A parallel algorithm for medical images registration based on B-splines

Zhiyong Zhou
Changchun Institute of Optics, Fine Mechanics and Physics,
Chinese Academy of Sciences
Changchun China

Tao Zhang Duojie Kuai
Suzhou institute of biomedical engineer and technology,
Chinese Academy of Sciences
Suzhou China

***Abstract*-Cubic B-splines is widely applied in non-rigid registration because of its approximation performance and fast computational characteristics. However, a small scale non-rigid deformation is needed to characterize by a large number of control points. Moreover, an iterative optimization strategy of the non-rigid registration algorithm and the normalized mutual information (NMI) cost a great quantity calculation. So, the process of the non-rigid registration is slowed by calculations of NMI in a iterative optimization strategy. In this paper, a parallel optimization algorithm based on cubic B-splines functions is proposed to parallelize the optimization algorithm of the non-rigid registration and the calculations of normalize mutual information. In practice, a fast algorithm of cubic B-splines is used and the control points are only distributed on the targets. Experiments show that the use of the fast algorithm and the parallel optimization strategy improves the non-rigid registration process of medical images.**

Keyword-Parallel algorithm; mutual information; gradient descent flow

I. Review

Free-form deformation mode based on B-splines, which has excellent approximation [1]8 and fast calculation properties [2], has been widely used because of its good registration result for non-rigid deformation with different scales. However, a large quantity of mark points is needed to characterize some small non-rigid deformation in organs or lesions. In the non-rigid registration processing, the normalized mutual information (NMI) and the coefficients of B-splines, both of them cost a lot of floating-point calculations, should be called frequently in an iterative algorithm. Moreover, medical images and a gray-scale joint histogram matrix, which always is a large matrix, needed to traverse when NMI is calculated. In practice, the NMI always occupies more than 80 percent of the whole registration time. So, a large number of floating-point calculations slow the process of the registration. Sometimes, NMI of improved form [3,4] or NMI with some spatial information[5] and other mathematical algorithms [3,6,7] could be used in order to improve the result of the registration.

Non-rigid registration algorithms are only concerned on the robustness, accuracy and precision while ignoring the importance of the computation time, what leads the algorithms and software not using in clinical.

With the development of the parallel computing technology, parallel computing technology has been used widely in the medical processing field. Parallel algorithms are very suitable for medical images non-rigid registration because of its parallelism and high calculation. Some parallel algorithms have been used in medical images non-rigid registration, and have been achieved quite good results [8-14]. What's more, there are scholars having obtained some meaningful conclusions [15] by analyzing different memory types and different parallel algorithm framework. Nowadays, some parallel algorithms with shared memory model have been ported to graphic processing unit (GPU), achieving excellent accelerating effect [15,16] with low-cost.

The main processing of the medical images non-rigid registration are NMI calculation, space transformation, image interpolation and so on. All of them have some characteristics: large floating-point calculation, large data and high parallelism, which are fit to speed up by a fine- grained data parallel algorithm. Furthermore, every mark point could only affect its neighborhood vector field in 4×4 neighborhood mark points according to B-splines is defined in a finite interval. So, a coarse-grained task parallel algorithm could be used to compute a gradient descent flow and displacements of all mark points.

II. Fast computing coefficients of B-splines

Multilevel B-splines functions [2] are used as basic functions for interpolation in free-form deformation mode [17]. A 3D deformation field is defined by mark points on the Cartesian coordinate system,

$$\mathbf{T}(x,y,z)=\sum_{l,m,n=0}^{3} B_l(u)B_m(v)B_n(w)\phi_{i+l,j+m,k+n} \quad (1)$$

Where, i,j,k are the indices of the mark points on x, y, z directions, u,v,w are relative position in its mark point grid on in x, y, z directions.

$$i=\left\lfloor \frac{x}{\delta_x} \right\rfloor -1, j=\left\lfloor \frac{y}{\delta_y} \right\rfloor -1, k=\left\lfloor \frac{z}{\delta_z} \right\rfloor -1 \quad (2)$$

$$u=\frac{x}{\delta_x}-(i+1), v=\frac{y}{\delta_y}-(j+1), w=\frac{z}{\delta_z}-(k+1) \quad (3)$$

Where, $\delta_x,\delta_y,\delta_z$ are mark point spacing on x, y, z directions.

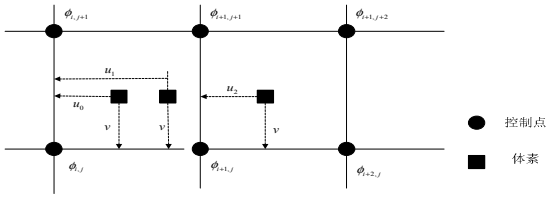

Fig 1 fast computing the coefficients of B-splines

In a 3D medical image, the B-splines coefficients i,u and $B_0(u),...,B_3(u)$ of the voxels on the same column could be pre-computed. Similarly, the B-splines coefficient j,v and $B_0(v),...,B_3(v)$ of the voxels on the same rows could be pre-computed. Therefore, the B-splines coefficients k,w and $B_0(w),...,B_3(w)$ also could be pre-computed. Fig 1 illustrates the underlying principles in the simplified 2-D case. In the same mark point grid cell $\phi_{i,j,k}$, the indices i,j,k and coefficients v,w are constants, only u is a variable. So, $B_m(v),B_n(w)$ and their effect to $\phi_{i+l,j+m,k+n}$ are also constants. (1) can be rewritten as

$$\mathbf{T}(x,y,z)=\sum_{l=0}^{3}B_l(u)\hat{\phi}_{i+l} \tag{4}$$

Where

$$\hat{\phi}_{i+l}=\sum_{m=0}^{3}\sum_{n=0}^{3}B_m(v)B_n(w)\phi_{i+l,j+m,k+n} \tag{5}$$

Assuming there are $P_x\times P_y\times P_z$ voxels and $N_x\times N_y\times N_z$ mark points in a 3D medical image. The computation before optimization is $O(P_xP_yP_z)$ while $O(P_xN_yN_z)$ after optimization. In practice, this algorithm could reduce about 50 percent time for computing the B-splines coefficients.

III. Deformation field optimization

In medical images, only a interesting region should be deformed, so the mark points could be distribution on a target region, not needed on the whole image. The background image dose not need any mark point and any processing. The mark points on a target region are called active point, which are needed to move, ones on background regions are called passivity points, which is fixed in registration process. The number of mark points needed to processing is substantially reduced by defining active points and passivity points.

A mark points set, $\Phi=\{\phi|\phi_{i,j,k},1\le i\le N_x,1\le i\le N_y,1\le i\le N_z\}$, $H^D_{i,j,k}$ is a local information entropy of sub-region $D_{i,j,k}$ in a fixed image and a moving image. $H^D_{i,j,k}$ is defined as

$$H^D_{i,j,k}=\frac{1}{\left|D_{i,j,k}\right|}\sum_{X\in D_{i,j,k}}p(X)\log p(X) \tag{6}$$

Where, X is a voxel in $D_{i,j,k}$, and $p(X)$ is a gray probability density.

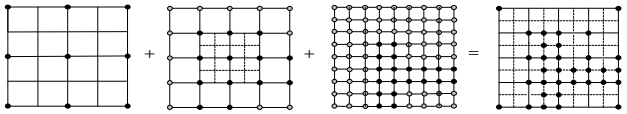

Fig 2 deformation field optimization, black are active points, gray are passivity points.

Because B-splines functions are defined on a finite interval, the support zone of the mark point $\phi_{i,j,k}$ is $D_{i,j,k}$, which also is a sub-region in the fixed image. The defined interval of the mark point $\phi_{i,j,k}$ is also its influence area.

τ is defined as entropy threshold, $H^D_{i,j,k}$ is the local entropy of the support region $D_{i,j,k}$ depending on the mark point $\phi_{i,j,k}$, the maximum value of local entropy is $H_{\max}$. The status of the mark point $\xi(\phi_{i,j,k})$ is defined as

$$\xi(\phi_{i,j,k})=\begin{cases}active\ point & H^D_{i,j,k}\ge\tau H_{\max}\\ passivity\ point & H^D_{i,j,k}<\tau H_{\max}\end{cases} \tag{7}$$

IV. Parallel algorithm

The bottlenecks of the non-rigid registration are data exchange between different threads and a access delay of the high-latency memory. The key steps of a good parallel algorithm are reduction communication time between threads and memory, balance between speedup ration and memory cost.

Assuming there are M thread blocks in the GPU, Ω is the region of the medical image. The thread i processes a sub-region $\Omega_i(1\le i\le M)$. Similarly, the fixed image R and the moving image F are separated by the same method, their sub-image is defined as R_i and F_i.

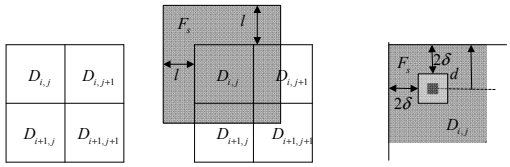

Fig 3 data separated

(x,y,z) is defined as a coordinate of a voxel in sub-region Ω_i. If (x,y,z) could meet the following condition

$$(x,y,z)\in\{(x,y,z)|(x,y,z)\in R_i\cap\mathbf{T}(x,y,z)\in F_i\} \tag{8}$$

The intensity probability density of voxels in sub-region can be paralleled computed.

$\omega_{i,j,k}$ is defined as B-splines function neighborhoods $4\delta_x\times4\delta_y\times4\delta_z$ of the mark point $\phi_{i,j,k}$. If $\omega_{i,j,k}$ could meet the following condition

$$\omega_{i,j,k}\in\{\omega_{i,j,k}\,|\,\omega_{i,j,k}\in R_i\cap\mathbf{T}(\omega_{i,j,k})\in F_i\} \tag{9}$$

The gradient descent $\partial C_{NMI}/\partial\phi_{i,j,k}$ can be paralleled computed.

A. Data separated and data parallel

Fig 3 illustrates a data separated method for paralleled computing NMI and gradient descent $\partial C_{NMI}/\partial\phi_{i,j,k}$ at the same time. v and l, whose units are millimeter, denote the maximum deformation and edge width in the moving image. The relationship of them is

$$l = d + 2\delta \tag{10}$$

Where, d is the maximum displacement length, $l \geq d$, and $\delta = \{\delta_x, \delta_y, \delta_z\}$.

Because $l = d + 2\delta$, the mark point $\phi_{i,j,k}$ meets both of (7) and (8).

$R(x, y, z)$ is the intensity of fixed image R, whose gray scales are M. $F(\mathbf{T}(x, y, z))$ is the moving image after space transformation, whose gray scales are N. Their joint gray probability density is $P_{R,F}$, which is defined as following

$$P_{R,F} = \begin{bmatrix} p(0,0) & p(0,1) & \dots & p(0,N-1) \\ p(1,0) & p(1,1) & \dots & p(1,N-1) \\ \dots & \dots & \dots & \dots \\ p(M,0) & p(M,1) & \dots & p(M-1,N-1) \end{bmatrix} \tag{11}$$

Normalized information mutual is

$$NMI(R,F) = \frac{H(F) + H(R)}{H(R,F)} \tag{12}$$

Where

$$\begin{aligned} H(R) &= \sum_{r \in R} -P_R(r) \log_2 P_R(r) \\ H(R,F) &= \sum_{r \in R} \sum_{f \in F} -P_{R,F}(r,f) \log_2 (r,f) \end{aligned} \tag{13}$$

Where, P_R, P_F are gray probability density of the fixed image and the moving image.

The global normalized information mutual is computed by a recursive binary tree reduction algorithm after all threads having computed a local gray probability density of a sub-image. The first step, every odd computing node sends its result $-P_R(r) \log_2 P_R(r)$ to the even computing node before it. So, both of the odd computing node and the even one before it have the same data. Then, merge the neighborhoods computing nodes, and repeat above steps until all data has been send to the first computing node, which store the global normalized information mutual. At last, the first node broadcast the result to every computing node.

B. Parallel computing

The mark points distribute uniformly in traditional algorithms. However, in the algorithm above description, the mark points distribute unevenly because of the deformation optimization. This cause a problem that the workload of each thread block is different. To solve the important problem, an algorithm, just like Greedy algorithm, is used for balance the work load in every thread block.

Φ_i^+ is defined as a active mark point set in the sub-region D_i. $\| \Phi_i^+ \|$ is a norm of Φ_i^+ denoting the number of active mark points in the sub-region D_i. N is the number of the thread blocks will be used. W is defined as a average number of active mark point in the sub-region D_i ,that $W = \sum_{i=1}^{M} \| \Phi_i^+ \| \Big/ N$. The task parallel algorithm is decrypting as following.

The thread blocks are speared into N groups, $G_i, 1 \leq i \leq N$, M groups being high workload groups, $G_m, 1 \leq m \leq M$,which defined as

$$g(\| \Phi_i^+ \|) > W \tag{14}$$

① Construct list L and push the $N - M$ low workload thread blocks into the list with ascending order according to the number of active mark points, $\left\| \Phi_i^+ \right\|$.

② Select the block G_i whose $W(G_i)$ is maximum. Here, $W(G_i) = \sum_{g_j \in G_i} \| \Phi_i^+ \| \Big/ G_i$, denoting the number of active mark points in G_i. Push g_j in the list head into the thread block groups and then delete it from the list. Repeat the operation until the list is empty.

Through above operation, the M block groups $G_i = \{g_{i,1}, g_{,2}, \dots g_{i,\|G_i\|}\}$ contain one high workload thread block $g_{i,1}$ and several low workload blocks $g_{i,2}, \dots, g_{i,\|G_i\|}$, here $1 \leq i \leq M$.

③ High workload distribute a sub-image $R_{i,1}, F_{i,1}$ and Φ_i^+ to low workload block $\{g_{i,2}, \dots, g_{i,\|\|G_i\|}\}$.

④ Decompose the active mark points set into $\lambda_{i,1}, \lambda_{i,2}, \dots \lambda_{i,\|G_i\|}$ satisfying the condition $\| \lambda_{i,1} \| = W(G_i)$ and $\| \lambda_{i,u} \cup \Phi_{i,u}^+ \| = W(G_i), 2 \leq u \leq \| G_i \|$ for each thread block.

⑤ $g_{i,1}$ and $g_{i,u}, (2 \leq u \leq \| G_i \|)$ calculate $\lambda_{i,1}$ and $\lambda_{i,u} \cup \Phi_{i,\|G_i\|}^+$ respectively.

⑥ $g_{i,1}$ in G_i collect the final result of $g_{i,u}$, here $2 \leq u \leq \| G_i \|$. At last, $g_{i,1}$ broadcast it to other thread blocks $G_j, i \neq j$.

Ⅴ. Result

The spatial resolution of the CT image in this paper is $0.621 \times 0.621 mm$, the image size is 512×512. The spatial resolution of the MR image is $1 \times 1 mm$,and the image size is 512×512.

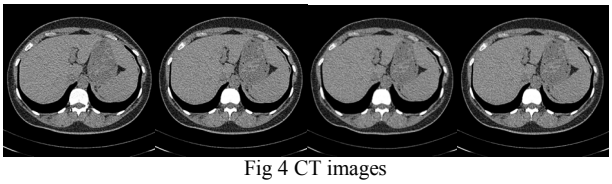

Fig 4 CT images

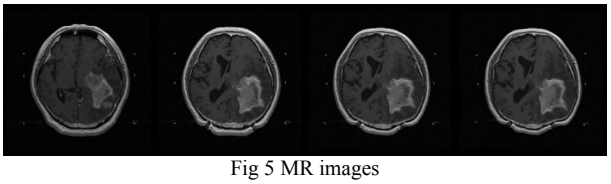

Fig 5 MR images

In fig 4 and fig5, the first image is fixed image, the second image is moving image, the third image is registered by a serial algorithm and the last one is registered by the parallel algorithm.

In this experiment, the frequency of the CPU in computer is 2.33GHz. The GPU is NVIDIA GeForce9500GT, which has 32 streaming multiprocessors, with the frequency 800MHz.

A. Parallel computing NMI

Fig 6 illustrates the time cost percentage among the NMI computing, the gradient descent computing and the communication with the different number of mark points. According to (10) and (12), computing NMI has to needs to traverse the both of the fixed image and, moving image and the joint gray histogram. Moreover, computing NMI, which requires a large number of floating-point operations, is often called in an iteration algorithm.

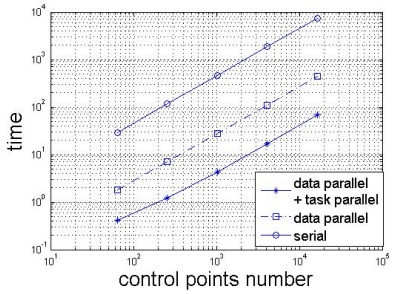

Fig 6 the percentage of the non-rigid registration process

Fig 6 indicates that computing NMI occupies about 90 percent of total time because of mass of mark points are needed to characterize the small deformation in the free deformation mode.

Furthermore, there is always a large quantity of gray scales in medical images that quite influence the calculation speed of NMI [18]. Tab 1 illustrates the time cost for NMI with a serial algorithm and a parallel algorithm.

Tab .1 time costs for NMI with a serial algorithm and a parallel algorithm

		CT images		MR image	
Gray	size	CPU(ms)	GPU(ms)	CPU(ms)	GPU(ms)
256	512×512×64	1128	96.51	1124	96.38
512	512×512×64	1756	246.32	1695	239.38
1024	512×512×64	9047	500.45	9039	508.65
2048	512×512×64	11563	967.84	11621	74.35
4096	512×512×64	470280	1909.12	469362	1917.76

A fine-grained data parallel algorithm is fit to compute NMI owing to its data intensive. What's more, little data is needed to exchange among threads, so a reduction an algorithm can be used to accelerate non-rigid registration process.

B. Task parallel

$\Phi = \{\phi_1, \phi_2, ..., \phi_{N-1}\}$ is defined as a mark points set of moving image,

$$C_{NMI}(R,F) = \arg\max_{}(R, T(F)) = \arg\max_{}(R, T(\Phi)) \quad (15)$$

So, a non-rigid registration can be seen a multidimensional parameter function. Φ^0 is a initial parameter set, and using gradient descent flow as a optimization algorithm. That, the process of the multidimensional parameter function can be optimized as following,

① calculate $\nabla C_{NMI} = \partial C_{NMI} / \partial \Phi^i$;

② update $\Phi : \Phi^{i+1} \leftarrow \Phi^i + \mu \frac{\nabla C_{NMI}}{\| \nabla C_{NMI} \|}$;

③ if $\| \nabla C_{NMI} \| = \| \frac{\partial C_{NMI}}{\partial \Phi^i} \| < \varepsilon$,then loop end； else it goes on. where,

$$\Phi^i = [\phi_0^i, \phi_1^i, ..., \phi_{N-1}^i] \quad (16)$$

$\frac{\partial C_{NMI}}{\partial \Phi}$ can be computed with a finite difference method.

$$\nabla C_{NMI} = \frac{\partial C_{NMI}}{\partial \Phi^j} = [\frac{\partial C_{NMI}}{\partial \phi_1^j}, \frac{\partial C_{NMI}}{\partial \phi_2^j}, ..., \frac{\partial C_{NMI}}{\partial \phi_{N-1}^j}] \quad (17)$$

Where,

$$\frac{\partial C_{NMI}}{\partial \phi_i^j} = \frac{\partial C_{NMI}(\phi_1^j, \phi_2^j, ..., \phi_i^j, ..., \phi_{N-1}^j)}{\partial \phi_i^j} \approx \frac{C_{NMI}(\phi_1^j, \phi_2^j, ..., \phi_i^j + \Delta\phi_i^j, ..., \phi_{N-1}^j) - C_{NMI}(\phi_1^j, \phi_2^j, ..., \phi_i^j, ..., \phi_{N-1}^j)}{\Delta \phi_i^j} \quad (18)$$

Summary, a task parallel algorithm actually is a parallel algorithm computing gradient descent $\frac{\partial C_{NMI}}{\partial \Phi}$.

Tab 2 time cost for gradient descent

	CT images		MR images	
mark points number	CPU(s)	GPU(s)	CPU(s)	GPU(s)
8×8×8	12.6	1.56	13.7	1.68
16×16×8	47.3	2.97	49.1	3.0
32×32×8	184.1	6.34	193.2	6.37
64×64×8	699.8	25.3	756.1	25.2
128×128×8	2442.4	105.4	2397.8	98.9

In practice, a task parallel algorithm contains some data parallel algorithm because gradient descent needs coordinate interpolation which is a vector operation.

C. Data parallel and task parallel

As 5.1 and 5.2 said, a data parallel algorithm and a task parallel algorithm can be used at the same time. Fig 7 illustrates the time cost of different parallel algorithm.

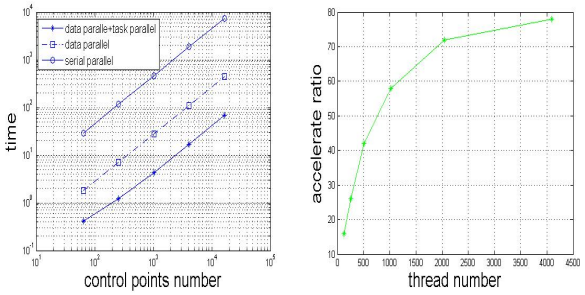

Fig 7 the time cost of different algorithm and thread number

The left picture denotes that a task parallel algorithm with a data parallel algorithm could achieve the best result, the right one shows the relationship between a speedup rate and threads number.

VI. Registration results

Fig 10.a illustrates the NMI with a different number of 128×128×8 mark points and fig 10.b illustrates the NMI with mark points that iterate 10, 20, 30, 40 and 50 times. The best accuracy of the CT image is 2.48mm and the best accuracy of MR image is 4mm when it iterate 50 times.

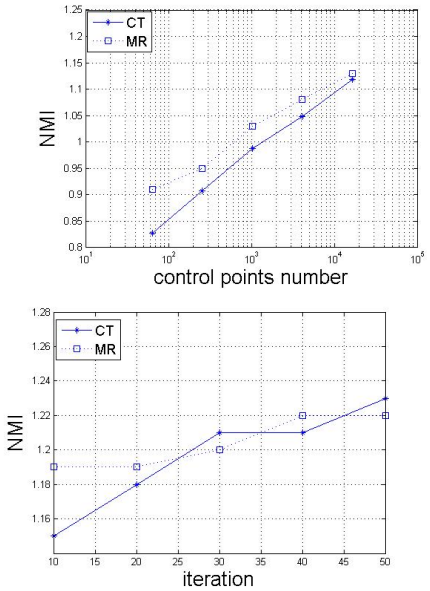

Fig 8 the results

The left image is explained the relationship between mark points number and measure, the right one the relationship between iteration number and measure.

VII. Conclusion

The optimization algorithm can reduce about a half calculation amount with optimization for coefficient of B-splines function owing to their relationships of each voxel in the same sub-image. Also, active mark points and passivity mark points are defined in order to distinguish the mark points disturbed on the background region which dose not need any processing. Moreover, a multi-scale optimization is used to extract active mark points from a mark points set, reducing the number of the mark points need to process.

There are some parallel processes in medical non-rigid registration, such as NMI, interpolation, space transformation and optimization, what are data intensive operations. So a parallel algorithm is fit to non-rigid registration. In this paper, a fine-grained data parallel algorithm is used with rectangle data separation, mark points optimization according to NMI computing feature. In addition, a coarse-grained task parallel algorithm is also used to calculate the gradient descent flow because the B-splines function can only influence its 4×4 neighborhood owing to its finite domain of definition. In the experiment, the parallel algorithm achieves a good result with reasonable memory space.

[1] T. Rohlfing, and Calvin Maurer, "Modeling liver motion and deformation during the respiratory cycle using intensity-based nonrigid registration of gated MR images", Medical Physics, vol. 31, pp. 427-432, 2004.

[2] S. Lee, G. Wolberg, and S.Y. Shin, "Scattered data interpolation with multilevel B-spline", IEEE Transactions on, Visualization and Computer Graphics, vol. 3, pp. 228-244, 1997.

[3] J. P. W. Pluim, J.B.A.Maintz, and M.A. Viergever, "f-information measures in medical image registration", IEEE Transactions on, Medical Imaging, vol. 23, pp. 1508-1516, 2004.

[4] X. Xiaoyan, and R. D. Dony, "Evaluation of hierarchical elastic medical image registration method", in Electrical and Computer Engineering, Canadian Conference on, Vol.3, pp. 1289-1292, 2004.

[5] S. S. Beauchemin, and J. L. Barron, "The computation of optical flow", ACM Comput. Surv, vol. 27, pp. 433-466, 1995.

[6] J. P. W. Pluim, J.B.A.Maintze, and M.A. Viergever, "Mutual-information-based registration of medical images: A survey", IEEE Transactions on Medical Imaging, vol. 22, pp. 986-1004, 2003.

[7] M. B. Skouson, Quji Guo, ang Zhi-Pei Liang, "A bound on mutual information for image registration", IEEE Transactions on, Medical Imaging, vol. 20, pp. 843-846, 2001.

[8] W. Plishker, O.Dandeker, S. Bhattacharyya, and R. Shekhar, "A taxonomy for medical image registration acceleration techniques", in Life Science Systems and Applications Workshop, LISA IEEE/NIH , pp. 160-163, 2007.

[9] Haifang Zhou, Xuejun Yang, Hengzhu Liu, and Yu Tang, "First evaluation of parallel methods of automatic global image registration based on wavelets", in Parallel Processing, International Conference on, pp. 129-136, 2005.

[10] M. P. Wachowiak, and T. M. Peters, "High-performance medical image registration using new optimization techniques", IEEE Transactions on, Information Technology in Biomedicine, vol. 10, pp. 344-353, 2006.

[11] S. Ourselin, R.Stefanescu, and X Pennec, "Robust Registration of Multi-modal Images: Towards Real-Time Clinical Applications", in Medical Image Computing and Computer-Assisted Intervention — MICCAI, vol1.pp. 2489-2494, 2002.

[12] S. Hastings, T.Kurc, S.Langella, U. Catalyurek, T.Pan, and J.Saltz, "Image processing for the rigid: a toolkit for building rigid-enabled image processing applications", in Cluster Computing and the rigid, Proceedings CCrigid , 3rd IEEE/ACM International Symposium on, pp. 36-43, 2003.

[13] T. Rohlfing, and C. R. Maurer, "Nonrigid image registration in shared-memory multiprocessor environments with application to brains, breasts, and bees", IEEE Transactions on , Information Technology in Biomedicine, vol. 7, pp. 16-25, 2003.

[14] M. P. Wachowiak, and T. M. Peters," Parallel Optimization Approaches for Medical Image Registration", in Medical Image Computing and Computer-Assisted Intervention – MICCAI, vol. 3216, pp. 781-788, 2004.

[15] G. E. Christensen, "MIMD vs. SIMD parallel processing: A case study in 3D medical image registration", Parallel Computing, vol. 24, pp. 1369-1383, 1998.

[16] P. Muyan-Ozcelik, D.J.Owens, Junyi Xia, and S.S Samant, "Fast Deformable Registration on the GPU: A CUDA Implementation of Demons", in Computational Sciences and Its Applications, ICCSA International Conference, pp. 223-233, 2008.

[17] T. W. Sederberg, and S. R. Parry, "Free-form deformation of solid geometric models", SIGGRAPH Comput. Graph., vol. 20, pp. 151-160, 1986.

[18] Jiarui Lin, Zhiyong Gao, Bangquan Xu, Yangxiezi Cao, and Zhan Yingjian, "The effection of grey levels on mutual information based medical image registration", IEEE in Nuclear Science Symposium Conference Record, vol.3, pp. 1575-1579, 2002.