Accelerating Mutual Information Computation for Nonrigid Registration on the GPU

Kei Ikeda, Fumihiko Ino, Kenichi Hagihara

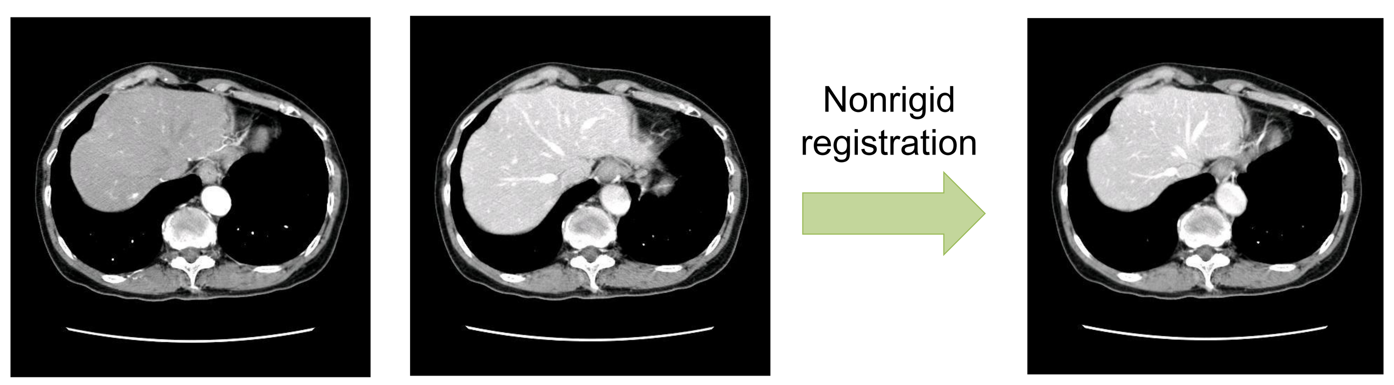
Graduate School of Information Science and Technology, Osaka University

{i-kei, ino}@ist.osaka-u.ac.jp

Background and goal

Nonrigid registration

- A technique for defining a geometric relation between each point in images
  - helps doctors in detecting cancers by monitoring changes in size
  - creates novel images by combining different modality images
    Examples of modality: CT, PET, and MR
- Nonrigid registration is a compute-intensive application because it deals with deformable objects, which require many degrees of freedom
- Many researchers accelerated registration using the GPU
  - Multimodal registration has not been supported by GPU implementations
  - Mutual information must be computed for multimodal registration



Reference image R Floating image F Aligned image

Our goal

- Acceleration of mutual information computation for nonrigid registration

Technical issue

- Fast mutual information computation with using shared memory
  - Shared memory is not large enough to deal with mutual information

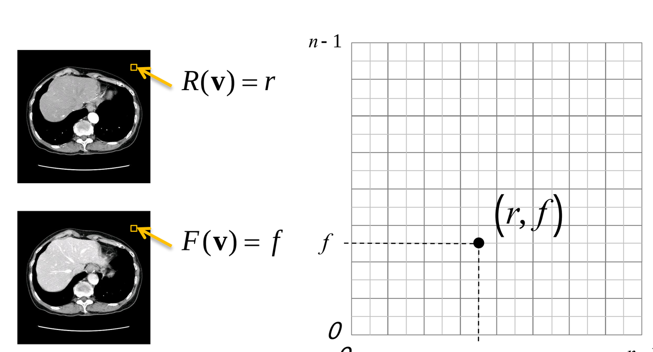
Nonrigid registration algorithm

Registration strategy

- Many algorithms solve registration problems through optimization of a similarity function, which represents how similar the images are
- Iterative methods such as steepest descent is used for optimization
- Objects are deformed at every optimization step according to similarity values
- Hierarchal data structure is usually employed to reduce computational cost

Normalized mutual information

- A robust similarity measure [1] used for multimodal registration
- Joint histograms must be constructed to obtain mutual information
  - A joint histogram is a 2-D matrix containing the number of intensities at the same position



Images Joint histogram

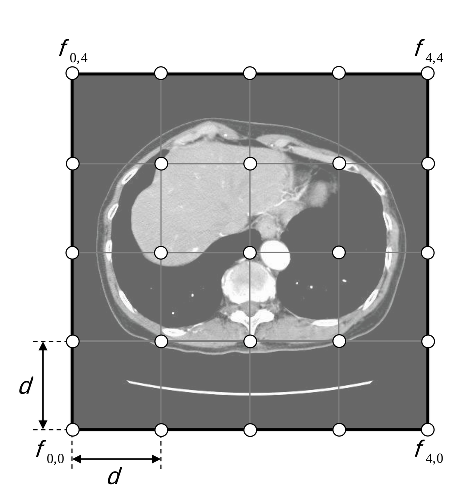
B-spline deformation model [2]

- Deformation is locally controlled by using a mesh of control points

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

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

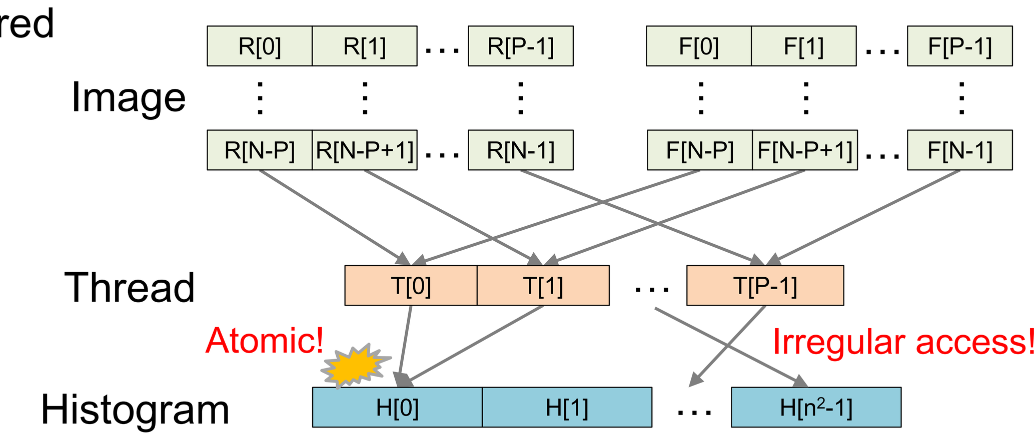
$B_l(u)$: l - th basis function of B - splines



Previous method

Technical issues in mutual information computation

- Joint histograms are not small enough to be stored in shared memory
  - A joint histogram for n grayscale levels contains n^2 bins
  - Data size reaches 256 KB if n=256 and each bin has 4-byte data
- Irregular access to histogram data
- Atomic operations are required to serialize simultaneous accesses to the same bin



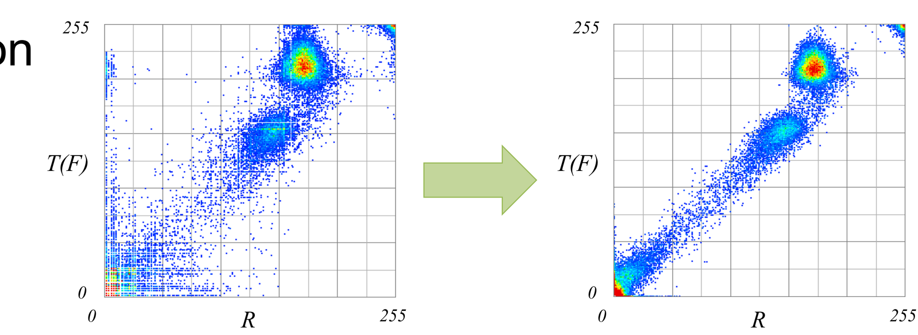
Shams' method [3]

- Each thread has its local joint histogram to avoid atomic operations
- Local histograms are then merged into a single histogram by parallel reduction
- Histograms are stored in global memory

Our method

Key idea for data size reduction

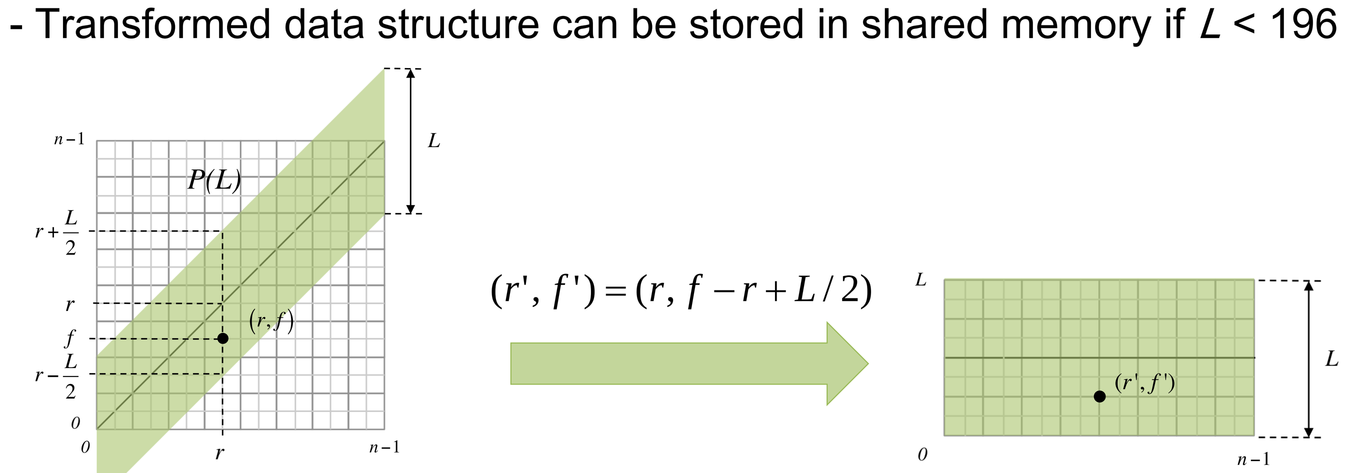
- Joint histograms are sparse data
- Our method reduces data size by eliminating empty bins far from the diagonal of the 2-D matrix
  - Plots in joint histograms come together around the diagonal, as the floating image converges to an optimal solution



Before registration After registration

Our data structure

- Bins within parallelogram $P(L)$ are transformed into dense data structure
- Data size of $P(L)$
  - nL in bytes if each bin has 8-bit data
  - Transformed data structure can be stored in shared memory if L < 196

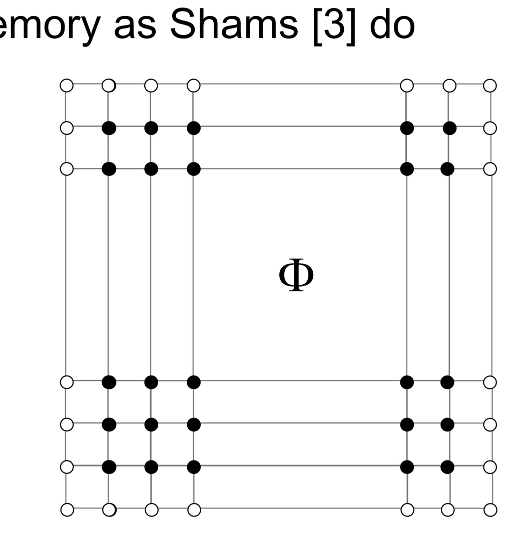


Proposed algorithm

- Our algorithm switches its behavior at every optimization step
  - If all plots exist within $P(L)$, bins within $P(L)$ are updated using shared memory
  - Otherwise, all bins are computed using global memory as Shams [3] do

Acceleration of B-spline deformation

- Data reuse
  - Our method stores $\hat{\varphi}_{i+l}$ in shared memory because it can be reused between different voxels (i.e., threads)
- Divergent branch elimination
  - Dummy control points are placed along the boundary to eliminate branches



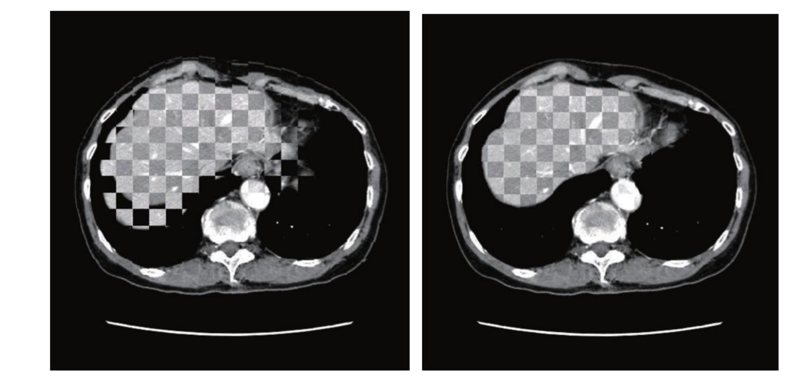
Experiments

Experimental setup

- Performance comparison with Shams' method [3] and a multi-threaded CPU implementation

Dataset

- 4 CT images of the liver
- 512x512x256 voxel with 256 grayscale levels
- Voxel size: 0.67x0.67x0.67 mm



Before After

Machine

- GPU: NVIDIA GeForce GTX 580 (VRAM capacity: 1.5 GB)
- CPU: Intel Core i5 2500K 3.3 GHz (4 cores, RAM capacity: 16 GB)
- Windows 7, CUDA 4.0, and driver 285.62

Parameter configuration

- 3-levels of hierarchy
- Our method uses shared memory at the 2nd and the 3rd levels

Hierarchy level	1	2	3
Voxel size (mm)	2.68	1.34	0.67
Volume size (voxel)	128x128x64	256x256x128	512x512x256
Control point spacing (mm)	42.88	21.44	10.72

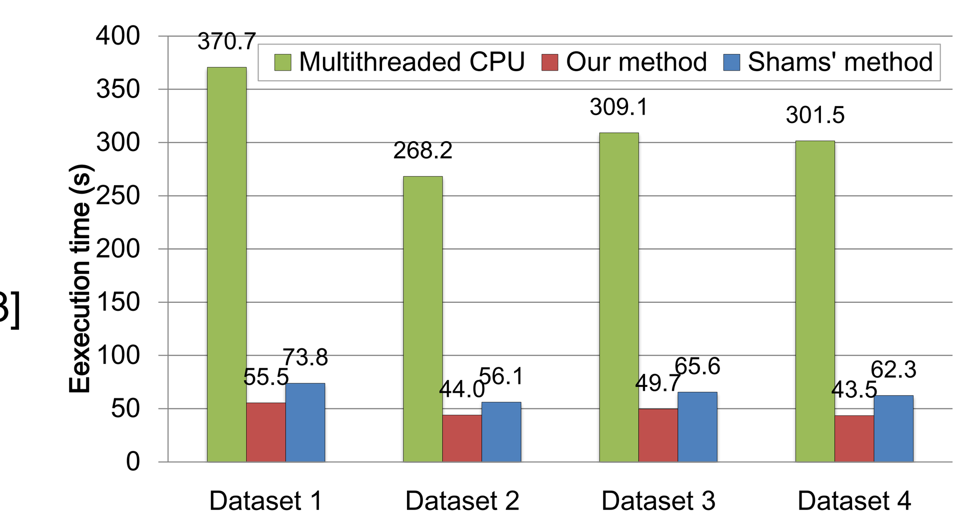
Performance results

Joint histogram computation

- 3X speedup over [3]

Nonrigid registration

- 1.3-1.4X speedup over [3]
- 6.1-6.9X speedup over the multi-threaded CPU implementation



Breakdown analysis of execution time

- Effective memory bandwidth increases from 9 GB/s to 27 GB/s
- But, the effective bandwidth is equivalent to 14% of peak bandwidth (192 GB/s)
- This lower efficiency is mainly due to the 3rd level, in which different threads frequently update the same bin as the registration process converges to a solution

	CPU version			Our method			Shams' method [3]		
Hierarchy level	1	2	3	1	2	3	1	2	3
Gradient computation	0.72	4.60	37.10	0.12	0.70	5.36	0.12	0.88	7.33
Similarity computation	0.03	0.31	0.91	0.01	0.02	0.13	0.01	0.07	0.67
Deformation	0.31	2.23	15.34	0.05	0.36	2.47	0.05	0.36	2.47
Others	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Total	1.07	6.97	53.36	0.19	1.09	7.97	0.19	1.32	10.48

References

[1] C. Studholme, D. L. G. Hill, D. J. Hawkes, An overlap invariant entropy measure of 3D medical image alignment, Pattern Recognition, 32(1):71–86, 1999.

[2] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. G. Hill, M. O. Leach, D. J. Hawkes, Nonrigid registration using free-form deformations: Application to breast MR images, IEEE Trans. Medical Imaging 18(8):712–721, 1999.

[3] R. Shams, P. Sadeghi, R. A. Kennedy, R. Hartley, Parallel computation of mutual information on the GPU with application to real-time registration of 3D medical images, Computer Methods and Programs in Biomedicine, 99(2):133–146, 2010.