A CONVEX PROGRAMMING APPROACH TO ANISOTROPIC SMOOTHING

Joachim Dahl, Søren Holdt Jensen* and Per Christian Hansen†*

*Department of Electronic Systems, Aalborg University
Niels Jernesvej 10, 9220 Aalborg Ø, Denmark
{joachim, shj}@es.aau.dk

†Informatics and Mathematical Modelling, Building 321
Technical University of Denmark, 2800 Lyngby, Denmark
pch@imm.dtu.dk

ABSTRACT

We develop a computationally efficient algorithm for the problem of anisotropic smoothing in images, *i.e.*, smoothing an image along its directional fields. Whereas similar problems are often posed and solved as PDE problems in the literature, we formulate the problem as a convex programming problem. We present efficient implementations of interior-point algorithms for solving large instances of this problem.

Index Terms— Optimization methods, quadratic programming, image processing

1. INTRODUCTION

The problem of smoothing an image mainly along its directional fields has been studied by several authors (*e.g.*, [1, 2]), using a non-linear diffusion process based on a partial differential equation (PDE). An interesting application of these ideas is the *inpainting* problem, where missing parts of an image can be reconstructed, *e.g.*, based on the surrounding directional fields.

In this correspondence we revisit the anisotropic smoothing problem, taking a new convex programming approach. One of the general advantages of formulating the problem in a convex programming setting is that this leads to straightforward implementations; in fact, many PDE based approaches formulate a discrete convex approximation, which is subsequently solved, *e.g.*, total variation based image processing algorithms (see [3] and references therein). Other PDE based problems are not as easily approximated by convex problems, which makes implementation difficult. The approach we suggest here directly exploits convexity of the *discrete* problem instead, and cannot be interpreted as a convex approximation of the problems in [1, 2]; instead it should be considered as an alternative approach based on convex programming.

Although convex programming problems are, in principle, readily solved using, *e.g.*, modern interior-point algorithms [4], limited progress has been made applying these techniques to image processing problems, where the large dimensions renders a typical implementation intractable. A notable exception is the second-order cone programming (SOCP) formulation of total variation based denoising in [5], where the authors demonstrate the advantages of implementing the large-scale methods directly using modern convex programming instead of usual smooth approximations.

Our approach and contribution is similar in spirit. We first demonstrate in Sec. 2 how the anisotropic smoothing problem can be written as a convex large-scale quadratically constrained quadratic programming (QCQP) problem. This problem has both sparse and non-sparse structure, which we exploit in the large-scale implementations of Sec. 3. With this implementation we can solve problems with more than 10^5 variables in a few minutes; we give such large-scale examples for fingerprint enhancement in Sec. 4. A key to this success is using iterative methods for solving the KKT systems in an interior-point algorithm.

2. CONVEX PROBLEM FORMULATION

In this section we show how smoothing along the directional fields of an image can be achieved by solving a convex optimization problem. Estimation of the directional fields (*e.g.*, [6]) or anisotropic diffusion (*e.g.*, [1, 2]) is achieve using local averaging of the gradient or directional fields by averaging the structure tensors (see below) in local neighborhoods.

Let X denote a discrete $m \times n$ grey-scale image with pixels X_{ij}, and let $x = \mathbf{vec}(X)$. We approximate the gradient at (i, j) by a finite difference,

$$\nabla X_{ij} = \begin{pmatrix} X_{i+1,j} - X_{i,j} \\ X_{i,j+1} - X_{i,j} \end{pmatrix} = B_{ij}^T x \qquad (1)$$

with the $mn \times 2$ matrix

$$B_{ij} = \begin{pmatrix} e_{i+1+(j-1)m} - e_{i+(j-1)m} & e_{i+jm} - e_{i+(j-1)m} \end{pmatrix}.$$

For ease of notation we assume that the image is periodically extended in both dimensions, so that (1) is valid for $1 \le i \le$

m, $1 \leq j \leq n$, see *e.g.*, [7] for different boundary conditions in image processing.

The *structure tensor* of the gradients at (i, j) averaged in a neighborhood is then

$$\mathcal{T}_{ij}(X) = \sum_{(k,l) \in N_Q(i,j)} \nabla X_{kl} \nabla X_{kl}^T \qquad (2)$$

where $N_Q(i,j) = \{(k,l) \mid \|(i-k, j-l)\|_\infty \leq Q\}$ defines the neighborhood. Note that the structure tensor encodes directions without orientation, so gradients of opposite direction are added coherently. The largest eigenvector of $\mathcal{T}_{ij}(X)$ gives the average gradient direction, and the smallest eigenvector gives the local direction of flow (see, *e.g.*, [6]).

The main idea of anisotropic smoothing is to encourage the gradients of the smoothed image to be aligned with the largest eigenvector of the structure tensors $\mathcal{T}_{ij}(X)$. If we partition a structure tensor as

$$\mathcal{T} = \begin{bmatrix} \mathcal{T}_{11} & \mathcal{T}_{12} \\ \mathcal{T}_{12} & \mathcal{T}_{22} \end{bmatrix}$$

we can define the *inertia tensor* as

$$\mathcal{T}' = \begin{bmatrix} \mathcal{T}_{22} & -\mathcal{T}_{12} \\ -\mathcal{T}_{12} & \mathcal{T}_{11} \end{bmatrix},$$

i.e., $\mathcal{T}' = \alpha \mathcal{T}^{-1}$ with $\alpha = \det(\mathcal{T}) > 0$ if $\mathcal{T}$ is non-singular (see [8] for a discussion of structure and inertia tensors). Note that $\mathcal{T}'$ is sometimes referred to as the adjugate of $\mathcal{T}$. Obviously, the largest eigenvector of $\mathcal{T}$ corresponds to the smallest eigenvector of $\mathcal{T}'$.

Assume next that the local structure tensors $\mathcal{T}_{ij}(Y)$ of a noisy image Y are estimated using (2) and subsequently converted to inertia tensors $\mathcal{T}'_{ij}(Y)$. Anisotropic smoothing can be achieved by attempting to make the gradients ∇X_{ij} orthogonal to the local directional fields characterized by the smallest eigenvectors of $\mathcal{T}'_{ij}(Y)$, *i.e.*, we can pose the problem

$$\begin{array}{ll} \text{minimize} & \sum_{ij} (\nabla X_{ij})^T \mathcal{T}'_{ij}(Y) \nabla X_{ij} \\ \text{subject to} & \|X - Y\|_F^2 \leq \delta \end{array}$$

which is a convex QCQP [9]. The constraint removes the trivial zero solution and forces the estimate to not depart too much from the noisy observation Y. Such a constraint is typical in, *e.g.*, total variation based image processing; see [3] and references therein.

One application of anisotropic smoothing is enhancement and inpainting of fingerprint images ([2, 3]). As fingerprints are ideally sparse and binary, we can add an additional ℓ_1-norm regularization term to emphasize this property of X. In Sec. 3 we will see how such a regularization can be added at no additional complexity. Thus, with $x = \mathbf{vec}(X)$ and $y = \mathbf{vec}(Y)$, the problem we propose to solve is

$$\begin{array}{ll} \text{minimize} & \frac{1}{2} x^T P x + \gamma \|x\|_1 \\ \text{subject to} & \frac{1}{2} \|x - y\|_2^2 \leq \delta \end{array} \qquad (3)$$

where $P = \sum_{ij} B_{ij} \mathcal{T}'_{ij}(Y) B_{ij}^T$. The dimension of P is mn which is often very large, but P is extremely sparse (see Fig. 2) which we exploit in the following implementation.

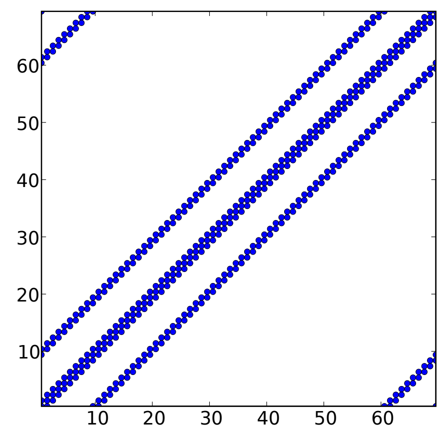

Fig. 1. Sparsity pattern of P for $m = 10$, $n = 7$.

3. LARGE-SCALE IMPLEMENTATION

We consider efficient large-scale implementations of (3) based on a primal-dual interior-point algorithm. The implementation is standard and covered in several texts (*e.g.*, [4]), so we just outline the basic steps.

At the heart of any interior-point algorithm is solving the KKT system for a differentiable problem. Thus we first rewrite (3) in epigraph form as

$$\begin{array}{ll} \text{minimize} & \frac{1}{2} x^T P x + \gamma \mathbf{1}^T t \\ \text{subject to} & -t \preceq x \preceq t \\ & \frac{1}{2} \|x - y\|_2^2 \leq \delta, \end{array} \qquad (4)$$

where $\mathbf{1}$ denotes the vector of all ones, and $\preceq$ denotes "elementwise $\leq$". The KKT system arises from linearizing first-order optimality conditions for (4) after introducing slack-variables for the inequalities. The slack-variables can immediately be eliminated from the KKT system, and for brevity we skip this initial step. The KKT system we solve is then

$$\begin{bmatrix} P + zI & 0 & x - y & I & -I \\ 0 & 0 & 0 & -I & -I \\ (x-y)^T & 0 & -d & 0 & 0 \\ I & -I & 0 & -D_u & 0 \\ -I & -I & 0 & 0 & -D_l \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta t \\ \Delta z \\ \Delta z_u \\ \Delta z_l \end{bmatrix} = r$$

where the scalar $d > 0$ and the positive definite diagonal scaling matrices D_u and D_l are given matrices that depend on the variables x, t and z, and arise from eliminating the slack variables, and r is a residual vector which is zero at the optimal point.

We next eliminate Δt, Δz_u and Δz_l to get a reduced KKT system

$$\begin{bmatrix} H & x-y \\ (x-y)^T & -d \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta z \end{bmatrix} = \begin{bmatrix} \tilde{r}_1 \\ \tilde{r}_2 \end{bmatrix} \tag{5}$$

where $H = P + zI + D_u - D_u^2 D_l^{-1}$. The sparse symmetric indefinite system (5) can be solved using sparse LU or LDL^T factorizations, but the amount of fill-in (using standard reorderings) makes this approach intractable for large images (say, when $mn > 10^5$). A more viable approach is then to solve (5) using an iterative method, *e.g.*, MINRES [10] which has small memory requirements.

If we resort to iterative methods, we can also further eliminate Δz and solve the symmetric positive definite system

$$(H + d^{-1}(x-y)(x-y)^T)\Delta x = \tilde{r}_1 + d^{-1}(x-y)\tilde{r}_2. \tag{6}$$

Since the coefficient matrix in (6) is the sum of a sparse matrix and a rank one matrix we can solve (6) efficiently using the method of conjugate gradients.

A critical component of iterative methods is efficient preconditioners. A cheap preconditioner with linear complexity, that works very well for all our tests is to solve

$$\widehat{H}\,\Delta x = \tilde{r}_1 + d^{-1}(x-y)\tilde{r}_2. \tag{7}$$

where $\widehat{H}$ is a tridiagonal approximation of H, *i.e.*, we solve (6) ignoring the rank-one term as well as elements outside the tridiagonal band. We also experimented with a preconditioner that directly solved (6) using the same tridiagonal approximation using the matrix inversion lemma, but the best results were obtained solving (7). Other preconditioners for structured matrices can be found in [11].

4. EXAMPLES

To illustrate our algorithm we use a low-quality fingerprint from the database in [12]; see Fig. 3a. Similar to the nonlinear diffusion processes in, *e.g.*, [1, 2, 3] we apply the algorithm repeatedly on the enhanced image, *i.e.*, we first compute an enhanced image of the noisy observation Y, and we then rerun the algorithm using the enhanced image as input Y, et cetera.

The dimension of this image is 480×300, *i.e.*, we have sparse QCQP with 144.000 variables. We implemented the algorithm (4) using the optimization tool [13], and the code for generating the plots in Fig. 3 is included in the example section of [13]. The figure clearly illustrates the effect of both the anisotropic smoothing, as well as the sparsity encouraged by the ℓ_1 regularization term. This gives rise to a thinning effect, which is desirable in algorithms for fingerprint recognition [12].

We solved the problems using the CG algorithm with the tridiagonal preconditioner in (7) to a tolerance level of 10^{-5} for the residual norm, which is a fairly high accuracy for such large-scale problems. For this tolerance level, the images in Fig. 3 took 404, 228, 223, and 181 seconds to compute, respectively, on a standard AMD Opteron 2.0GHz PC, which is quite remarkable considering the size of the problem. We observed that solving (5) using MINRES was in general slower than solving (6) using CG.

For comparison we also solved the problems without a preconditioner, to a lower accuracy of 10^{-3}. Fig. 2 shows the number of Newton steps and conjugate gradient iterations in the algorithm with and without a preconditioner. We observe the fundamental problem with the conjugate gradient method, namely that close to the optimal solution the KKT system becomes increasingly ill-conditioned, and without an efficient preconditioner convergence of the CG algorithm slows down significantly.

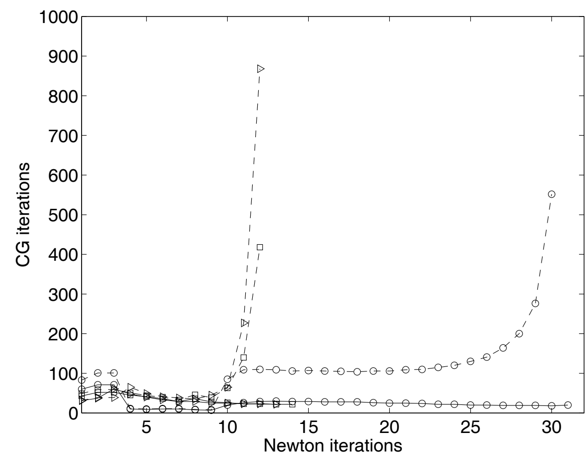

Fig. 2. Number of CG iterations at different Newton iterations for the examples in Fig. 3; solid lines for the tridiagonal preconditioner, and dashed lines without a preconditioner.

5. CONCLUSIONS

We formulated a discrete anisotropic smoothing problem with an ℓ_1 regularization term as a convex quadratically constrained quadratic programming problem (QCQP). Similar problems have previously been derived from partial differential equations (PDEs) combined with non-linear diffusion processes in the literature.

In principle, a convex programming approach has the advantage of offering a straightforward implementation and the ability to add additional heuristic constraints and regularization terms, but in practice the large dimensions involved with image processing make implementation issues non-trivial. So far interior-point algorithms have only scarcely been applied to problems in image processing, but with the large-scale implementations in this correspondence we hope to demonstrate that such an approach is indeed tractable.

Fig. 3. Example of anisotropic smoothing of a low-quality fingerprint image. From left to right: noisy original, and images after 1, 2 and 3 iterations.

Fig. 4. Subregions of the images in Fig. 3.

6. REFERENCES

[1] J. Weickert, "Coherence-enhancing diffusion filtering," *Int. J. Comp. Vision*, vol. 31, no. 2/3, pp. 111–127, 1999.

[2] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, "Filling-in by joint interpolation of vector fields and gray levels," *IEEE Trans. Image Proc.*, vol. 10, no. 8, pp. 1200–1211, 2001.

[3] T.F. Chan and J.J. Shen, *Image Processing and Analysis*, SIAM, 2005.

[4] S.J. Wright, *Primal-Dual Interior-Point Methods*, SIAM, 1997.

[5] D. Goldfarb and W. Yin, "Second-order cone programming methods for total variation based image restoration," *SIAM J. Scientific Computing*, vol. 27, no. 2, pp. 622–645, 2005.

[6] A.M. Bazen and S.H. Gerez, "Systematic methods for the computation of the directional fields and singular points of fingerprints," *IEEE. Trans. Pattern Anal. and Mach. Intelligence*, vol. 24, no. 7, pp. 905–919, 2002.

[7] P.C. Hansen, J.G. Nagy, and D.P. O'Leary, *Deblurring Images: Matrices, Spectra, and Filtering*, SIAM, 2006.

[8] B. Jähne, *Spatio-Temporal Image Processing*, vol. 751, Lecture notes in Computer science. Springer, 1993.

[9] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[10] C.C Paige and M. A. Saunders, "Solution of sparse indefinite systems of linear equations," *SIAM J. Numer. Anal.*, vol. 12, no. 4, 1975.

[11] M. Ng, *Iterative Methods for Toeplitz Systems*, Oxford University Press, 2004.

[12] D. Maltoni, D. Maio, A.K. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*, Springer, 2003.

[13] J. Dahl and L. Vandenberghe, "CVXOPT: A python package for convex optimization," available from `www.ee.ucla.edu/~vandenbe/cvxopt`.