

Image Deformation using Velocity Fields: An Exact Solution

Jeff Orchard

University of Waterloo, Waterloo Ontario N2L 3G1, Canada,
jorchard@cs.uwaterloo.ca

Abstract. In image deformation, one of the challenges is to produce a deformation that preserves image topology. Such deformations are called “homeomorphic”. One method of producing homeomorphic deformations is to move the pixels according to a continuous velocity field defined over the image. The pixels flow along solution curves. Finding the pixel trajectories requires solving a system of differential equations (DEs). Until now, the only known way to accomplish this is to solve the system approximately using numerical time-stepping schemes. However, inaccuracies in the numerical solution can still result in non-homeomorphic deformations. This paper introduces a method of solving the system of DEs exactly over a triangular partition of the image. The results show that the exact method produces homeomorphic deformations in scenarios where the numerical methods fail.

1 Introduction

Digital images can be “warped” or deformed using a non-rigid deformation. Each pixel can have its own displacement vector indicating where it moves to. There are different methods of determining these displacement vectors. One family of methods, which includes elastic deformation [1] and thin plate splines [2], uses a set of image control points that are moved, and the pixels in the image move with them as if attached to a rubber sheet. These methods work well for small control-point displacements, but large displacements can lead to non-homeomorphic deformations [3]. That is, the spatial transformation may not conserve local topology and therefore not be invertible. In applications such as medical imaging and remote sensing, a non-homeomorphic deformation is not desirable.

Deforming an image usually proceeds by traversing the pixels of the **deformed** image, determining a colour (or intensity) for each pixel. If one knows where the pixel was moved from in the original image, the new pixel’s colour can be found by interpolating the original image at that location. This process requires the inverse of the deformation. For this reason, one usually stores the inverse of the desired deformation. The deformation map consists of a displacement vector for each pixel. In this context, the vector indicates where the pixel was displaced from.

There is no efficient way to find the inverse of a deformation defined using control-point displacement maps. Simply negating the displacement vectors does not suffice.

Instead of defining the deformation using control points, the pixels can be moved by flowing along a velocity field [4]. A continuous velocity field is defined over the image, typically by interpolating between a finite set of velocity vectors assigned in the image. These methods are often called “flow methods”. The velocity fields can be derived in conjunction with other constraints such as smoothness and incompressibility [5–7].

However, given a velocity field, the operation of deriving the pixel trajectories can be challenging, and amounts to solving a system of differential equations (DEs) for each pixel. How easy the system is to solve depends on the type of DEs. Some systems are solvable, while others have no known closed-form solution (i.e. we cannot write down the solution in terms of combinations of transcendental functions). For such situations, we can approximate the solution using numerical time-stepping schemes [4]. This, too, has problems. The solution is only an approximation and can contain inaccuracies.

In this paper, we show that by performing linear interpolation on triangular cells in the image, the resulting system of DEs is linear and can be solved analytically. Thus, we derive homeomorphic deformations while avoiding the inaccuracies of numerical approximate solutions.

2 Methods

2.1 Velocity Fields

Let $\mathbf{f} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ be a continuous function that represents the velocity field over a 2D image. Hence, $\mathbf{f}(x, y)$ is the velocity vector at location (x, y) . We can model deformations as the movement of pixels as they follow this velocity field. Determining the path of a pixel is then a matter of solving the system of DEs

$$\begin{bmatrix} \frac{dx(t)}{dt} \\ \frac{dy(t)}{dt} \end{bmatrix} = \mathbf{f}(x(t), y(t)) , \quad (1)$$

where $\mathbf{f}(x(t), y(t))$ is a vector-valued velocity function. Note the introduction of the parameter t . For a particular deformation, the system is integrated (backward or forward in time) to a predetermined stop time. The final resting point of each pixel is where that pixel is displaced to in the deformation.

A velocity field is better than a control-point displacement map because a velocity field is easily inverted by simply reversing the velocity vectors. A velocity field also offers continuous deformations so that intermediate deformations can be attained.

If velocity vectors are defined in our image (for example, at a subset of pixel locations), we can define a continuous velocity field over the domain of the image by interpolating between these velocity vectors. Bilinear interpolation can

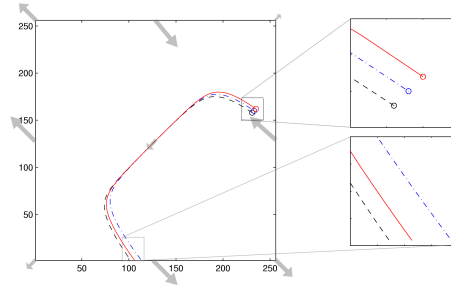


Fig. 1. Crossing pixel trajectories resulting from using a numerical solver. The gray arrows represent the velocity vectors at the pixel locations where they are specified. The velocity field is continuously interpolated between these vectors.

be used to interpolate between velocity vectors defined on a rectangular grid. In this case, the system of DEs that governs the motion of pixels in the image is

$$\begin{cases} x'(t) = Ax(t) + By(t) + Cx(t)y(t) + D \\ y'(t) = Ex(t) + Fy(t) + Gx(t)y(t) + H \end{cases}, \quad (2)$$

where $x'(t)$ and $y'(t)$ are time derivatives, and A, \dots, H are constants based on nearby velocity vectors, \mathbf{f} . This system is the same as the interacting species model, and is non-linear because of the “ $x(t)y(t)$ ” corss-terms. There is no known closed-form general solution to (2).

An alternative is to find an approximate solution to the system of DEs using a numerical method. That is, the movement of each pixel through the velocity field is estimated in discrete time steps using a numerical scheme. However, depending on the characteristics of the velocity field, this can be problematic. Figure 1 shows three numerical solution trajectories that cross each other. The underlying velocity field is continuous, so the solution curves should not cross. Hence, the error introduced by the time-stepping method changed the topology of the solutions.

Consider performing linear interpolation on triangular cells instead of rectangular cells. Since each of the x - and y -components of the velocity field can be uniquely represented as a linear function in a triangular cell, the resulting system of DEs is linear,

$$\begin{cases} x'(t) = Ax(t) + By(t) + D \\ y'(t) = Ex(t) + Fy(t) + H \end{cases}. \quad (3)$$

Linear systems of DEs are easy to solve [8], so the movement of the pixels through the velocity field can be known exactly, and written as a combination of transcendental functions.

2.2 Implementation

For a given image and velocity field, the deformation proceeds by first propagating each pixel through the velocity field. The propagation of the trajectories runs from an initial time t_0 until a final time t_f . During that time, each pixel flows with the velocity field to another location (unless it happens to rest on an equilibrium point). The displacement map for the deformation is simply a record of where each pixel rests at time t_f .

A system has been built in MATLAB (MathWorks Inc., Natick, Massachusetts) to derive image deformations by finding the exact solution of pixel trajectories. The image is broken into triangles, the vertices of which are nine pixels from the image¹. The following is an outline of the methodology for finding the exact trajectory of a pixel through a triangular cell.

Suppose we are starting from the point (x_i, y_i) at time t_i in triangle T .

1. The three velocity vectors assigned to the vertices of T are used to define two linear functions over T : one for the velocity's x -component, and one for its y -component.
2. Compute the coefficients for the triangle's system of DEs using the linear functions from step 1.
3. Find and store the general solution for the system of DEs. This involves finding the (generalized) eigenvalues and eigenvectors of the system matrix [8]. This general solution still has two unknown constants of integration.
4. Use the initial data (initial point (x_i, y_i) and initial time t_i) to determine values for the unknown constants.
5. Two cases remain: either the trajectory stays inside triangle T for the remainder of the propagation, or the trajectory intersects one of the triangle's edges at some time $t_{\text{exit}} < t_f$.
 - (a) If the trajectory remains inside T , simply evaluate the solution at $t = t_f$.
 - (b) If the trajectory leaves T , find the exit time t_{exit} and evaluate the exit location $(x_{\text{exit}}, y_{\text{exit}})$. Use this exit point as a new initial point. Find out which triangle the trajectory is entering, and go back to step 1 until the final time has been reached.

For the sake of brevity, many of the details are omitted here. It is worth noting that finding the time at which the solution curve intersects one of the triangle's edges results in a nonlinear algebraic equation in t that is not generally solvable. However, fast and accurate solvers exist for such algebraic equations. We use MATLAB's `fzero` function, in conjunction with multiple samples of the trajectory, to find the exit time if it exists.

We also implemented a numerical method for propagating the solution through the triangular cells. The method is written as a MATLAB script. Its velocity function is derived from the same triangulation as the exact method above, but instead of seeking an exact solution, it uses MATLAB's `ode45` function, an implementation of a 4th/5th-order Runge-Kutta time-stepping method [9].

¹ Our implementation is not limited to pixel locations for triangle vertices. The vertices can occur at any location.

2.3 Experiments

The exact method and the numerical method are compared through a series of experiments. For testing purposes, a 256×256 image is partitioned into a triangle mesh of eight triangles. Each triangle is an isosceles right-triangle, measuring 128 pixels on the short sides (see Fig. 2). Velocity vectors are defined at the nine vertices of the triangle mesh.

In each experiment, three different sets of results are produced for the numerical method, each using a different maximum time step. In MATLAB's numerical DE solvers, the user can specify the maximum time step. In our experiments, we derived three test values based on the Courant-Friedrichs-Levy (CFL) condition [10]. We took the CFL condition to mean that the time step should be short enough so that a single step, taken at the maximum velocity in the velocity field, cannot be larger than 128 pixels (the width of a triangle). We will refer to this maximum time step as the CFL step. In our experiments, the tested maximum time-step sizes are 60%, 70% and 80% of the CFL step size.

In the first experiment, we compare the trajectories of three pixels through a challenging velocity field that makes the trajectories converge toward each other, and then diverge again. The purpose is to see if changing the step-size constraint changes the solution significantly, and to see if the solution trajectories cross over each other.

In the second experiment, the numerical and exact methods are each used to produce a displacement map. Each displacement map has two components: an x -component and a y -component. For each pixel, the corresponding value in the x -component gives the x -coordinate of the location where that pixel came from. The y -component is defined similarly.

3 Results

Figures 2(a)-(c) contrast several pixel trajectories for the numerical method under different maximum time-step sizes. Notice that, despite the same underlying velocity field, the numerical methods give qualitatively different solutions for different step-size constraints. In particular, the order of the trajectory curves is not preserved in (a) and (c). Solution curves cannot cross over each other when the velocity field is continuous. Even the strictest step-size solution (Fig. 2(a)) gives solutions that are qualitatively different from those of the exact method.

The displacement maps for the exact and numerical methods are shown in Fig. 3. The figure shows the x -component for the displacement maps corresponding to the numerical method with max time-step set to 60% of the CFL step, and the exact method. The y -components are not shown here, but exhibit similar results. Since the underlying velocity field is continuous, we expect the displacement map to be smooth. However, notice the banding in the displacement map for the numerical method. The artifact is not present in the displacement map for the exact method.

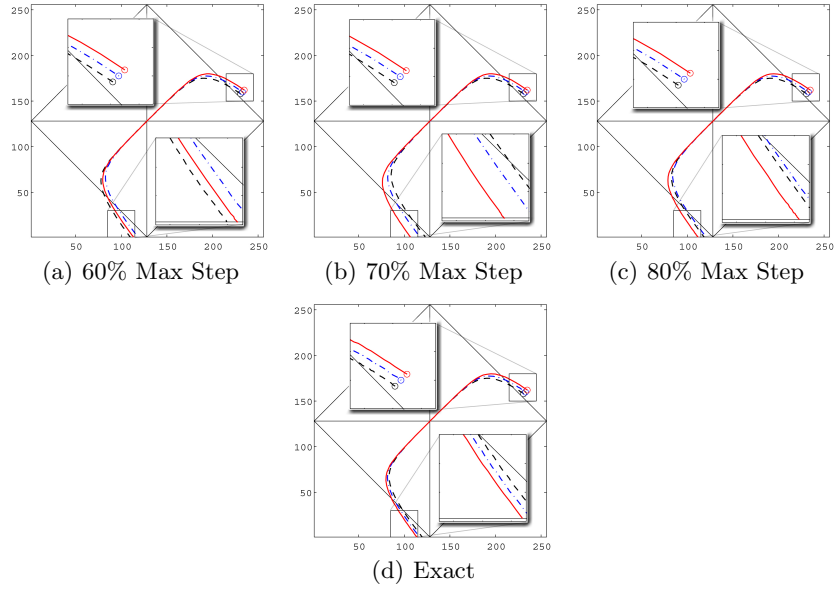


Fig. 2. Pixel trajectories for numerical and exact solution methods. Each trajectory starts at the circle marker. The insets magnify the starting and ending of each trajectory.

4 Discussion and Conclusions

The velocity fields chosen for these experiments are somewhat pathological, specifically designed to exhibit, in an obvious way, the sort of errors that the numerical methods can produce. However, such errors can occur in more subtle scenarios. The examples in this paper illustrate that simply taking a shorter time step does not necessarily lead to a qualitatively correct solution.

The banding observed in Fig. 3 seems to be related to the maximum step size because the other numerical results showed similar bands of different widths.

The “exact” method is not truly exact since the point where the trajectories exit one triangle (and enter the next) have to be located numerically. However, finding the solution of a single algebraic equation is much easier than numerically solving a system of DEs. The algebraic equation gives concrete feedback about the error in the solution. The feedback available to numerical DE solvers is far more nebulous.

Methods exist to remove redundant triangles from a triangulation [11]. If a region of the velocity field is highly linear, it can be modeled by a single triangle. Removing redundant triangles can help speed up the processing time of these methods.

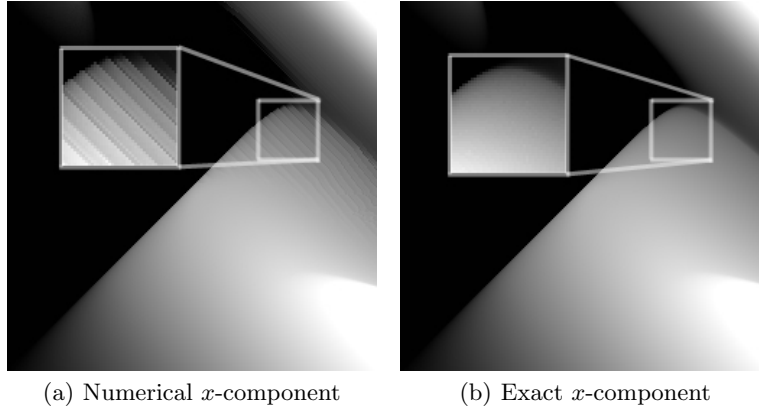


Fig. 3. Displacement maps for the numerical method (with maximum time step set to 60% of the CFL step size), and for the exact method. The inset shows a contrast-enhanced magnification of part of the map.

The deformation methods outlined in this paper are for 2D images. However, they can easily be extended to higher dimensions. Higher-order linear systems of DEs are still quite easy to solve exactly.

A 3D application for this type of exact solution method is in the field of magnetic resonance (MR) diffusion tensor tractography. In this field, information about the preferential direction of diffusion of water in the brain is collected using an MR scanner. Since water in a nerve fiber bundle preferentially diffuses along the fiber’s axis, following these diffusion vectors can give doctors information about long-range nerve connections in the brain. This technique, known as “diffusion tensor tractography”, is similar in nature to following the velocity vectors in image deformation. Current methods use discrete time-stepping techniques to trace the nerve fiber bundles through a 3D dataset [12–14]. Often, bundles converge with other bundles, and then separate again. Numerical methods to approximate the fiber tracts are subject to the same inaccuracy issues as the numerical methods studied in this paper [12]. Although the spatial resolution of a diffusion tensor image is somewhat coarse compared to the size of the nerve fiber bundles, an exact solution might yield more consistent and possibly more accurate results than the numerical methods.

Finally, more research needs to be done to investigate other methods of interpolation to see if the resulting system of DEs can be solved exactly. So far we have briefly considered Fourier basis functions and radial basis functions, but believe that neither option leads to a solvable system.

Acknowledgment

We would like to thank the Natural Science and Engineering Research Council (NSERC) of Canada for their financial support.

References

1. Bajcsy, R., Kocacic, S.: Multiresolution elastic matching. *Computer Vision, Graphics and Image Processing* **46** (1989) 1–12
2. Bookstein, F.L.: Principal warps: thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **11** (1989) 567–585
3. Christensen, G.E.: Deformable shape models for anatomy. PhD thesis, Washington University, St. Louis, Missouri (1994)
4. Christensen, G.E., Rabbit, R.D., Mill, M.I.: Deformable templates using large deformation kinematics. *IEEE Transactions on Image Processing* **5** (1996) 1435–1447
5. Beg, M.F., Miller, M., Trounev'e, A., Younes, L.: Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *Journal of Computer Vision* **61** (2005) 139–157
6. Christensen, G.E., Joshi, S.C., Miller, M.I.: Volumetric transformation of brain anatomy. *IEEE Transactions on Medical Imaging* **16** (1997) 864–877
7. Joshi, S.C., Miller, M.I.: Landmark matching via large deformation diffeomorphisms. *IEEE Transactions on Image Processing* **95** (2000) 1357–1370
8. Nagle, R.K., Saff, E.B.: *Fundamentals of Differential Equations*. Second edn. Benjamin Cummings (1989)
9. Burden, R.L., Faires, J.D.: *Numerical Analysis*. Fourth edn. PWS-Kent (1989)
10. Haberman, R.: *Elementary Applied Partial Differential Equations*. Prentice Hall (1987)
11. Schroeder, W.J., Zarge, J.A., Lorensen, W.E.: Decimation of triangle meshes. *Computer Graphics* **26** (1992) 65–70
12. Bassar, P.J., Pajevic, S., Pierpaoli, C., Duda, J., Aldroubi, A.: In vivo fiber tractography using DT-MRI data. *Magnetic Resonance in Medicine* **44** (2000) 625–632
13. Pajevic, S., Aldroubi, A., Bassar, P.J.: A continuous tensor field approximation of discrete DT-MRI data for extracting microstructural and architectural features of tissue. *Journal of Magnetic Resonance* **154** (2002) 85–100
14. Poupon, C., Clark, C.A., Frouin, V., Regis, J., Bloch, I., Bihan, D.L., Mangin, J.F.: Regularization of diffusion-based direction maps for the tracking of brain white matter fascicles. *NeuroImage* **12** (2000) 184–195