CATEGORY: MEDICAL IMAGING - MI10

POSTER P5304 | CONTACT NAME Nathan Ellingwood: nathan-ellingwood@uiowa.edu

GPU TECHNOLOGY CONFERENCE

# Accelerating Multi-level Deformable Image Registration Methods for Lung Images with GPU Computing


Nathan Ellingwood[1], Youbing Yin[2], Matthew Smith[4], Ching-Long Lin[1,2,3]

[1]IIHR-Hydroscience & Engineering, [2]Department of Mechanical & Industrial Engineering, [3]Department of Applied Mathematical and Computational Sciences, The University of Iowa, Iowa City, IA, USA, [4]Department of Mechanical Engineering, National Cheng Kung University, Tainan City, Taiwan



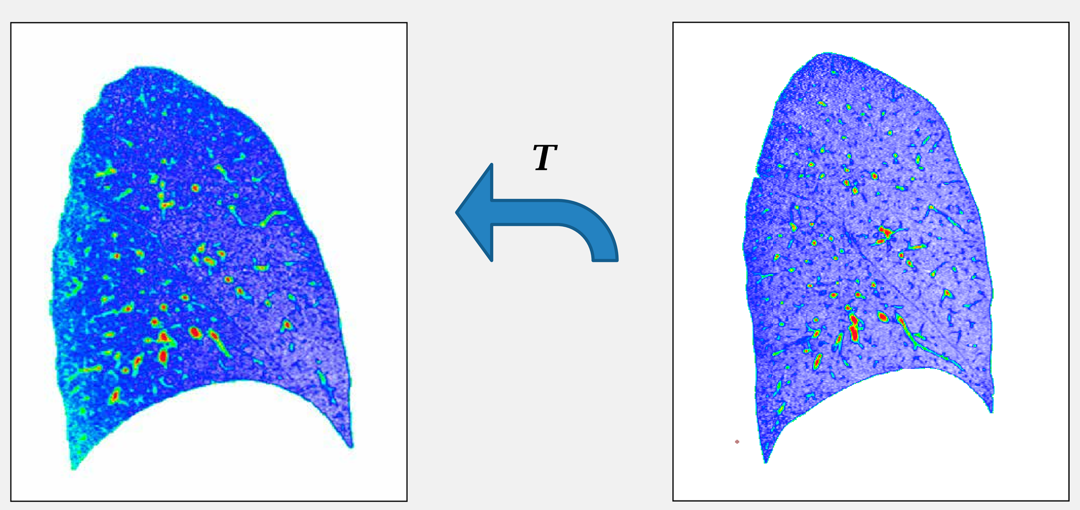
## Abstract

In this work we introduce a novel Diffeomorphic Multi-level Transform Composite method (DMTC) for accelerated computation of nonrigid mass-preserving registration of lung images with large deformation on Graphics Processing Units (GPUs). The proposed method dramatically reduces computational time when compared to its single- and multi-threaded CPU counterparts, with the speedup factor ranging from 6 to 112. The significance of this work is two-fold. First, the DMTC achieves computation and memory efficiencies on GPUs, and together with the mass-preserving measure improves the accuracy of registration. Second, the improved computational efficiency is essential in analyzing data for population-based studies and translational science. This GPU implementation can be easily adapted for use with other non-mass-preserving similarity measures.


## Registration Model

### Cubic B-spline Transformation Model:

$$T[x,\phi] = x + \sum_{l=0}^{3}\sum_{m=0}^{3}\sum_{n=0}^{3} B_l(u)B_m(v)B_n(w)\,\phi_{i+l,j+m,k+n}$$

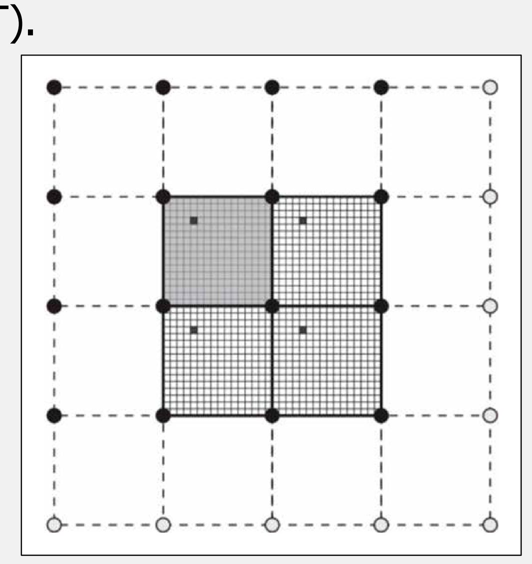



Human lung at FRC          Human lung at TLC

**Figure 1** Transformation from TLC (moving image) to FRC (fixed image)

### Grid Alignment:

B-Spline weight values repeat across each tile created by grid alignment, allowing pre-computation of weights for single tile and storage in lookup table (LUT).



32 x 32 Voxel Image Grid
5 x 5 Control Node Grid

**Legend**:
Small squares = voxels
Grey circles = control nodes
Bold squares = tiles

**Figure 2** 2D example of grid alignment

### SSTVD Similarity Criterion:

$$C(\phi) = \sum_{x\in\Omega} v_f(x)\left[\tilde{I}_f(x) - J_T(x,\phi)\tilde{I}_m(T[x,\phi])\right]^2, \quad \tilde{I}(x) = \frac{I(x)-HU_{air}}{HU_{tissue}-HU_{air}}$$

### Control Node Constraints Enforce Positive Jacobian:

$$\phi_x < \frac{\delta_x}{K}, \quad \phi_y < \frac{\delta_y}{K}, \quad \phi_z < \frac{\delta_z}{K}, \quad K = 2.479472335$$

## Methods

### Diffeomorphic Multi-level Transform Composite (DMTC):

Constraints create conflicting needs – capturing large deformations requires large control node spacing, local deformations require small spacing. A Multi-level method is employed to satisfy both. A composite transform defines the final transformation.

| | Previously Reported | DMTC |
|---|---|---|
| **Composite Transform $T_c$** | $T_c = T_i \circ T_{i-1} \circ \cdots \circ T_0$ | $T_c = T_0 \circ T_1 \circ \cdots \circ T_i$ |
| **Warping Image $\omega$** | $\omega = T_{i-1} \circ \cdots \circ T_0$ | $\omega = T_0 \circ T_1 \circ \cdots \circ T_{i-1}$ |
| **$T_c$ (in terms of $\omega$)** | $T_c[x,\phi] = T_i[\omega(x),\phi]$ | $T_c[x,\phi] = \omega(T_i[x,\phi])$ |

### Composite Transform Example (2x2 voxels):

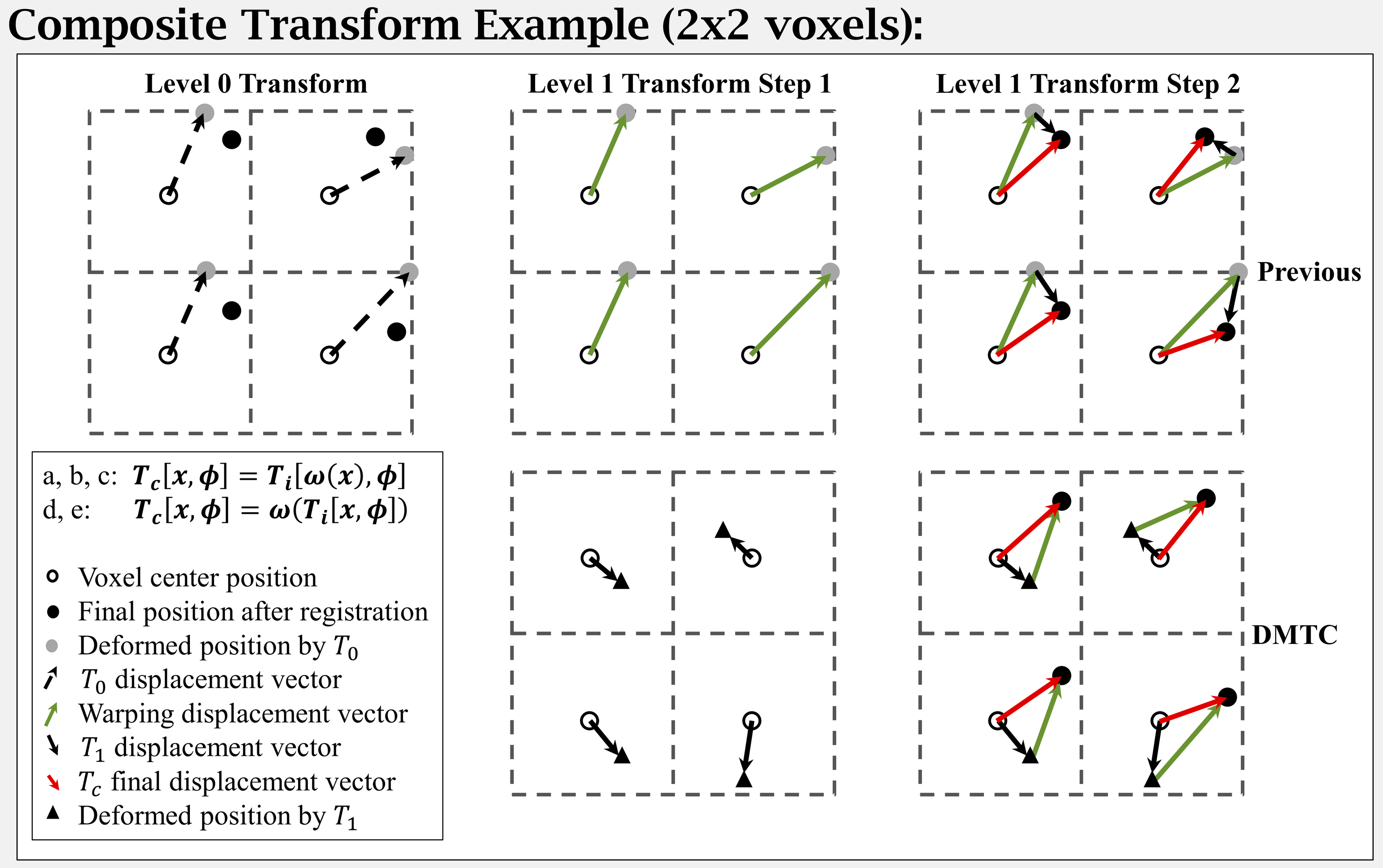



**Figure 3** Top row: Typical composite transform first applies warping image displacement to original position (step 1). The displaced position serves as input to the current transformation (step 2). Bottom row: The DMTC takes the original position as input to the current transform (step 1). The displaced position is then input to the (interpolated) warping image (step 2). Regular voxel spacing is preserved.

### GPU Implementation using CUDA:

- Tiles assigned to blocks for parallelization.
- Each 3D tile is partitioned into sub-tiles of 4x4x4 voxels, each voxel assigned to one of 64 threads.
- Sum reduction used to compute partial sums of sub-tile's contribution to cost and cost-gradient, partial sums and running total stored in shared memory.
- Final contribution of a tile to the cost-gradients of its surrounding control nodes stored in bin data structure, consisting of 64 slots per node in 3D. Sum reduction performed on bins to attain final values.

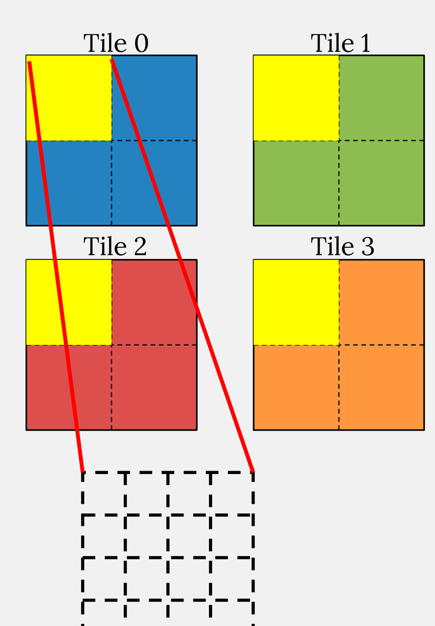



**Figure 4(a)** 2D tiling example with 4 tiles assigned to threads and partitioned into sub-tiles of 4x4=16 voxels, each of which is assigned to a thread.

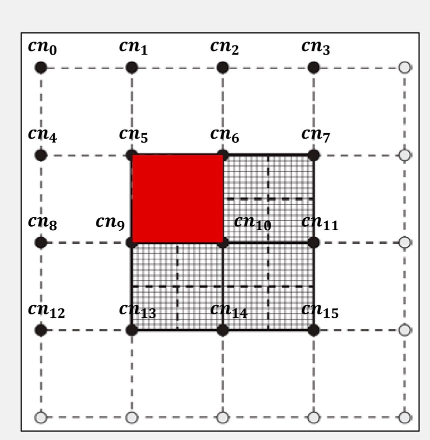


**Figure 4(b)** 2D example showing 16 control nodes surrounding tile 0. Each control node assigned a bin consisting of 16 slots, each slot uniquely identified with a surrounding tile for storage of partial cost-gradient sum.

## Results and Conclusions

### Benefits of DMTC and Tiling:

- Pre-compute B-Spline weights LUT for one tile one time per registration level.
- Pre-compute several variables for cost and cost-gradient at initialization by splitting current transform $T_i$ and warping terms.
- Minimizes CPU-GPU communication, eliminates redundant computation, reduces memory storage needs to fit GPU architecture limitations.
- Tiling allows use of fast shared memory and avoids slower global memory.

| Version | Total Time (min) | Total Cost and Cost-gradient Time (min) |
|---|---|---|
| **K20X GPU** | 2.9 | 1.1 |
| **12T CPU** | 12.9 | 11.7 |
| **6T CPU** | 23.7 | 22.6 |
| **1T CPU** | 112.5 | 111.3 |

**Table 1** Total times for 8 level image pyramid ($32^3$ to $256^3$ voxels). Intel Xeon E5-2620 6-core CPU and Nvidia Tesla K20X GPU were used.

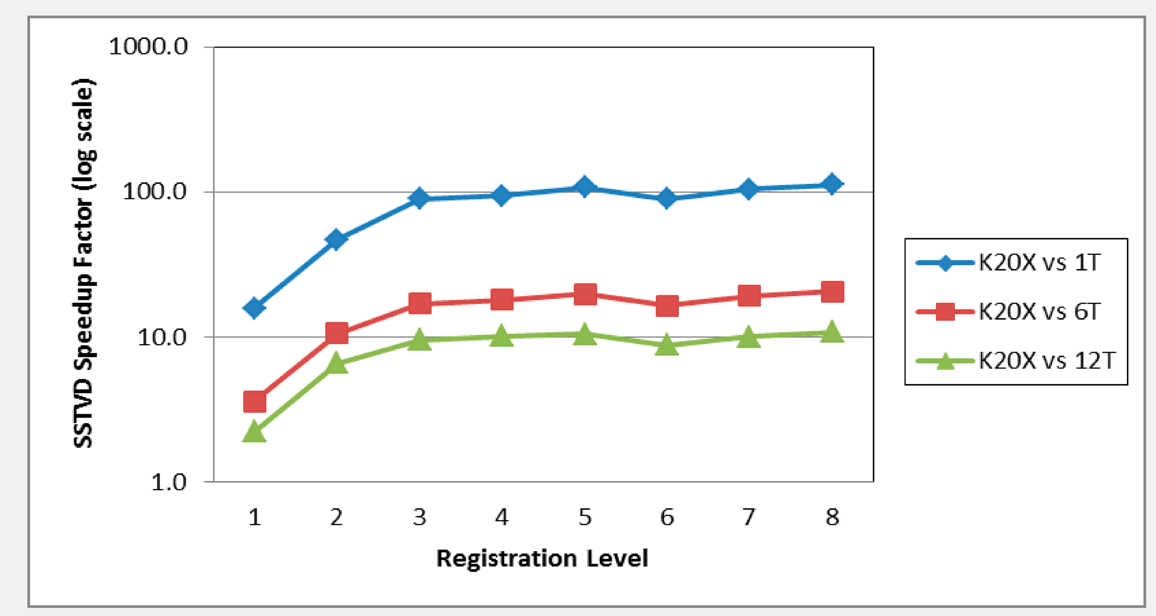



**Figure 5** Speedup factors per level, $speedup = t_1/t_0,\ t_1 > t_0$

| Level | K20X GPU | 12T CPU | 6T CPU | 1T CPU |
|---|---|---|---|---|
| 1 | 0.013 | 0.03 | 0.05 | 0.21 |
| 2 | 0.005 | 0.03 | 0.05 | 0.24 |
| 3 | 0.019 | 0.18 | 0.32 | 1.70 |
| 4 | 0.018 | 0.18 | 0.33 | 1.70 |
| 5 | 0.13 | 1.33 | 2.49 | 13.51 |
| 6 | 0.15 | 1.34 | 2.50 | 13.52 |
| 7 | 1.03 | 10.29 | 19.78 | 106.99 |
| 8 | 0.95 | 10.26 | 19.72 | 106.94 |

**Table 2** SSTVD time per iteration by registration level for GPU and CPU implementations

## References


Yin, Youbing, Eric A. Hoffman, and Ching-Long Lin. Mass preserving nonrigid registration of CT lung images using cubic B-spline. Medical physics 36.9 (2009): 4213-4222.

Shackleford, J. A., Kandasamy, N., & Sharp, G. C. (2010). On developing B-spline registration algorithms for multi-core processors. Physics in Medicine and Biology, 55(21), 6329.

Rueckert, D., Sonoda, L. I., Hayes, C., Hill, D. L., Leach, M. O., & Hawkes, D. J. (1999). Nonrigid registration using free-form deformations: application to breast MR images. Medical Imaging, IEEE Transactions on, 18(8), 712-721.

Choi, Y., & Lee, S. (2000). Injectivity conditions of 2D and 3D uniform cubic B-spline functions. Graphical models, 62(6), 411-427.


## Acknowledgements


This work supported in part by NIH grants R01 HL094315, U01 HL114494, and S10 RR022421.