Image Registration under Varying Illumination: Hyper-Demons Algorithm

Mehran Ebrahimi and Anne L. Martel

Department of Medical Biophysics, University of Toronto
Imaging Research, Sunnybrook Health Sciences Centre
Toronto, Ontario, Canada
mehran.ebrahimi@sri.utoronto.ca, anne.martel@sri.utoronto.ca

Abstract. The goal of this paper is to present a novel recipe for deformable image registration under varying illumination, as a natural extension of the demons algorithm. This generalization is derived directly from the optical-flow constraints in a variational formulation. Furthermore, our approach provides a new mathematical interpretation of the demons algorithm via fixed-point iterations in a consistent framework.

1 Introduction

Since the appearance of Thirion's work [13,14], the so-called *demons method* has become a popular deformable registration technique in medical imaging. A reason for this success may be attributed to the simplicity, speed, and performance of Thirion's proposed technique. Over the years, the demons method has been extended by several authors [6,2,15,16] who have proposed different variations.

Although Thirion's seminal work [13,14] intuitively borrows physical ideas of thermodynamics, deeper mathematical insight of these ideas has been introduced in [10,11] which focuses on understanding the algorithm.

In this paper, we shall provide a novel mathematical interpretation of the demons algorithm, originating directly from the optical flow constraints [7,5,17]. The optical flow equations [7] inspired Thirion to introduce his so-called *forces* in his proposed scheme. Furthermore, to the best of our knowledge, none of these existing extensions of the demons algorithm focuses directly on the varying illumination problems introduced by changes in image intensity such as those due to contrast enhancement or to changes in coil sensitivity between images. Our formulation shall address this varying illumination problem.

To proceed, we need to rigorously introduce the required background material which we will be employed in our formulation.

In Section 2, we will introduce an intensity-based deformation model. For consistency, we adapt our notation from [10] and shall mention our distinctive differences when necessary. In Section 3, we briefly cover the background on optical flow constraints and associated methods, including the demons algorithm. Section 4 is dedicated to developing our extension. Finally, we will present various computational experiments and concluding remarks in Section 5.

D. Cremers et al. (Eds.): EMMCVPR 2009, LNCS 5681, pp. 303–316, 2009.

2 An Intensity-Based Deformation Model

An *intensity-based deformation process* is modeled via a real-valued multivariate intensity function $E\big(X(x,t),t\big)$ in which $t \in [0,1]$ is an artificial time-step, $x = (x_1, x_2, \ldots, x_d)$ represents the initial coordinate of a particle in the image domain $\Omega \subset \mathbb{R}^d$, and the function

$$X : \Omega \times [0,1] \to \mathbb{R}^d, \quad X = (X_1, X_2, \ldots, X_d)$$

defines the path of the particle [10]. Based on the fact that the coordinate of the particle at time $t = 0$ is x, we write $X(x,0) = x$. We also assume that the coordinate of the particle at time $t = 1$ is defined by the transformation $\phi(x)$ meaning that $X(x,1) = \phi(x)$. The known measurements of such deformation process are the target image $f(x)$ and the source image $g(x)$ defined in $L^2(\Omega)$, under the two main assumptions

$$E\mid_{t=0} = f(X(x,0)) = f(x),$$

$$E\mid_{t=1} = g(X(x,1)) = g \circ \phi(x).$$

[We have slightly modified the model presented in [10] (pp. 159) that assumes $E\mid_{t=1} = g(x)$.] We are now able to define deformations both at a time instance and over a time interval. More formally, we define a time dependent instantaneous deformation vector

$$\frac{dX}{dt} = (u_1, u_2, \ldots, u_d) = u$$

and the total displacement vector by

$$U(x) = X(x,1) - X(x,0) = \phi(x) - x. \tag{1}$$

Equation (1) yields $\phi(x) = U(x) + x$. Furthermore, by the fundamental theorem of calculus we can write

$$\int_0^1 [\frac{dE}{dt}]dt = E\mid_{t=1} - E\mid_{t=0} = g \circ \phi - f, \tag{2}$$

$$\int_0^1 u\ dt = \int_0^1 [\frac{dX}{dt}]dt = X(x,1) - X(x,0) = U(x). \tag{3}$$

These equations will be used in the following sections.

3 Optical Flow Constraints and Associated Methods

3.1 Conserved Intensity

In the simplest case, the optical flow equation (see for example Horn-Schunck [7]) relies on the conservation (or constancy) of intensity at a time instance t

and a spatial location x, i.e., $\frac{dE}{dt} = 0$. Hence,

$$\begin{aligned} \frac{dE}{dt} = \frac{dE(X(x,t),t)}{dt} &= E_t + \sum_{i=1}^{d} E_{X_i} \frac{dX_i}{dt} \\ &= E_t + \sum_{i=1}^{d} u_i E_{X_i} = 0. \end{aligned}$$

This equation contains d unknowns $\{u_i\}_{1 \leq i \leq d}$ and does not possess a unique solution. A possibility to overcome this ill-posed problem is to regularize the solution by exploiting spatial correspondence in the image domain Ω at any time instance t. Horn and Schunck [7] use a uniform smoothness constraint in the regularization expression and form

$$\arg\min_{\{u_i\}} \int_{x \in \Omega} \left[\lambda^2 \sum_{i=1}^{d} \sum_{j=1}^{d} (\frac{\partial u_i}{\partial x_j})^2 \right] + \left[E_t + \sum_{i=1}^{d} u_i E_{X_i} \right]^2 dx.$$

To compute the d unknown instantaneous deformation vectors $\{u_i\}_{1 \leq i \leq d}$, the corresponding Euler-Lagrange equations are computed, a discretization of time with $\Delta t = 1$ is used and the solution is estimated iteratively. It is worth mentioning that the original model of [7] does not distinguish between x and X.

3.2 Non-conserved Intensity

Gennert-Negahdaripour [5] propose a non-conserved intensity model in which the rate of change of intensity is modeled as a linear (polynomial of degree $m = 1$) expression of intensity with spatially-varying coefficients c_k as

$$\frac{dE}{dt} = \sum_{k=0}^{m} c_k E^k.$$

The corresponding minimization is

$$\begin{aligned} \min_{\{u_i\}\{c_k\}} \int_{\Omega} \left[\lambda^2 \sum_{i=1}^{d} \sum_{j=1}^{d} (\frac{\partial u_i}{\partial x_j})^2 + \sum_{k=0}^{m} \lambda_k^2 \sum_{j=1}^{d} (\frac{\partial c_k}{\partial x_j})^2 \right] + \\ \left[E_t + \sum_{i=1}^{d} u_i E_{X_i} - \sum_{k=0}^{m} c_k E^k \right]^2 dx, \end{aligned} \tag{4}$$

which contains $d + m + 1$ unknowns that are the instantaneous deformation vectors $\{u_i\}_{1 \leq i \leq d}$ and the coefficients of the intensity shift $\{c_k\}_{0 \leq k \leq m}$. The approach in [5] is very similar to Horn-Schunck's. However, the original model in [5] only addresses a linear intensity shift, i.e., $m = 1$, and does not consider the case where $d > 2$ which involves larger matrix inversions.

A series of methods introduced in [1,4,9] propose constructing a very large system of equations directly based on the expression (4). Iterative solvers (e.g., conjugate gradient method) are used to minimize the objective functional and estimate the deformation vectors. However, these methods require enormous memory and their hierarchical extensions are not obvious in general.

3.3 Demons Method

The demons method proposed by Thirion [13,14] introduces a notation adapted from Maxwell's demons in thermodynamics. As a special type of demon-type Thirion introduces "a complete grid of demons" in which the "pushing forces" are computed for demons placed at every pixel of the image and computed based on the minimum-norm solution of $\frac{dE}{dt} = E_t + \sum_{i=1}^{d} u_i E_{X_i} = 0$. Thirion approximates such force expressions as $u_i = \frac{-f_{x_i}(g-f)}{\epsilon^2 + \sum_{i=1}^{d} f_{x_i}^2}$ that leads to Alg. 1 for a fixed number of iterations. It is worth mentioning that no distinction between instantaneous and total deformations (u and U) is made in the original approach [13,14]. Also, ϵ is replaced by $f - g$ which may cause instabilities. Some of these issues have been clarified in [10].

Alg. 1. Demons algorithm, complete grid of demons

```
read target image $f(x)$ and source image $g(x)$.
choose a linear smoothing operator $G$, a number $\epsilon$, and some large $N$.
set    ${U_i}^{(\mathbf{1})}(x) = 0$,   for   $1 \leq i \leq d$ and $x \in \Omega$.
for $\mathbf{n} = 1 : N$ do                                        // Iterations
    $\phi(x) = U^{(\mathbf{n})}(x) + x$
    ${U_i}^{(\mathbf{n+1})} = G\left[{U_i}^{(\mathbf{n})} - \frac{f_{x_i}\left[g \circ \phi - f\right]}{\epsilon^2 + \|\nabla f\|^2}\right]$,    $1 \leq i \leq d$
end
return the displacement $U^{(\mathbf{n+1})}$, and the registered source image $g \circ \phi$.
```

4 Building a General Model

4.1 The Problem Set-Up

We plan to derive Alg. 1 and include the intensity correction terms in the demons algorithm. To do so, we revisit the Gennert-Negahdaripour model [5]

$$\min_{\{u_i\}\{c_k\}} \int_{\Omega} \left[\lambda^2 \sum_{i=1}^{d} \sum_{j=1}^{d} (\frac{\partial u_i}{\partial x_j})^2 + \sum_{k=0}^{m} \lambda_k^2 \sum_{j=1}^{d} (\frac{\partial c_k}{\partial x_j})^2\right] +$$

$$\left[E_t + \sum_{i=1}^{d} u_i E_{X_i} - \sum_{k=0}^{m} c_k E^k\right]^2 dx.$$

Due to an anticipated improved representation, we change the scale of the coefficients defining $\alpha_k = \lambda/\lambda_k$, $w_k = -c_k/\alpha_k$ and obtain the new functional

$$L = \int_\Omega \lambda^2 \Big[\sum_{i=1}^{d} \sum_{j=1}^{d} (\frac{\partial u_i}{\partial x_j})^2 + \sum_{k=0}^{m} \sum_{j=1}^{d} (\frac{\partial w_k}{\partial x_j})^2 \Big] + \\ \Big[E_t + \sum_{i=1}^{d} u_i E_{X_i} + \sum_{k=0}^{m} \alpha_k w_k E^k \Big]^2 dx.$$

This change of variable, will enable us to use a simplified version of Sherman-Morrison-Woodbury matrix inversion lemma [18,12] in deriving the solution in the next section. The new objective is to minimize L with respect to $\{u_i\}_{1\le i\le d}$ and $\{w_k\}_{0\le k\le m}$. The corresponding Euler-Lagrange equations of this minimization yields

$$\forall \;\; 1 \le i \le d,$$

$$\lambda^2 (\sum_{j=1}^{d} \frac{\partial^2 u_i}{\partial {x_j}^2}) = E_{X_i} \Big[E_t + \sum_{i=1}^{d} u_i E_{X_i} + \sum_{k=0}^{m} \alpha_k w_k E^k \Big], \qquad (5)$$

$$\forall \;\; 0 \le k \le m,$$

$$\lambda^2 (\sum_{j=1}^{d} \frac{\partial^2 w_k}{\partial {x_j}^2}) = \alpha_k E^k \Big[E_t + \sum_{i=1}^{d} u_i E_{X_i} + \sum_{k=0}^{m} \alpha_k w_k E^k \Big]. \qquad (6)$$

Similar to [7], assume that for any v we approximate the Laplacian of v with

$$\nabla^2 v = \sum_{j=1}^{d} \frac{\partial^2 v}{\partial {x_j}^2} \approx \kappa^2 (Gv - v)$$

in which G is some linear smoothing operator and κ is a constant real number. One can verify that assuming G to be a Gaussian with zero mean and standard deviation $\frac{1}{\sqrt{\ln 4}}$ and $\kappa = 2$ leads to the corresponding approximations presented in [7]. We also define the set of vectors

$$\mathbf{M} = (E_{X_1}, E_{X_2}, \ldots, E_{X_d}, \alpha_0, \alpha_1 E^1, \ldots, \alpha_m E^m)$$

$$\mathbf{V} = (u_1, u_2, \ldots, u_d, w_0, w_1, \ldots, w_m)^T$$

$$\bar{\mathbf{V}} = G\big[\mathbf{V}\big] = (Gu_1, Gu_2, \ldots, Gu_d, Gw_0, Gw_1, \ldots, Gw_m)^T$$

which will be used in finding the solution.

4.2 Deriving the Solution

The set of Euler-Lagrange Equations (5) and (6) can be summarized as

$$(\lambda\kappa)^2(\bar{\mathbf{V}} - \mathbf{V}) = \mathbf{M}^T(E_t + \mathbf{M}\mathbf{V}).$$

Arranging the terms to form a linear equation of $\mathbf{V}$ yields

$$\big[\mathbf{M}^T\mathbf{M} + (\lambda\kappa)^2\mathbf{I}\big]\mathbf{V} = (\lambda\kappa)^2\bar{\mathbf{V}} - E_t\mathbf{M}^T. \tag{7}$$

Solving Equation (7) for $\mathbf{V}$ using Lemma 1 (simplified Sherman-Morrison-Woodbury matrix inversion formula [18,12]), for $\epsilon = \lambda\kappa$ yields [See the proofs in the Appendix]

$$\mathbf{V} = \bar{\mathbf{V}} - \frac{\mathbf{M}^T(E_t + \mathbf{M}\bar{\mathbf{V}})}{(\lambda\kappa)^2 + \mathbf{M}\mathbf{M}^T}. \tag{8}$$

Lemma 1. If $\mathbf{M}$ is a row-vector, then for any nonzero ϵ

$$[\mathbf{M}^T\mathbf{M} + \epsilon^2\mathbf{I}]^{-1} = \frac{1}{\epsilon^2}[\mathbf{I} - \frac{\mathbf{M}^T\mathbf{M}}{\epsilon^2 + \mathbf{M}\mathbf{M}^T}].$$

Finally, taking G from both sides of Equation (8) yields

$$\bar{\mathbf{V}} = G\Big[\mathbf{V}\Big] = G\Big[\bar{\mathbf{V}} - \frac{\mathbf{M}^T(E_t + \mathbf{M}\bar{\mathbf{V}})}{(\lambda\kappa)^2 + \mathbf{M}\mathbf{M}^T}\Big].$$

Hence, we wish to approximate $\bar{\mathbf{V}}$ that satisfies

$$\bar{\mathbf{V}} = G\Big[\bar{\mathbf{V}} - \frac{\mathbf{M}^T(E_t + \mathbf{M}\bar{\mathbf{V}})}{(\lambda\kappa)^2 + \mathbf{M}\mathbf{M}^T}\Big].$$

Momentarily, renaming $\bar{\mathbf{V}}$ as $\mathbf{V}$ we find an estimate of $\mathbf{V}$ that instead satisfies

$$\mathbf{V} = G\Big[\mathbf{V} - \frac{\mathbf{M}^T(E_t + \mathbf{M}\mathbf{V})}{(\lambda\kappa)^2 + \mathbf{M}\mathbf{M}^T}\Big]. \tag{9}$$

From now on, we remember that any approximation of $\mathbf{V}$ requires a deconvolution step with respect to G (if G is a convolution operator) due to this change of variable. Such deconvolution is not performed assuming $\mathbf{V}$ is smooth enough. To estimate $\mathbf{V}$ that satisfies Equation (9) converting to the original notation we would like to find $\{u_i\}_{1\le i\le d}$, $\{w_k\}_{0\le k\le m}$ that

$\forall\ \ 1 \le i \le d,$

$$u_i = G\Bigg[u_i - \frac{E_{X_i}\Big[E_t + \sum_{i=1}^d u_i E_{X_i} + \sum_{k=0}^m \alpha_k w_k E^k\Big]}{(\lambda\kappa)^2 + \sum_{i=1}^d (E_{X_i})^2 + \sum_{k=0}^m(\alpha_k E^k)^2}\Bigg], \tag{10}$$

$\forall\ \ 0 \le k \le m,$

$$w_k = G\Bigg[w_k - \frac{\alpha_k E^k \Big[E_t + \sum_{i=1}^{d} u_i E_{X_i} + \sum_{k=0}^{m} \alpha_k w_k E^k\Big]}{(\lambda\kappa)^2 + \sum_{i=1}^{d}(E_{X_i})^2 + \sum_{k=0}^{m}(\alpha_k E^k)^2}\Bigg]. \tag{11}$$

Now we are ready to move from instantaneous deformation vectors to total displacements. Integrating Equation (10) with respect to time t over $[0,1]$ yields

$$\int_0^1 u_i \; dt = G\Bigg[\int_0^1 u_i \; dt -$$
$$\int_0^1 \frac{E_{X_i}\Big[E_t + \sum_{i=1}^{d} u_i E_{X_i} + \sum_{k=0}^{m} \alpha_k w_k E^k\Big]}{(\lambda\kappa)^2 + \sum_{i=1}^{d}(E_{X_i})^2 + \sum_{k=0}^{m}(\alpha_k E^k)^2} dt\Bigg],$$

for any $1 \leq i \leq d$. Hence, from Equation (3), $\frac{dE}{dt} = E_t + \sum_{i=1}^{d} u_i E_{X_i}$, and we obtain

$$U_i = G\Bigg[U_i - \int_0^1 \frac{E_{X_i}\Big[\frac{dE}{dt} + \sum_{k=0}^{m} \alpha_k w_k E^k\Big]}{(\lambda\kappa)^2 + \sum_{i=1}^{d}(E_{X_i})^2 + \sum_{k=0}^{m}(\alpha_k E^k)^2} dt\Bigg]. \tag{12}$$

Choosing $\Delta t = 1$ and approximating the integrals using left Riemann sum evaluating E and its partial derivatives yields for any $1 \leq i \leq d$

$$U_i = G\Bigg[U_i - \frac{f_{x_i}\Big[\int_0^1 \frac{dE}{dt} dt + \sum_{k=0}^{m} \alpha_k f^k \int_0^1 w_k dt\Big]}{(\lambda\kappa)^2 + \sum_{i=1}^{d}(f_{x_i})^2 + \sum_{k=0}^{m}(\alpha_k f^k)^2}\Bigg].$$

[Note that the choice of $\Delta t = 1$ is the extreme case of Riemann sum where the interval is partitioned to only one element. The authors do not claim that this provides a superior approximation. This approximation is employed for consistency with [7].] Hence, using Equation (2) yields for any $1 \leq i \leq d$

$$U_i = G\Bigg[U_i - \frac{f_{x_i}\Big[g \circ \phi - f + \sum_{k=0}^{m} \alpha_k f^k W_k\Big]}{(\lambda\kappa)^2 + \sum_{i=1}^{d}(f_{x_i})^2 + \sum_{k=0}^{m}(\alpha_k f^k)^2}\Bigg],$$

in which $W_k = \int_0^1 w_k dt$. Similarly, if we proceed the same approach for w_k, $0 \leq k \leq m$, starting from Equation (11) we obtain the following pair of equations.

$\forall \;\; 1 \leq i \leq d,$

$$U_i = G\Bigg[U_i - \frac{f_{x_i}\Big[g \circ \phi - f + \sum_{k=0}^{m} \alpha_k f^k W_k\Big]}{(\lambda\kappa)^2 + \|\nabla f\|^2 + \sum_{k=0}^{m}(\alpha_k f^k)^2}\Bigg], \tag{13}$$

$\forall \;\; 0 \leq k \leq m,$

$$W_k = G\Bigg[W_k - \frac{\alpha_k f^k\Big[g \circ \phi - f + \sum_{k=0}^{m} \alpha_k f^k W_k\Big]}{(\lambda\kappa)^2 + \|\nabla f\|^2 + \sum_{k=0}^{m}(\alpha_k f^k)^2}\Bigg], \tag{14}$$

in which $\phi(x) = U(x) + x$. The solution of these two equations will be approximated numerically using the *fixed-point iterations*.

4.3 Hyper-Demons Algorithm

Approximating the solution of Equations (13) and (14) using fixed-point iterations, we obtain the Hyper-Demons Alg. 2. Assuming the normalization factors $[\alpha_0, \ldots, \alpha_m]$ to be all zero makes Alg. 2 equivalent to the demons Alg. 1. Hence, the demons algorithm is a special case of our proposed hyper-demons algorithm. A simple case of the algorithm with 0-degree polynomial is given in Alg. 3.

Alg. 2. Hyper-Demons Algorithm

```
read target image $f(x)$ and source image $g(x)$.
choose a linear smoothing operator $G$, a number $\epsilon$, polynomial degree $m$,
        normalizing factors $[\alpha_0, \ldots, \alpha_m]$, and some large $N$.
set   $U_i{}^{(\mathbf{1})}(x) = W_k{}^{(\mathbf{1})}(x) = 0$,  for any $1 \le i \le d$,  $0 \le k \le m$,  $x \in \Omega$.
for $\mathbf{n} = 1 : N$ do                                    // Fixed point iterations
    $\phi(x) = U^{(\mathbf{n})}(x) + x$
    $U_i{}^{(\mathbf{n+1})} = G\left[U_i{}^{(\mathbf{n})} - \frac{f_{x_i}\left[g \circ \phi - f + \sum_{k=0}^{m} \alpha_k f^k W_k{}^{(\mathbf{n})}\right]}{\epsilon^2 + \|\nabla f\|^2 + \sum_{k=0}^{m} (\alpha_k f^k)^2}\right]$,   $1 \le i \le d$
    $W_k{}^{(\mathbf{n+1})} = G\left[W_k{}^{(\mathbf{n})} - \frac{\alpha_k f^k\left[g \circ \phi - f + \sum_{k=0}^{m} \alpha_k f^k W_k{}^{(\mathbf{n})}\right]}{\epsilon^2 + \|\nabla f\|^2 + \sum_{k=0}^{m} (\alpha_k f^k)^2}\right]$,  $0 \le k \le m$.
end
return the displacement $U^{(\mathbf{n+1})}$, and the registered source image $g \circ \phi$.
```

Alg. 3. Hyper-Demons Algorithm: 0-Degree Contrast Polynomial

```
read target image $f(x)$ and source image $g(x)$.
choose a linear smoothing operator $G$, a number $\epsilon$, a normalizing factor $\alpha$,
        and some large number $N$.
set   $U_i{}^{(\mathbf{1})}(x) = W^{(\mathbf{1})}(x) = 0$,  for any $1 \le i \le d$,  $x \in \Omega$.
for $\mathbf{n} = 1 : N$ do                                    // Fixed point iterations
    $\phi(x) = U^{(\mathbf{n})}(x) + x$
    $U_i{}^{(\mathbf{n+1})} = G\left[U_i{}^{(\mathbf{n})} - \frac{f_{x_i}\left[g \circ \phi - f + \alpha W^{(\mathbf{n})}\right]}{\epsilon^2 + \|\nabla f\|^2 + \alpha^2}\right]$,   $1 \le i \le d$
    $W^{(\mathbf{n+1})} = G\left[W^{(\mathbf{n})} - \frac{\alpha\left[g \circ \phi - f + \alpha W^{(\mathbf{n})}\right]}{\epsilon^2 + \|\nabla f\|^2 + \alpha^2}\right]$.
end
return the displacement $U^{(\mathbf{n+1})}$, and the registered source image $g \circ \phi$.
```

Hyper-Demons algorithm with right or central approximation: The terms f_{x_i}, ∇f, and f^k in the hyper-demons algorithm is a consequence of a left approximation of E_{X_i}, ∇E, and E^k in Equation (12) for t over $[0, 1]$. If we wish to use the right approximation, we need to replace the terms f_{x_i}, ∇f, and f^k in the Hyper-Demons algorithm respectively by $g_{x_i} \circ \phi$, $(\nabla g) \circ \phi$, and $(g \circ \phi)^k$. Similarly, the terms f_{x_i}, ∇f, and f^k in the Hyper-Demons algorithm are respectively replaced by $\frac{f_{x_i}+g_{x_i}\circ\phi}{2}$, $\frac{\nabla f+(\nabla g)\circ\phi}{2}$, and $(\frac{f+g\circ\phi}{2})^k$ starting from a central approximation.

5 Results and Conclusions

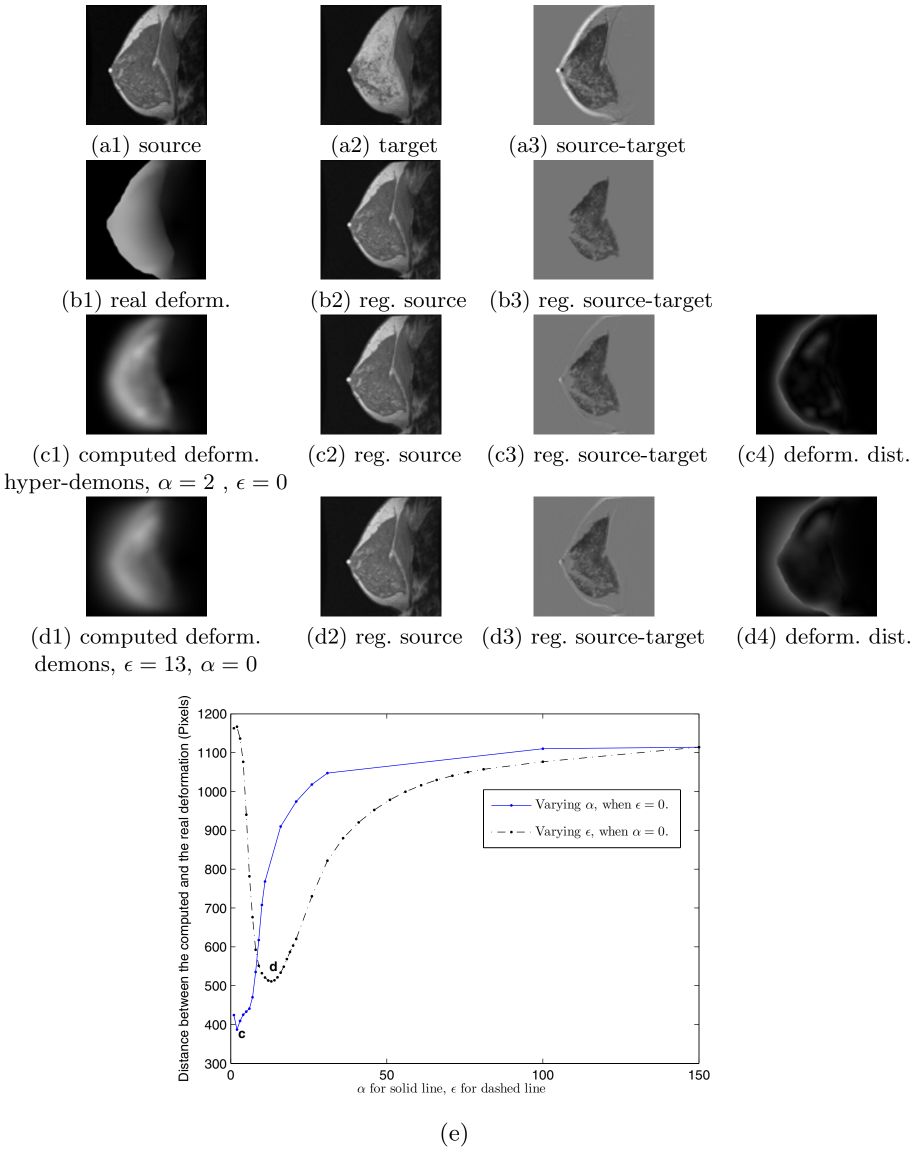
5.1 Theoretical Results

Our approach presents a novel and distinct interpretation of the demons algorithm compared to previous works [11,10] in this direction. Furthermore, our approach presents a generalization of the Gennert-Negahdaripour formulation [5] in a natural fashion, because we based our approach on their model yet included arbitrary image dimensions and arbitrary degrees of the polynomial. Our approach generalizes the demons algorithm to include the intensity shift, because assuming $\alpha_k \to 0$, $0 \leq k \leq m$, corresponds to $\lambda_k = \infty$, $0 \leq k \leq m$ in the Gennert-Negahdaripour formulation [5] and yields the original demons algorithm [13,14]. Furthermore, the algorithm resulting from the "right" and "central" approximations corresponds to similar ones in the literature [11,2,15,16] obtained from different point of views.

5.2 Computational Results

To better understand the introduced algorithm, we start from two 8-bit, 256×256 images shown in Fig. 1(a1-a2). The target image is generated by adding contrast enhancement to the source image and the source image is then deformed using a finite element model (FEM) [9]. The magnitude of this deformation is shown is Fig. 1(b1) and the intensity change is shown in Fig. 1(b3). If we use the central-scheme demons algorithm over various integer values of the parameter ϵ and compute the euclidean distance between the computed and the real deformation we yield the dashed curve shown in Fig. 1(e). It turns out that the minimum of such deformation distance occurs at $\epsilon = 13$ marked by "d" on Fig. 1(e). The magnitude of the corresponding deformation, the registered source, the difference between the registered source and the target, and the distance between the computed deformation and the real deformation are shown in Fig. 1(d1-d4) respectively. In all of our experiments in this section, $N = 100$ iterations were performed, G is chosen as a Gaussian of size 15×15 and of standard deviation $\sigma = 5$, and cubic interpolation is used in the algorithms when necessary.

Now if we use the central-scheme hyper-demons algorithm with zero-degree intensity shift, i.e. $m = 0$, and choose $\epsilon = 0$ varying α with the same parameters as the previous experiment we yield the solid curve plotted in Fig. 1(e).



(a1) source (a2) target (a3) source-target

(b1) real deform. (b2) reg. source (b3) reg. source-target

(c1) computed deform. hyper-demons, $\alpha = 2$, $\epsilon = 0$ (c2) reg. source (c3) reg. source-target (c4) deform. dist.

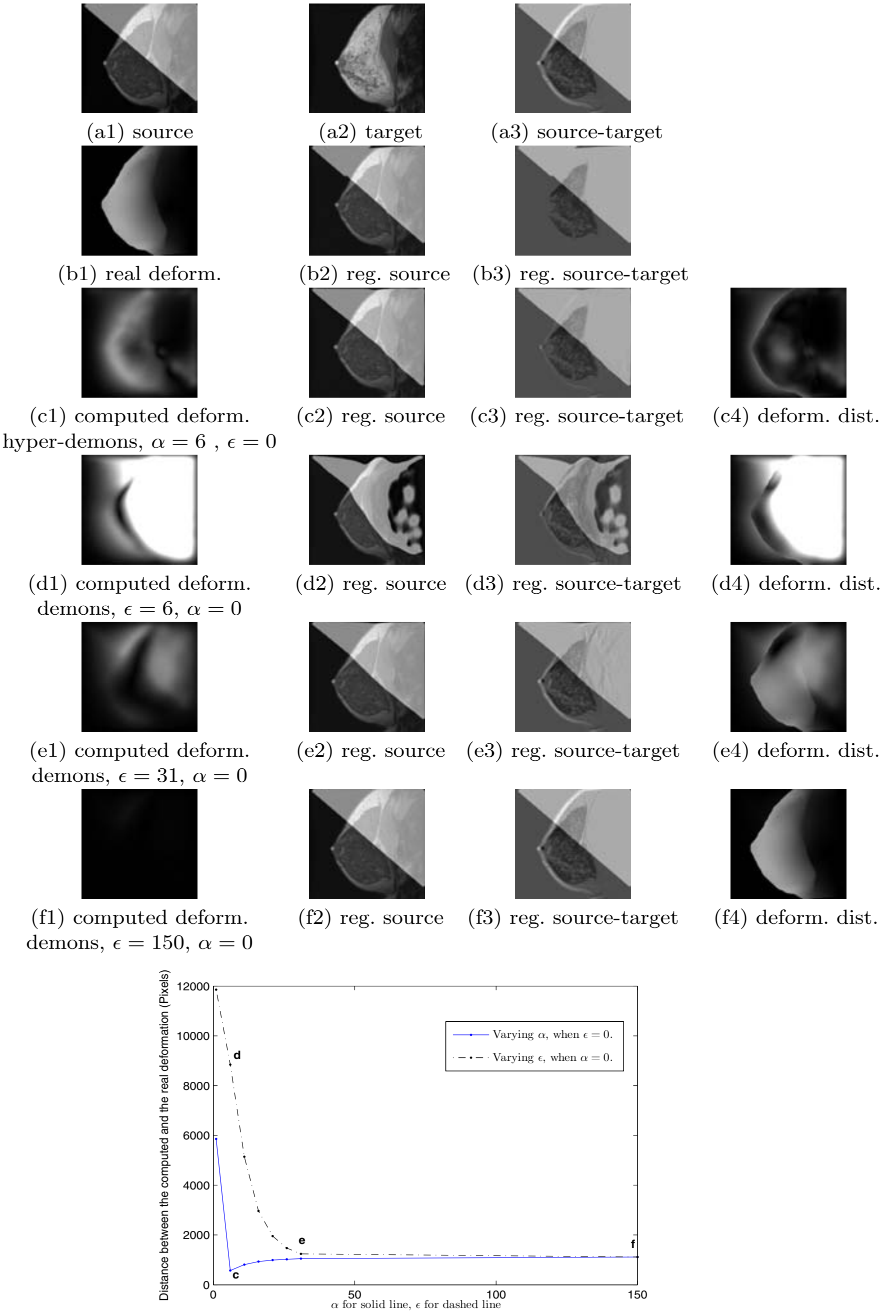
(d1) computed deform. demons, $\epsilon = 13$, $\alpha = 0$ (d2) reg. source (d3) reg. source-target (d4) deform. dist.

Distance between the computed and the real deformation (Pixels)
1200
1100
1000
900
800
700
600
500
400
300
0
50
100
150
α for solid line, ϵ for dashed line
Varying α, when ϵ = 0.
Varying ϵ, when α = 0.
c
d

(e)

Fig. 1. Comparison of the demons and hyper-demons algorithms for simulated data

The minimum deformation distance occurs at $\alpha = 2$ marked as "c" on the plot. The corresponding deformation related images in this case are shown Fig. 1(c1-c4). It can be observed that the "best" deformation computed via changing α (i.e., a special case of hyper-demons) is slightly more precise that the "best" deformation vector computed varying ϵ (i.e., the demons algorithm).



(a1) source (a2) target (a3) source-target

(b1) real deform. (b2) reg. source (b3) reg. source-target

(c1) computed deform. hyper-demons, $\alpha = 6$, $\epsilon = 0$ (c2) reg. source (c3) reg. source-target (c4) deform. dist.

(d1) computed deform. demons, $\epsilon = 6$, $\alpha = 0$ (d2) reg. source (d3) reg. source-target (d4) deform. dist.

(e1) computed deform. demons, $\epsilon = 31$, $\alpha = 0$ (e2) reg. source (e3) reg. source-target (e4) deform. dist.

(f1) computed deform. demons, $\epsilon = 150$, $\alpha = 0$ (f2) reg. source (f3) reg. source-target (f4) deform. dist.

Distance between the computed and the real deformation (Pixels)
12000
10000
8000
6000
4000
2000
0
0
50
100
150
α for solid line, ε for dashed line
Varying α, when ε = 0.
Varying ε, when α = 0.
d
c
e
f

(g)

Fig. 2. Comparison of the demons and hyper-demons algorithms for simulated data

To further observe the role of α in the algorithm we add a ramp-shaped intensity shift of value 160 to the source image denoted in Fig. 2(a1) and perform similar experiments. It can be observed that the central-scheme demons algorithm is not capable of estimating the motion in this case due to this rather large intensity shift. The corresponding curve in plotted via dashed line in Fig. 2(g). It can be seen that curve does not attain a minimum. Three corresponding nominal values of $\epsilon = 6$, 31, and 150 are simply picked and their corresponding deformation-related images are displayed in Fig. 2 (d1-d4, e1-e4,f1-f4). These three locations are labeled as "d","e", and "f" on the plot of Fig. 2(g). Now if we repeat the experiment for the central-scheme hyper-demons algorithm with zero-degree intensity shift and choose $\epsilon = 0$ varying α we obtain the solid curve on the plot of Fig. 2(g). The minimum of this curve is attained at $\alpha = 6$ which is labeled as "c" on the plot. The corresponding deformation-related figures of this $\alpha = 6$, $\epsilon = 0$ case is given in Fig. 2(c1-c4). Note that the deformation difference plotted in Fig. 2(c4) is much smaller compared to the corresponding differences displayed in Fig. 2(d4,e4,f4). It can be deduced that for no value of ϵ the correct motion can be estimated in this case using the traditional demons algorithm due to the large intensity change (or varying illumination). However, the hyper-demons algorithm is effectively capable of estimating the motion.

5.3 Concluding Remarks

We mathematically derived the demons algorithm [13,14] from the past methods [5,7] in a consistent framework. This provided a new interpretation and a novel extension of the demons algorithm. It is worth mentioning that our approach does not separate the intensity correction stage from the registration as opposed to some of the existing works in the literature (e.g., [6]). A focus of our work was to address image registration problem under varying illuminations. This is particularly relevant for the registration of dynamic contrast enhanced images. A recent, general, and related work to this manuscript will appear in [3].

Acknowledgements

This research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) in the form of a Post-Doctoral Fellowship for Mehran Ebrahimi. This work was also supported by the Terry Fox Foundation for Cancer Research. The authors would like to thank Dr. Kristy Brock of the Princess Margaret Hospital for the FEM simulations.

References

1. Barber, D.C., Hose, D.R.: Automatic segmentation of medical images using image registration: diagnostic and simulation applications. Journal of Medical Engineering & Technology 29(2), 53–63 (2005)

2. Cachier, P., Bardinet, E., Dormont, D., Pennec, X., Ayache, N.: Iconic feature based nonrigid registration: the pasha algorithm. Computer Vision and Image Understanding 89(2-3), 272–298 (2003)
3. Ebrahimi, M., Martel, A.L.: A general pde-framework for registration of contrast enhanced images. In: MICCAI (2009)
4. Froh, M.S., Barber, D.C., Brock, K.K., Plewes, D.B., Martel, A.L.: Piecewise-quadrilateral registration by optical flow - applications in contrast-enhanced mr imaging of the breast. In: MICCAI, vol. 2, pp. 686–693 (2006)
5. Gennert, M.A., Negahdaripour, S.: Relaxing the brightness constancy assumption in computing optical flow. Technical Report 975, MIT AI Lab. Memo (1987)
6. Guimond, A., Roche, A., Ayache, N., Meunier, J.: Three-dimensional multimodal brain warping using the demons algorithm and adaptive intensity corrections. IEEE Transactions on Medical Imaging 20(1), 58–69 (2001)
7. Horn, B.K.P., Schunck, B.G.: Determining optical flow. Artificial Intelligence 17, 185–203 (1981)
8. Lai, S.-H.: Computation of optical flow under non-uniform brightness variations. Pattern Recognition Letters 25(8), 885–892 (2004)
9. Martel, A.L., Froh, M.S., Brock, K.K., Plewes, D.B., Barber, D.C.: Evaluating an optical-flow-based registration algorithm for contrast-enhanced magnetic resonance imaging of the breast. Phys. Med. Biol. 52, 3803–3816 (2007)
10. Modersitzki, J.: Numerical Methods for Image Registration. Oxford University Press, Oxford (2003)
11. Pennec, X., Cachier, P., Ayache, N.: Understanding the "demon's algorithm": 3d non-rigid registration by gradient descent. In: Taylor, C., Colchester, A. (eds.) MICCAI 1999. LNCS, vol. 1679, pp. 597–605. Springer, Heidelberg (1999)
12. Sherman, J., Morrison, W.J.: Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. Annals of Mathematical Statistics 21(1), 124–127 (1950)
13. Thirion, J.P.: Fast non-rigid matching of 3d medical images. In: Medical Robotics and Computer Aided Surgery (MRCAS), Baltimore, pp. 47–54 (1995)
14. Thirion, J.P.: Image matching as a diffusion process: an analogy with maxwell's demons. Medical Image Analysis 2(3), 243–260 (1998)
15. Vercauteren, T., Pennec, X., Malis, E., Perchant, A., Ayache, N.: Insight into efficient image registration techniques and the demons algorithm. In: Karssemeijer, N., Lelieveldt, B. (eds.) IPMI 2007. LNCS, vol. 4584, pp. 495–506. Springer, Heidelberg (2007)
16. Vercauteren, T., Pennec, X., Perchant, A., Ayache, N.: Non-parametric diffeomorphic image registration with the demons algorithm. In: Ayache, N., Ourselin, S., Maeder, A. (eds.) MICCAI 2007, Part II. LNCS, vol. 4792, pp. 319–326. Springer, Heidelberg (2007)
17. Weickert, J., Bruhn, A., Brox, T., Papenberg, N.: A survey of variational optic flow methods for small displacement, in Mathematical Models for Registration and Applications to Medial Imaging. Springer, Heidelberg (2005)
18. Woodbury, M.A.: Inverting modified matrices. Memorandum rept. 42, Statistical Research Group, Princeton University, Princeton, NJ (1950)

Appendix

Proof of Lemma 1

Proof. Note that $\mathbf{M}\mathbf{M}^T$ is a scalar if $\mathbf{M}$ is a row-vector. Multiply $[\mathbf{M}^T\mathbf{M}+\epsilon^2\mathbf{I}]$ with its "potential" inverse $\frac{1}{\epsilon^2}[\mathbf{I}-\frac{\mathbf{M}^T\mathbf{M}}{\epsilon^2+\mathbf{M}\mathbf{M}^T}]$ from left (and then right) and obtain the identity matrix $\mathbf{I}$. The left-hand-side calculations are given below. The right-hand-side case is shown similarly.

$$\begin{aligned}
&\frac{1}{\epsilon^2}[\mathbf{I}-\frac{\mathbf{M}^T\mathbf{M}}{\epsilon^2+\mathbf{M}\mathbf{M}^T}]\times[\mathbf{M}^T\mathbf{M}+\epsilon^2\mathbf{I}]\\
&=\frac{1}{\epsilon^2}\Big[\mathbf{M}^T\mathbf{M}-\frac{\mathbf{M}^T\mathbf{M}\mathbf{M}^T\mathbf{M}}{\epsilon^2+\mathbf{M}\mathbf{M}^T}+\epsilon^2\mathbf{I}-\frac{\epsilon^2\mathbf{M}^T\mathbf{M}}{\epsilon^2+\mathbf{M}\mathbf{M}^T}\Big]\\
&=\frac{1}{\epsilon^2}\Big[\frac{(\epsilon^2+\mathbf{M}\mathbf{M}^T)\mathbf{M}^T\mathbf{M}-\mathbf{M}\mathbf{M}^T\mathbf{M}^T\mathbf{M}-\epsilon^2\mathbf{M}^T\mathbf{M}}{\epsilon^2+\mathbf{M}\mathbf{M}^T}+\epsilon^2\mathbf{I}\Big]\\
&=\frac{1}{\epsilon^2}\Big[\mathbf{0}+\epsilon^2\mathbf{I}\Big]=\mathbf{I}.
\end{aligned}$$

Proof of Equation (8)

Choosing $\epsilon=\lambda\kappa$, Equation (7) gives

$$\big[\mathbf{M}^T\mathbf{M}+\epsilon^2\mathbf{I}\big]\mathbf{V}=\epsilon^2\bar{\mathbf{V}}-E_t\mathbf{M}^T.$$

Hence, using Lemma 1

$$\begin{aligned}
\mathbf{V}&=\big[\mathbf{M}^T\mathbf{M}+\epsilon^2\mathbf{I}\big]^{-1}\big[\epsilon^2\bar{\mathbf{V}}-E_t\mathbf{M}^T\big]\\
&=\frac{1}{\epsilon^2}[\mathbf{I}-\frac{\mathbf{M}^T\mathbf{M}}{\epsilon^2+\mathbf{M}\mathbf{M}^T}]\big[\epsilon^2\bar{\mathbf{V}}-E_t\mathbf{M}^T\big]\\
&=\frac{1}{\epsilon^2}\big[\epsilon^2\bar{\mathbf{V}}-\frac{\epsilon^2\mathbf{M}^T\mathbf{M}\bar{\mathbf{V}}}{\epsilon^2+\mathbf{M}\mathbf{M}^T}-E_t\mathbf{M}^T+\frac{E_t\mathbf{M}^T\mathbf{M}\mathbf{M}^T}{\epsilon^2+\mathbf{M}\mathbf{M}^T}\big]\\
&=\frac{1}{\epsilon^2}\big[\epsilon^2\bar{\mathbf{V}}-\frac{\epsilon^2\mathbf{M}^T\mathbf{M}\bar{\mathbf{V}}}{\epsilon^2+\mathbf{M}\mathbf{M}^T}-\frac{E_t(\epsilon^2+\mathbf{M}\mathbf{M}^T)\mathbf{M}^T}{\epsilon^2+\mathbf{M}\mathbf{M}^T}+\frac{E_t\mathbf{M}\mathbf{M}^T\mathbf{M}^T}{\epsilon^2+\mathbf{M}\mathbf{M}^T}\big]\\
&=\frac{1}{\epsilon^2}\big[\epsilon^2\bar{\mathbf{V}}-\frac{\epsilon^2\mathbf{M}^T\mathbf{M}\bar{\mathbf{V}}}{\epsilon^2+\mathbf{M}\mathbf{M}^T}-\frac{\epsilon^2\mathbf{M}^TE_t}{\epsilon^2+\mathbf{M}\mathbf{M}^T}\big]\\
&=\bar{\mathbf{V}}-\frac{\mathbf{M}^T(E_t+\mathbf{M}\bar{\mathbf{V}})}{\epsilon^2+\mathbf{M}\mathbf{M}^T}.
\end{aligned}$$

[Note that $\mathbf{M}\mathbf{M}^T$ is a scalar and has commuted its order in the above computation.]