Abstract. We show the primal dual method studied in [7], which is an efficient method for minimizing large non-differentiable convex functions, can also be a practical method for minimizing certain large non-differentiable nonconvex functions for which the nonconvex component is smooth and can be made strongly convex by the addition of a quadratic. Under some standard assumptions to ensure boundedness of the primal-dual iterates produced by this algorithm, we show that all accumulation points are critical points as long as the distance between the iterates goes to zero. In our numerical examples, we focus on an application where we want to increase the sparsity of a variable that is constrained to be nonnegative and sum to one. Convex $l_1$ regularization doesn’t help in this case because the $l_1$ norm is constrained to equal one. Instead we choose to encourage sparsity by adding a concave quadratic function to an otherwise convex function. In particular, we show how to apply this strategy to a nonlocal patch-based image inpainting problem.

Key words. nonlocal image inpainting, primal-dual methods, difference of convex optimization, sparsity promotion

AMS subject classifications. 90C90, 90C59, 65K10, 49N45

1. Introduction. There are many existing convex inpainting models, but they tend to be based on propagating local information into the unknown region and therefore aren’t well suited for filling in areas far from the boundary. For example, total variation inpainting [4] works well for piecewise constant images and when missing pixels are close to known pixels. It is good for geometry inpainting and interpolation but not for texture inpainting.

Greedy approaches for exemplar-based texture inpainting have been successfully considered by many authors. The idea is closely related to the texture synthesis technique of [6] that sweeps through the unknown pixels, greedily setting each to be the value from the center of the image patch that best agrees with its known neighboring pixels.

Previous variational methods for texture inpainting have also been proposed. A variational model proposed in [5] and extended in [2] is based on a correspondence map $\Gamma$, which maps from the unknown region to the known region such that the value $u$ at the location $x$ is given by $u(x) = u(\Gamma(x))$. The functionals to be minimized essentially require that $u(x - y)$ be close to $u(\Gamma(x) - y)$ for $y$ in a neighborhood of 0, and that $\Gamma$ should locally behave like a translation operator. A more easily computable variational approach in [1]
E. Esser and X. Zhang

minimizes a nonlocal means type energy but with dynamic weights that are determined by also minimizing an entropy term. It produces the same kinds of updates for the dynamic weights as the nonlocal total variation wavelet inpainting method in [12]. These variational approaches are all based on non-convex models.

The approach proposed here is inspired by the method of Arias, Caselles and Sapiro in [1]. We first consider a small set of unknown patches that cover the inpainting region. Each unknown patch will be a c-weighted average of known patches with the weights c to be determined by minimizing a convex functional. All patches are of uniform size. To yield a good solution, the weights c should be sparse, since each unknown patch should be a weighted average only of very similar patches. For many examples, the ideal situation would be for the weights to be binary where each unknown patch would exactly equal a known patch. Unknown pixels are defined to be a weighted average of contributing pixels from overlapping unknown patches with these weights fixed in advance and emphasizing more heavily pixels near patch centers. The proposed functional consists of two terms. The first term penalizes at each pixel in or near the inpainting region the sum of the squares of the differences between the values of the contributing pixels and the value at the current pixel. This encourages the unknown patches to agree with each other where they overlap and to agree with any known data they overlap. The second term regularizes the weights by treating them like correspondence maps. The weights for a single unknown patch correspond to the locations of the centers of known patches. Weights for a neighboring unknown patch shifted by v should more likely than not correspond to the same previous locations also shifted by v. The second term of the functional enforces this by penalizing the $l_1$ norm of the differences of these weights, wherever these differences are defined.

The model proposed so far is convex and of the general form

$$\min_{c \in C} H(c) + \sum_{i=1}^{m} J_i(A_ic)$$

for closed proper convex functions $H$ and $J_i$ and a bounded convex set $C$. The precise definition of the functional is in Section 2. For simplification of notation, we denote $Ac = (A_1c, \ldots, A_mc)$ and $J(Ac) = \sum_{i=1}^{m} J_i(A_ic)$. It can be efficiently minimized by a variety of primal dual methods for convex optimization including a modified version of the PDHG method [14] that we will refer to as the PDHGM algorithm [8, 7, 3]. It solves for a minimizer $c^*$ of the primal problem (P0) by finding a saddle point $(c^*, p^*)$ of

$$L(c, p) = \langle p, Ac \rangle - J^*(p) + H(c) \quad \text{for } c \in C$$
via the following iterations:

\[ c^{k+1} = \arg \min_{c \in C} H(c) + \langle A^T (2p^k - p^{k-1}), c \rangle + \frac{1}{2\alpha} \|c - c^k\|^2 \] (1.1a)

\[ p_i^{k+1} = \arg \min_{p_i} J_i^*(p) - \langle A_i c^{k+1}, p_i \rangle + \frac{1}{2\delta} \|p_i - p_i^k\|^2 \text{ for } i = 1, \ldots, m \] (1.1b)

for \( 0 < \delta < \frac{1}{\|A\|^2} \). However, a global minimum of the convex inpainting model is not always a great solution. The recovered image tends to be somewhat blurry and averaged out, especially away from the boundary. The method can still work reasonably well for simple examples with repetitive structure. An example of this is given in Section 3. The blurriness occurs when the weights don’t converge to a sparse enough solution. The unknown patches can therefore end up being averages of too many known patches. This causes a loss of contrast in the unknown patches, which in turn can actually help them agree with each other where they overlap. Moreover, having many nonzero weights can still be consistent with the correspondence constraint. Although there would almost surely be disagreement near the boundary for non-sparse weights, this isn’t enough by itself to enforce sparsity.

It’s difficult to encourage \( c \) to be sparse while maintaining convexity of the model. The constraint on \( c \) helps by requiring that the weights in each weighted average be nonnegative and sum to one. To further encourage sparsity of \( c \), the convex model can be modified by adding a nonconvex term of the form \( \gamma(\langle c, 1 \rangle - \|c\|^2) \). This is analogous to how a double well potential is used to enforce the binary constraint in phase field approaches for image segmentation, except here it suffices to use a quadratic function because \( c \) is already constrained to lie between zero and one. Under the nonnegative and sum to one constraints, this penalty is equivalent to maximizing \( \gamma \|c\|^2 \) on the unit simplex for each unknown patch. This modification promotes sparsity and can lead to much better solutions with binary weights as demonstrated in Section 2.2.

With the addition of a concave penalty, the resulting model is no longer convex. However, the nonconvex objective can naturally be written as a difference of convex functions. There are some standard difference of convex (DC) programming methods for finding critical points by solving a sequence of convex problems with the concave term linearized about the previous iterate [9]. By replacing the convex function \( H \) with a concave function \( G(c) \) in (P0), we can write the difference of convex objective as

\[ \min_{c \in C} \left( \sum_{i=1}^m J_i(A_i c) + \rho \|c\|^2 \right) - \left( -G(c) + \rho \|c\|^2 \right) \] (P1)

and proceed by iterating

\[ c^{n+1} = \arg \min_{c \in C} \sum_{i=1}^m J_i(A_i c) + \rho \|c\|^2 + \langle c, v^n - 2\rho c^n \rangle, \text{ where } v^n \in \partial G(c^n). \] (1.2)
Since the problem is nonconvex, this cannot be guaranteed to converge to a 
global minimizer, but as discussed in [10], it can be shown that any convergent 
subsequence of \( \{c^n\} \) converges to a critical point of (P1) [13, 11]. The convex 
subproblem in each iteration can be solved using PDHGM. A downside of this 
approach is that each iteration is as expensive as applying PDHGM to (P0).

Although relatively few iterations of the DC method seem to be needed 
in practice, it still represents a substantial increase in computational time. 
It would be preferable to solve the problem with one pass of a PDHGM- 
like method. We therefore propose to apply PDHGM directly to (P1) as 
if \( G \) were convex, subject to some additional constraints on the algorithm 
parameters to make sure the subproblems in each iteration are well posed. 
The resulting numerical scheme is not guaranteed to converge, but it can 
work well in practice and is substantially faster than the difference of convex 
approach. Theoretically, if the distance between iterates goes to zero, then 
any accumulation point is a critical point of the nonconvex function. The 
theoretical properties of PDHGM applied to nonconvex problems are discussed 
in Section 4. We use this method to produce the inpainting examples in Section 
3. The detailed formulation of the inpainting model is presented in Section 2.

2. Nonlocal Patch-Based Inpainting Model. In this section, first a 
convex variational model for nonlocal image inpainting is proposed. It uses 
patches from anywhere in the known part of the image to fill in a large unknown 
area. Later in this section, a nonconvex modification to the model is proposed.

2.1. Notation and Formulation of Model. The formulation of the 
inpainting model is notationally heavy despite being based on simple ideas. 
Figure 2.1 shows a picture of the setup, and the key notation is defined in the 
list in Section 2.1.1.
2.1.1. Notation.

$h \in \mathbb{R}^{m_r \times m_c}$ \quad \text{Original image}

$\Omega$ \quad \text{Inpainting region: Set of pixels $(i, j)$ such that $h(i, j)$ is unknown}

$p_s$ \quad \text{Patch size (assume $p_s = 6n + 3$ for simplicity)}

$\Omega_u$ \quad \text{Region of unknown patches: Set of pixels $(i, j)$ covered by unknown patches. This should strictly contain $\Omega$.}

$v$ \quad \text{Index for pixels in $\Omega_u$, $v = 1, ..., |\Omega_u|$}

$\Omega_{up}$ \quad \text{Subset of $\Omega_u$ consisting of a grid of pixels spaced apart by $\frac{2p_s}{3}$, corresponding to the unknown patches that will be solved for}

$\Omega_o$ \quad \text{Overlap region: $\Omega_o = \Omega_u \cap \Omega^c$}

$\Omega_p$ \quad \text{Region of known patches: Set of pixels for which corresponding patches are contained in $\Omega^c$}

$P \in \mathbb{R}^{p_s^2 \times |\Omega_p|}$ \quad \text{Matrix of vectorized known patches. $P(q, p)$ is the $q^{\text{th}}$ pixel in the $p^{\text{th}}$ patch, $q = 1, ..., p_s^2$, $p = 1, ..., |\Omega_p|$}

$u \in \mathbb{R}^{|\Omega_u|}$ \quad \text{Value at pixels in $\Omega_u$ constrained so $u(v) = h(v)$ for $v \in \Omega_o$}

$S$ \quad \text{Set of valid $u$: $\{u : u(v) = h(v) \text{ for } v \in \Omega_o\}$}

$c \in \mathbb{R}^{|\Omega_p| \times |\Omega_{up}|}$ \quad \text{Weights for representing $\Omega_{up}$ patches as weighted averages of $\Omega_p$ patches. Must constrain $c(p, m) \geq 0$ and $\sum_p c(p, m) = 1 \quad \forall m$, $m = 1, ..., |\Omega_{up}|$}

$C$ \quad \text{Set of valid $c$: $\{c : c(p, m) \geq 0 \text{ and } \sum_p c(p, m) = 1 \quad \forall m\}$}

$Pc \in \mathbb{R}^{p_s^2 \times |\Omega_{up}|}$ \quad \text{Matrix product of $P$ and $c$ is matrix of vectorized $\Omega_{up}$ patches}

$\beta(q)$ \quad \text{Vectorized 2D Gaussian weights (standard deviation $\frac{p_s}{3}$) defined on a single patch,}

$\beta_v(q)$ \quad \text{Normalized weights $\beta_v(q) = \frac{\beta(q)}{\sum_{Q_v} \beta(q)}$ where $Q_v = \{q : \text{there exists a $\Omega_{up}$ patch whose $q^{\text{th}}$ pixel overlaps } v\}$}
2.1.2. Definition of Functional. The proposed functional will initially be defined in terms of $c$ and $u$. Later, the expression for $u$ in terms of $c$ will be substituted in. The constraints on $c$ and $u$ will be handled by introducing indicator functions $g_C(c)$ and $g_S(u)$ for the sets $C$ and $S$.

$$g_C(c) = \begin{cases} 0 & \text{if } c \in C \\ \infty & \text{otherwise} \end{cases} \quad g_S(u) = \begin{cases} 0 & \text{if } u \in S \\ \infty & \text{otherwise} \end{cases}$$

The first term of the functional can be written

$$\sum_{n=1}^{\left|\Omega_u\right|} \sum_{\text{contributing}(q,m)} (\beta_v(q)((Pc)(q,m) - u(v)))^2,$$

where the contributing $(q, m)$ indices are those for which the $q^{th}$ pixel in the $m^{th} \Omega_{up}$ patch overlaps pixel $v$. This can be more conveniently rewritten as

$$\|A(c) - B(u)\|_F^2,$$

where $\| \cdot \|_F$ is the Frobenius norm and $A : \mathbb{R}^{\left|\Omega_p\right| \times \left|\Omega_{up}\right|} \rightarrow \mathbb{R}^{s \times \left|\Omega_u\right|}$ and $B : \mathbb{R}^{\left|\Omega_u\right|} \rightarrow \mathbb{R}^{s \times \left|\Omega_u\right|}$ are linear operators defined as follows.

$$A(c)(q, v) = \begin{cases} \beta_v(q)(Pc)(q,m) & \text{if there exists } m \text{ such that pixel } q \text{ of the } \Omega_{up} \text{ patch at } m \text{ overlaps } v \\ 0 & \text{otherwise} \end{cases}$$

$$B(u)(q, v) = \begin{cases} \beta_v(q)u(v) & \text{if there exists } m \text{ such that pixel } q \text{ of the } \Omega_{up} \text{ patch at } m \text{ overlaps } v \\ 0 & \text{otherwise} \end{cases}$$

Note that $u$ is only compared to pixels that come from weighted averages of known patches. This is a potential weakness of this choice of data fidelity term. It would be better to directly compare $u$ to the information in the known patches in this weighted average, but we don’t because to do so would be much more computationally intensive.

The correspondence term of the functional will be defined as

$$\sum_{m=1}^{\left|\Omega_{up}\right|} \sum_{\tilde{m} \sim m} \sum_{p=1}^{\left|\Omega_p\right|} \left|c(p,m) - c(\tilde{p}(p, \tilde{m}, m), \tilde{m})\right| \quad \text{if defined}$$

$$0 \quad \text{otherwise},$$

where $\tilde{p}(p, \tilde{m}, m)$ denotes the index for the $\Omega_p$ patch shifted from $p$ by the same amount $\tilde{m}$ is shifted from $m$, defined when contained in $\Omega_p$. These differences can be more conveniently written in terms of a linear operator $D : \mathbb{R}^{\left|\Omega_p\right| \times \left|\Omega_{up}\right|} \rightarrow \mathbb{R}^e$, with $e$ the total number of differences taken. With this notation the correspondence term can be rewritten as $\beta \|D(c)\|_1$. 
The proposed functional is then
\[ G(c, u) = g_C(c) + g_S(u) + \frac{\mu}{2} \| A(c) - B(u) \|_F^2 + \beta \| D(c) \|_1, \] (2.1)
which is to be minimized with respect to \( u \) and \( c \). Note, however, that \( u \) can be easily solved for in terms of \( c \).

\[ u(v) = \begin{cases} h(v) & v \in \Omega_o \\ ((B^*B)^{-1}B^*A(c))(v) & v \in \Omega, \end{cases} \]

where \( B^* \) denotes the adjoint of \( B \). How can we ensure \( B^*B \) is invertible?

This formula for \( u \) can be thought of as taking a weighted average at each unknown pixel of the contributing pixels from overlapping patches. Pixels closer to the center of the patches are more heavily weighted according to the Gaussian weights \( \beta \). This weighted averaging update for \( u \) is very similar to the one used in [1]. When plugging the formula for \( u \) back into \( \| A(c) - B(u) \|_F^2 \), it makes sense to break the term into two parts, one defined on \( \Omega_o \) and the other defined on \( \Omega \). To that end, define

\[ X_{\Omega_o}(q, v) = \begin{cases} 1 & v \in \Omega_o \\ 0 & \text{otherwise} \end{cases}, \quad X_{\Omega}(q, v) = \begin{cases} 1 & v \in \Omega \\ 0 & \text{otherwise} \end{cases}. \]

Also define

\[ h_0(v) = \begin{cases} h(v) & v \in \Omega_o \\ 0 & \text{otherwise} \end{cases}. \]

Plugging the expression for \( u \) into \( G \) yields
\[ g_C(c) + \frac{\mu}{2} \| X_{\Omega_o} \cdot A(c) - X_{\Omega_o} \cdot B(h_0) \|_F^2 + \frac{\mu}{2} \| X_{\Omega} \cdot (I - B(B^*B)^{-1}B^*)A(c) \|_F^2 + \beta \| D(c) \|_1, \]

where \( \cdot \) denotes componentwise multiplication of matrices. Let

\[ f = X_{\Omega_o} \cdot B(h_0) \]

and define linear operators \( A_{\Omega_o} \) and \( A_{\Omega} \) such that

\[ A_{\Omega_o}(c) = X_{\Omega_o} \cdot A(c) \]

and

\[ A_{\Omega}(c) = X_{\Omega} \cdot (I - B(B^*B)^{-1}B^*)A(c). \]

Now we can define a convex functional in terms of \( c \),
\[ F(c) = g_C(c) + \frac{\mu_{\Omega_o}}{2} \| A_{\Omega_o}(c) - f \|_F^2 + \frac{\mu_{\Omega}}{2} \| A_{\Omega}(c) \|_F^2 + \beta \| D(c) \|_1, \] (2.2)
and attempt to solve the inpainting problem by finding a minimizer.
2.2. Nonconvex Modification to Inpainting Functional. A modification to \( F(2.2) \) intended to improve the sparsity of the weights is discussed in this section. We show that adding a nonconvex term to encourage sparser or even binary weights can lead to better quality solutions.

Motivated by the phase field approach for segmentation that enforces a binary constraint by introducing a double well potential, we consider a similar strategy for making \( c \) sparser or even binary. Any background or reference on this functional? The double well potential strategy would be to add a term of the form

\[ \gamma \sum_{p,m} c_{p,m}^2 (1 - c_{p,m})^2 \]

to (2.2). Since the normalization constraint \( c \in C \) already forces \( 0 \leq c_{p,m} \leq 1 \), we instead choose to add the quadratic function

\[ \gamma \sum_{p,m} c_{p,m} (1 - c_{p,m}), \]

which can be rewritten as

\[ \gamma \langle c, 1 \rangle - \gamma \|c\|_F^2. \]

The resulting nonconvex functional is defined by

\[
F_\gamma(c) = g_C(c) + \gamma \langle c, 1 \rangle - \gamma \|c\|_F^2 + \frac{\mu_\Omega}{2} \|A_{\Omega_o}(c) - f\|_F^2 + \frac{\mu_\Omega}{2} \|A_\Omega(c)\|_F^2 + \beta \|D(c)\|_1,
\]

which is of the general form of (P1).


3.1. Algorithms. In this section, we demonstrate how to apply the modified PDHG algorithm. To minimize \( F_\gamma \), using modified PDHG, define

\[
G(c) = g_C(c) + \gamma \langle c, 1 \rangle - \gamma \|c\|_F^2,
\]

\[
\tilde{A} = \begin{bmatrix} A_{\Omega_o} \\ A_\Omega \\ D \end{bmatrix}
\]

and

\[
J(\tilde{A}(c)) = J_{\Omega_o}(A_{\Omega_o}(c)) + J_\Omega(A_\Omega(c)) + J_D(D(c)),
\]

where

\[
J_{\Omega_o}(z_{\Omega_o}) = \frac{\mu_\Omega}{2} \|z_{\Omega_o} - f\|_F^2,
\]

\[
J_\Omega(z_\Omega) = \frac{\mu_\Omega}{2} \|z_\Omega\|_F^2
\]

and

\[
J_D(z_D) = \beta \|z_D\|_1.
\]
With the addition of dual variables $p_\Omega, p_\Omega, p_D$, time step parameters $\alpha, \delta$ and optional scaling parameters $s_\Omega, s_\Omega, s_D$, the PDHGM iterations from (1.1) are given by

$$c^{k+1} = \arg\min_c g_C(c) + \gamma\langle c, 1 \rangle - \gamma\|c\|_F^2 + \frac{1}{2\alpha}\|c - \left(c^k - \alpha \frac{A^*_\Omega(2p^k_\Omega - p^{k-1}_\Omega)}{s_\Omega}, \frac{A^*_\Omega(2p^k_\Omega - p^{k-1}_\Omega)}{s_\Omega}\right)\|_F^2$$

$$-\alpha \frac{A^*_\Omega(2p^k_\Omega - p^{k-1}_\Omega)}{s_\Omega} - \alpha \frac{D^*(2p^k_D - p^{k-1}_D)}{s_D}$$

$$p^{k+1}_\Omega = \arg\min_{p_\Omega} J^*_\Omega\left(\frac{p^k_\Omega}{s_\Omega} + \frac{1}{2\delta}\|p_\Omega - (p^k_\Omega + \frac{\delta A_\Omega(c^{k+1})}{s_\Omega})\|_F^2\right)$$

$$p^{k+1}_D = \arg\min_{p_D} J^*_D\left(\frac{p^k_D}{s_D} + \frac{1}{2\delta}\|p_D - (p^k_D + \frac{\delta A_D(c^{k+1})}{s_D})\|_F^2\right)$$

where the initialization is arbitrary.

Each of the above minimizers can be explicitly solved by the following formulas,

$$c^{k+1} = \Pi_C \left(\left(1 - \frac{1}{1 - 2\alpha \gamma}\right) \left(c^k - \alpha \frac{A^*_\Omega(2p^k_\Omega - p^{k-1}_\Omega)}{s_\Omega} - \alpha \frac{A^*_\Omega(2p^k_\Omega - p^{k-1}_\Omega)}{s_\Omega}\right)\right)$$

$$-\alpha \frac{D^*(2p^k_D - p^{k-1}_D)}{s_D} - \gamma\alpha$$

$$p^{k+1}_\Omega = \frac{p^k_\Omega + \frac{\delta A_\Omega(c^{k+1})}{s_\Omega} - f}{\frac{\mu_\Omega s^2_\Omega}{\mu_\Omega s^2_\Omega} + 1}$$

$$p^{k+1}_D = \frac{p^k_D + \frac{\delta A_D(c^{k+1})}{s_D}}{\frac{\mu D s^2_D}{\mu D s^2_D} + 1}$$

$$p^{k+1}_D = \Pi\{z : ||z||_\infty \leq \beta s_D\}\left(p^k_D + \frac{\delta}{s_D}D(c^{k+1})\right)$$

where $\Pi_C$ and $\Pi\{z : ||z||_\infty \leq \beta s_D\}$ denote orthogonal projection onto $C$ and $\{z : ||z||_\infty \leq \beta s_D\}$ respectively. The projection $\Pi_C(c)$ amounts to projecting each column of $c$ onto the positive face of the $l_1$ unit ball. This ensures the weights are nonnegative and normalized. To ensure $\frac{1}{1 - 2\alpha \gamma} > 0$, we require $\gamma$ to satisfy $0 \leq \gamma < \frac{1}{2\alpha}$. This guarantees that the objective functional for the $c^{k+1}$ update remains convex.

Note that the convergence theory for PDHGM [7, 3] only holds in the convex case corresponding to $\gamma = 0$. When $\gamma \neq 0$, $F_\gamma(c)$ is not convex,
yet the method can still work empirically. In practice, it doesn’t find global minimizers of $F_\gamma$, but it is able to produce good solutions with binary weights as demonstrated in Section 3.2.

To minimize $F_\gamma$, using the difference of convex approach (1.2), we can write

$$F_\gamma(c) = (g(c) + \frac{\mu_{\Omega_o}}{2} \|A_{\Omega_o}(c) - f\|^2 + \frac{\mu_{\Omega}}{2} \|A_{\Omega}(c)\|^2 + \|D(c)\|_1 + \rho \|c\|^2 - (-\gamma \langle c, 1 \rangle + \gamma \|c\|^2 + \|c\|^2))$$

and iteratively solve the convex problems

$$c^{n+1} = \arg\min_{c \in C} \frac{\mu_{\Omega_o}}{2} \|A_{\Omega_o}(c) - f\|^2 + \frac{\mu_{\Omega}}{2} \|A_{\Omega}(c)\|^2 + \beta \|D(c)\|_1 + \rho \|c\|^2 + \langle c, (1-2c^n) - 2\rho c^n \rangle.$$  

The PDGHM method for updating $c^{n+1}$ requires exactly the same iterations as in (3.1), but with the $c^{k+1}$ step replaced by

$$c^{k+1} = \Pi_C \left( \left( \frac{1}{1 + 2\alpha \rho} \right) \left( c^k - \alpha \frac{A_{\Omega_o}^*(2p_{\Omega_o}^k - p_{\Omega_o}^{k-1})}{s_{\Omega_o}} - \alpha \frac{A_{\Omega}^*(2p_{\Omega}^k - p_{\Omega}^{k-1})}{s_{\Omega}} - \alpha \frac{D^*(2p_{D}^k - p_{D}^{k-1})}{s_{D}} - \alpha (\gamma (1 - 2c^n) - 2\rho c^n) \right) \right).$$

### 3.2. Numerical Results

The convex inpainting model works best for simple images with repeating structure and we show its successful application to the problem of inpainting a missing portion of a brick wall in Figures 3.1 and 3.2. These examples also demonstrate the effect of the correspondence term $\|D(c)\|_1$, which encourages information in the recovered image to have similar spatial correspondence as information in the known part of the image. In the extreme case where the weights are binary and the correspondence term equals zero, the recovered data would simply be a copy of a contiguous block of known image. Figure 3.1 shows the inpainting result without the correspondence term. The parameters $\mu_{\Omega_o}$ and $\mu_{\Omega}$ were both set equal to one.

The scaling parameters, which affect the efficiency of the numerical scheme but don’t change the model, were chosen to be $s_{\Omega_o} = 100000$ and $s_{\Omega} = 10000$. Figure 3.2 shows the result with the correspondence term included. For this example, $\beta = 1000$ and $s_{D} = 100$. As can be seen in the figures, the addition of the correspondence term makes it possible to better reproduce the repeating structure of the image even when far from the boundary.

It is better to combine the two figures into one and maybe more explanation on the image "weighted average on overlap ($A(c)$)"

For more complicated images like the picture of grass in Figure 3.3, the addition of the correspondence term is not always able to encourage recovery of more detail in the inpainting region. In this example, with 15 by 15 patches, it’s difficult to find known patches that agree well with the boundary information. When that happens, weights minimizing the convex functional
Fig. 3.1. Inpainting brick wall using $15 \times 15$ patches but without including the correspondence term.

Fig. 3.2. Inpainting brick wall using $15 \times 15$ patches and including the correspondence term.

tend to be less sparse. That’s because when the unknown patches end up being averages of many known patches, they become more nearly constant and
therefore agree well with patches they overlap. Figure 3.3 shows an example of such an unsatisfactory over-averaged inpainting result. Modifications to the functional that address this drawback are discussed in the next section.

![Fig. 3.3. Inpainting grass using $15 \times 15$ patches and including the correspondence term](image)

The nonconvex modification of the inpainting model discussed in Section 2.2 is tested on two example images, an image of grass and the brick wall image, both missing a large rectangular region in the center. In both examples, the weights end up being binary. Although the solutions are not global minimizers of $F_\gamma(c)$, they look more natural than the global minimizers of the convex model.

For both examples, $\mu_{\Omega_o} = 1$, $\mu_{\Omega} = 1$, $\beta = 1000$, $s_{\Omega_o} = 100000$, $s_\Omega = 10000$ and $s_D = 100$.

The brick example in Figure 3.4 was computed in 400 iterations. For the first 200 iterations we set $\alpha = 1000$, $\delta = .001$ and $\gamma = \frac{0.1}{2\alpha}$. For the last 200 iterations we set $\alpha = 100$, $\delta = .0001$ and $\gamma = \frac{0.05}{2\alpha}$.

The grass example in Figure 3.5 was computed in 700 iterations. Similar to the brick example, for the first 500 iterations we set $\alpha = 1000$, $\delta = .001$ and $\gamma = \frac{0.1}{2\alpha}$. For the last 200 iterations we set $\alpha = 100$, $\delta = .0001$ and $\gamma = \frac{0.05}{2\alpha}$.

These parameters were not exhaustively optimized and better parameter selections may well lead to improved performance of the model.

We use a $196 \times 196$ mosaic image to compare the PDHGM and difference of convex methods in Figures 3.6 and 3.7. For both methods, the patches were $15$ by $15$ and the parameters were set to $\mu_{\Omega_o} = 1$, $\mu_{\Omega} = 1$, $\beta = 1000$, $s_{\Omega_o} = 330000$, $s_\Omega = 150000$ and $s_D = 2$, $\alpha = .005$ and $\delta = 50$. Here the scaling
parameters $s_{\Omega_o}$, $s_{\Omega}$ and $s_D$ were chosen to approximate the spectral norms of $A_{\Omega_o}$, $A_{\Omega}$ and $D$ respectively. This seems to be a good rule for defining the scaling parameters in general. For both methods $\gamma$ was fixed at 10 for all iterations. Note that $2\alpha \gamma = .1$. The PDHGM result is shown in Figure 3.6.
and required 200 iterations to compute. The DC results is shown Figure 3.7. It required 10 outer iterations before the weights became binary, therefore taking about 10 times as long as the PDHGM method.

Fig. 3.6. Inpainting mosaic using PDHGM to minimize the nonconvex model with $15 \times 15$ patches

Fig. 3.7. Inpainting mosaic using the a DC method to minimize the nonconvex model with $15 \times 15$ patches

The computed minimizers of the nonconvex $F_\gamma$ are usually not global min-
imizers. Even the global minimizers of the convex $F$ for the same examples have lower energy in terms of $F_{\gamma}$. The fact that the computed minimizers of $F_{\gamma}$ lead to better solutions suggests that perhaps we shouldn’t be looking for a global minimizer of that functional. Since the computed minimizers of $F_{\gamma}$ are binary for the examples tested, this indicates that we may be more interested in computing minimizers of $F$ subject to an additional constraint that restricts $c$ to be binary. Our procedure for minimizing $F_{\gamma}$ may be a practical approach for approximating solutions to that nonconvex problem.

4. Convergence Analysis. It is well known that it is difficult to obtain general convergence results for nonconvex minimization problems. In this section, we will analyze the theoretical results for the algorithm (3.1).

For simplicity of notation we refer to the general minimization problem (P1):

$$\min_{x \in X} G(x) + \sum_{i=1}^{m} J_i(A_ix)$$

where $G(x)$ is a nonconvex but Lipschitz smooth functional, $J_i$ are proper convex functionals, $A_i$ are bounded linear operators for $i = 1, \cdots, m$ and $X$ is a bounded closed convex set in $\mathbb{R}^n$. This includes the inpainting model (2.3) as a special case. In the previous example, we have some special forms for $J$ and $G$, where $G(\cdot) + \frac{1}{2\tau} \lVert \cdot \rVert^2$ is strongly convex for some small enough $\tau$ and $J_i(\cdot)$ are of form $\frac{\mu_i}{2} \lVert \cdot \rVert^2$ or $\mu_i \lVert \cdot \rVert_1$ norm where $\lVert \cdot \rVert, \lVert \cdot \rVert_1$ denote the usual $\ell^2$ and $\ell^1$ norm respectively.

Remarks: We can see that if $G(x) = \gamma \langle x, 1 \rangle - \gamma \|x\|^2$, then $G(x) + \frac{1}{2\tau} \|x\|^2$ becomes strongly convex under a suitable choice of $\tau < \frac{1}{2\gamma}$. For any $x$, we have $\nabla G(x) = \gamma - 2\gamma x$, thus $\forall x, y \in \mathbb{R}^n$, and we have

$$\langle x - y, \nabla G(x) - \nabla G(y) \rangle = -2\gamma \|x - y\|^2.$$

Denote $\delta_X(x)$ as the indicator function of the constraint set $X$ that has value 0 for $x \in X$ and $+\infty$ otherwise. The normal cone operator of $X$ is the subdifferential of $\delta_X(x)$:

$$N_X(x) = \{s \in \mathbb{R}^n | \langle s, z - x \rangle \leq 0 \quad \forall z \in X \}$$

Note that since $X$ is closed convex set, then $\delta_X(x)$ is a closed convex function and $N_X$ is maximal monotone.

For simplicity, we write $A = \begin{pmatrix} A_1 \\ \vdots \\ A_m \end{pmatrix}$. Using the Fenchel conjugate functions, we consider the primal dual form:

$$\min_{x \in X} \max_{y \in Y} L(x; y) := \delta_X(x) + G(x) + \langle y, Ax \rangle - \sum_{i=1}^{m} J_i^*(y_i)$$
where \( y = (y_i)_{i=1}^m \) and \( \langle y, Ax \rangle = \sum_{i=1}^m \langle A_i x, y_i \rangle \). We denote the vector function
\[
J^* = \left( J^*_1, \cdots, J^*_m \right)
\]
Let
\[
T_1(x, y) = \begin{pmatrix} N_X(x) + A^T y \\ \partial J^*(y) - Ax \end{pmatrix}, T_2(x, y) = \begin{pmatrix} \nabla G(x) \\ 0 \end{pmatrix}
\]
and \( T(x, y) = T_1(x, y) + T_2(x, y) \). Then the first-order necessary condition for a minimizer \( x^* \) is
\[
0 \in T(x^*, y^*)
\]
where \( y^* \in \partial J(Ax^*) \) (i.e., \( y^*_i \in \partial J_i(A_i x^*) \)).

Recall that the PDHGM algorithm applied to (P1) is as follows:

\[
x^{k+1} = \arg \min_{x \in X} G(x) + \langle A^T (2y^k - y^{k-1}), x \rangle + \frac{1}{2\tau} \|x - x^k\|^2
\]
\[
y^{k+1}_i = \arg \min_{y_i} J^*_i(y) - \langle A_i x^{k+1}, y_i \rangle + \frac{1}{2\sigma} \|y_i - y^k_i\|^2 \quad \text{for } i = 1, \cdots, m
\]

By the assumption on \( G \) and \( J \), we can see that \( x^{k+1} \) and \( y^{k+1}_i \) are uniquely defined by the above iteration.

In another form, if we assume
\[
M = \begin{pmatrix}
\frac{1}{\tau} I & A_1^T \\
A_1 & \frac{1}{\sigma} I \\
\vdots & \vdots \\
A_m & \frac{1}{\sigma} I
\end{pmatrix} = \begin{pmatrix}
\frac{1}{\tau} I & A^T \\
A & \frac{1}{\sigma} I
\end{pmatrix}
\]
Here we choose suitable parameter \( \sigma \) and \( \tau \) such that the matrix \( M \) is positive definite. The conditions will be specified in the next section. The above algorithm can be reformulated as

\[
T(x, y) + M \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{4.1}
\]

With some assumptions on the parameter \( \tau \) and \( \sigma \), we can obtain strong convexity of each subproblem, we have a unique sequence \( (x^{k+1}, y^{k+1}) \) satisfies the above equation:

\[
T(x^{k+1}, y^{k+1}) + M \begin{pmatrix} x^{k+1} - x^k \\ y^{k+1} - y^k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}
\]

Let \( z^k := (x^k, y^k), Z = X \times Y \), then we have equivalently:

\[
z^{k+1} = (T + M)^{-1} M z^k \tag{4.2}
\]
Lemma 1. The set of minima $X^*$ of (P1) is nonempty and compact.

Proof. Since $X$ is compact, and by Weierstrass’s Theorem, the set of solution of the minimization problem (P1) is nonempty and compact. □

Lemma 2. If $\tau \sigma \|A^TA\| < 1$, then $M$ is positive definite.

In the following, we assume that the condition in Lemma 2 is satisfied.

Lemma 3. The sequence $z^k$ generated by PDHGM is bounded.

Proof. Since $x^k \in X$ for every $k$, and $X$ is bounded, we get immediately the boundedness of $x$. By the optimality condition, we have

$$\partial J_i^*(y_i^{k+1}) - A_i x^{k+1} + \frac{1}{\sigma}(y_i^{k+1} - y_i^k) \ni 0$$

that is:

$$y_i^{k+1} = (I + \sigma \partial J_i^*)^{-1}(y_i^k + \sigma A_i x^{k+1})$$

If $J_i(\cdot) = \frac{\lambda_i}{2}\|\cdot\|^2$, then $J_i^*(\cdot) = \frac{1}{2\lambda_i}\|\cdot\|^2$, and

$$y_i^{k+1} = (1 + \frac{\sigma}{\lambda_i})^{-1}(y_i^k + \sigma A_i x^{k+1}) = \rho(y_i^k + \sigma A_i x^{k+1})$$

where $\rho = (1 + \frac{\sigma}{\lambda_i})^{-1} < 1$. Recursively, we have

$$y_i^{k+1} = \rho^k y_i^1 + \sum_{i=1}^{k} \rho^i \sigma A_i x^{k+1-i}$$

$$\|y_i^{k+1}\| \leq \|y_i^1\| + \frac{1}{1-\rho}\sigma\|A_i\|\|x^{k+1}\| < \infty.$$  

If $J_i(\cdot) = \lambda_i\|\cdot\|_1$, then $J_i^*(\cdot) = \delta_C(\cdot)$, where $C = \{\|x\|_\infty < \lambda_i\}$, thus

$$\|y_i^{k+1}\| = \|\Pi_C(y_i^k + \sigma A_i x^{k+1})\| < \infty$$

Therefore, we conclude that the sequence $z^{k+1} = (x^{k+1}, y^{k+1})$ is bounded. □

Similar to the convex case, we give some estimate of the error of the sequence to the solution.

Lemma 4. Let $z^* = (x^*, y^*)$ be a zero of $T$, then

$$\|z^{k+1} - z^*\|^2_M \leq \|z^k - z^*\|^2_M - \|z^k - z^{k+1}\|^2_M + 4\gamma\|z^{k+1} - x^*\|^2$$

Proof. Since each subproblem is strongly convex, then for any $z \in Z$,

$$\langle z - z^{k+1}, T_1(z^{k+1}) + T_2(z^{k+1}) + M(z^{k+1} - z^k) \rangle = 0$$

For any $z^* \text{ s.t. } T(z^*) = 0$, we have $T_1(z^*) = -T_2(z^*)$, which leads to

$$\langle z^* - z^{k+1}, T_1(z^{k+1}) - T_1(z^*) + T_2(z^{k+1}) - T_2(z^*) + M(z^{k+1} - z^k) \rangle = 0$$
Since $T_1$ is monotone, we have
\[ \langle z^* - z^{k+1}, T_2(z^*) - T_2(z^{k+1}) \rangle = \langle z^{k+1} - z^*, T_1(z^{k+1}) - T_1(z^*) \rangle \geq 0 \]
Therefore
\[ \langle z^{k+1} - z^*, M(z^{k+1} - z^*) \rangle \leq \langle z^* - z^{k+1}, T_2(z^{k+1}) - T_2(z^*) \rangle \]
Using the fact that
\[ \langle z^{k+1} - z^*, T_2(z^{k+1}) - T_2(z^*) \rangle = \langle \nabla G(x^{k+1} - \nabla G(x^*), x^{k+1} - x^* \rangle = -2\gamma \|x^{k+1} - x^*\|^2 \]
we get
\[ \langle z^{k+1} - z^*, M(z^{k+1} - z^*) \rangle \leq 2\gamma \|x^{k+1} - x^*\|^2. \]
Moreover
\[ \|z^k - z^{k+1}\|_M^2 \leq \|z^k - z^*\|_M^2 - \|z^{k+1} - z^*\|_M^2 + 2\langle z^{k+1} - z^*, M(z^{k+1} - z^*) \rangle \]
\[ \leq \|z^k - z^*\|_M^2 - \|z^{k+1} - z^*\|_M^2 + 4\gamma \|x^{k+1} - x^*\|^2 \]
\[ \square \]
\[ \text{THEOREM 1.} \, \text{Let} \, \sigma \tau \|AT\| < 1, \, \text{then the sequence} \, z^k \, \text{has a convergent subsequence}. \]
• If $z^{k+1} = z^k$, then $z_k$ is a critical point of (P1).
• If $\|z^{k+1} - z^k\|_M \to 0$, then any accumulation point of $z^k$ is a stationary point of (P1).

\[ \text{Proof.} \, \text{From the iteration formula (4.2), we have the first statement immediately.} \, \text{Since} \, \sigma \tau \|AT\| < 1, \, \text{then} \, M \, \text{is positive definite. From Lemma 3, we can easily see that there exists a convergent subsequence. Let} \, (x^*, y^*) \, \text{be any accumulation point and let} \, (x_k, y_k) \, \text{be a subsequence converging to it. Since} \, \lim_{l \to \infty} T(x_k, y_k) \supseteq \lim_{l \to \infty} -M(x_k - x^{k-1}, y_k - y^{k-1}). \, \text{If the condition} \, \|z^{k+1} - z^k\|_M \to 0 \, \text{satisfied, we conclude that} \]
\[ 0 \in T(x^*, y^*) \]
Thus $x^*$ is a zero of (4.2). \[ \square \]

\[ \text{Remarks:} \]
From Lemma 4, we can see that if $\gamma = 0$, then $\|z^{k+1} - z^k\|_M \to 0$ and we can conclude that for the convex model any accumulation point is a stationary point. Unfortunately for general $\gamma > 0$, we can not show that $\|z^{k+1} - z^k\|_M \to 0$ necessarily holds.
5. Conclusions. In this paper, we have proposed a convex model for non-local patch-based inpainting and a nonconvex modification to promote sparsity of weights constrained to a unit simplex. This modification yields better inpainting results by requiring unknown patches to be weighted averages of very few known patches, ideally just a copy of a single known patch. We proposed two methods for minimizing this model. One is a difference of convex approach that works but can be slow because it requires repeatedly solving an expensive convex problem. As a faster alternative, we proposed applying a modified PDHG method directly to the nonconvex problem. Empirically, this can be used to produce good solutions in less time, but convergence is not guaranteed unless the distance between iterates can be shown to go to zero in the limit. Future work should continue investigating if there are parameter choices that can guarantee this condition as well the properties of a method where the strength of the sparsity penalty is gradually increased, as this approach tends to lead to better empirical results.

REFERENCES