

# Fast Optimal Mass Transport for Dynamic Active Contour Tracking on the GPU

Gallagher Pryor, Tauseef ur Rehman, Shawn Lankton, Patricio A. Vela and Allen Tannenbaum

**Abstract**—In computational vision, visual tracking remains one of the most challenging problems due to noise, clutter, occlusion, and dynamic scenes. No one technique has yet managed to solve this problem completely, but those that employ control-theoretic filtering techniques have proven to be quite successful. In this work, we extend one such technique by Niethammer *et al.* in which implicitly represented dynamically evolving contours are filtered using a geometric observer framework. The effectiveness of the observer hangs upon the solution of two major problems: (1) the calculation of accurate curve velocities and (2) the determination of diffeomorphic correspondence maps between curves for geometric interpolation. We propose the use of novel image registration techniques such as image warping and optimal mass transport for the solution of these problems which increase the performance of the framework and reduce algorithmic complexity. One major drawback to the original scheme, as it relies on PDE solutions, is its computational burden restricting it from real time use. We show that the framework can, in fact, run in near real time by implementing our additions to the framework on the graphics processing unit (GPU) and show better execution times for these algorithms than reported in recent literature.

## I. INTRODUCTION

In computational vision, visual tracking is the act of consistently locating a desired feature in each image of an input sequence. Visual tracking is a critical step in numerous machine vision applications such as surveillance, driver assistance, human-computer interactions, etc. However, it is a challenging problem due to noise, clutter, occlusion, and dynamic scenes. Numerous approaches and techniques exist for the solution of the tracking problem and some of the most successful employ control-theoretic filtering techniques.

In this paper we build on one such control-theoretic tracking framework by Niethammer *et al.* [1] who proposed a deterministic observer framework for visual tracking based on non-parametric implicit (level-set) curve descriptions. This work is novel in that it provides a general framework for the filtering of implicit curves and additional state without the need for parameterizations which are complex to implement in practice and have their own particular drawbacks.

The above referenced work involves a continuous-discrete observer with continuous-time system dynamics and discrete-time measurements. The proposed framework is general in nature and multiple simulation models can be

This work was supported in part by grants from NSF, AFOSR, ARO, MURI, MRI-HEL, and NIH

Tauseef ur Rehman, Patricio Vela, Shawn Lankton, Allen Tannenbaum are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0250, {tauseef,slankton,tannenba}@ece.gatech.edu.

Gallagher Pryor is with College of Computing, Georgia Institute of Technology donovang@cc.gatech.edu

incorporated. Measurements can be performed through static segmentation techniques and optical flow computations.

Unfortunately, discrete-time measurements lead to the problem of geometric curve interpolation and the filtering of quantities propagated along with implicitly represented estimated curves. Interpolation and filtering are intimately linked to the correspondence problem between curves. In [1], this is done using a Laplace equation approach, establishing a one-to-one correspondence between measured and estimated curves to determine unique "distances" between points and to exchange information between them for dynamic filtering. The distance measurements allow for a geometric interpolation between measured and estimated curves, facilitating geometric, intuitively tunable gains for position filtering.

In our work, we use the Monge-Kantorovich formulation of the optimal mass transport (OMT) problem for the purpose of establishing the diffeomorphic correspondence maps between the estimated and measured curves instead of the Laplace equation approach which requires domain decompositions in order compute the correct map. The OMT method, on the other hand, is an elegant solution to the correspondence problem and is applied globally on the implicit representations of the curve. Although in the  $L^2$  case OMT is an NP-hard problem we benefit from a fast and near-realtime, multi-resolution and multigrid implementation of the OMT algorithm on the GPU [2], [3]. Another significant contribution is the use of a registration warp as a velocity measurement instead of the classical Horn & Schunck [4] optical flow used by Niethammer *et al.* [1]. This warp is also a 2D velocity field over every point in the image like its predecessor but has accurate magnitudes making advection unnecessary in the framework and velocity measurements accurate.

The paper is organized in the following sections. Section II briefly reviews curve evolution equations and the general observer structure as proposed in [1]. Section III briefly describes the areas of contribution of this work. In Sections IV to VII we describe our proposed velocity measurement based upon image warping, our proposed error correction scheme based upon optimal mass transport, and the GPU implementation of the framework. Results are presented in Section VIII. Conclusions are given and future work is discussed in Section IX.

## II. BACKGROUND

### A. Curve Evolution

A planar curve evolution may be described as the time-dependent mapping:  $\mathcal{C}(p, t) : S^1 \times [0, \tau) \mapsto \mathbb{R}^2$ , where  $p \in$

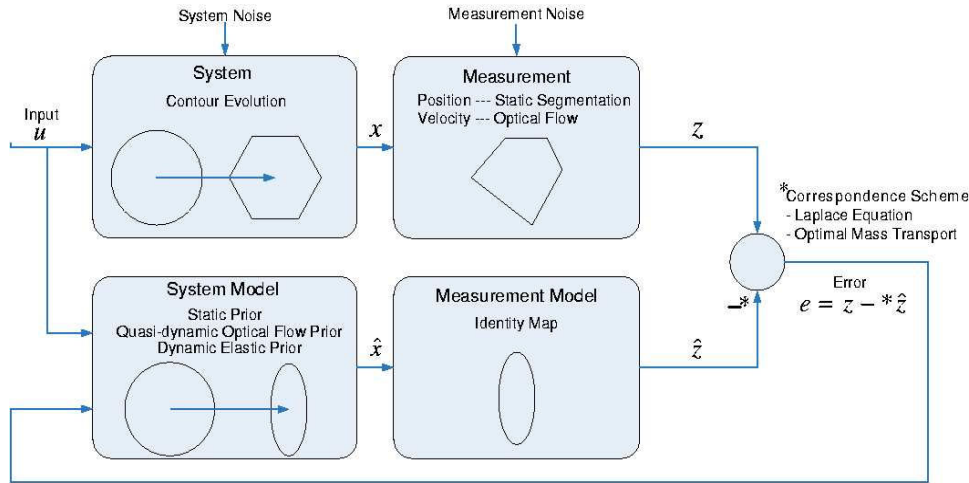


Fig. 1. Geometric Observer Structure

$[0, 1]$  is the curve's parameterization on the unit circle  $S^1$ ,  $\mathcal{C}(p, t) = [x(p, t), y(p, t)]^T$ , and  $\mathcal{C}(0, t) = \mathcal{C}(1, t)$ . Define the interior and the exterior of a curve  $\mathcal{C}$  on the domain  $\Omega \subset \mathbb{R}^2$  as

$$\begin{aligned} \text{int}(\mathcal{C}) &:= \{ \mathbf{x} \in \Omega : (\mathbf{x} - \mathbf{x}_c)^T \mathcal{N} > 0, \forall \mathbf{x}_c \in \mathcal{C} \}, \\ \text{ext}(\mathcal{C}) &:= \Omega \setminus \overline{\text{int}(\mathcal{C})}, \end{aligned}$$

where  $\mathcal{N}$  is the unit inward normal to  $\mathcal{C}$ . To avoid tracing individual curve particles over time  $\mathcal{C}$  can be represented implicitly by a level set function  $\Psi : \mathbb{R}^2 \times [0, \tau) \rightarrow \mathbb{R}$  [5], where

$$\Psi(0, t)^{-1} = \text{trace}(\mathcal{C}(\cdot, t)).$$

There is no unique level set function  $\Psi$  for a given curve  $\mathcal{C}$ . Frequently,  $\Psi$  is chosen to be a signed distance function, defined as follows:

$$\begin{cases} \|\nabla \Psi\| = 1, \text{ almost everywhere,} \\ \Psi(\mathbf{x}) = 0, \forall \mathbf{x} \in \mathcal{C}, \\ \Psi(\mathbf{x}) < 0, \forall \mathbf{x} \in \text{int}(\mathcal{C}), \\ \Psi(\mathbf{x}) > 0, \forall \mathbf{x} \in \text{ext}(\mathcal{C}). \end{cases}$$

Given a curve evolution equation

$$\mathcal{C}_t = \mathbf{v},$$

where  $\mathbf{v}$  is a velocity vector, the corresponding level set evolution equation is [5]

$$\Psi_t + \mathbf{v}^T \nabla \Psi = 0.$$

The unit inward normal,  $\mathcal{N}$ , and the signed curvature,  $\kappa$ , are given by

$$\mathcal{N} = -\frac{\nabla \Psi}{\|\nabla \Psi\|}, \quad \kappa = \nabla \cdot \frac{\nabla \Psi}{\|\nabla \Psi\|}.$$

### B. General Observer Structure

Assume that the system to be observed evolves in continuous time and that measurements of the system's states become available at discrete time instants  $k \in \mathbb{N}_0^+$ , i.e.,

$$\begin{aligned} \begin{pmatrix} \mathcal{C} \\ \mathbf{q} \end{pmatrix}_t &= \begin{pmatrix} \mathbf{v}(\mathcal{C}, \mathbf{q}, t) \\ \mathbf{f}(\mathcal{C}, \mathbf{q}, t) \end{pmatrix} + \mathbf{w}(t) \\ z_k &= \mathbf{h}_k \left( \begin{pmatrix} \mathcal{C} \\ \mathbf{q} \end{pmatrix} \right) + \mathbf{v}_k = \begin{pmatrix} \mathcal{C}(t_k) \\ \mathbf{q}(t_k) \end{pmatrix} + \mathbf{s}_k(t), \end{aligned}$$

where  $\mathbf{w}$  and  $\mathbf{s}_k$  are the system and measurement noises respectively,  $\mathcal{C}$  represents the curve position, and  $\mathbf{q}$  denotes additional states transported along with  $\mathcal{C}$  (e.g., velocities). Assume further a simulation and measurement model of the form

$$\begin{pmatrix} \hat{\mathcal{C}} \\ \hat{\mathbf{q}} \end{pmatrix}_t = \begin{pmatrix} \hat{\mathbf{v}}(\hat{\mathcal{C}}, \hat{\mathbf{q}}, t) \\ \hat{\mathbf{f}}(\hat{\mathcal{C}}, \hat{\mathbf{q}}, t) \end{pmatrix}, \quad \hat{z}_k = \begin{pmatrix} \hat{\mathcal{C}}(t_k) \\ \hat{\mathbf{q}}(t_k) \end{pmatrix},$$

where the hat denotes estimated quantities. Assuming  $\mathbf{h}_k = id$  (the identity map), this corresponds to a completely measurable state  $\mathbf{x} = [\mathcal{C}, \mathbf{q}]^T$ . The proposed continuous-discrete observer is formally

$$\begin{aligned} \begin{pmatrix} \hat{\mathcal{C}} \\ \hat{\mathbf{q}} \end{pmatrix}_t &= \begin{pmatrix} \hat{\mathbf{v}}(\hat{\mathcal{C}}, \hat{\mathbf{q}}, t) \\ \hat{\mathbf{f}}(\hat{\mathcal{C}}, \hat{\mathbf{q}}, t) \end{pmatrix}, \quad \hat{z}_k = \begin{pmatrix} \hat{\mathcal{C}}(t_k) \\ \hat{\mathbf{q}}(t_k) \end{pmatrix}, \\ \begin{pmatrix} \hat{\mathcal{C}}_k(+)) \\ \hat{\mathbf{q}}_k(+)) \end{pmatrix} &= \begin{pmatrix} \hat{\mathcal{C}}_k(-)) \\ \hat{\mathbf{q}}_k(-)) \end{pmatrix} +^* \begin{pmatrix} K_k^{\mathcal{C}} *^* (\mathcal{C}_k -^* \hat{\mathcal{C}}_k(-)) \\ \mathbf{K}_k^{\mathbf{q}} *^* (\mathbf{q}_k -^* \hat{\mathbf{q}}_k(-)) \end{pmatrix}, \end{aligned} \quad (1)$$

where  $(-)$  denotes the time just before a discrete measurement,  $(+)$  the time just after the measurement, and  $K_k^{\mathcal{C}}$  (a scalar) and  $\mathbf{K}_k^{\mathbf{q}}$  (a diagonal matrix of scalars) are the decoupled error correction gains for the curve position  $\mathcal{C}$  and the additional state quantities  $\mathbf{q}$  respectively. The operators  $-^*$ ,  $+^*$ ,  $*^*$  denote subtraction, addition, and multiplication of curves and of the additional state quantities propagated with the curves respectively. They are a crucial part of the proposed observer framework and are implemented implicitly in the error correction scheme described in Section VI. Figure 1 shows the overall observer structure as given in

Equation (1) and model assumptions associated with each component of the observer.

### III. CONTRIBUTIONS

We extend the geometric observer framework of Niethammer *et al.* [1] (given above) with image registration techniques such as optimal mass transport in place of the previous methods utilized for curve correspondence and velocity computation. These modifications lead to more accurate tracking performance and overall algorithmic simplification as well as real time implementations on GPU hardware. Specifically, the improvements can be divided into three areas and are described below:

#### A. Registration Warp Velocity Measurement

In the original model, classical optical flow (which in general gives incorrect velocity magnitudes) is utilized both as a motion prior and to measure curve velocity. We propose an image warp based on a brightness constancy assumption as a motion prior and curve velocity measurement. This leads to accurate estimation of velocity field magnitudes and orientations, thereby increasing performance and removing the need for advection of state during prediction.

#### B. Optimal Mass Transport Error Correction Scheme

In the original model, a non-trivial procedure was needed to carry out filtering of both curve and velocity information involving complex domain decomposition, the solution of the laplacian, and advectons. We solve this problem globally using optimal mass transport as a registration technique between curves. This results in a mapping between curves, removing the need for advection to transport information for error correction. It also alleviates any need for complex domain decomposition and leads to more accurate tracking performance.

#### C. Computational Efficiency.

We leverage the full multigrid PDE solution scheme on the graphics processing unit (GPU; now available on most consumer computers) to solve the PDEs employed in the geometric observer framework resulting in computation times that outperform implementations documented in the recent literature. This shows that the framework is capable of functioning as a real time vision system and that some PDE methods that were previously impractical for real time use are now reasonable to consider in such systems.

In what follows, we describe how the aforementioned contributions are integrated into the observer framework.

### IV. MEASUREMENTS

#### A. Velocity Measurements

Previously in [1], optical flow was used as the velocity measurement  $v$  corresponding to the curve velocity state within the geometric observer. In this strategy, a classical Horn & Schunck optical flow [4] is calculated, which estimates the 2D velocities of all image points from frame

to frame which are then used as the measurement of curve velocity. However, optical flow is in general inaccurate with respect to flow magnitude and Horn & Schunck optical flow can be inaccurate in terms of flow direction. While both of these facts can lead to poor performance, the former leads to the necessity of advection to transport information for curve prediction; without correct magnitudes information must be transported in a particular direction until it reaches its destination.

As a solution to these problems, we propose the use of a registration warp as a motion prior. Like optical flow, this warp is a velocity field over every point in the image domain, but has more accurate magnitudes which are suitable for directly mapping one image to another, making advection unnecessary and velocity measurements accurate.

Given two frames of imagery  $I_1$  and  $I_2$ , solving for the registration warp  $u : \mathbb{R}^2 \mapsto \mathbb{R}^2$  entails minimization of the following energy:

$$\mathcal{L}(I) = \iint (I_1(x + v(x)) - I_2(x))^2 dx$$

In other words, we wish to find a  $u$  that transforms  $I_1$  so as to match  $I_2$  as closely as possible in terms of squared error.

Such a quadratic minimization can be sufficiently dealt with via the Gauss-Newton method. Leaving derivations aside, the resulting algorithm for computing  $v$  is as follows:

- 1) Set  $v = 0$ .
- 2) Calculate Horn & Schunck optical flow by minimizing the following with respect to  $\hat{v}$ ,

$$\iint (I_1(x+v) - I_2(x) + \langle \Delta I_1, \hat{v} \rangle)^2 + \lambda \|\Delta \hat{v}\|^2 dx,$$

where  $\lambda$  is a constant determining the required smoothness of the resulting field.

- 3) Set  $v = v + \rho \hat{v}$ , where  $\rho$  is the predefined time step for the Gauss-Newton method.
- 4) If not yet converged, go to (2).

Previously, such minimization would be considered impractical for tracking due to the computational cost involved with step (2). However, this is not the case when computed using a full multigrid scheme on a small, parallel computing architecture such as the graphical processing unit which is now available in most consumer computers (see Section VII for details on the GPU implementation of the warping algorithm).

#### B. Curve Position Measurement

Static measurements of curve position provide information about the actual system and allow the model state to converge to the true system states. Motion models used in visual tracking cannot possibly be rich enough to capture the variability seen in deformable objects in natural video sequences. Therefore, it is important that the observer prediction be supplemented with measurements so the observer can adapt to real-life changes in the system.

The estimate model is a function of the curve position and dynamics as it evolves through time. The measurement

of the actual system, however, is independent of the model. Therefore, the static curve measurements can be obtained using any segmentation technique with no modification to the observer. Most segmentation methods are either based on local image information from an edge detector, or statistics of regions of image data defined by the curve. Additionally, shape priors can be learned and included in the segmentation algorithm to improve accuracy if detailed shape information is available a priori.

We create our static measurements in a two step process. First, background subtraction is used as a pre-processing step to help separate the target from background clutter. Second, the resulting image is segmented to isolate the target.

In order to obtain a smooth curve  $C$  that represents this segmentation, we employ the active contour technique proposed by Chan and Vese [6]. This method attempts to minimize the variance over the region inside the curve  $C$ ,  $C_{inside}$  and outside the curve,  $C_{outside}$ . This is accomplished by minimizing the following energy with respect to  $C$  where  $u$  and  $v$  represent the mean image intensity over  $C_{inside}$  and  $C_{outside}$  respectively:

$$\int_{C_{inside}} (I - u)^2 dA + \int_{C_{outside}} (I - v)^2 dA + \mu \int ds$$

, where  $\mu$  is a weighting factor on the smoothness of the curve. This energy is at a minimum when the interior and exterior are modeled best by  $u$  and  $v$ . The Chan-Vese segmentation technique is advantageous in that it is robust to noise since image data is integrated over large regions, it's global use of image information makes it robust to different initializations, and its simplicity allows for real time implementation.

## V. MOTION PRIORS

In [1], three major motion priors were defined: A static motion prior,

$$\hat{C}_t = \mathbf{0},$$

a quasi-dynamic optical flow prior,

$$\hat{C}_t = (\mathbf{v}_{OF} \cdot \mathcal{N})\mathcal{N},$$

and a dynamic elastic prior,

$$\mu \hat{C}_{tt} = \left( \frac{1}{2} \mu \|\hat{C}_t\|^2 + a \right) \kappa \mathcal{N} - (\nabla a \cdot \mathcal{N})\mathcal{N} - \frac{1}{2} \mu (\|\hat{C}_t\|)_s \mathcal{I}.$$

For our work, we utilize the latter, as it models the dynamics of the estimated curve  $\hat{C}$ .

## VI. ERROR CORRECTION

The observer framework proposed in this paper requires a methodology to associate a predicted curve state to a measured curve state. This amounts to establishing correspondences between points on the measured and the predicted curves. The correspondence map between measured and estimated curves should be diffeomorphic.

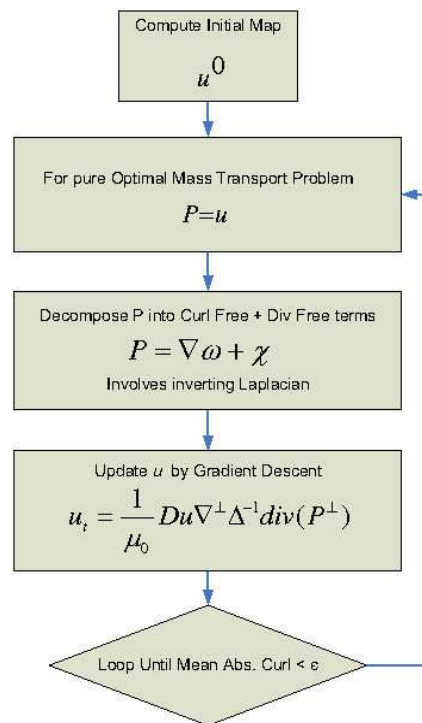


Fig. 2. Optimal Mass Transport Algorithm

### A. The Optimal Mass Transport Problem

Motivated by our earlier work [2], [3] one-to-one correspondences between curves can be established using the Optimal Mass Transport technique. The optimal mass transport problem was first formulated by a French engineer Gaspier Monge in 1781, and was given a modern formulation in the work of Kantorovich [7] and, therefore, is now known as the Monge-Kantorovich problem. The original problem concerned finding the optimal way to move a pile of soil from one site to another in the sense of minimal transportation cost. Hence, the Kantorovich-Wasserstein distance is also commonly referred to as the Earth Mover's Distance (EMD).

1) *Formulation of the Problem:* We will briefly provide an introduction to modern formulation of the Monge-Kantorovich problem. We assume we are given, a priori, two sub-domains  $\Omega_0$  and  $\Omega_1$  of  $R^d$  with smooth boundaries, and a pair of positive density functions,  $\mu_0$  and  $\mu_1$  defined on  $\Omega_0$  and  $\Omega_1$  respectively. We assume that,

$$\int_{\Omega_0} \mu_0 = \int_{\Omega_1} \mu_1 \quad (2)$$

This ensures that we have the same total mass in both the domains. We now consider *diffeomorphisms*  $\tilde{u}$  from  $\Omega_0$  to  $\Omega_1$  which map one density to other in the sense that,

$$\mu_0 = |D\tilde{u}| \mu_1 \circ \tilde{u} \quad (3)$$

which we call the mass preservation (MP) property, and write  $\tilde{u} \in MP$ . Equation (3) is called the Jacobian equation. Here,  $|D\tilde{u}|$  denotes the determinant of the Jacobian map  $D\tilde{u}$ , and  $\circ$  denotes composition of functions. It basically implies that

if a small region in  $\Omega_0$  is mapped to a larger region in  $\Omega_1$ , then there must be a corresponding decrease in density in order for the mass to be preserved. There may be many such mappings, and we want to pick an optimal one in some sense. Accordingly, we define the squared  $L^2$  Monge-Kantorovich distance as following:

$$d_2^2(\mu_0, \mu_1) = \inf_{\tilde{u} \in MP} \int \|\tilde{u}(x) - x\|^2 \mu_0(x) dx \quad (4)$$

The *optimal MP map* is a map which minimizes this integral while satisfying the constraint given by Equation (3). The Monge-Kantorovich functional, Equation (4), is seen to place a penalty on the distance the map  $\tilde{u}$  moves each bit of material, weighted by the material's mass. A fundamental theoretical result [8], [9], is that there is a unique optimal  $\tilde{u} \in MP$  transporting  $\mu_0$  to  $\mu_1$ , and that  $\tilde{u}$  is characterized as the gradient of a convex function  $\omega$ , i.e.,  $\tilde{u} = \nabla\omega$ . This theory translates into a practical advantage, since it means that there are no non-global minima to stall our solution process.

2) *Computing the Transport Map*: We will describe here only the algorithm for finding the optimal mapping  $\tilde{u}$ . The details of this method can be found in [10]. The basic idea for finding the optimal warping function is first to find an initial MP mapping  $u^0$  and update it iteratively to decrease an energy functional. When the pseudo time  $t$  goes to  $\infty$ , the optimal  $u$  will be found, which is  $\tilde{u}$ . Basically there are two steps. The first step in this algorithm is to find an initial mass preserving mapping. This can be done for general domains using the method of Moser [11] or the algorithm proposed in [10]. The latter method can simply be interpreted as the solution of a one-dimensional Monge-Kantorovich problem in the x-direction followed by the solution of a family of one-dimensional Monge-Kantorovich problems in y-direction. The second step is to adjust the initial mapping found above iteratively using gradient descent in order to minimize the functional defined in Equation (4), while constraining  $u$  so that it continues to satisfy Equation (3). This process iteratively removes the curl from the initial mapping  $u$  and, thereby, finds the polar factorization of  $u$ . For details on this technique, please refer to [10]. The overall algorithm is summarized graphically in Figure 2.

### B. Curve Interpolation

Optimal mass transport can be employed to find correspondences between curves in the following way. Let a curve  $C$  be represented as a binary mask as follows:

$$C_{binary}(x) = \begin{cases} 1, & x \in C_{inside} \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

Given two curves,  $C^1$  and  $C^2$ , a correspondence map can be calculated between these curves by finding a mass-preserving mapping (via the algorithm described above) between their implicit representations  $C_{binary}^1$  and  $C_{binary}^2$ .

The resulting mapping  $\tilde{u}^*$  provides a one-to-one correspondence from the implicit representation  $C_{binary}^1$  to  $C_{binary}^2$ . However, for the purposes of filtering, an intermediate

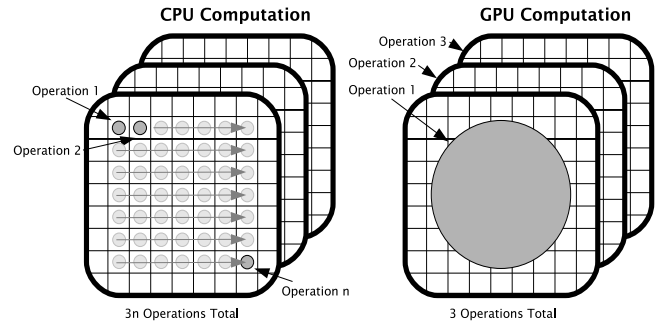


Fig. 3. **CPU versus GPU solutions of PDEs.** While the CPU computes updates on data grids one element at a time, the GPU is capable of updating entire grids in one pass due to their massively parallel architecture.

mapping between curves is required. Such an intermediate mapping is easily recovered through the following warping map  $u(x, w)$ ,

$$u^*(x, w) = x + w(\tilde{u}^*(x) - x), \quad (6)$$

where  $w \in [0, 1]$  the interpolation parameter such that  $u^*(x, w = 0)$  is the identity map and  $u^*(x, w = 1)$  is the original  $\tilde{u}^*$  which maps all the way from  $C_{binary}^1$  to  $C_{binary}^2$ .

### C. Error Correction

The error correction scheme builds on the results of subsections VI-A and VI-B. The observer structure dictates that error correction (the combination of measurement and estimate) for the state corresponding to curve position be computed as:

$$\hat{C}_k(+) = \hat{C}_k(-) + * K_k^C * (C_k - * \hat{C}_k(-)),$$

which amounts to curve interpolation via the method from subsection VI-B where  $w = K_k^C$ .

For the error correction of additional state quantities, state information needs to be exchanged and compared between the measured and the predicted curves i.e. moved to appropriate positions in the image domain which correspond to the filtered curve  $\hat{C}_k(+)$  to facilitate a pointwise linear combination of values for filtering. This information exchange is performed, again, via the method from subsection VI-B with  $w = K_k^C$  to transport velocity measurements  $q_i$  &  $\hat{q}_i(-)$  to the newly estimated (interpolated) curve  $\hat{C}_k(+)$ ,

$$\begin{aligned} p_i &= q_i(u^*(x, K_k^C)) \\ \hat{p}_i(-) &= \hat{q}_i(-)(u_{inv}^*(x, 1 - K_k^C)), \end{aligned}$$

where  $u_{inv}^*$  is the inverse map corresponding to  $u^*$  and  $p_i$  &  $\hat{p}_i(-)$  are the velocities on the measured and predicted curves, respectively, transported appropriately to the corrected curve. We can now perform pointwise filtering on the velocity states via,

$$\hat{q}_i(+) = \hat{p}_i(-) + * K_k^q * (p_i - * \hat{p}_i(-)),$$

where  $\hat{q}_i(+)$  is the new velocity state estimate.

## VII. GPU IMPLEMENTATION

In this paper, the basic building blocks of all the given algorithms (optical flow for image warping, optimal mass transport, and active contour evolution) are the solutions of PDEs. Typically, such problems are computationally intensive and prohibitive for real time tracking implementations. However, in the past few years, it has been shown that graphical processing units (GPUs), which are now standard in most consumer computers and are normally applied to graphics processing for gaming, are particularly suited for several types of parallelizable problems including the solution of PDEs [12], [13]. In our work, we implemented all PDE solvers on the GPU utilizing the full multigrid method, making two algorithms practical for tracking which were not practical before: optical flow for image warping and optimal mass transport. In this section, we give details on the GPU, our implementation, and the computational advantages achieved.

The GPU can be considered a massively parallel coprocessor and dedicated memory interfacing to the CPU over a standard bus. Modern GPUs are comprised of up to 128 symmetric processing units running up to speeds of 1.35Ghz. Their advantage over the CPU in this sense is that while the CPU can execute only one or two threads of computation at a time, the GPU can execute over two orders of magnitude more. From a PDE perspective, while the CPU computes updates on data grids one element at a time, the GPU computes updates on *entire grids* at a time (Figure 3).

Both the optical flow and optimal mass transport algorithms were implemented on the GPU using a full multigrid solver. In this method, a fine-to-coarse hierarchy of equation systems with excellent error reduction properties [3] is created and the PDEs are solved using classical iterative methods at different resolution grids. The principle grid operations are grid interpolation, relaxation, and restriction and unlike the CPU, the GPU computes each update in one render pass, increasing speed dramatically.

On a modest Dual Xeon 1.6Ghz machine with an nVidia GeForce 8800 GX GPU (3DMark score of 7200) optical flow speeds of 153 FPS were achieved on imagery of size  $128^2$ . This is in contrast to the optical flow framerate of 97 FPS achieved on specialized FPGA hardware recently reported in [14] on comparably sized imagery. Note that the former implementation will run on most modern consumer level PC's without modification.

On the same machine, similar gains in speed were observed for the optimal mass transport algorithm. On a  $512^2$  grid, 100 iterations of the latest OMT solver required 15.25 seconds (full multigrid in C) while the GPU version required 1.59 seconds representing almost an order of magnitude speed improvement. This gap continues to increase with grid size (Figure 4).

These results show that certain algorithms involving PDEs which would be considered impractical for use in real time control are now reasonable to consider in the design of vision systems. Additionally, these results show that the scheme

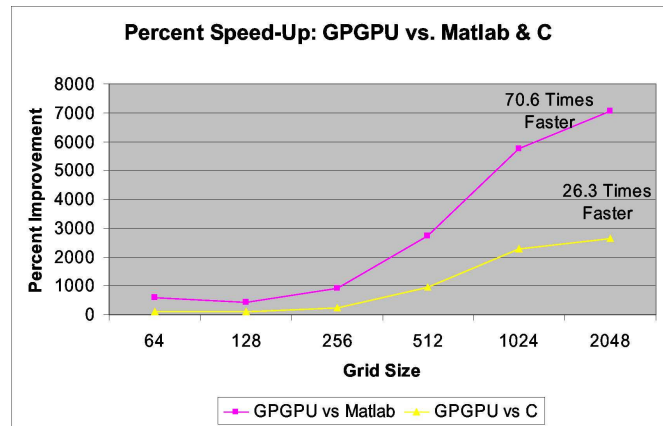


Fig. 4. The GPU realizes an increasing advantage in speed over the CPU as grid size increases for the optimal mass transport solver with final performance ratios reaching well beyond an order of magnitude.

presented in this paper is capable of real time performance.

## VIII. RESULTS

Figures 5 and 6 show tracking results from both the tracker with proposed extensions and the original tracker on a single blob and a walking person, respectively. The error correction gains  $K_k^c$  for the curve position and  $K_k^q$  for the curve's normal velocity were both 0.8 for the blob tracking sequence and 0.3 and 0.5, respectively, for the walking person sequence. Initial conditions (the bold solid curves) were chosen far from the initial measurement curves (dash-dotted curves).

In the blob tracking case, both trackers converge to the correct solution and handle the topological change correctly. This shows that the addition of the optimal mass transport for curve interpolation is capable of handling the same changes in shape the original interpolation scheme could without the need for domain decomposition. In the walking person sequence, the proposed tracker is capable of handling complex shape changes such as those brought on by the moving legs and arms while the original tracker struggles with this difficulty and never recovers.

## IX. CONCLUSIONS AND FUTURE WORKS

### A. Conclusions

This paper proposes an extension to the work of Niethammer *et al.* [1] by introducing novel image registration techniques as a velocity measurement and optimal mass transport as a curve interpolation scheme. These additions lead to more accurate tracking performance as well as algorithmic simplifications in the framework. Additionally, we show that the framework and related PDE techniques for image analysis are practical for real time vision applications without the requirement of specialized hardware on consumer level graphical processing units. In fact, we show that the speeds achieved on the GPU exceed those reported in recent literature.

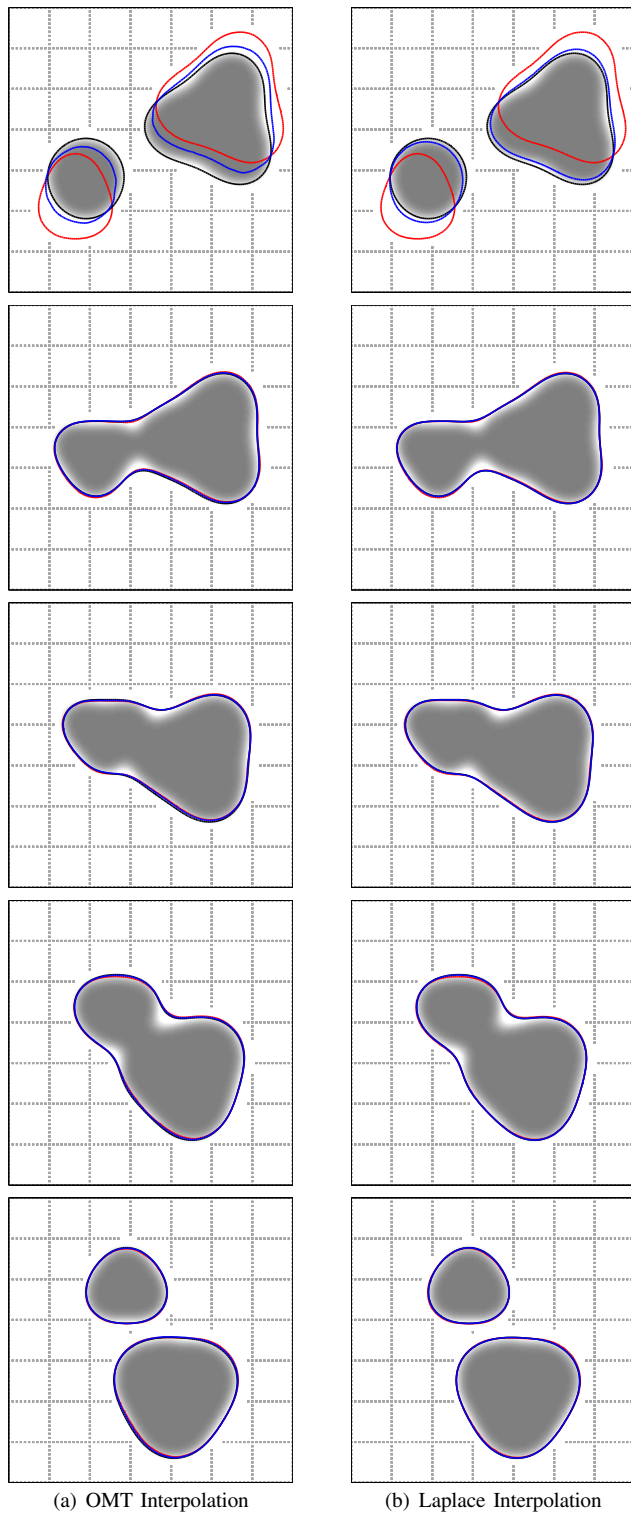


Fig. 5. Comparison of tracking results between the OMT based tracker and the Laplace based tracker on a blob sequence with topological changes. The black curve is the measurement, the red curve is the prediction, and the blue curve is the estimate. Note that both trackers have comparable performance in this simplest of cases showing that our improvements yield no decrease in algorithm capabilities versus the original.

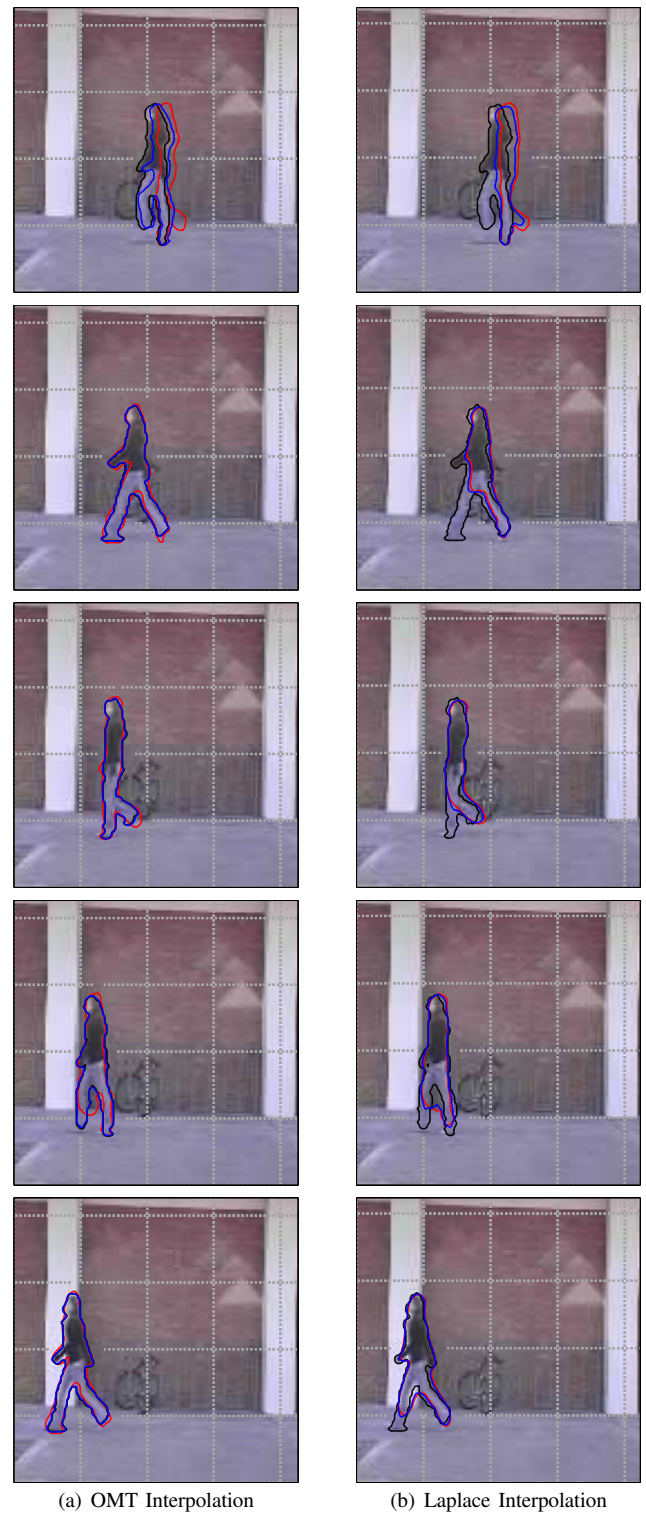


Fig. 6. Comparison of tracking results between the OMT based tracker and the Laplace based tracker on a walking person. The black curve is the measurement, the red curve is the prediction, and the blue curve is the estimate. Note that (top) the proposed extensions are capable of handling the complex shape change due to the moving legs and arm while the original algorithm is not able to recover from this initial difficulty.

### B. Future Works

The measurement model applied in this work is a combination of simple background subtraction and Chan-Vese segmentation. Through the utilization of more robust, layered measurement models such as those in our work [15] the applicability of this framework can be expanded now that it is capable of real time performance. Additionally, now that the computational load of the observer framework has been reduced, it is possible to incorporate a particle filter for non-deterministic estimation.

Additionally, in this work, the interpolation parameter  $\omega$  was manually set and kept static. An interesting avenue of research would be the investigation of statistical methods for the adaptive tuning of this parameter.

### REFERENCES

- [1] M. Niethammer, P. A. Vela, and A. Tannenbaum, "Geometric observers for dynamically evolving curves," in *Proceedings of the IEEE Conference on Decision and Control, and the European Control Conference*, 2005, pp. 6071–6077.
- [2] T. ur Rehman and A. Tannenbaum, "Multigrid optimal mass transport for image registration and morphing," in *Proceedings of SPIE Conference on Computational Imaging V*, vol. 6498, 2007.
- [3] T. ur Rehman, G. Pryor, and A. Tannenbaum, "GPU enhanced multigrid optimal mass transport for image registration and morphing," 2007, IEEE International Conference on Image Processing: In submission.
- [4] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.
- [5] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, ser. Applied Mathematical Sciences. Springer-Verlag, 2003, vol. 153.
- [6] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.
- [7] L. V. Kantorovich, "On a problem of monge," *Uspekhi Matematicheskikh Nauk.*, vol. 3, pp. 225–226, 1948.
- [8] Y. Brenier, "Polar factorization and monotone rearrangement of vector-valued functions," *Communications on Pure and Applied Mathematics*, vol. 64, pp. 375–417, 1991.
- [9] W. Gangbo and R. McCann, "The geometry of optimal transportation," *Acta Mathematica*, vol. 177, pp. 113–161, 1996.
- [10] S. Haker, L. Zhu, A. Tannenbaum, and S. Angenent, "Optimal mass transport for registration and warping," *International Journal of Computer Vision*, vol. 60, no. 3, pp. 225–240, 2004.
- [11] J. Moser, "On the volume elements on a manifold," *Transactions of the American Mathematical Society*, vol. 120, pp. 286–294, 1965.
- [12] J. Bolz and et al., "Sparse matrix solvers on the GPU: Conjugate gradients and multigrid," in *Proceedings of SIGGRAPH*, vol. 22, 2003, pp. 917–924.
- [13] G. Nolan and et al., "A multigrid solver for boundary value problems using programmable graphics hardware," in *Proceedings of SIGGRAPH*, 2003, pp. 102–111.
- [14] J. Diaz, E. Ross, F. Pelayo, E. Ortigosa, and S. Mota, "Fpga-based real time optical flow system," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, pp. 274–279, 2006.
- [15] G. Pryor, P. Vela, J. Rehman, and A. Tannenbaum, "Layered active contours for tracking," 2007, IEEE Conference on Computer Vision and Pattern Recognition: In submission.