *10th International Conference on High Performance Computing (HiPC'03)*.

Yasuhiro Kawasaki received the BE and ME Degrees in Information and Computer Sciences from Osaka University, Osaka, Japan, in 2002 and 2004, respectively. He is currently pursuing his PhD at the Department of Computer Science, Graduate School of Information Science and Technology, Osaka University. His current research interests include high performance computing, grid computing, and systems architecture and design. He received the Best Paper Award at the *10th International Conference on High Performance Computing (HiPC'03)*.

Takahito Tashiro received his BE and ME Degrees in Information and Computer Sciences from Osaka University, Osaka, Japan, in 2002 and 2004, respectively. He is currently pursuing his PhD Degree at the Department of Multimedia Engineering, Graduate School of Information Science and Technology, Osaka University.

Yoshikazu Nakajima received his BE Degree from Fukui University, Fukui, Japan, in 1992 and his PhD Degree in Computer Science from Osaka University, Osaka, Japan, in 1997. He then joined the Information Technology R&D Center, Mitsubishi Electric Corporation, Kamakura, Japan. From 2001 to 2005, he was an Assistant Professor at the Graduate School of Medicine, Osaka University. He is currently an Associate Professor of the Intelligent Modelling Laboratory at the University of Tokyo. His research interests include computer vision, image processing, and image based methods for medical and clinical applications.

Yoshinobu Sato received his BS, MS and PhD Degrees in Information and Computer Sciences from Osaka University, Osaka, Japan, in 1982, 1984, 1988 respectively. From 1988 to 1992, he was a Research Engineer at the NTT Human Interface Laboratories, Yokosuka, Japan. In 1992, he joined the Division of Functional Diagnostic Imaging of Osaka University Medical School as a Faculty Member. He is currently an Associate Professor in the Graduate School of Medicine and Graduate School of Engineering Science at Osaka University, where he leads a group conducting research on 3-D image analysis and surgical navigation systems.

Shinichi Tamura received his BS, MS and PhD Degrees in Electrical Engineering from Osaka University, Osaka, Japan, in 1966, 1968 and 1971, respectively. He is currently a Professor at the Graduate School of Medicine and Graduate School of Engineering Science of Osaka University. He has published over 200 papers in scientific journals and received several paper awards from journals including *Pattern Recognition* and *Investigative Radiology.* His current research activities include works in the field of medical image processing and its applications. He is currently an Associate Editor of *Pattern Recognition* and a Vice Editor-in-Chief of *Medical Imaging Technology.*

Kenichi Hagihara received his BE, ME and PhD Degrees in Information and Computer Sciences from Osaka University, Osaka, Japan, in 1974, 1976 and 1979, respectively. From 1994 to 2002, he was a Professor in the Department of Informatics and Mathematical Science, Graduate School of Engineering Science, Osaka University. Since 2002, he has been a Professor in the Department of Computer Science, Graduate School of Information Science and Technology, Osaka University. His research interests include the fundamentals and practical application of parallel processing. He received the Best Paper Award at the *10th International Conference on High Performance Computing (HiPC'03).*

1 Introduction

Image registration (Hajnal et al., 2001) is a technique for finding point correspondences between two different images, usually taken at different times, from different viewing points, and/or in different modalities. This technique plays an increasingly important role in surgery (Gueziec et al., 1998; Joskowicz et al., 1998). For example, registration of preoperative images to intraoperative images is essential to perform surgical procedures according to a preoperative surgical plan. This is because surgical plans are developed in the coordinate system relative to the preoperative data, while the surgical procedure is performed in the coordinate system relative to the patient (realworld). In this case, the registration technique relates the data coordinate system with the patient coordinate system, enabling image-guided or robot-assisted surgery (Gueziec et al., 1998; Herring et al., 1998; Weese et al., 1997), which minimises surgical complications and provides for better surgical outcomes.

For this purpose, many researchers have tackled the problem of 2-D/3-D registration (Lemieux et al., 1994), which estimates the location and orientation of a 3-D volume with respect to the patient coordinate system using one or more 2-D projected images. The reasons for aligning a 3-D volume with 2-D images are the limitations of current 3-D imaging systems, such as helical Computed Tomography (CT) scans, which have more spatial information but require more acquisition time and radiation exposure as compared to 2-D imaging systems. Due to these limitations, the intraoperative data are usually 2-D X-ray fluoroscopy or ultrasound images, whereas the preoperative data are 3-D CT volumes, motivating us to deal with 2-D/3-D registration.

One key challenge for 2-D/3-D registration is to develop a fast, accurate, and robust algorithm. Prior algorithms can be classified into two groups: feature-based and intensity-based approaches. The intensity-based approach has been shown to provide more accurate and robust results than the feature-based approach (McLaughlin et al., 2002; West et al., 1999). The feature-based approach uses geometrical features such as contours (Feldmar et al., 1997; Lavallée and Szeliski, 1995) and surfaces

(Herring et al., 1998; Maurer et al., 1998). It provides fast alignments but needs us to manually find correct contours in the projected 2-D images or to extract precise surfaces from the 3-D volume. This intervention is a serious drawback because precise features, which are manually selected, are essential to obtain accurate registration results. On the other hand, the intensity-based approach (Lemieux et al., 1994; Penny et al., 1998; Weese et al., 1997; Zollei et al., 2001) compares the intensity values between the real projected 2-D image and the Digitally Reconstructed Radiograph (DRR) which is generated from the volume. It requires a large amount of computation to iteratively produce DRRs until a best match between the real image and the DRR is found. Thus, there is a tradeoff between computation time and manual intervention.

In this paper, we present the design and implementation of a parallel 2-D/3-D registration method, aiming at achieving fast, accurate and robust alignments for computer-assisted surgery. Our method parallelises an intensity-based algorithm to reduce computation time without either degrading the quality of alignment or requiring manual intervention. The key contribution of our method is to provide fast and robust alignments by means of data-parallel and speculative processing, respectively. We also demonstrate that exploiting two parallelisms on our cluster, namely, data and speculative parallelism in registration tasks, reduces computation time sufficiently to utilise the registration technique during surgery.

The paper is organised as follows: Section 2 describes the intensity-based registration algorithm employed in our parallel method. The design aspects of our method are presented in Section 3 with a theoretical performance analysis. Section 4 shows experimental results using a cluster of 64 PCs. Section 5 presents related work. Finally, Section 6 concludes the paper.

2 Intensity-based 2-D/3-D registration

The intensity-based algorithm employed in our method has the following three advantages:

- automated registration by comparing a real projected image and a DRR (Lemieux et al., 1994)
- robust registration using an information-based similarity measure (Penny et al., 1998)
- accurate registration using biplane 2-D images (Li et al., 1994) and Region Of Interest (ROI) (Weese et al., 1997).

In addition to the earlier advantages mentioned above, our method aims at providing fast and robust registration by means of data-parallel and speculative processing.

Before describing each advantage, we first define the 2-D/3-D registration problem. To make it easier, we present a definition for a single image rather than biplane images. Given a volume V and a real projected image I_F (see Figure 1), the 2-D/3-D registration technique computes the rigid transformation parameter $\mathbf{T}$ that relates the coordinate system of the volume V and that of the imaging (patient) environment. Here, the rigid transformation is given by $\mathbf{T} = (T_X, T_Y, T_Z, \theta_X, \theta_Y, \theta_Z)$, where the first and last three parameters are the translations and rotations of V.

Figure 1 Overview of intensity-based 2-D/3-D registration. In this case, it aligns a CT volume of a real spine to a fluoroscopy image of the spine

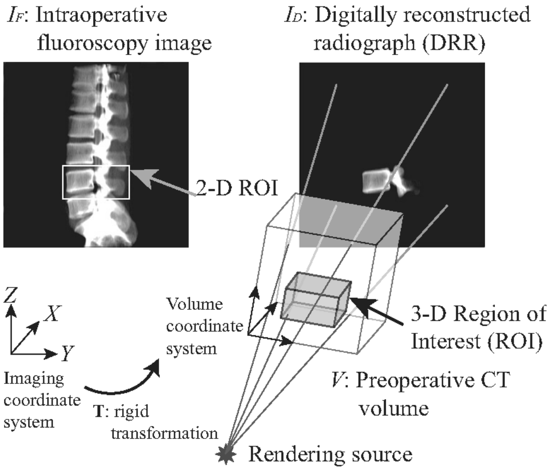
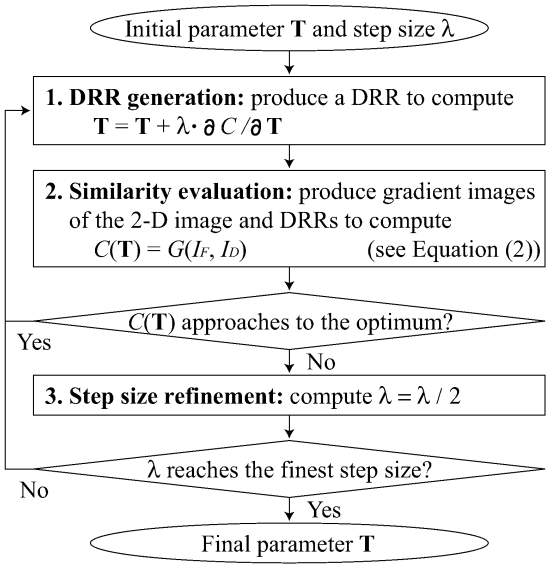

I_F: Intraoperative fluoroscopy image
I_D: Digitally reconstructed radiograph (DRR)
2-D ROI
Z
X
Y
Imaging coordinate system
Volume coordinate system
T: rigid transformation
3-D Region of Interest (ROI)
V: Preoperative CT volume
Rendering source

Figure 2 briefly presents the intensity-based algorithm. The algorithm resolves the registration problem into an optimisation problem. That is, in order to register the volume V to the 2-D image I_F, the algorithm optimises a cost function C associated with the location and orientation of V, where C represents the similarity measure between the 2-D image I_F and the DRR I_D, which is generated from V. Furthermore, this optimisation is performed in a hierarchical manner in order to reduce the amount of computation. This hierarchy is controlled by the step size λ of the optimisation.

Figure 2 Intensity-based 2-D/3-D registration algorithm. The algorithm maximises a similarity measure through the use of the steepest descent optimisation. This optimisation is performed in a coarse-to-fine manner


Initial parameter **T** and step size λ
1. DRR generation: produce a DRR to compute $\mathbf{T} = \mathbf{T} + \lambda \cdot \partial C / \partial \mathbf{T}$
2. Similarity evaluation: produce gradient images of the 2-D image and DRRs to compute $C(\mathbf{T}) = G(I_F, I_D)$ (see Equation (2))
$C(\mathbf{T})$ approaches to the optimum?
Yes
No
3. Step size refinement: compute $\lambda = \lambda / 2$
λ reaches the finest step size?
No
Yes
Final parameter **T**

The algorithm consists of the following four technologies.

DRR generation

As illustrated in Figure 1, a ray casting method (Levoy, 1988) generates the DRR I_D. Image intensity $I_D(i,j)$ at point (i, j) on the DRR I_D is computed by accumulating the intensities of the voxels that ray $\mathbf{r}(i,j)$ penetrates, where $\mathbf{r}(i,j)$ represents a ray that penetrates point (i, j) from the rendering source.

Similarity measure

We use Gradient Correlation (GC) for our algorithm according to Penny's experimental study (Penny et al., 1998) on six similarity measures. Although they found that pattern intensity (Weese et al., 1997) and gradient difference were the most robust measure for their registration scenario, these measures require intensity correction (Penny et al., 1998) to minimise the difference between the two images I_F and I_D, because they use a difference image created by subtracting I_D from I_F. In contrast, GC focuses on edge information in the images, so it essentially does not require intensity correction to minimise the difference. Furthermore, they also show that GC provides a small failure rate (5%) for clinical datasets, and it is the most robust measure that assumes no intensity correction. Therefore, we use GC as the similarity measure between the two images I_F and I_D.

$$C(\mathbf{T}) = G(I_F, I_D). \tag{1}$$

Here, GC $G(A, B)$ between images A and B is given by

$$G(A,B) = \frac{1}{2}\left[N\left(\frac{\partial A}{\partial i}, \frac{\partial B}{\partial i}\right) + N\left(\frac{\partial A}{\partial j}, \frac{\partial B}{\partial j}\right)\right], \tag{2}$$

where $\partial A/\partial i$ and $\partial A/\partial j$ ($\partial B/\partial i$ and $\partial B/\partial j$) are the gradient images of A (B, respectively), representing the derivative of the intensity in the horizontal and vertical axes of the image, and $N(A, B)$ is Normalised Cross Correlation (NCC) defined over two images A and B:

$$N(A,B) = \frac{\sum_{i,j}(A(i,j)-\overline{A})(B(i,j)-\overline{B})}{\sqrt{\sum_{i,j}(A(i,j)-\overline{A})^2}\sqrt{\sum_{i,j}(B(i,j)-\overline{B})^2}}, \tag{3}$$

where $\overline{A}$ and $\overline{B}$ are the mean values of the images.

The gradient images are produced by means of the first derivative of a Gaussian. This filter has the advantage that it reduces and smoothes noise in images, improving the robustness of alignment. In summary, the intensity values at point (i,j) on the gradient images $\partial A/\partial i$ and $\partial A/\partial j$ are given by convolution with the first derivative Gaussian filters $F_I(i,j)$ and $F_J(i,j)$:

$$\frac{\partial A(i,j)}{\partial i} = F_I(i,j) \times A(i,j),$$
$$\frac{\partial A(i,j)}{\partial j} = F_J(i,j) \times A(i,j),$$

where

$$F_I(i,j) = \frac{-i}{2\pi\sigma^4} e^{-(i^2+j^2)/2\sigma^2},$$

$$F_J(i,j) = \frac{-j}{2\pi\sigma^4} e^{-(i^2+j^2)/2\sigma^2},$$

and σ is the standard deviation of the distribution and is proportional to the kernel size, namely the size of the neighbourhood on which the filter operates. The remaining gradient images $\partial B/\partial i$ and $\partial B/\partial j$ also can be generated in the same manner.

Summarising the above description, the algorithm mainly consists of three computation phases:

- DRR generation
- gradient image generation
- NCC computation.

Optimisation

In order to find the optimal transformation parameter **T** that maximises the cost function *C*, the algorithm employs the steepest descent optimisation technique (Press et al., 1988) during registration process:

$$\mathbf{T} = \mathbf{T} + \lambda \frac{\partial C}{\partial \mathbf{T}}. \tag{4}$$

This optimisation stops if a local optimum has been found. The gradient $\partial C/\partial \mathbf{T}$ of the cost function is estimated by using the finite-difference approximation (Press et al., 1998). As we mentioned earlier, this optimisation is performed from coarse to fine resolution by decreasing the step size λ. Because the transformation **T** consists of six independent parameters, the computation phases (a)–(c) are repeated 13 times to approximate the gradient $\partial C/\partial \mathbf{T}$ at each optimisation step: one repetition for current transformation **T** and 12 repetitions for finite differences $\mathbf{T} \pm \lambda\Delta$ of each parameter.

Biplane images and ROI

Generally, a single projected image is not sufficient for accurate registration in the 3-D space, because it essentially is not sensitive enough to estimate the precise position in the depth direction. A straightforward solution to this problem is to use biplane images (Li et al., 1994). To do this, the algorithm optimises the sum of two GCs, each computed from one of the pairs of biplane images and DRRs.

Furthermore, the algorithm supports ROI specification to minimise computation time and improve registration accuracy (Weese et al., 1997). As illustrated in Figure 1, the ROI must be specified such that it includes the anatomy to be aligned. Given such a ROI, the algorithm is allowed to process only inside the ROI, minimising the amount of computation. This ROI specification can be done automatically and quickly by an intensity parser (Lorenz et al., 1997).

3 Parallelising 2-D/3-D registration

In this section we present the design and implementation of our parallel method. We then show a theoretical performance analysis of our method.

3.1 Design aspects

To accelerate the registration process, we can exploit three parallelisms as follows.

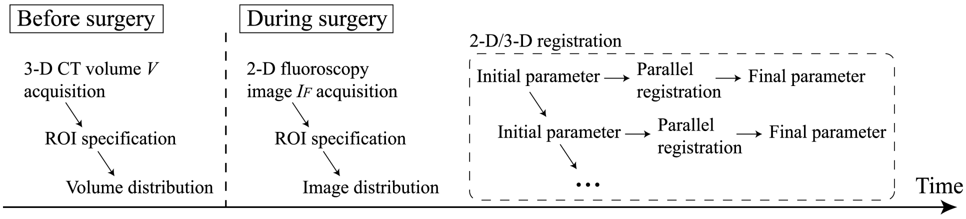
- *Speculative parallelism.* In the registration algorithm, speculative parallelism can be exploited by simultaneously processing the same registration task with different initial parameters. This is important to prevent unsuccessful registrations (due to local optimums), because the surgery cannot progress until the alignment has been correctly achieved. Otherwise, the surgery must be performed without the surgical plan. To prevent such undesirable situations, an appropriate transformation must be given as the initial parameter **T**. However, in general, initial parameters are experimentally determined according to the surgeon's experience. Therefore, speculative processing contributes to improvement in the robustness of our method.
- *Data parallelism*. Exploiting this parallelism accelerates a single registration task. It can easily be established by using image parallelism (Molnar et al., 1994), where processors take the responsibility for each subtask associated with a small part of the 2-D image. The details of this workload distribution are presented later in Section 3.2.
- *Task parallelism.* This parallelism also contributes to the acceleration of a single registration task. It exists in the finite-difference approximation, where the computation phases (a)–(c) are repeated 13 times. However, this means that the speedup derived by this parallelism is limited by a small factor of 13. Furthermore, load balancing is probably not easy if it is exploited, because 13 cannot divide the number of processors, usually chosen to be a power of two.

From the above discussion, we have decided to exploit speculative parallelism and data parallelism. Exploiting these parallelisms then raises another question to be answered: given P processors, how many processors should be used to exploit each parallelism? The idea for this issue is to estimate an appropriate number of processors for data-parallel processing and then assign the remaining processors to speculative processing. The appropriate number is determined by the speedup estimated by the theoretical analysis presented later in Section 3.4.

In addition to the computation phases (a)–(c), Input/Output (I/O) operations also might become a performance bottleneck after parallelisation. However, I/O issues are not critical in our cluster environment for the following two reasons. Firstly, the largest input data, namely the volume V, are the preoperative data. Therefore, it can be distributed to processors before surgery, allowing us to assume that processors have loaded it into their local memory when registration tasks are submitted. Secondly, the remaining data I_F are small enough to be broadcast rapidly in our cluster. For example, it takes only about 119 ms to broadcast a 1024×1024 pixel image while the succeeding optimisation process takes more than 10s, as presented later. Thus, although the intraoperative image IF needs to be broadcast just before starting the registration process, it is not critical for our well-connected computing environment. Therefore, we assume that all processors have the entire data, V and I_F, in their local memory.

Summarising the design aspects mentioned above, Figure 3 shows a timeline view of a typical surgical procedure using our parallel method.

Figure 3 Timeline view of typical surgical procedure using our parallel method



3.2 Workload distribution

We now show how our method exploits image parallelism. A good solution to this issue balances workload among processors and minimises the amount of messages transmitted between processors and the number of sends and receives. To find such a solution, we first investigate the characteristics of computation phases (a)–(c) with respect to available parallelism, load balancing, and data access pattern. Table 1 shows these characteristics with a preliminary timing result measured on a single processor machine.

- *DRR generation.* The intensity value at any point (i, j) can independently be computed with the values at other points, because different rays can cast independently. The workload associated with each point is nonuniform due to the different number of penetrated voxels. Points around the DRR edge tend to have less workload. In addition to this image parallelism, we can also use object parallelism (Molnar et al., 1994), where processors take the responsibility for each subtask associated with a small part of the volume and then merge locally rendered DRRs into a final DRR. This object-parallel scheme allows processors to load only a small portion of the volume, but it requires communication to generate the final DRR. As mentioned earlier, we assume that all processors have the entire volume, so that our method uses an image-parallel scheme to prevent communication in this most time-consuming phase.
- *Gradient image generation.* As in DRR generation, different points can independently be processed to obtain their intensities on the gradient image. The convolution for point (i, j) requires all intensities $A(i + \alpha, j + \beta)$ such that $-\lfloor K/2 \rfloor \leq \alpha, \beta \leq \lfloor K/2 \rfloor$, where K denotes the kernel size of the filter. Note here that this means any point on the gradient image requires DRR generation of its surrounding $K \times K$ neighbourhood, because the gradient images are generated from the DRR I_D as well as the image I_F. With regard to load balancing, this computation phase has uniform workload, because the same kernel size K is used for every point. Note also that the kernel size K is usually a relatively large number, which increases the amount of messages under an inappropriate workload distribution scheme. For example, we use $K = 19$ pixels for typical 2-D ROI sizes ranging from 200 × 200 to 400 × 400 pixels.

- *NCC computation*. NCC computation can be approached as a reduction problem, because equation (3) can be rewritten as

$$N(A,B) = \frac{\sum_{i,j}(A(i,j)B(i,j) - \overline{A}\overline{B})}{\sqrt{\sum_{i,j}(A(i,j)^2 - \overline{A}^2)}\sqrt{\sum_{i,j}(B(i,j)^2 - \overline{B}^2)}}. \tag{5}$$

This equation indicates that NCC can be computed from six local sums: the local sums of the number of points; intensities $\sum A(i,j); \sum B(i,j)$; squared intensities $\sum A(i,j)^2; \sum B(i,j)^2$; and multiplied intensities $\sum A(i,j)B(i,j)$. These sums can independently be computed if processors are responsible for the same point (i, j) on images A and B. The workload is perfectly balanced if the same number of points is assigned to each processor. However, communication is required to reduce local sums into a global sum.

Table 1 Summary of computation phases with respect to (1) available parallelism; (2) workload associated with each point on the 2-D image; (3) data access required for each point and (4) sequential time measured using a spine dataset on a single node of our cluster

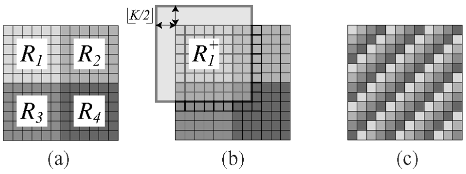
Computation phase	*Parallelism*	*Workload*	*Data required for intensity* $A(i,j)$	*Time* (*s*)
(a) DRR generation	Image/object*	Nonuniform	Penetrated voxels	993.7
(b) Gradient image generation	Image	Uniform	Surrounding $K \times K$ neighbourhood intensities	67.2
(c) NCC computation	Image**	Uniform	Corresponding intensity $B(i,j)$	2.5

*, **Communication is required to produce the final DRR and to perform reduction operations, respectively.

According to the analysis mentioned above, our method employs a 2-D block distribution scheme with the overlap region, as shown in Figure 4. Here, the overlap size is given by the kernel size *K*, allowing processors to produce gradient images without any communication. As compared with other distribution schemes such as 1-D/2-D disjoint block and cyclic schemes, our scheme has the following advantage/disadvantages:

- the advantage of less communication, achieved by the overlap region
- the disadvantage of more computation, due to the redundant DRR generation for the overlap region
- the disadvantage of imbalanced workload, as compared with the cyclic scheme.

Figure 4 Workload distribution: (a) 2-D disjoint block; (b) 2-D block with the overlap region and (c) cyclic distribution schemes. Our method employs (b)



If the overlap region is not given, communication is required for block boundaries in order to obtain intensities of neighbour points computed by other processors. This communication becomes a significant performance bottleneck in the case where many processors are responsible for the neighbour points. In this case, processors need to gather the intensities from many processors and also have to scatter their own intensities to others, but it is not easy to realise both fast scatter and gather operations at the same time. Due to this complex communication pattern, the cyclic scheme possibly results in poor performance.

Furthermore, the 1-D/2-D block scheme without the overlap will also suffer from this situation as the number of processors P increases, because the kernel size K is relatively large compared to the block size, which decreases as P increases.

Note here that an appropriate distribution scheme depends on the target environment such as the hardware configuration and the input data. For example, our scheme may fail to provide fast registration on slow processors connected with low-latency network, and the block scheme without any overlap may yield higher performance if the kernel size K is relatively small compared to the block size. Therefore, the above quantitative analysis is important to find the best distribution scheme for the target environment.

3.3 Proposed method

We denote by $\mathcal{R} = \{(i, j) \mid 1 \leq i \leq S_I,\ 1 \leq j \leq S_J\}$ the domain of the ROI specified on the 2-D image, where S_I and S_J are the horizontal and vertical sizes of the ROI, respectively. Let $\mathcal{R}_p$, where $1 \leq p \leq P$, be the pth subdomain partitioned by the 2-D disjoint block scheme such that $\mathcal{R} = \bigcup_{p=1}^{P} \mathcal{R}_p$ and $\mathcal{R}_p \bigcap \mathcal{R}_q = \emptyset$, for all $1 \leq p \leq q \leq P$. Let $\mathcal{R}_p^+$ be the p-th subdomain with its overlap region.

Given P processors, our parallel method aligns t he volume V to the image I_F as follows.

1 *Data load.* For all $1 \leq p \leq P$, processor p loads the volume V from its local disk into main memory and waits for registration tasks to be submitted. Then, processor 1 serves as a gateway receiving a registration task with its input data: the projected image I_F, the initial parameter $\mathbf{T}$, and the initial step size λ. After this, the gateway broadcasts these input data to all processors.

2 *DRR generation.* For all $1 \leq p \leq P$, processor p locally generates a DRR for subdomain $\mathcal{R}_p^+$.

3 *Gradient image generation.* For all $1 \leq p \leq P$, processor p locally generates the gradient images for disjoint subdomain $\mathcal{R}_p$.

4 *NCC computation.* For all $1 \leq p \leq P$, processor p locally computes six local sums from subdomain $\mathcal{R}_p$. Then, every processor participates in a reduction communication to combine local sums from all processors and distribute the global sum back to all processors. After this communication, every processor has six global sums, so that locally computes NCC.

5 *Optimisation.* Repeat 2–4 13 times to update the parameter $\mathbf{T}$ by using the steepest descent optimisation. Repeat this step until a local optimum has been found.

3.4 Theoretical performance analysis

As we mentioned earlier, the objective of this theoretical analysis is to estimate the balancing point between speculative and data-parallel processing with respect to the number of processors. To do this, we estimate the speedup $U = T_{SEQ}/T_{PAR}$ on P processors, where T_{SEQ} and T_{PAR} denote the sequential and parallel registration time, respectively. Then, we consider $\lceil U \rceil$ processors to be an appropriate number for data-parallel processing, because the efficiency U/P usually results in a lower value when using more than U processors for data-parallel processing. This makes the remaining $\max(P - \lceil U \rceil, 0)$ processors participate in speculative processing.

To make the analysis easier, we focus on the body of the time-consuming loop, namely the three computation phases (a)–(c). The remaining phases such as the data load phase are disregarded in the analysis, because they can be processed in rapid time, as compared with the three phases. We also analyse a single iteration of the loop, because the iterative nature of our method allows us to substitute the speedup for a single iteration for the whole iteration. Furthermore, we assume that the workload of DRR generation is balanced between processors.

Table 2 shows the notations used in the analysis. Let t_1, t_2 and t_3 be the sequential time for DRR generation, gradient image generation and NCC computation at an optimisation step, respectively. Then, a sequential step takes

$$T_{SEQ} = t_1 + t_2 + t_3. \qquad (6)$$

Table 2 Notations used in the theoretical performance analysis

Symbol	*Description*
S_I and S_J	Horizontal and vertical size of the ROI
K	Kernel size of the filter
P	Number of processors
P_I and P_J	Number of processors in the horizontal and vertical direction such that $P = P_I P_J$
t_1, t_2 and t_3	Sequential times for DRR generation, gradient image generation, and NCC computation, respectively
L	Communication latency between nodes

Let P_I and P_J be the number of processors in the horizontal and vertical directions of the image, respectively, such that $P = P_I P_J$. In the DRR generation phase, each processor takes the responsibility for a subdomain $\mathcal{R}_p^+$ of size $(\lceil S_I/P_I \rceil + K - 1) \times (\lceil S_J/P_J \rceil + K - 1)$, while the entire domain for this phase contains $(S_I + K - 1) \times (S_J + K - 1)$ pixels, which are sequentially processed in t_1 time. Assuming that the workload is balanced among processors, we can estimate the parallel time for DRR generation. On the other hand, during the local computation of the last two phases, each processor is responsible for a $\lceil S_I/P_I \rceil \times \lceil S_J/P_J \rceil$ portion $\mathcal{R}_p$ of the entire domain $\mathcal{R}$ of size $S_I \times S_J$. In addition to this local computation, NCC computation takes additional time for an all reduce communication (Message Passing Interface Forum, 1994). A tree-structured communication strategy efficiently realises this in $2L \log_2 P$ time, where L represents the communication latency between two nodes in the tree. The final computation using the global sums is rapid enough to ignore.

Summing up each time, a parallel optimisation step takes

$$T_{PAR} = \frac{\left(\lceil S_I / P_I \rceil + K - 1\right) + \left(\lceil S_J / P_J \rceil + K - 1\right)}{(S_I + K - 1)(S_J + K - 1)} t_1 + \frac{\left(\lceil S_I / P_I \rceil \lceil S_J / P_J \rceil\right)}{S_I S_J} (t_2 + t_3) + 2L \log_2 P. \tag{7}$$

Thus, given time and size information on a sequential optimisation step, equations (6) and (7) estimate the speedup *U* so that determine the appropriate number of processors.

4 Experimental results

To evaluate the performance of our parallel method, we have implemented it using the C++ language and Message Passing Interface (MPI) standard (Message Passing Interface Forum, 1994).

4.1 Experimental setup

We used a cluster of 64 PCs, each equipped with two Pentium III 1-GHz processors. The interconnection between nodes is a Myrinet switch (Boden et al., 1995), yielding a bandwidth of 2 Gb/s. Our implementation runs on a Linux operating system with the MPICH-SCore library (O'Carroll et al., 1998), a fast MPI implementation.

We performed registration tasks using datasets of a femur phantom and a real spine (see Table 3). The biplane images are generated as the front (coronal) view and the side (sagittal) view of the body. The kernel size *K* of the Gaussian filter was experimentally determined as $K = 19$ pixels ($\sigma = 3$).

Table 3 Dataset specification

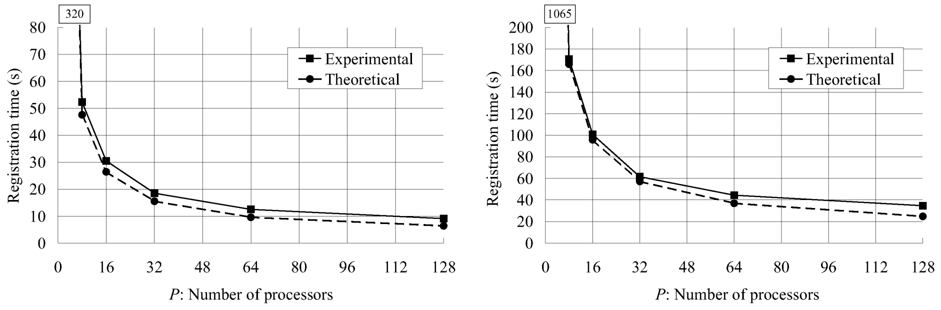
	Femur phantom	*Real spine*
3-D volume size	256 × 256 × 367 voxels	512 × 512 × 204 voxels
File size	45 MB	102 MB
ROI size	53 × 47 × 54 voxels	299 × 299 × 47 voxels
2-D image size	640 × 512 pixels	1024 × 1024 pixels
File size	320 KB	2 MB
ROI size (front)	353 × 276 pixels	340 × 204 pixels
ROI size (side)	344 × 272 pixels	336 × 200 pixels

As presented earlier in Figure 3, we first produced the CT volume and distributed it with its ROI information to each node before running our registration program. This distribution takes 1.7 s and 3.8 s on the Myrinet network for the femur and the spine datasets, respectively. On the other hand, the 2-D fluoroscopy images are produced immediately before registration and then are broadcast by the registration program itself. It takes 37 ms and 119 ms to broadcast each dataset, respectively.

4.2 Timing results

Figure 5 shows experimental and theoretical timing results on different numbers of processors. Here, theoretical values are derived using equation (7) and multiplying the number of iterations required for optimisation. We can see that our implementation, running on $P = 128$, reduces computation time for the spine dataset from 17 m (1065 s) to 35 s. It also achieves a shorter time of 9 s for the femur dataset with a smaller ROI. Times of less than 60 s are compatible with the time constraints required for surgery. Thus, our parallel method enables us to utilise the registration technique during surgery without degrading the quality of alignment.

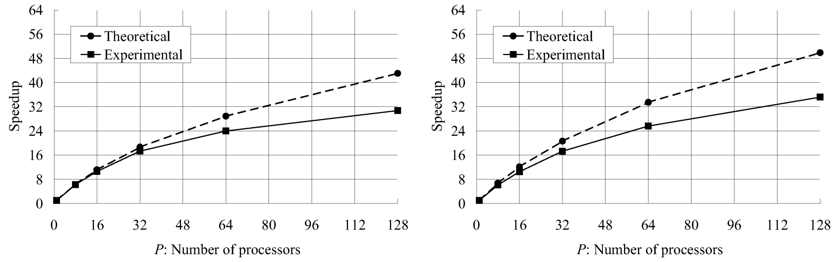
Figure 5 Registration time: (a) for femur phantom and (b) for real spine on different numbers of processors. Sequential registration takes 320 s and 1065 s for each dataset



(a) (b)

Figure 6 shows experimental and theoretical speedups on different numbers of processors. For both datasets the maximum speedup of our implementation reaches about a factor of 32 when $P = 128$. Thus, the speedup does not increase so well when $P \geq 32$. This is due to the kernel size K of the filter, which is relatively large when compared to the size of disjoint blocks. For example, when using 128 processors ($\langle P_i, P_j \rangle = \langle 16, 8 \rangle$) for the femur dataset, the size of disjoint blocks becomes $\lceil S_I/P_I \rceil \times \lceil S_J/P_J \rceil = 23 \times 35$ pixels while that of overlapping blocks becomes $(\lceil S_I/P_I \rceil + K - 1) \times (\lceil S_J/P_J \rceil + K - 1) = 41 \times 53$ pixels. This means that each processor performs about 2.7 times more computation due to redundant DRR generation as compared with under disjoint distribution schemes. Thus, the speedup for a single registration task results in a lower value as P (P_I and P_J) increases under the fixed ROI size S_I and S_J.

Figure 6 Speedup: (a) for femur phantom and (b) for real spine on different numbers of processors



(a) (b)

4.3 Discussion

On workload distribution

If we change our distribution scheme to a 1-D block scheme with overlap, the size of disjoint blocks and that of overlapping blocks on $P = 128$ become 3×276 and 21×294 pixels, respectively. Therefore, this 1-D scheme requires about 7.5 times more computation, resulting in a lower speedup. Moreover, since the vertical length of 1-D blocks becomes shorter than the kernel size K of the filter, processors need to communicate with more processors to obtain intensities of vertical neighbours, having a more complex communication pattern with network contention.

Although our overlapping scheme requires redundant computation for DRR generation, this disadvantage is covered by the advantage of less communication. If a 2-D disjoint block scheme is employed, every processor needs to communicate its boundary data with its vertical/horizontal/diagonal neighbours. Though this can be implemented by repeating shift communication operations, these operations could be a performance bottleneck. For example, when using 128 processors for the femur dataset, this scheme causes 2.7 KB ($41 \times 53 - 23 \times 35$ pixels, each in 2 bytes) of incoming data and the same amount of outgoing data at every processor, which must be sequentially processed in eight shift communication operations.

On theoretical analysis

With respect to the accuracy of our analysis, the prediction error ranges from 3% to 30% for both datasets. The error is less than 5% when $P < 32$ but increases with P. Thus, the accuracy decreases as P increases. We guess that this lower accuracy is due to the assumption made in the DRR generation phase. That is, though our analysis assumes uniform workload in this phase, it becomes more imbalanced as P increases. Actually, the error in this phase dominates the entire error.

Although the maximum prediction error of 30% seems a high value, our analysis provides sufficient information to determine the number of processors that should be engaged with speculative processing. That is, it estimates the speedup on 128 processors to be about 48, which suggests that at most 48 processors should be used for data-parallel processing. In this case, the registration task can be started simultaneously with three or four different initial parameters in order to prevent unsuccessful alignments with a small performance loss.

Thus, the prediction accuracy could be improved by another more detailed analysis; for example, by taking account of the workload in the DRR generation phase. However, we think that the appropriate number of processors can be estimated well by our analysis, which requires only time and size information on a single sequential optimisation step.

On speculative processing

In our experiments, we found that the speedup was limited by a relatively smaller value, as compared with P. In this situation, where the speedup is theoretically limited by a small value, using more processors for data-parallel processing results in a lower utilisation of computing nodes. To deal with this, our method tries to raise the speedup by means of speculative processing. This strategy will lead to a higher speedup if optimisation is repeatedly processed with different initial parameters due to unsuccessful alignments.

Another important motivation to exploit speculative parallelism comes from the fact that the registration algorithm sometimes fails to align objects due to local optimums. Therefore, our strategy will also improve the confidence of registration technique, providing more robust alignment for a wide variety of clinical scenarios.

5 Related work

There are many papers reporting experiences in using High Performance Computing (HPC) resources to realise intraoperative assistances based on compute-intensive applications. To the best of our knowledge, such applications include rigid/nonrigid image registration (Ino et al., 2005; Rohlfing and Maurer, 2003; Warfield et al., 1998, 2000), biomechanical simulation (Kawasaki et al., 2004), and image visualisation (Liao et al., 2003). All of these applications exploit data parallelism in the 2-D image space or the 3-D object space. In contrast, the key contribution of this work is the development of a 2-D/3-D rigid registration method that exploits both data parallelism and speculative parallelism. This combination is applicable to many optimisation-based registration algorithms, accelerating registration tasks with more robustness.

Recently, computational Grids are also emerging as an attractive HPC platform. Hastings et al. (2003) show a toolkit that allows rapid and efficient development of biomedical image analysis applications in a distributed environment. Their toolkit exploits data and coarse grain task parallelism in a chain of processing operations that begins with data acquisition and ends with data visualisation. Stefanescu et al. (2004) present a grid service that accelerates nonrigid registration tasks by exploiting image parallelism.

Although clusters generally have more tightly-coupled computing nodes than Grids, we believe that our method, which requires less communication but more computation, also could provide a fast registration service on Grids.

6 Conclusions and future work

We have presented a parallel method for 2-D/3-D registration, aiming at realising intra-operative alignment. Our method exploits data and speculative parallelism in an intensity-based algorithm, so that we can perform fast, accurate, and robust registration during surgery. Our implementation on a cluster of 64 PCs aligns a 299 × 299 × 47 voxel volume to 340 × 204 pixel images in a few tens of seconds, a clinically compatible time.

In the future, our parallel implementation could be improved by exploiting task parallelism in order to achieve further acceleration. Although we currently avoid this to have better load balancing, exploiting this parallelism certainly accelerates a registration task. We are also planning to integrate our implementation into a Grid-enabled environment in order to provide a fast, accurate, and robust 2-D/3-D registration service through the Internet to hospitals.

Acknowledgements

This work was partly supported by JSPS Grant-in-Aid for Scientific Research on Priority Areas (16016254 and 17032007) by the Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

Boden, N.J., Cohen, D., Felderman, R.E., Kulawik, A.E., Seitz, C.L., Seizovic, J.N. and Su, W-K. (1995) 'Myrinet: a gigabit-per-second local area network', *IEEE Micro*, Vol. 15, No. 1, pp.29–36.

Feldmar, J., Ayache, N. and Betting, F. (1997) '3D-2D projective registration of free-form curves and surfaces', *Computer Vision and Image Understanding*, Vol. 65, No. 3, pp.403–124.

Gueziec, A., Kazanzides, P., Williamson, B. and Taylor, R.H. (1998) 'Anatomy-based registration of CT-scan and intraoperative X-ray images for guiding a surgical robot', *IEEE Trans. Medical Imaging*, Vol. 17, No. 5, pp.715–728.

Hajnal, J.V., Hill, D.L. and Hawkes, D.J. (Eds.) (2001) *Medical Image Registration*, CRC Press, Boca Raton, FL.

Hastings, S., Kurc, T., Langella, S., Catalyurek, U., Pan, T. and Saltz, J. (2003) 'Image processing for the grid: a toolkit for building grid-enabled image processing applications', *Proc. 3rd IEEE/ACMInt. Symp. Cluster Computing and the Grid (CCGrid'03)*, Tokyo, Japan, pp.36–13.

Herring, J.L., Dawant, B.M., Maurer, C.R., Muratore, D.M., Galloway, R.L. and Fitzpatrick, J.M. (1998) 'Surface-based registration of CT images to physical space for image-guided surgery of the spine: a sensitivity study', *IEEE Trans. Medical Imaging*, Vol. 17, No. 5, pp.743–752.

Ino, F., Ooyama, K. and Hagihara, K. (2005) 'A data distributed parallel algorithm for nonrigid image registration', *Parallel Computing*, Vol. 31, No. 1, pp.19–43.

Joskowicz, L., Milgrom, C., Simkin, A., Tockus, L. and Yaniv, Z. (1998) 'FRACAS: a system for computer-aided image-guided long bone fracture surgery', *Computer Aided Surgery*, Vol. 3, No. 6, pp.271–288.

Kawasaki, Y., Ino, F., Mizutani, Y., Fujimoto, N., Sasama, T., Sato, Y., Sugano, N., Tamura, S. and Hagihara, K. (2004) 'High-performance computing service over the Internet for intraoperative image processing', *IEEE Trans. Information and Technology in Biomedicine*, Vol. 8, No. 1, pp.36–46

Lavallée, S. and Szeliski, R. (1995) 'Recovering the position and orientation of free-form objects from image contours using 3D distance maps', *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 17, No. 4, pp.378–390.

Lemieux, L., Jagoe, R., Fish, D.R., Kitchen, N.D. and Thomas, D.G.T. (1994) 'A patient-to- computed-tomography image registration method based on digitally reconstructed radiographs', *Medical Physics*, Vol. 21, No. 11, pp.1749–1760.

Levoy, M. (1988) 'Display of surfaces from volume data', *IEEE Computer Graphics and Applications*, Vol. 8, No. 3, pp.29–37.

Li, S., Pelizzari, C.A. and Chen, G.T.Y. (1994) 'Unfolding patient motion with biplane radio graphs', *Medical Physics*, Vol. 21, No. 9, pp.1369–1512.

Liao, H., Hata, N., Iwahara, M., Sakuma, I. and Dohi, T. (2003) 'An autostereoscopic display system for image-guided surgery using high-quality integral videography with high performance computing', *Proc. 6th Int. Conf. Medical Image Computing and Computer-Assisted Intervention (MICCAI'03), PartII*, Montréal, Canada, pp.247–255.

Lorenz, C., Buzug, T.M., Fassnacht, C. and Weese, J. (1997) 'Automated detection and segmentation of lumbar vertebrae in CT and CTA image based on a grey-value profile parser', *Proc. Computer Assisted Radiology and Surgery: 11th Int. Congress and Exhibition (CARS '97)*, Berlin, Germany, pp.209–214.

Maurer, C.R., Maciunas, R.J. and Fitzpatrick, J.M. (1998) 'Registration of head CT images to physical space using a weighted combination of points and surfaces', *IEEE Trans. Medical Imaging*, Vol. 17, No. 5, pp.753–761.

McLaughlin, R.A., Hipwell, J., Hawkes, D.J., Noble, J.A., Bryne, J.V. and Cox, T. (2002) 'A comparison of 2D-3D intensity-based registration and feature-based registration for neurointer-ventions', *Proc. 5th Int. Conf. Medical Image Computing and Computer-Assisted Intervention (MICCAI'02), Part II*, Tokyo, Japan, pp.517–524.

Message Passing Interface Forum (1994) 'MPI: a message-passing interface standard', *Int. J. Supercomputer Applications and High Performance Computing*, Vol. 8, Nos. 3–4, pp.159–416.

Molnar, S., Cox, M., Ellsworth, D. and Fuchs, H. (1994) 'A sorting classification of parallel rendering', *IEEE Computer Graphics and Applications*, Vol. 14, No. 4, pp.23–32.

O'Carroll, F., Tezuka, H., Hori, A. and Ishikawa, Y. (1998) 'The design and implementation of zero copy MPI using commodity hardware with a high performance network', *Proc. 12th ACM Int. Conf Supercomputing (ICS'98)*, Melbourne, Australia, pp.243–250.

Penny, G.P., Weese, J., Little, J.A., Desmedt, P., Hill, D.L.G. and Hawkes, D.J. (1998) 'A comparison of similarity measures for use in 2-D-3-D medical image registration', *IEEE Trans. Medical Imaging*, Vol. 17, No. 4, pp.586–595.

Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P. (1988) *NUMERICAL RECIPES in C: The Art of Scientific Computing*, Cambridge University Press, Cambridge, UK.

Rohlfing, T. and Maurer, C.R. (2003) 'Nonrigid image registration in shared-memory multi processor environments with application to brains, breasts, and bees', *IEEE Trans. Information Technology in Biomedicine*, Vol. 7, No. 1, pp.16–25.

Stefanescu, R., Pennec, X. and Ayache, N. (2004) 'Grid powered nonlinear image registration with locally adaptive regularization', *Medical Image Analysis*, Vol. 8, No. 3, pp.325–342.

Warfield, S.K., Ferrant, M., Gallez, X., Nabavi, A., Jolesz, F.A. and Kikinis, R. (2000) 'Real-time biomechanical simulation of volumetric brain deformation for image guided neurosurgery', *Proc. High Performance Networking and Computing Conf. (SC2000)*, CD-ROM, Dallas, TX, USA, p.16.

Warfield, S.K., Jolesz, F.A. and Kikinis, R. (1998) 'A high performance computing approach to the registration of medical imaging data', *Parallel Computing*, Vol. 24, Nos. 9–10, pp.1345–1368.

Weese, J., Penney, G.P., Desmedt, P., Buzug, T.M., Hill, D.L.G. and Hawkes, D.J. (1997) ' Voxel-based 2-D/3-D registration of fluoroscopy images and CT scans for image-guided surgery', *IEEE Trans. Information Technology in Biomedicine*, Vol. 1, No. 4, pp.284–293.

West, J., Fitzpatrick, J.M., Wang, M.Y., Dawant, B.M., Maurer, C.R., Kessler, R.M. and Maciunas, R.J. (1999) 'Retrospective intermodality registration techniques for images of the head: Surface-based versus volume-based', *IEEE Trans. Medical Imaging*, Vol. 18, No. 2, pp.144–150.

Zollei, L., Grimson, E., Norbash, A. and Wells, W. (2001) '2D-3D rigid registration of X-ray fluoroscopy and CT images using mutual information and sparsely sampled histogram estimators', *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR '01)*, Vol. 2, Kauai, HI, USA, pp.696–703.