A Convex Model for Image Registration

Ernie Esser

January 2010

Abstract

Variational methods for image registration generally involve minimizing a nonconvex functional with respect to the unknown displacement between two given images. A linear approximation of the image intensities is often used to obtain a convex approximation to the model, but it is only valid for small deformations. Algorithms such as gradient descent can get stuck in undesirable local minima of the nonconvex functional. Here, instead of seeking a global minimum of a nonconvex functional, and without making a small deformation assumption, we introduce and work with a different, convex model for the registration problem. In particular we consider a graph-based formulation that requires minimizing a convex function on the edges of the graph instead of working directly with the displacement field. The corresponding displacement can be inferred from the edge function. The convex model generally involves many more variables, but its global minimum can be a better solution than a local minimum of the nonconvex model. We use a variant of the primal dual hybrid gradient (PDHG) algorithm [23] to numerically solve several example problems.

1 Introduction

A classical approach for registering given images u, $\phi : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ is to minimize the Horn-Schunck model, [12]

$$\frac{1}{2}\|\phi(x + v(x)) - u(x)\|^2 + \frac{\gamma}{2}\|\nabla v_1\|^2 + \frac{\gamma}{2}\|\nabla v_2\|^2, \tag{1}$$

with respect to the displacement field v. The goal this paper is to reformulate this as a convex minimization problem, but without linearizing the fidelity term, which would require a small deformation assumption, and also while avoiding the interpolation difficulty that arises when discretizing $\phi(x + v)$. Rather than discretizing a continuous model, we will work in a discrete setting throughout. Given images $u \in \mathbb{R}^{m_r \times m_c}$ and $\phi \in \mathbb{R}^{n_r \times n_c}$, consider defining a graph as in Figure 1. The nodes correspond to pixel centers in u and ϕ, and the edges are defined to connect each pixel in u to a neighborhood of pixels in ϕ. The unknown edge weights will correspond to the coefficients of a weighted average. The main idea behind the convex reformulation is then to replace $\phi(x + v)$, v_1 and v_2 in (1) with weighted averages of intensities and pixel locations. The discrete interpolation corresponding to $\phi(x + v)$ will be a weighted average of intensities of ϕ with the weights corresponding to edge weights on the graph. Similarly the displacement will be modeled as the difference between the pixel locations in u and the same weighted averages of the pixel locations in ϕ.

A crucial requirement for the weighted average approach to be justified is that the weights be localized. For the interpolation to make sense, it should only depend on nearby pixels. Therefore

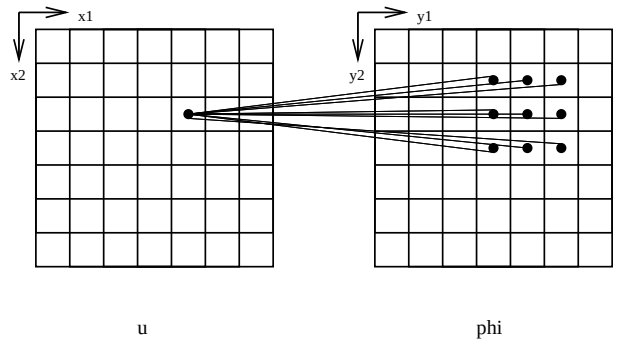

Figure 1: Construction of edges $e_{i,j}$

the edge weights should be zero outside a small neighborhood around the weighted average of the ϕ pixel locations. Directly enforcing this would unfortunately correspond to a nonconvex constraint. However, it is possible to indirectly encourage the weights to cluster by adding a convex term to the functional requiring the weights to be spatially smooth. Details are in the following section.

There are many other approaches that aim to convexify or at least partially convexify Horn-Schunck related models for image registration. Many authors make use of a multiscale approach, working from coarse to fine images, since applying low pass filters to the images can make the energy more convex [14]. As already mentioned, one can obtain a convex approximation to (1) by linearizing $\frac{1}{2}\|\phi(x+v(x))-u(x)\|^2$ to obtain $\frac{1}{2}\|\phi(x)+\langle\nabla\phi(x),v(x)\rangle-u(x)\|_2^2$. A multiscale approach can be used to get around the drawback that the linearization is only valid for small deformations. It's shown in [5] that applying a coarse to fine strategy to either the original nonconvex functional or to the linearized version amounts to essentially the same thing. A coarse to fine approach is also used in [21] in a real time algorithm for computing optical flow. Another very interesting approach is the discrete functional lifting method in [10], which recovers a global minimum of the nonconvex functional by solving an equivalent convex problem. It is related to the discrete and continuous functional lifting approaches of [13] and [17].

The organization of this paper is as follows. In Section 2, the weighted average based convex model for image registration is defined and discussed. Section 3 explains how a variant of the primal dual hybrid gradient (PDHG) method [23, 9] can be used to minimize the resulting functional. Numerical results for several registration examples are presented in Section 4. Section 5 discuses extensions of the convex model such as incorporating l_1 and TV terms into the functional, modifications of the numerical scheme as well as other applications of the numerical approach to similar models.

2 Formulation of Convex Registration Model

Let the images $u \in \mathbb{R}^{m_r \times m_c}$ and $\phi \in \mathbb{R}^{n_r \times n_c}$ be given. Assume that the pixel intensities are not changed by the optimal displacement. This is the key assumption behind the data fidelity term in (1). Also assume we are given a guess $\nu = (\nu_1, \nu_2)$ of the displacement field and upper bounds $r_1, r_2 \geq 0$ such that if v^* is the true displacement, then $\|\nu_1 - v_1^*\|_\infty \leq r_1$ and $\|\nu_2 - v_2^*\|_\infty \leq r_2$. The images are allowed to be at different resolutions, but we assume that the dimensions of the

pixels are known and are such that there are no major scale differences between the two images. Let $M = m_r m_c$ be the number of pixels in u and $N = n_r n_c$ the number of pixels in ϕ. Consider a graph $G(\mathcal{V}, \mathcal{E})$ with a node for each pixel in u and ϕ. We can write $\mathcal{V}$ as $\mathcal{V} = \mathcal{V}_u \bigcup \mathcal{V}_\phi$ where $\mathcal{V}_\phi = \{1, ..., N\}$ indexes the nodes from ϕ and $\mathcal{V}_u = \{N+1, ..., M+N\}$ indexes the nodes from u. We will consider $u \in \mathbb{R}^M$ to be a function on the nodes in $\mathcal{V}_u$ and similarly $\phi \in \mathbb{R}^N$ a function on the nodes in $\mathcal{V}_\phi$. Also define $x_1,\ x_2 \in \mathbb{R}^M$ to be functions on nodes in V_u, and similarly define y_1 and y_2 to be functions on the nodes in V_ϕ such that (x_1^i, x_2^i) is the location of the center of the pixel corresponding to node $i \in V_u$, and (y_1^j, y_2^j) is the location of the center of the pixel corresponding to node $j \in V_\phi$. Now let there be an edge between node $i \in \mathcal{V}_u$ and node $j \in V_\phi$ if $|y_1^j - (x_1^i + \nu_1^i)| \leq r_1$ and $|y_2^j - (x_2^i + \nu_2^i)| \leq r_2$. Denote the total number of edges by e. Figure 1 illustrates how these edges connect each node i to a group of nodes j.

Note that if r is too large, the number of edges defined could be so large as to cause memory problems in a numerical implementation. One way of avoiding this is to limit the size of the allowed deformation. Some alternatives include only defining edges on a subset of the nodes satisfying the displacement bounds, using a coarse to fine multiscale approach, or designing a transformation that gives a lower dimensional approximation of the edge weights. Here, we will make use of the multiscale approach in numerical implementations.

Assume for simplicity that any $(w_1, w_2) \in \mathbb{R}^2$ satisfying $|w_1 - (x_1^i + \nu_1^i)| \leq r_1$ and $|w_2 - (x_2^i + \nu_2^i)| \leq r_2$ for some $i \in \mathcal{V}_u$ also satisfies $\min_j(y_1^j) \leq w_1 \leq \max_j(y_1^j)$ and $\min_j(y_2^j) \leq w_2 \leq \max_j(y_2^j)$ (ϕ can be padded if this doesn't hold). Let $c \in \mathbb{R}^e$ be a function on the edges such that $c_{i,j} \geq 0$ and $\sum_{j \sim i} c_{i,j} = 1$ for each $i \in \mathcal{V}_u$. Now we model the displacement v to be the difference between the c-weighted average of y and x.

$$(v_1^i, v_2^i) = \left((\sum_{j \sim i} c_{i,j} y_1^i - x_1^i), (\sum_{j \sim i} c_{i,j} y_2^i - x_2^i) \right) \tag{2}$$

These weighted averages can be represented more compactly in terms of the edge-node adjacency matrix $[Q\ R] \in \mathbb{R}^{e \times (M+N)}$ for the graph G, where Q corresponds to the nodes in V_ϕ and R corresponds to the nodes in V_u. For each edge $e_{i,j}$ where $i \in \mathcal{V}_u$ and $j \in \mathcal{V}_\phi$, $Q_{e,j} = -1$ and $R_{e,i} = 1$. All other entries of Q and R equal zero. Let $\operatorname{diag}(c)$ denote the diagonal matrix with the vector c along the diagonal. The operation of taking c-weighted averages of a function on $\mathcal{V}_\phi$ for each node $i \in \mathcal{V}_u$ can be represented in matrix notation by $-(R^T \operatorname{diag}(c) R)^{-1} R^T \operatorname{diag}(c) Q$. The constraint that the weights on the edges coming out of each node i sum to 1 can be written as $R^T c = 1$ or $R^T \operatorname{diag}(c) R = I$. Thus the c-weighted averages of ϕ can be written as $-R^T \operatorname{diag}(c) Q\phi$. Now define

$$A_\phi = -R^T \operatorname{diag}(Q\phi).$$

This means $A_\phi c$ represents the c-weighted averages of ϕ. Similarly, define

$$A_{y_1} = -R^T \operatorname{diag}(Q y_1)$$

and

$$A_{y_2} = -R^T \operatorname{diag}(Q y_2).$$

$A_{y_1} c$ and $A_{y_2} c$ represent the c-weighted averages of y_1 and y_2 respectively. In terms of these matrices, the displacements $v_1, v_2 \in \mathbb{R}^M$ are modeled by

$$v_1 = A_{y_1} c - x_1 \qquad , \qquad v_2 = A_{y_2} c - x_2.$$

Let X denote the convex set that the weights c are constrained to lie in. This set is defined by

$$X = \left\{ c \in \mathbb{R}^e : c_{i,j} \geq 0 \qquad \text{and} \qquad \sum_{j \sim i} c_{i,j} = 1 \right\}. \tag{3}$$

Let g_X be the indicator function for X defined by

$$g_X(c) = \begin{cases} 0 & \text{if } c \in X \\ \infty & \text{otherwise} \end{cases}. \tag{4}$$

We also need to define the discretized gradient D that will act on vectorized images in $\mathbb{R}^M$. The convention for vectorizing a m_r by m_c matrix will be to stack the columns so that the (r, c) element of the matrix corresponds to the $(c-1)m_r + r$ element of the vector. Consider a new graph, $G_D(\mathcal{V}_u, \mathcal{E}_D)$ with nodes in $\mathcal{V}_u$ and edges that correspond to forward differences as is shown in Figure 2. Index the M nodes by $(c-1)m_r + r$ and the $e_D = 2m_r m_c - m_r - m_c$ edges arbitrarily.

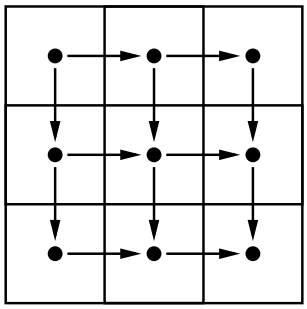

Figure 2: Graph for Defining D

Now define $D \in \mathbb{R}^{e_D \times M}$ to be the edge node adjacency matrix for the graph G_D. For each edge ξ with endpoint indices (i, j), $i < j$, define

$$D_{\xi,k} = \begin{cases} -1 & \text{for } k = i \\ 1 & \text{for } k = j \\ 0 & \text{for } k \neq i, j \end{cases}. \tag{5}$$

The matrix D thus defined can be interpreted as the discrete gradient, and likewise $-D^T$ represents the discrete divergence operator. The graph definition implicitly assumes Neumann boundary conditions.

With this notation we can define the weighted average analogue to (1),

$$\frac{1}{2}\|A_\phi c - u\|_2^2 + \frac{\eta}{2}\|D(A_{y_1} c - x_1)\|_2^2 + \frac{\eta}{2}\|D(A_{y_2} c - x_2)\|_2^2 + g_X(c). \tag{6}$$

We will assume that the displacement is smooth and therefore keep the regularization terms that penalize the l_2 norm squared of its gradient. It will be convenient for notational reasons to define

$$R_1(z) = \frac{\eta}{2}\|D(z - x_1)\|_2^2$$

and

$$R_2(z) = \frac{\eta}{2}\|D(z - x_2)\|_2^2$$

so that these regularization terms can be rewritten as $R_1(A_{y_1}c) + R_2(A_{y_2}c)$. For applications where discontinuities are expected in the displacement field, total variation regularization terms could be used here instead.

Since the displacement is assumed to be smooth, the weights themselves should also be spatially smooth. Moreover enforcing this should encourage the weights to be localized, which needs to happen for the model of the displacement to make sense. For simplicity, assume every node $i \in \mathcal{V}_u$ has the same number of edges connecting it to nodes in $\mathcal{V}_\phi$ and moreover that those nodes in $\mathcal{V}_\phi$ form a rectangle of dimension w_r by w_c that will be referred to as the search window. To be consistent with the bounds r_1 and r_2, we can take $w_r = 2r_2 + 1$ and $w_c = 2r_1 + 1$. There are $W = w_r w_c$ weights for each node in $\mathcal{V}_u$. Note that the total number of edge weights is $e = MW$. Let w be an index for the weights in the search window and define $\mathcal{X}_w \in \mathbb{R}^{M\times e}$ to be a row selector ($\mathcal{X}_w\mathcal{X}_w^T = I$) for the w^{th} weight. So $\mathcal{X}_w$ applied to c returns a vector of just those M weights that correspond to the index w. Now we can encourage spatial smoothness of the weights by adding

$$\sum_{w=1}^{W} \|D\mathcal{X}_w c\|_2$$

to the functional. It will later be helpful to rewrite this in terms of an indicator function g_B for the unit l_2 ball

$$B = \{p : \|p\|_2 \leq 1\}$$

because the numerical approach will involve projecting onto this set. Since the Legendre transform of a norm can be interpreted as the indicator function for the unit ball in the dual norm,

$$\|D\mathcal{X}_w c\|_2 = g_B^*(D\mathcal{X}_w c),$$

where g_B^* denotes the Legendre transform of g_B.

The quadratic data fidelity term doesn't robustly handle outliers. To remedy this, we will replace the quadratic term with convex constraints that control both the local error and the average error for $A_\phi c - u$. To control the local error, we can require that $\|\frac{A_\phi c - u}{\tau}\|_\infty \leq 1$ for some data dependent $\tau \in \mathbb{R}^M$ and where the division by τ is understood to be componentwise. To control the average error we can require that $\|A_\phi c - u\|_2 \leq \epsilon$ for some $\epsilon \geq 0$. These constraints can be added to the functional as indicator functions for the appropriate convex sets. Let

$$T_2 = \{z : \|z - u\|_2 \leq \epsilon\}$$

and let

$$T_\infty = \{z : \|\frac{z - u}{\tau}\|_\infty \leq 1\}.$$

Let g_{T_2} and g_{T_∞} be the indicator functions for T_2 and T_∞. Another possibility is to use the l_1 norm for the data fidelity term, which is considered in Section 5.1.

Altogether, the proposed convex functional for image registration is given by

$$F(c) = g_X(c) + \sum_{w=1}^{W} g_B^*(D\mathcal{X}_w c) + g_{T_2}(A_\phi c) + g_{T_\infty}(A_\phi c) + R_1(A_{y_1}c) + R_2(A_{y_2}c). \tag{7}$$

A minimizer exists as long as the set

$$\{c : c \in X, \|A_\phi c - u\|_2 \leq \epsilon, \|\frac{A_\phi c - u}{\tau}\|_\infty \leq 1\}$$

is nonempty.

3 Numerical Approach

To solve (7) we will use a variant of the PDHG algorithm of [23] discussed in [9], where it is denoted PDHGMp.

3.1 PDHGMp Method

PDHG [23] is an effective method for producing explicit iterations to solve certain large scale non-differentiable convex minimization problems. It can be applied for example to problems of the form

$$\min_u \sum_{i=1}^{N} J_i(A_i u) + H(u), \tag{8}$$

where J_i and H are closed proper convex functions. Let $A = \begin{bmatrix} A_1 \\ \vdots \\ A_N \end{bmatrix}$ and define J so that (8) can be written as

$$\min_u J(Au) + H(u). \tag{9}$$

The main idea of the PDHG algorithm is to work with an equivalent saddle point formulation of the problem, replacing $J(Au)$ with $\sup_p \langle p, Au\rangle - J^*(p)$ to get

$$\min_u \sup_p \langle p, Au\rangle - J^*(p) + H(u). \tag{10}$$

Then it alternates proximal steps that maximize and minimize penalized forms of the saddle function. Writing $p = \begin{bmatrix} p_1 \\ \vdots \\ p_N \end{bmatrix}$, it follows from the definition of the Legendre transform that

$$J^*(p) = \sum_{i=1}^{N} J_i^*(p_i).$$

The resulting PDHG algorithm is to iterate

$$u^{k+1} = \arg\min_u H(u) + \langle \sum_{i=1}^{N} A_i^T p_i^k, u\rangle + \frac{1}{2\alpha}\|u - u^k\|_2^2$$

$$p_i^{k+1} = \arg\min_{p_i} J_i^*(p_i) - \langle p_i, A_i u^{k+1}\rangle + \frac{1}{2\delta}\|p_i - p_i^k\|_2^2 \qquad i = 1, ..., N\ .$$

PDHGMp, the method we will use here, additionally replaces p_i^k in the u^{k+1} update with $2p_i^k - p_i^{k-1}$. By an equivalence with the split inexact Uzawa methods in [22, 9], this modified PDHG algorithm converges as long as a minimizer of F exists and the parameters α and δ satisfy a stability condition.

Theorem 3.1. *[22] Let $\alpha > 0$, $\delta > 0$ and $0 < \delta < \frac{1}{\alpha\|A\|^2}$. Let (p^k, u^k) satisfy the PDHGMp iterations. Also let u^* be optimal for (9). Then*

- $J(Au^k) \to J(Au^*)$
- $H(u^k) \to H(u^*)$

and all convergent subsequences of (p^k, u^k) converge to a saddle point of (10).

3.2 Application of PDHGMp to F

Since F is in the form of (8), the PDHGMp method can be directly applied. However, the relative scaling of the matrices $D\mathcal{X}_w$, A_ϕ, A_{y_1} and A_{y_2} can affect the numerical performance. We therefore introduce scaling factors s_w, s_{ϕ_2}, s_{ϕ_∞}, s_{y_1} and s_{y_2} and define

$$\begin{aligned}
\tilde{g}_B^*(z_w) &= g_B^*(s_w z_w) \\
\tilde{g}_{T_2}(z_{\phi_2}) &= g_{T_2}(s_{\phi_2} z_{\phi_2}) \\
\tilde{g}_{T_\infty}(z_{\phi_\infty}) &= g_{T_\infty}(s_{\phi_\infty} z_{\phi_\infty}) \\
\tilde{R}_1(z_{y_1}) &= R_1(s_{y_1} z_{y_1}) \\
\tilde{R}_2(z_{y_2}) &= R_2(s_{y_2} z_{y_2})
\end{aligned}$$

so that F can be equivalently written as

$$F(c) = g_X(c) + \sum_{w=1}^{W} \tilde{g}_B^*(\frac{D\mathcal{X}_w c}{s_w}) + \tilde{g}_{T_2}(\frac{A_\phi c}{s_{\phi_2}}) + \tilde{g}_{T_\infty}(\frac{A_\phi c}{s_{\phi_\infty}}) + \tilde{R}_1(\frac{A_{y_1} c}{s_{y_1}}) + \tilde{R}_2(\frac{A_{y_2} c}{s_{y_2}}). \tag{11}$$

To apply PDHGMp, let

$$A = \begin{bmatrix} \frac{D\mathcal{X}_1}{s_1} \\ \vdots \\ \frac{D\mathcal{X}_w}{s_W} \\ \frac{A_\phi}{s_\phi} \\ \frac{A_\phi}{s_\phi} \\ \frac{A_{y_1}}{s_{y_1}} \\ \frac{A_{y_2}}{s_{y_2}} \end{bmatrix}, \qquad p = \begin{bmatrix} p_1 \\ \vdots \\ p_W \\ p_{\phi_2} \\ p_{\phi_\infty} \\ p_{y_1} \\ p_{y_2} \end{bmatrix}, \qquad H(c) = g_X(c)$$

and

$$J(Ac) = \sum_{w=1}^{W} \tilde{g}_B^*(\frac{D\mathcal{X}_w c}{s_w}) + \tilde{g}_{T_2}(\frac{A_\phi c}{s_{\phi_2}}) + \tilde{g}_{T_\infty}(\frac{A_\phi c}{s_{\phi_\infty}}) + \tilde{R}_1(\frac{A_{y_1} c}{s_{y_1}}) + \tilde{R}_2(\frac{A_{y_2} c}{s_{y_2}}).$$

The initialization is arbitrary. One iteration of PDHGMp applied to $F(c)$ and including the scale factors consists of the following minimization steps:

$$c^{k+1} = \arg\min_c g_X(c) + \frac{1}{2\alpha}\left\| c - \left(c^k - \alpha \sum_{w=1}^{W} \frac{\mathcal{X}_w^T D^T (2p_w^k - p_w^{k-1})}{s_w} - \alpha \frac{A_\phi^T (2p_{\phi_2}^k - p_{\phi_2}^{k-1})}{s_{\phi_2}} \right.\right.$$
$$\left.\left. -\alpha \frac{A_\phi^T (2p_{\phi_\infty}^k - p_{\phi_\infty}^{k-1})}{s_{\phi_\infty}} - \alpha \frac{A_{y_1}^T (2p_{y_1}^k - p_{y_1}^{k-1})}{s_{y_1}} - \alpha \frac{A_{y_2}^T (2p_{y_2}^k - p_{y_2}^{k-1})}{s_{y_2}} \right) \right\|_2^2$$

$$p_w^{k+1} = s_w \arg\min_{p_w} g_B(p_w) + \frac{s_w^2}{2\delta} \left\| p_w - \frac{(p_w^k + \frac{\delta D \mathcal{X}_w c^{k+1}}{s_w})}{s_w} \right\|_2^2 \qquad \text{for } w = 1, ..., W$$

$$p_{\phi_2}^{k+1} = s_{\phi_2} \arg\min_{p_{\phi_2}} g_{T_2}^*(p_{\phi_2}) + \frac{s_{\phi_2}^2}{2\delta} \left\| p_{\phi_2} - \frac{(p_{\phi_2}^k + \frac{\delta A_\phi c^{k+1}}{s_{\phi_2}})}{s_{\phi_2}} \right\|_2^2$$

$$p_{\phi_\infty}^{k+1} = s_{\phi_\infty} \arg\min_{p_{\phi_\infty}} g_{T_\infty}^*(p_{\phi_\infty}) + \frac{s_{\phi_\infty}^2}{2\delta} \left\| p_{\phi_\infty} - \frac{(p_{\phi_\infty}^k + \frac{\delta A_\phi c^{k+1}}{s_{\phi_\infty}})}{s_{\phi_\infty}} \right\|_2^2$$

$$p_{y_1}^{k+1} = s_{y_1} \arg\min_{p_{y_1}} R_1^*(p_{y_1}) + \frac{s_{y_1}^2}{2\delta} \left\| p_{y_1} - \frac{(p_{y_1}^k + \frac{\delta A_{y_1} c^{k+1}}{s_{y_1}})}{s_{y_1}} \right\|_2^2$$

$$p_{y_2}^{k+1} = s_{y_2} \arg\min_{p_{y_2}} R_2^*(p_{y_2}) + \frac{s_{y_2}^2}{2\delta} \left\| p_{y_2} - \frac{(p_{y_2}^k + \frac{\delta A_{y_2} c^{k+1}}{s_{y_2}})}{s_{y_2}} \right\|_2^2$$

There are simple to compute, explicit formulas for each of the minimization steps. A helpful tool for writing down some of the formulas is the Moreau decomposition [16, 6].

Theorem 3.2. *[6] Let $f \in \mathbb{R}^m$, J a closed proper convex function on $\mathbb{R}^n$, and $A \in \mathbb{R}^{n \times m}$. Then*

$$f = \arg\min_{u \in R^m} J(Au) + \frac{1}{2\alpha} \|u - f\|_2^2 + \alpha A^T \arg\min_{p \in \mathbb{R}^n} J^*(p) + \frac{\alpha}{2} \left\| A^T p - \frac{f}{\alpha} \right\|_2^2. \tag{12}$$

Substituting formulas for the minimization steps and using the Moreau decomposition to simplify the last four updates, the PDHGMp method applied to $F(c)$, still including the scale factors,

is to iterate

$$
\begin{aligned}
c^{k+1} &= \Pi_X \left(c^k - \alpha \sum_{w=1}^{W} \frac{\mathcal{X}_w^T D^T (2p_w^k - p_w^{k-1})}{s_w} - \alpha \frac{A_\phi^T (2p_{\phi_2}^k - p_{\phi_2}^{k-1})}{s_{\phi_2}} \right. \\
&\qquad \left. -\alpha \frac{A_\phi^T (2p_{\phi_\infty}^k - p_{\phi_\infty}^{k-1})}{s_{\phi_\infty}} - \alpha \frac{A_{y_1}^T (2p_{y_1}^k - p_{y_1}^{k-1})}{s_{y_1}} - \alpha \frac{A_{y_2}^T (2p_{y_2}^k - p_{y_2}^{k-1})}{s_{y_2}} \right) \\
p_w^{k+1} &= s_w \Pi_B \left(\frac{(p_w^k + \frac{\delta D \mathcal{X}_w c^{k+1}}{s_w})}{s_w} \right) \qquad \text{for } w = 1, ..., W \\
p_{\phi_2}^{k+1} &= p_{\phi_2}^k + \frac{\delta A_\phi c^{k+1}}{s_{\phi_2}} - \frac{\delta}{s_{\phi_2}} \Pi_{T_2} \left((p_{\phi_2}^k + \frac{\delta A_\phi c^{k+1}}{s_{\phi_2}}) \frac{s_{\phi_2}}{\delta} \right) \\
p_{\phi_\infty}^{k+1} &= p_{\phi_\infty}^k + \frac{\delta A_\phi c^{k+1}}{s_{\phi_\infty}} - \frac{\delta}{s_{\phi_\infty}} \Pi_{T_\infty} \left((p_{\phi_\infty}^k + \frac{\delta A_\phi c^{k+1}}{s_{\phi_\infty}}) \frac{s_{\phi_\infty}}{\delta} \right) \\
p_{y_1}^{k+1} &= p_{y_1}^k + \frac{\delta A_{y_1} c^{k+1}}{s_{y_1}} - (I + \frac{\eta s_{y_1}^2}{\delta} D^T D)^{-1} \left(\eta s_{y_1} D^T D x_1 + p_{y_1}^k + \frac{\delta A_{y_1} c^{k+1}}{s_{y_1}} \right) \\
p_{y_2}^{k+1} &= p_{y_2}^k + \frac{\delta A_{y_2} c^{k+1}}{s_{y_2}} - (I + \frac{\eta s_{y_2}^2}{\delta} D^T D)^{-1} \left(\eta s_{y_2} D^T D x_2 + p_{y_2}^k + \frac{\delta A_{y_2} c^{k+1}}{s_{y_2}} \right)
\end{aligned}
$$

Here Π_X, Π_B, Π_{T_2} and Π_{T_∞} denote the orthogonal projections onto the convex sets X, B, T_2 and T_∞ respectively. Formulas for the latter three are

$$\Pi_B(p) = \frac{p}{\max(\|p\|_2, 1)},$$

$$\Pi_{T_2}(z) = u + \frac{z-u}{\max(\frac{\|z-u\|_2}{\epsilon}, 1)}$$

and

$$\Pi_{T_\infty}(z) = u + \frac{z-u}{\max(\frac{|z-u|}{\tau}, 1)},$$

where for Π_{T_∞} the max and division are understood in a componentwise sense. Although there isn't a formula for $\Pi_X(c)$, it can still be computed efficiently with complexity of $O(MW \log(W))$. In describing this projection, it is helpful to reindex c. Computing $\Pi_X(c)$, where $c \in \mathbb{R}^{MW}$, amounts to orthogonally projecting M vectors in $\mathbb{R}^W$ onto the positive face of the l_1 unit ball in $\mathbb{R}^W$. Let $c_m \in \mathbb{R}^W$ for $m = 1, ..., M$ denote those vectors. Then the elements $c_{m,w}$ of c_m must project either to zero or to $c_{m,w} - \theta_m$, where θ_m is such that $\sum_{w=1}^W c_{m,w} = 1$ and $c_{m,w} \geq 0$. Therefore the projection can be computed once the thresholds θ_m are found, which can be done by using a bisection strategy to determine how many elements project to zero. Finally note that $D^T D$ denotes minus the discrete Laplacian corresponding to Neumann boundary conditions. The corresponding discrete Poisson equation can be efficiently solved with $O(M \log(M))$ complexity using the discrete cosine transform.

3.3 Discussion of Parameters

There are some necessary conditions on the parameters. The parameters $\epsilon \in \mathbb{R}$ and $\tau \in \mathbb{R}^M$, which appear in the definition of the sets T_2 and T_∞, must be large enough so that a minimizer exists. Also,

α and δ must be positive and satisfy $\alpha\delta < \frac{1}{\|A\|^2}$. Adjusting ϵ and τ changes the underlying model and will affect the solution. Changing α and δ, as long as they satisfy the stability requirement, only affects the rate of convergence.

The scaling factors s_w, s_{ϕ_2}, s_{ϕ_∞}, s_{y_1} and s_{y_2} also affect the rate of convergence but mostly they alter the relative weight that each term of the functional has on each iteration. Adjusting these factors doesn't change the model or the eventual solution, but it can for example make the numerical solution satisfy the data constraint in fewer iterations at the cost of it taking more iterations for the weights to become smooth. Or vice versa, the scaling factors can encourage early iterates to be smooth at the cost of more iterations being needed to satisfy the data fidelity constraints. A natural approach to defining the scaling parameters is to try to give the five terms (thinking of the sum over w as a single term) in $J(Ac)$ roughly equal weight.

The parameter η does affect the model in the sense of altering the relative importance of the smooth displacement regularizer and the smooth weights regularizer. If η is too large, then a smooth displacement might come at the cost of having insufficiently smooth and therefore possibly nonlocal weights. If η is too small, the weights themselves will be spatially smooth, but although the smoothness of the displacement is indeed encouraged by smoothness of the weights, it is not always sufficiently enforced that way. Moreover, it empirically takes many more iterations to get a reasonable solution when η is too small. For some examples, choosing η too small can again result in nonlocal weights. Since the displacement model assumes local weights, it's therefore important to choose η well to avoid errors in the registration.

3.4 Multiscale Approach

A downside of the model (7) is the large number of variables involved. Although it's a convex registration model that allows for large deformations, the number of edge weights in the graph formulation can be impractically large. By practical necessity, a coarse to fine multiscale approach is used for the numerical examples in Section 4. A dimension reduction idea is mentioned in Section 5.5 but not implemented.

The multiscale approach works by downsampling the original images by a factor of two as many times as is necessary for the number of pixels within the maximum displacement estimate not to be impractically large. The effect of one level of downsampling is illustrated in Figure 3. The convex registration problem is first solved for the low resolution images. Then the resulting displacement solution given by $(A_{y_1}c - x_1, A_{y_2}c - x_2)$ is upsampled by a factor of two and used as an initial guess when solving the convex registration problem for the next finer resolution. If one assumes the global coarse solution is close, say within half a coarse pixel, to the global fine solution, then small search windows suffice for all the successive applications of the method to the finer resolution images. For example, after the coarse problem is solved, a three by three window of weights could be used for the remaining problems.

An advantage of the multiscale approach is that by downsampling enough times, the size of the search window of weights can be made small enough to automatically satisfy the localized weights assumption. And the number of extra variables can be made small enough so that the computation remains efficient. However, the coarse solutions aren't guaranteed to be close to the true global minimum of the functional at the fine scale. In fact, downsampling too many times will result in poor solutions. So it is best to employ the multiscale strategy as little as possible, only as much as is practically necessary to achieve reasonable computation times.

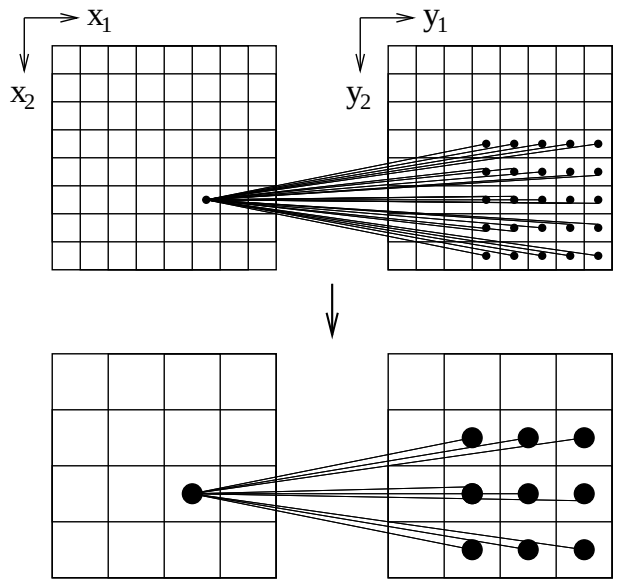

Figure 3: Effect of downsampling on resolution and search window size

4 Numerical Examples

In this section, the performance of the model and numerical approach is illustrated for three examples.

The first example is a synthetic image of the letter E which is to be registered with a translated, rotated version. See Figure 4. This example illustrates that the model succeeds in filling in the large homogeneous regions with a smooth displacement field.

The second example is low resolution digital photo of two pencils on a desk which is to be registered with a version that has undergone a large translation and also some rotation. See Figure 5. The initial displacement is chosen to align the rightmost pencil in u with the leftmost pencil in ϕ. This would be a local minimum for the classical registration model (1), but the global solution of the convex model correctly registers the images despite the large translation and challenging initialization. The pencil example will also be used to illustrate the problems that occur if η is too small or if the fidelity terms are too weak. These both result in nonlocal weights and a poor solution, whereas with well chosen parameters the weights do indeed localize and the solution is good.

The third example is a brain scan where a section in the middle has been deformed, leaving the outer regions unchanged. See Figure 6. This example is from [18, 15] and the ground truth is known. A comparison with the ground truth displacement is plotted in Figure 7. In addition to showing that the model can successfully register this medical image, the example is also used to illustrate the need for the smoothing regularizer on the weights. If η is chosen too large, the displacement should still of course be smooth but the weights themselves may not be smooth enough. It's then possible for nonlocal weights to conspire to satisfy the data fidelity constraints and yield a smooth displacement while still giving the wrong solution.

4.1 Parameter Definitions

Most of the parameters are chosen similarly for the numerical examples. The parameter η that balances the smooth displacement and smooth weights regularizers is usually chosen to be 1 except for the examples that illustrate what goes wrong when η is too small or too large. The weights $\tau \in \mathbb{R}^m$ for the l_∞ data constraint are data dependent, small in homogeneous regions and large near discontinuities. For all the examples τ_i is defined by taking the difference of the maximum and minimum intensities in the three by three neighborhood around the i^{th} pixel, multiplying by .75 and adding .5. This ensures $|A_\phi c - u|$ is never much larger that could be expected from interpolation errors. Recall that $A_\phi c - u$ is the difference between the c-weighted averages of ϕ and u. For the l_2 data fidelity constraint, the best choice of ϵ depends on the problem. The scaling parameters are designed to normalize the matrices $\sum_{w=1}^{W} \mathcal{X}_w^T D^T$, A_ϕ^T, $A_{y_1}^T$ and $A_{y_2}^T$ so that they have roughly the same operator norms. For all the numerical examples, define

$$\begin{aligned} s_w &= 2\sqrt{10W} \\ s_{\phi_2} &= \|\phi\|_\infty \sqrt{5W} \\ s_{\phi_\infty} &= s_{\phi_2} \\ s_{y_1} &= n_c \Delta_{y_c} \sqrt{5W} \\ s_{y_2} &= n_r \Delta_{y_r} \sqrt{5W}, \end{aligned}$$

where Δ_{y_c} and Δ_{y_r} denote the dimensions of a single pixel in ϕ. The choice of numerical parameters α and δ can greatly affect the rate of convergence. It is best for $\alpha\delta$ to be close to the stability bound $\frac{1}{\|A\|^2}$. An upper bound, a for $\|A\|^2$ is

$$a = \frac{8W}{s_w^2} + \| \frac{A_\phi A_\phi^T}{s_{\phi_2}^2} + \frac{A_\phi A_\phi^T}{s_{\phi_\infty}^2} + \frac{A_{y_1} A_{y_1}^T}{s_{y_1}^2} + \frac{A_{y_2} A_{y_2}^T}{s_{y_2}^2} \|,$$

which is straightforward to compute because $A_\phi A_\phi^T$, $A_{y_1} A_{y_1}^T$ and $A_{y_2} A_{y_2}^T$ are diagonal matrices. Reasonable choices for α and δ are $\alpha = \frac{.995}{s_{\phi_2}\sqrt{a}}$ and $\delta = \frac{.995 s_{\phi_2}}{\sqrt{a}}$.

4.2 Multiscale Implementation and Stopping Condition

To speed up the numerical implementation, a coarse to fine multiscale approach as described in 3.4 is used for all the following examples. The letter E and pencil examples are both initially downsampled twice. The brain example is initially downsampled three times. The downsampling is always by a factor of two. Once a coarse solution is obtained and the displacement computed, bilinear interpolation is used on the coarse displacement field to obtain an initial guess for the next finer resolution. Since the coarse solution is assumed to be close to the true solution, small search windows, (usually 3×3 or 5×5), are used for all the finer resolution registration problems with good initial displacement estimates.

The stopping condition used for the E and pencil examples is

$$\|c^{k+1} - c^k\|_\infty \leq \frac{.002}{W}, \tag{13}$$

where W is the total number of weights in each search window. To speed up computation for the brain example, .002 was replaced by .004.

scale	iterations
2	3890
1	2348
0	1552

Table 1: Iterations required for registering E example

4.3 Results

For the letter E example, Figure 4 shows the original 128×128 images u and ϕ, the c-weighted averages of ϕ and the computed displacement. The parameters used are $\alpha = \frac{.995}{s_{\phi_2}\sqrt{a}}$, $\delta = \frac{.995 s_{\phi_2}}{\sqrt{a}}$, $\epsilon = \frac{\sqrt{M}}{2}$ and $\eta = 1$. The maximum displacement (r_1, r_2) is set to $(16, 8)$. For all scales beyond the coarsest, three by three search windows are used. The number of iterations required at each scale to satisfy the stopping condition (13) is given in Table 1, where scale refers to the number of downsamplings.

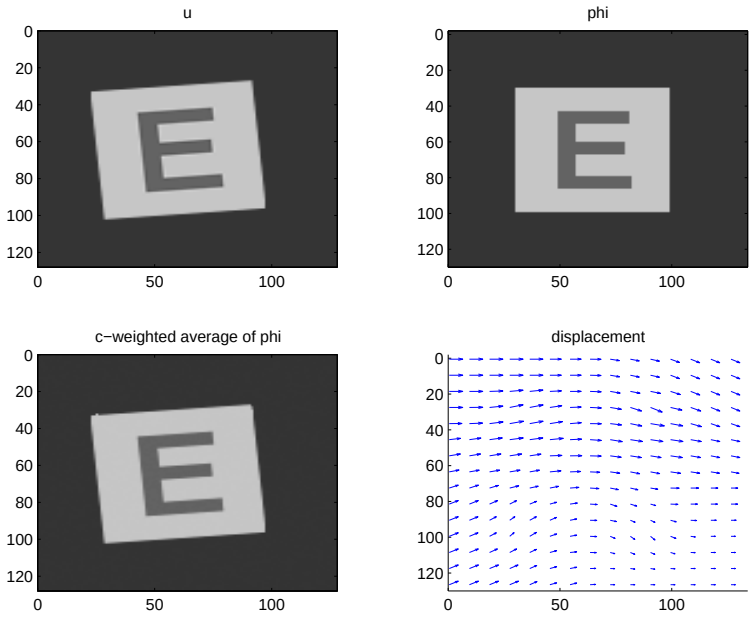

Figure 4: Registration of rotated and translated letter E

The pencil registration result plotted in Figure 5 uses the same parameters as the E example except that five by five search windows are used after the coarse solution is obtained. Recall that the initial displacement is chosen to make this problem challenging by lining up the right pencil in u with the left pencil in ϕ. The displacement plot shows the location of the 46×38 image u, bordered in black, relative to the 79×98 image ϕ and draws the displacement field at a few points. The maximum displacement (r_1, r_2) is set to $(32, 16)$. Table 2 shows the number of iterations required.

Some parameters are changed for the brain example, where u and ϕ are both 186×197 images. The maximum displacement (r_1, r_2) is set to $(10, 10)$. Since the displacement is expected to be

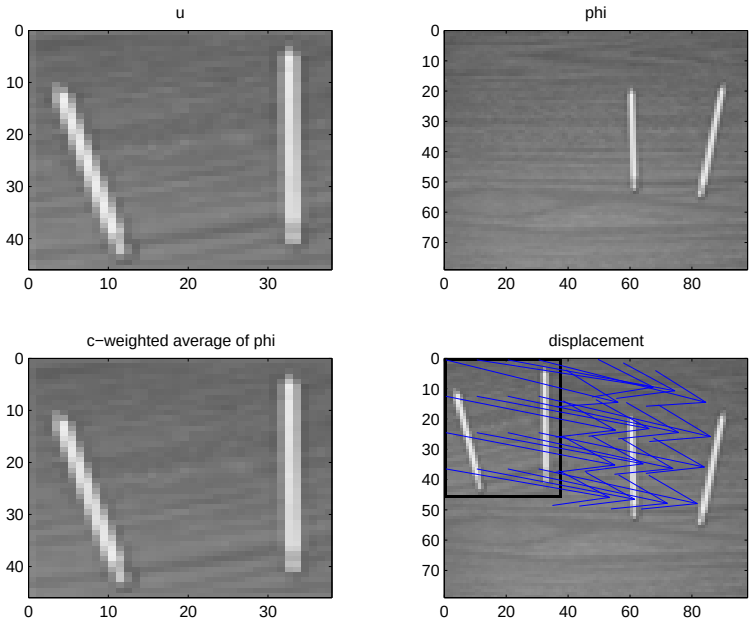

Figure 5: Registration of Low Resolution Photo of Two Pencils

scale	iterations
2	14276
1	6560
0	10403

Table 2: Iterations required for registering pencil example

mostly zero, a smaller ϵ is used, namely $\epsilon = \frac{\sqrt{M}}{16}$. Adjusting α and δ to $\alpha = \frac{.0995}{s_{\phi_2}\sqrt{a}}$, $\delta = \frac{9.95 s_{\phi_2}}{\sqrt{a}}$ sped up the rate of convergence slightly for this example. Also, at the four scales computed in the multiscale approach, a three by three search window was used for the finest while five by five search windows were used for the two intermediate scales after the coarsest. The registration result is shown in Figure 6 and the number of iterations required are listed in Table 3. For this example, the ground truth displacement is known and compared to the computed displacement in Figure 7. The root mean square errors relative to the ground truth for the displacement components v_1 and v_2 are .3995 and .5748 respectively.

Although the convex registration model was successful on the previous examples, it can fail if the parameters are not well chosen. Three examples of what can go wrong are when the data fidelity constraints are too weak, when η is too small or even when η is too large.

The pencil example will be used to illustrate the first two potential problems. The solution of the coarse, twice downsampled, problem suffices to demonstrate this. Figure 8 shows the c-weighted average of ϕ, the displacement, the weights corresponding to index $w = 77$ and the weights corresponding to pixel $m = 102$ that result from choosing $\epsilon = 100\sqrt{M}$ or choosing $\eta = 10^{-12}$. Since the l_∞ constraint was already weak, the c-weighted average is not a good approximation to u when

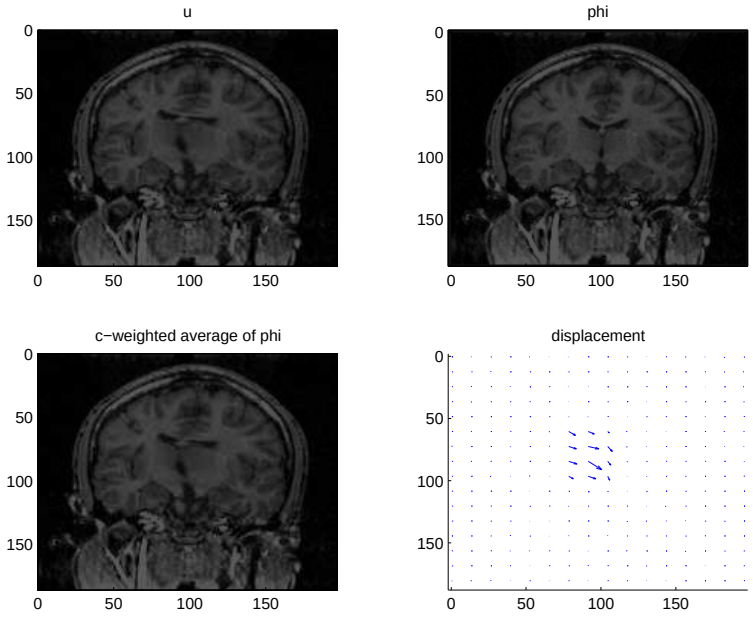

Figure 6: Registration of brain images

scale	iterations
3	3914
2	4919
1	6813
0	11303

Table 3: Iterations required for registering brain example

ϵ is large. Moreover, the computed displacement is poor. With $\epsilon = \frac{\sqrt{M}}{2}$ but η too small, the c-weighted average accurately approximates u but the resulting displacement is not smooth or accurate. Part of the reason for these poor results is the nonlocal weights. Weights that should be concentrated on one pencil instead appear on both, which means the weighted averages of the locations in ϕ are not even close to where the weights are large. The nonlocal weights that result from poor parameter choices are also illustrated in Figure 8.

A large value of η actually works fine for the pencil example, but not for the brain example. Looking at the coarse, three times downsampled problem, Figure 9 compares the good brain image registration result for $\eta = 1$ to the poor result when $\eta = 10^{12}$. In both examples, $\epsilon = \frac{\sqrt{M}}{16}$ and the stopping condition was $\|c^{k+1} - c^k\|_\infty \leq \frac{.0001}{W}$. In the case where η is too large, the displacement is zero in many places where it shouldn't be. Figure 9 shows that for $m = 247$ corresponding to such a location, the c_m weights are more localized for the $\eta = 1$ case than for the large η case. The figure also shows that example c_w for $w = 19$ and $w = 13$ are less smooth in the large η case.

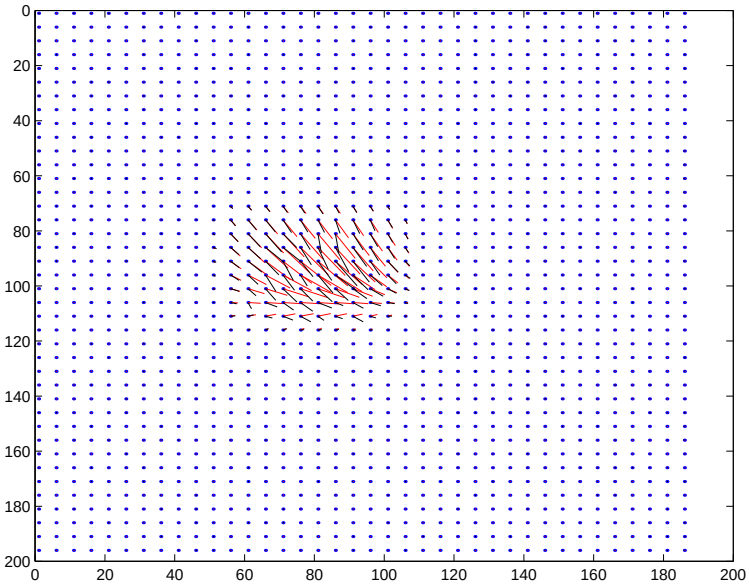

Figure 7: Comparison of brain registration to ground truth displacement

5 Modifications and Other Applications

5.1 Using Other Norms

Some image registration problems are better modeled using the l_1 norm or the TV seminorm for either the data fidelity or the regularization terms. This is the case when one expects discontinuities in the displacement field or a sparse difference between the original and registered images. If the images have slight changes in intensity, it makes sense to register the gradients of u and ϕ. Even with all these possible changes to the model, the PDHGMp method is still applicable and the overall numerical approach is very similar.

The optical flow problem of registering successive frames in a video sequence is an example where it would make sense to use the l_1 norm for data fidelity and total variation to regularize the displacement field. The convex registration model, however, is not ideal for optical flow problems. When such problems already satisfy a small deformation assumption, the convexity of the model isn't really needed. Moreover, speed of the algorithm is especially important for video applications and the convex model is slow because of the many extra variables it must take into account. Nevertheless, it serves as a good illustration of how to substitute other norms into (7) and still apply PDHGMp to minimize the functional.

Consider the following TV-l_1 model for optical flow,

$$F(c) = g_X(c) + \sum_{w=1}^{W} \|\mathcal{X}_w c\|_{TV} + g_T(A_\phi c) + \eta\|A_{y_1} c - x_1\|_{TV} + \eta\|A_{y_2} c - x_2\|_{TV},$$

where g_T is the indicator function for $T = \{z : \|z - u\|_1 \leq \epsilon\}$. The total variation seminorm is

defined by

$$\|u\|_{TV} = \sum_{q_1=1}^{m_r} \sum_{q_2=1}^{m_c} \sqrt{(D_1^+ u_{q_1,q_2})^2 + (D_2^+ u_{q_1,q_2})^2},$$

where D_i^+ represents a forward difference in the i^{th} index and we assume Neumann boundary conditions. From the definition of the discrete gradient D (5) we can define a matrix E by

$$E_{\eta,k} = \begin{cases} 1 & \text{if } D_{\eta,k} = -1, \\ 0 & \text{otherwise} \end{cases}$$

and use this to define a norm $\|\cdot\|_E$ on $\mathbb{R}^{e_D}$,

$$\|w\|_E = \left\| \sqrt{E^T(w^2)} \right\|_1,$$

so that

$$\|Du\|_E = \|u\|_{TV}.$$

Note that the dual norm $\|p\|_{E^*} = \|\sqrt{E^T(p^2)}\|_\infty$. See [9] for details. Let g_B now be defined as the indicator function for $B = \{p : \|p\|_{E^*} \leq 1\}$. Then the functional can be rewritten as

$$F(c) = g_X(c) + \sum_{w=1}^{W} g_B^*(D\mathcal{X}_w c) + g_T(A_\phi c) + \eta g_B^*(D(A_{y_1} c - x_1)) + \eta g_B^*(D(A_{y_2} c - x_2)).$$

Applying PDHGMp analogous to the way it was applied in Section 3.2, again with scaling factors, yields the following iterations,

$$\begin{aligned}
c^{k+1} &= \Pi_X \left(c^k - \alpha \sum_{w=1}^{W} \frac{\mathcal{X}_w^T D^T (2p_w^k - p_w^{k-1})}{s_w} - \alpha \frac{A_\phi^T (2p_\phi^k - p_\phi^{k-1})}{s_\phi} \right. \\
&\qquad \left. - \alpha \frac{A_{y_1}^T D^T (2p_{y_1}^k - p_{y_1}^{k-1})}{s_{y_1}} - \alpha \frac{A_{y_2}^T D^T (2p_{y_2}^k - p_{y_2}^{k-1})}{s_{y_2}} \right) \\
p_w^{k+1} &= s_w \Pi_B \left(\frac{(p_w^k + \frac{\delta D \mathcal{X}_w c^{k+1}}{s_w})}{s_w} \right) \qquad \text{for } w = 1, ..., W \\
p_\phi^{k+1} &= p_\phi^k + \frac{\delta A_\phi c^{k+1}}{s_\phi} - \frac{\delta}{s_\phi} \Pi_T \left((p_\phi^k + \frac{\delta A_\phi c^{k+1}}{s_\phi}) \frac{s_\phi}{\delta} \right) \\
p_{y_1}^{k+1} &= \eta s_{y_1} \Pi_B \left(\frac{(p_{y_1}^k + \frac{\delta D(A_{y_1} c^{k+1} - x_1)}{s_{y_1}})}{\eta s_{y_1}} \right) \\
p_{y_2}^{k+1} &= \eta s_{y_2} \Pi_B \left(\frac{(p_{y_2}^k + \frac{\delta D(A_{y_2} c^{k+1} - x_2)}{s_{y_2}})}{\eta s_{y_2}} \right),
\end{aligned}$$

where $\Pi_B(p) = \frac{p}{E \max\left(\sqrt{E^T(p^2)}, 1\right)}$ with the max and division understood in a componentwise sense. Although there isn't an explicit formula for Π_T, the orthogonal projection onto T, its computation is very similar to that for Π_X as described in Section 3.2 and can be computed with $O(M \log M)$ complexity.

5.2 Different Multiscale Strategies

Although the convex registration model presented here doesn't require a multiscale numerical approach, it is impractically slow without it. Even with the coarse to fine approach, it took 20 to 30 minutes for the E and pencil examples and several hours for the brain example. Most of this time was needed for computing the solutions at the finest scales. However, after solving the coarse problem, a small deformation assumption is satisfied for all problems at finer resolutions. Therefore, it might make sense to solve the convex model only for the coarsest problem in the multiscale approach, switching perhaps to a more efficient linearized version of something like (1) for the finer scales.

5.3 More Implicit Numerical Methods

The special structure of the matrices A_ϕ, A_{y_1}, A_{y_2} and $D\mathcal{X}_w$ suggests that a more implicit algorithm than PDHGMp might work well. Since $A_\phi A_\phi^T$, $A_{y_1} A_{y_1}^T$ and $A_{y_2} A_{y_2}^T$ are diagonal, terms like $I + A_\phi^T A_\phi$, $I + A_{y_1}^T A_{y_1}$ and $I + A_{y_2}^T A_{y_2}$ are easy to invert using the Sherman Morrison Woodbury formula. A term like $I + \sum_w \mathcal{X}_w^T D^T D \mathcal{X}_w$ is also easy to deal with because with the proper indexing it is a block diagonal matrix with I minus the discrete Laplacian as each block. So, with the addition of some extra variables, the application of split Bregman [11] yields simple iterations. The equivalent application of the alternating direction method of multipliers (ADMM) [7] to the Lagrangian

$$\begin{aligned} L = & \, g_X(c) \\ & + \sum_{w=1}^{W} \left(g_B^*(z_w) + \langle p_w, D\mathcal{X}_w u_1 - z_w \rangle \right) + \langle r_1, c - u_1 \rangle \\ & + g_{T_2}(z_{\phi_2}) + \langle p_{\phi_2}, A_\phi u_2 - z_{\phi_2} \rangle + \langle r_2, c - u_2 \rangle \\ & + g_{T_\infty}(z_{\phi_\infty}) + \langle p_{\phi_\infty}, A_\phi u_3 - z_{\phi_\infty} \rangle + \langle r_3, c - u_3 \rangle \\ & + R_1(z_{y_1}) + \langle p_{y_1}, A_{y_1} u_4 - z_{y_1} \rangle + \langle r_4, c - u_4 \rangle \\ & + R_2(z_{y_2}) + \langle p_{y_2}, A_{y_2} u_5 - z_{y_2} \rangle + \langle r_5, c - u_5 \rangle \end{aligned}$$

was attempted but found to be slightly less efficient for the examples tested. With the introduction of scaling parameters similar to those used for PDHGMp, ADMM required similar numbers of iterations to meet the same stopping criteria. However, the more implicit ADMM iterations were more time consuming and memory intensive. It may still be possible to improve the performance with better parameter choices or by working with a different Lagrangian formulation that involves fewer variables but requires solving slightly more complicated linear systems for some of the subproblems.

5.4 Constraint Relaxation

It's possible to slightly speed up the iterations for PDHGMp applied to (11) by splitting up the constraint $c \in X$ into two separate constraints. Recall that this normalization constraint enforces that for each pixel $m \in \mathbb{R}^m$, the corresponding vector of weights $c_m \in \mathbb{R}^W$ has to be nonnegative and sum to one. Consider adding a new variable s constrained to equal c such that s is constrained to be nonnegative and each vector c_m is constrained to sum to one. Numerically this can be handled by introducing indicator functions for these constraints into the model and applying the

split inexact Uzawa method from [22] to the Lagrangian

$$g_{\{s\geq 0\}}(s) + g_{\{\sum_w c_{m,w}=1\}}(c) + J(z) + \langle p, Ac - z\rangle + \langle \lambda, c - s\rangle,$$

where p and λ are Lagrange multipliers for the $Ac = z$ and $s = c$ constraints. Compared to the PDHGMp implementation, the c update is replaced by

$$c_m^{k+1} = \frac{1}{W} + (c^k - \alpha A^T(2p^k - p^{k-1}) - \alpha\lambda^k)_m - \text{mean}((c^k - \alpha A^T(2p^k - p^{k-1}) - \alpha\lambda^k)_m) \qquad m = 1, ..., M,$$

$$s^{k+1} = \max(c^{k+1} + \frac{\lambda^k}{\delta}, 0),$$

the p updates are identical, and there is an additional update for the multiplier λ,

$$\lambda^{k+1} = \lambda^k + \delta(c^{k+1} - s^{k+1}).$$

The projection onto X which could be computed with complexity $O(MW\log(W))$ has now been replaced with two simpler projections that have complexity $O(MW)$. This does indeed speed up each iteration, but more iterations are also required, especially when M is large. Overall the method tends not to be any faster unless W is large or M is small.

5.5 Dimension Reduction

A possible idea for speeding up the method without resorting to a multiscale approach is to try to approximate c by a linear transformation of a lower dimensional vector s. The motivation for this kind of dimension reduction is that the $c_w \in \mathbb{R}^M$ represent smooth images and can therefore be well approximated by, for example, the low frequency terms in its representation via the discrete cosine transform (DCT). Putting these $l < M$ low frequency DCT basis vectors in the columns of $\Psi \in \mathbb{R}^{M\times l}$, we can try to represent $c_w = \Psi s_w$ for $w = 1, ..., W$. A difficulty with this approach is finding feasible constraints on s so that $c \in X$. The situation can be somewhat simplified by using the above constraint relaxation idea, but we still must have $\Psi s_w \geq 0$ and $\Psi \sum_w s_w = 1$. Assuming $\Psi^T\Psi = I$, the overall constraints on s could be written

$$\sum_{w=1}^{W} s_w = \Psi^T 1 \qquad , \qquad \Psi s_w \geq 0 \qquad \forall w.$$

If this approach is to succeed, it will likely be necessary to redesign Ψ so that the constraints on s are feasible and also computable without reintroducing the larger vector c that we are trying to approximate in the first place.

5.6 Application to Convexified Multiphase Segmentation

Even if the convex model presented here is ultimately not practical for image registration applications, the numerical approach should be useful for some similar problems. An example is the convexified multiphase segmentation model proposed in [20] and discussed in [4, 3], which seeks to minimize a functional of the form

$$g_{\triangle}(u) + \sum_{i=1}^{N} \left(\|u_i\|_{TV} + \langle u_i, f_i\rangle\right),$$

where

$$\triangle(u) = \{u = (u_1, ..., u_N) : u_i \in \mathbb{R}^M, \sum_{i=1}^{N} u_i = 1, u_i \geq 0\}$$

and $g_\triangle$ is the indicator function for $\triangle$. This is precisely the same kind of constraint that the weights c in the registration model must satisfy. PDHGMp can be applied in the same way it was used here, with the analogous orthogonal projection onto X, or in this notation $\triangle$, at each iteration. The PDHGMp iterations are given by easy to compute orthogonal projections

$$\begin{aligned} u^{k+1} &= \Pi_\triangle \left(u^k - \alpha \sum_{i=1}^{N} (\mathcal{X}_i^T D^T (2p_i^k - p_i^{k-1}) + f_i) \right) \\ p_i^{k+1} &= \Pi_{\{p:\|p\|_{E^*} \leq 1\}} (p_i^k + \delta D \mathcal{X}_i u^{k+1}) \qquad \text{for } i = 1, ..., N, \end{aligned}$$

where $\mathcal{X}_i$ is a row selector for the u_i labels and $\|\cdot\|_{E^*}$ is the dual norm for $\|\cdot\|_E$ defined in Section 5.1. PDHGMp should be more efficient for this segmentation application than it was for the registration problem because the number of segmentation phases is likely far less than the number of weights in the convex registration model for a large deformation problem.

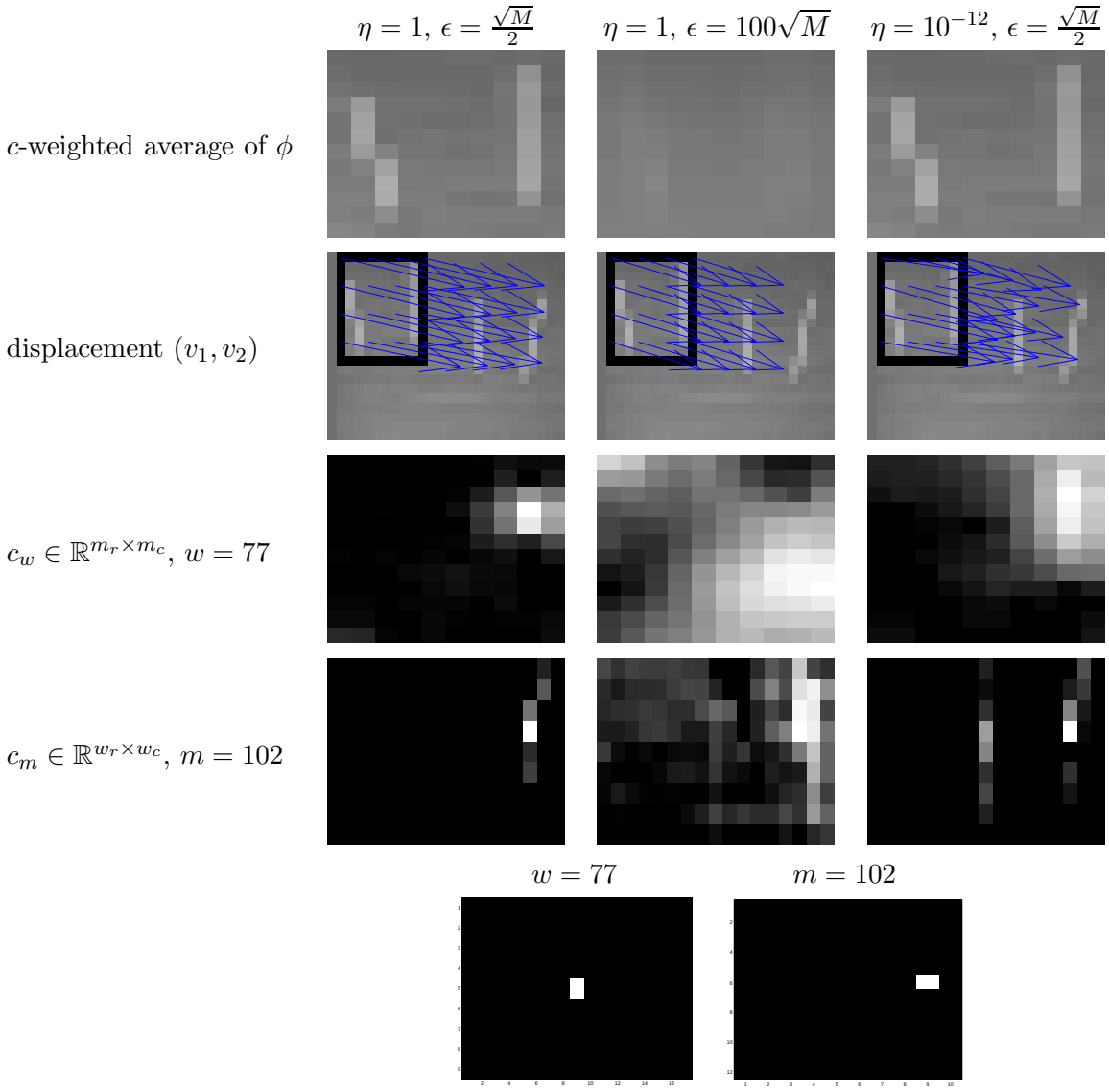

Figure 8: Comparison of coarse pencil registration results

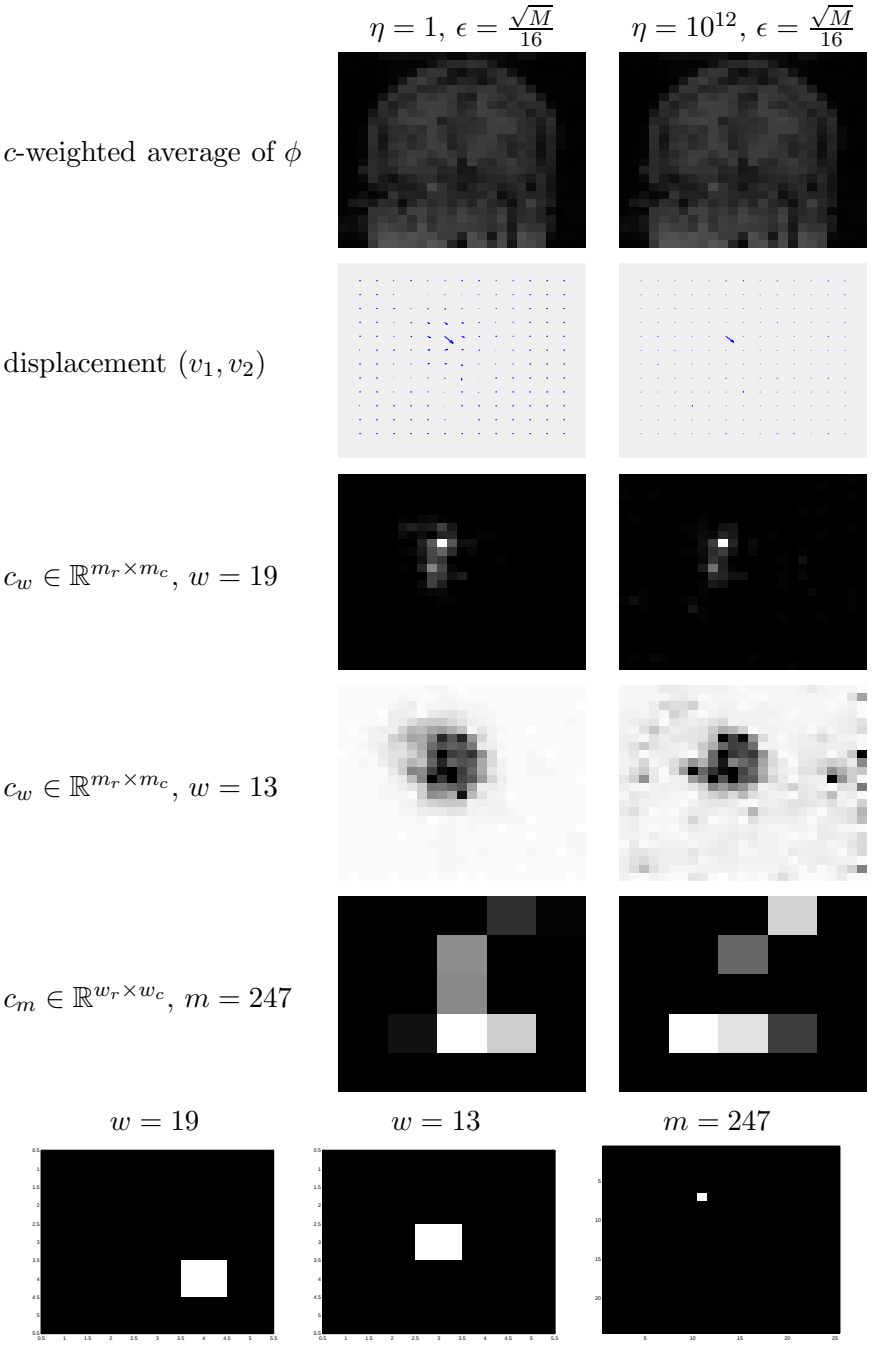

Figure 9: Comparison of coarse brain image registration results

References

[1] G. Aubert, and P. Kornprobst, *Mathematical Problems in Image Processing*, Springer, Applied Mathmetical Sciences, Vol 147, 2006.

[2] D. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Athena Scientific, 1996.

[3] E. Bae, J. Yuan, and X. Tai, *Global Minimization for Continuous Multiphase Partitioning Problems Using a Dual Approach*, UCLA CAM Report [09-75], Sept. 2009.

[4] E. Brown, T.F. Chan, and X. Bresson, *Convex Formulation and Exact Global Solutions for Multi-phase Piecewise Constant Mumford-Shah Image Segmentation*, UCLA CAM Report [09-66], August 2009.

[5] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, *High Accuracy Optical Flow Estimation Based on a Theory for Warping*, ECCV, Springer LNCS 3024, Vol. 4, pp. 25-36, 2004.

[6] P. Combettes, and W. Wajs, *Signal Recovery by Proximal Forward-Backward Splitting*, Multiscale Model. Simul., 2006.

[7] J. Eckstein and D. Bertsekas, *On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators*, Mathematical Programming 55, North-Holland, 1992.

[8] A. Elmoataz, O. Lezoray, and S. Bougleux, *Nonlocal Discrete Regularization on Weighted Graphs: A framework for Image and Manifold Processing*, IEEE, Vol. 17, No. 7, July 2008.

[9] E. Esser, X. Zhang, and T. F. Chan, *A General Framework for a Class of First Order Primal-Dual Algorithms for TV Minimization*, UCLA CAM Report [09-67], August 2009.

[10] T. Goldstein, X. Bresson, and S. Osher, *Global Minimization of Markov Random Fields with Applications to Optical Flow*, UCLA CAM Report [09-77], Sept. 2009.

[11] T. Goldstein and S. Osher, *The Split Bregman Algorithm for L1 Regularized Problems*, UCLA CAM Report [08-29], April 2008.

[12] B. Horn and B. Schunck, *Determining Optical Flow*, Artificial Intelligence, 17:185-203, 1981.

[13] H. Ishikawa, *Exact Optimization for Markov Random Fields with Convex Priors*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 25:1333-1336, 2003.

[14] M. Lefébure and L. Cohen, *Image Registration, Optical Flow and Local Rigidity*, Journal of Mathematical Imaging and Vision 14: 131-147, 2001.

[15] T. Lin, C. Le Guyader, I. Dinov, P. Thompson, A. Toga, and L. Vese, *A Landmark-Based Image Registration Model using a Nonlinear Elasticity Smoother for Mapping Mouse Atlas to Gene Expression Data*, UCLA CAM Report [09-51], June 2009.

[16] J. J. Moreau, *Proximité et dualité dans un espace hilbertien*, Bull. Soc. Math. France, 93, 1965, pp. 273-299.

[17] T. Pock, T. Schoenemann, G. Graber, H. Bischof, and D. Cremers, *A Convex Formulation of Continuous Multi-Label Problems*, ECCV, Springer-Verlag, Proceedings of the 10th European Conference on Computer Vision, pp. 792-805, 2008.

[18] H. Tagare, D. Groisser, and O. Skrinjar, *A geometric theory of symmetric registration*, In Proceedings of CVPRW'06 IEEE, 2006.

[19] I. Yanovsky, *Unbiased Nonlinear Image Registration*, PhD thesis, UCLA CAM Report [08-30], May 2008.

[20] C. Zach, D. Gallup, J.-M. Frahm, and M. Niethammer, *Fast global labeling for real-time stereo using multiple plane sweeps*, Vision, Modeling and Visualization Workshop (VMV), 2008.

[21] C. Zach, T. Pock, and H. Bischof, *A Duality Based Approach for Realtime TV-L^1 Optical Flow*, Springer LNCS, Proceedings of the 29th DAGM Symposium on Pattern Recognition, 2007.

[22] X. Zhang, M. Burger, and S. Osher, *A Unified Primal-Dual Algorithm Framework Based on Bregman Iteration*, UCLA CAM Report [09-99], November 2009.

[23] M. Zhu, and T. F. Chan, *An Efficient Primal-Dual Hybrid Gradient Algorithm for Total Variation Image Restoration*, UCLA CAM Report [08-34], May 2008.