

# A SELF-REFERENCING LEVEL-SET METHOD FOR IMAGE RECONSTRUCTION FROM SPARSE FOURIER SAMPLES <sup>1</sup>

*Jong Chul Ye, Yoram Bresler and Pierre Moulin*

Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, IL 61801  
Email: {jong,yoram,moulin}@ifp.uiuc.edu

## ABSTRACT

We address image estimation from sparse Fourier samples. The problem is formulated as joint estimation of the supports of unknown sparse objects in the image, and pixel values on these supports. The domain and the pixel values are alternately estimated using the level-set method and the conjugate gradient method, respectively. Our level-set evolution shows a unique switching behavior, which stabilizes the level-set evolution. Furthermore, the trade-off between the stability and the speed of evolution can be easily controlled by the number of the conjugate gradient steps, hence removing the re-initialization steps in conventional level set approaches.

## 1. INTRODUCTION

In Fourier imaging problems, the goal is to reconstruct an image from samples of its Fourier transform. Examples include synthetic aperture radar (SAR), radio astronomy, and magnetic resonance imaging (MRI). If sufficient number of Fourier samples are available, the reconstruction can be easily obtained using the inverse Fourier transform. However, in many practical situations, owing to physical or economic limitations, only a small number of Fourier samples is available. Our motivating example is a passive radar imaging problem [1], where the sampling pattern in Fourier space is determined by the locations of the radars and commercial TV/FM stations, and the trajectory of the airborne targets.

With sparse Fourier data, the reconstruction problem becomes ill-posed, because it does not admit a unique solution. However, if the image can be assumed to consist of objects supported on a small unknown set  $D$  on a known background (Fig. 1(a)), a unique solution exists, provided the sampling pattern satisfies appropriate so-called universality conditions [2–4]. These conditions are typically satisfied by random sampling patterns, such as those arising in our motivating radar imaging example. The computation of the reconstruction requires, however, the solution of a difficult nonlinear problem.

An alternative approach to such problems developed over the past decade is based on edge preserving regularization (EPR). EPR has been extensively employed for many ill-posed inverse problems such as computed tomography (CT) and positron emission tomography (PET) [5]. In another deterministic form of EPR known as ARTUR [6], the

analog edge map of the image is iteratively updated, reducing smoothing across the edges. Delaney and Bresler [7] provided a convergence proof for a generalized version of ARTUR, and showed excellent image reconstruction results in limited-angle tomography problems.

EPR does not make explicit use of the prior knowledge of sparse objects on a homogeneous background, and therefore may not offer optimum performance with sparse data. Instead, in this paper we explicitly use this prior knowledge, and formulate the problem as the estimation of the support  $D$  as well as the pixel values in it. To solve the associated nonlinear problem, we derive a gradient-based nonlinear optimization technique, where we alternately estimate the domain  $D$  and the pixel values using a level-set evolution method and the conjugate gradient (CG) technique, respectively.

Related work on domain estimation algorithm using level-set approaches has appeared in [8], which considers a binary image. Our approach addresses a more general problem, where both support domain and pixel values are unknown, and are estimated jointly. In addition to reconstructing non-constant objects, our approach allows to reconstruct objects on a known non-homogeneous background, neither of which is possible using the approach in [8]. The latter feature is important in tasks of “change detection” in a reference image acquired at an earlier time. Furthermore, in our level-set approach the embedding level function is initialized using the direct Fourier inversion estimate of pixel value, rather than the conventional signed distance function. This formulation has several unique advantages. First, the initialization of the level-set and the necessary extension of the evolution velocity are simplified. Second, the resultant level-set evolution shows a switching behavior with significant stabilizing effects, hence the level function is never re-initialized. Finally, by combining with CG steps of pixel value update, the stability and the speed of the level-set evolution can be easily controlled by the number of the CG steps.

## 2. PROBLEM FORMULATION

Let  $\Omega \subset \mathbb{R}^2$  denote the image domain and the open set  $D$  a subset (which may be disconnected) of  $\Omega$ . Suppose the unknown image  $v(\mathbf{x})$  ( $\mathbf{x} \in \Omega$ ) has the following form:

$$u(\mathbf{x}) = \begin{cases} v(\mathbf{x}), & \mathbf{x} \in D \\ 0 & \mathbf{x} \in \Omega \setminus D \end{cases} \quad (1)$$

where  $v(\mathbf{x})$  is an unknown pixel value at  $\mathbf{x} \in D$ . Thus, the objects may have arbitrary values over their support. The

<sup>1</sup>This work was supported in part by a grant from DARPA under Contract F49620-98-1-0498, administered by AFOSR, and by a NSF Research Infrastructure Grant No. CDA 96-24396.

model (1) includes more general cases of known background since in those cases the unknown image  $u(\mathbf{x})$  of (1) can be regarded as the unknown difference image. The noiseless measurement in the Fourier imaging problem is given by:

$$F^D v(\mathbf{f}) = \int_D \exp(-j\mathbf{f} \cdot \mathbf{x}') v(\mathbf{x}') d\mathbf{x}' \quad , \quad \mathbf{f} \in \Phi \subset \mathbb{R}^2 \quad (2)$$

where  $\mathbf{f} \in \mathbb{R}^2$  is the frequency,  $\Phi$  denotes the set of 2-D frequency sample locations, and  $d\mathbf{x}'$  denotes the differential surface element. Suppose we have noisy samples  $y(\mathbf{f})$ ,  $\mathbf{f} \in \Phi$ . Our goal is to fit (2) to the noisy samples while penalizing the length of the boundary of the domain  $D$ . The resultant penalized least-squares estimator of  $D$  and  $v$  can be computed by

$$\min_{D,v} \left( \frac{1}{2} \|y - F^D v\|_{\Phi}^2 + \alpha \int_{\Gamma} d\Gamma \right) \quad (3)$$

where  $\|z\|_{\Phi}^2 = \int_{\Phi} |z(\mathbf{f})|^2 d\mathbf{f}$ , the second integral is a line-integral along the boundary  $\Gamma = \partial D$  (hence the length of the domain boundary), and  $\alpha$  is a regularization constant. This form of penalized likelihood can be motivated by various ways, including one based on Risannen's minimum description length (MDL) [9].

### 3. ALTERNATING MINIMIZATION

To solve the optimization problem (3), we use an alternating minimization of the cost function with respect to the pixel value  $v$  and the domain  $D$ . For fixed  $D$ , the optimization problem is simple, admitting the linear least-squares solution

$$\hat{v} = \left( F^D \right)^{\dagger} y \quad (4)$$

where  $A^{\dagger}$  denotes the pseudo inverse of  $A$  and  $\hat{v}$  is defined only on  $D$ . Because of the large dimensions of the matrix involved in the discrete form of (4), the estimator  $\hat{v}$  of (4) is preferably obtained using the conjugate gradient algorithm, without ever having to form the matrix itself.

### 4. THE SHAPE DERIVATIVES

#### 4.1. Shape Deformation

The main technical difficulty of the optimization problem (3) is the estimation of the shape  $D$ . This is a problem of shape optimization consisting of finding the geometries which minimizes a cost functional while satisfying certain of constraints. Usually, such problems are solved by using iterative gradient-based nonlinear optimization techniques. Calculation of the gradient requires the sensitivity analysis of functionals with respect to the shape deformation. Such a shape deformation is described by constructing a family of shapes  $D_t, D_0 = D$  that are perturbations of  $D$  for  $0 \leq t < \epsilon$ ,  $t$  being a fictitious time parameter. These admissible domains  $D_t$  are assumed to be subsets of a larger fixed domain  $\Omega$  and their boundary is assumed to retain the same regularity  $C^k$  as that of the original domain.

The next question is how to define this family of perturbations. One could use ideas developed in continuum mechanics where a family of transformations  $T_t : \Omega \rightarrow \mathbb{R}^2$  for  $t \in [0, \epsilon]$  is constructed such that [10]

$$D_t = T_t(D) \quad , \quad (5)$$

where  $T_t(D)$  is defined by

$$T_t(D) = \{ \mathbf{x}_t : \mathbf{x}_t = T_t(\mathbf{x}), \mathbf{x} \in D \} \quad . \quad (6)$$

The perturbation of the identity

$$T_t(\mathbf{x}) = \mathbf{x} + t\eta(\mathbf{x}) \quad (7)$$

where  $\eta$  is of class  $C^k$ , is a possible form for these transformations. One could also use a more general method known as the speed method [10]. Instead of being defined by its transformation  $T_t$ , the deformation is defined by its speed field  $V(t, \mathbf{x}) \in C([0, \epsilon], V^K(\Omega))$  where  $C([0, \epsilon], V^K(\Omega))$  denotes the space of functions continuously differentiable with respect to  $t \in [0, \epsilon]$  and  $k$ -continuously differentiable with respect to  $\mathbf{x}$  with compact support included in  $\Omega$  and such that these functions leave the boundary  $\partial\Omega$  unchanged in order that the admissible domain  $D_t$  remain a subset of  $\Omega$ , that is:

$$V(t, \mathbf{x}) \cdot n(\mathbf{x}) = 0 \quad , \quad \text{for a.e. } \mathbf{x} \in \partial\Omega \quad (8)$$

$$V(t, \mathbf{x}) = 0 \quad , \quad \text{for all singular points } \mathbf{x} \quad , \quad (9)$$

where  $n(\mathbf{x})$  denote the outer-normal unit vector on the boundary  $\partial\Omega$ , and a singular point is a point where  $n(\mathbf{x})$  is not defined. In fact, these two definitions of the perturbation are equivalent. Indeed, under appropriate regularity conditions, it is possible to associate a unique velocity field  $V(t, \mathbf{x})$  to a given transformation, and vice versa [10].

Now, let  $\mathbf{x}(t)$  denote the solution to the system of ordinary differential equation

$$\frac{\partial \mathbf{x}(t)}{\partial t} = V(t, \mathbf{x}(t)), \quad \mathbf{x}(0) \in D \quad . \quad (10)$$

Then, the following result from shape analysis allows us to use a gradient-based approach for the domain update:

**Proposition 1 (Sokolowski and Zolesio [10])** *Let the domain functions  $Q(D)$  and  $L(D)$  be given by*

$$Q(D) = \int_D Y d\mathbf{x} \quad , \quad L(D) = \int_{\Gamma} Y d\Gamma \quad , \quad (11)$$

where  $\Gamma$  is of class  $C^1$ . Suppose  $D_t$  is deformed by (10) where  $V(t, \mathbf{x}) \in C([0, \epsilon], V^1(\Omega))$ . Then, the domain functions  $Q(D)$  and  $L(D)$  are shape differentiable,

$$Q'(D, V) = \left( \frac{\partial}{\partial t} \int_{D_t} Y d\mathbf{x} \right)_{t=0} = \int_{\Gamma} Y F d\Gamma \quad (12)$$

$$L'(D, V) = \int_{\Gamma} \kappa Y F d\Gamma, \quad (13)$$

where  $F = \langle V(0), n \rangle$  denotes the component of the velocity normal to the boundary  $\Gamma$  at  $t = 0$ ,  $n$  denotes outer normal vector on  $\Gamma$ , and  $\kappa$  denotes the curvature of  $\Gamma$ .

#### 4.2. Shape Derivatives of the Cost Functional

We wish to obtain a closed form for the derivatives of the cost functional (3) with respect to a perturbation of the geometry. Using Proposition 1, we can show that the first term of (3)

$$C(D) = \frac{1}{2} \|y - F^D v\|_{\Phi}^2 \quad (14)$$

has the shape derivative

$$Re \left[ \int_{\Gamma} \left( \int_{\Phi} (F^D v(\mathbf{f}) - y(\mathbf{f}))^* e^{j\mathbf{f}\cdot\mathbf{x}} d\mathbf{f} \right) v(\mathbf{x}) F(\mathbf{x}) d\Gamma \right], \quad (15)$$

where  $\mathbf{x} \in \Gamma$  and the superscript  $*$  denotes the complex conjugate and  $Re[\cdot]$  denotes the real part of its argument. Similarly, the shape derivative of the second term of (3) is given by

$$\int_{\Gamma} \kappa F d\Gamma. \quad (16)$$

Therefore, the Cauchy-Schwartz inequality tells us that the steepest descent direction of the domain update for the minimization problem (3) is given by a normal boundary speed function  $F(x)$  (boundary flow) of the form

$$F = -(E + \alpha\kappa) \chi_{\Gamma} \quad (17)$$

$$E = Re \left[ v^* (F^D)^* (F^D v - y) \right] \quad (18)$$

where  $\chi_{\Gamma}$  denotes the indicator function of the boundary  $\Gamma$ .

## 5. SELF-REFERENCING LEVEL-SET METHOD

### 5.1. Shape Evolution by Level-Sets

The problem of minimizing the cost function (3) with respect to the domain  $D$  has now been reduced to deforming or evolving the boundary  $\Gamma$  by the boundary flow  $F$ . This will provide a steepest descent procedure for the minimization of (3). However, rather than evolve the boundary directly, we adopt the powerful level-set approach pioneered by Sethian and Osher [11]. They described the evolution of boundary  $\Gamma_t$  in (10) and the domain  $D_t$  as a zero-level set of the three-dimensional embedding function  $\phi(t, \mathbf{x}) : \mathbb{R}^+ \times \mathbb{R}^2 \rightarrow \mathbb{R}$ :

$$\begin{aligned} \Gamma_t &= \{ \mathbf{x} \in \mathbb{R}^2 | \phi(t, \mathbf{x}) = 0 \} \\ D_t &= \{ \mathbf{x} \in \mathbb{R}^2 | \phi(t, \mathbf{x}) < 0 \}. \end{aligned} \quad (19)$$

For a given boundary flow  $F$ , the evolution of the domain is then given by

$$\frac{\partial \phi}{\partial t} = -F |\nabla \phi|. \quad (20)$$

Eq. (20) is a Hamilton-Jacobi-type equation in continuum mechanics [11]. Using (20), topological changes of the domain evolution can be handled easily. Furthermore, the level-set algorithm can be implemented in a fixed Cartesian grid throughout the iterative process. For our problem,  $F$  is given by (17).

### 5.2. The Velocity Extension Problem

A somewhat hidden aspect of the level-set method (20) is that  $F$  is assumed defined on all  $\Omega$ , not just on  $\Gamma$ . Thus, not only is the interface embedded in a higher dimensional function, but the normal speed of the interface is itself embedded in a higher dimensional function:

$$\frac{\partial \phi}{\partial t} = -F_e |\nabla \phi| \quad (21)$$

where  $F_e$  is some normal boundary speed field which, at the zero level-set, equals the given speed  $F = \langle V(0), n \rangle$ . This new speed field  $F_e$  is known as the ‘‘extension velocity’’, and must be chosen judiciously.

The freedom to choose the extension velocity is equivalent to the freedom to choose the embedding of the boundary curve  $\Gamma$  into the level-set function  $\phi$ . Indeed, for a given boundary  $\Gamma$  at time  $t$ , (19) specifies little more than the zero level curve of  $\phi(t, \cdot)$ . The rest of  $\phi(t, \cdot)$  can be either specified directly, giving rise (through the evolution equation (21)) to a natural velocity extension  $F_e$ , or indirectly, by specifying  $F_e$ , and letting  $\phi$  be determined by (21).

There are various approaches to constructing the extension velocity. Sethian’s original level-set calculations [11] were concerned with interface problems with geometric propagation speeds such as curvature flow, leading to a natural construction of an extension velocity by using the geometry of the given level-set. In these flows, the evolution equation (20) converges to a correct boundary for any choice of the embedding function that satisfies certain smoothness conditions. Based on this, the signed distance function is usually used as a level function owing to its simplicity. Adalsteinsson and Sethian [12] proposed a fast extension method of the velocity which preserves the signed distance in narrow band around the zero level curve.

However, neither the signed distance, nor this choice of velocity extension are effective for our problem, for the following reasons. If the signed distance is used as a level function, the level function must be re-initialized after several iterations to preserve the signed distance. Usually this is done by solving another PDE of the level function [11]. This procedure is time-consuming, but it is essential to stabilize the level-set evolution. However, the more frequent the re-initialization iteration, the more the zero level set may stray from the expected position [11]. The advantage of our approach over such re-initialization is the ‘‘switching’’ behavior of our level-set evolution, which keeps the level function stable avoiding the problematic re-initialization step.

We are aware that in the narrow-band level set method [11], the velocity extension method provides a way to update an interface such a way that the signed distance function is maintained, and the front is never re-initialized. However, in comparison to narrow-band extension methods, our velocity extension is easily obtained from the level function evolution, reducing the computation and simplifying the algorithm. In the following, we explain the details of our new level-set formulation.

### 5.3. Velocity Extension

Recall the normal velocity field (17). One may think that a simple extension of (18) to the whole domain  $\Omega$  is obtained by removing the indicator function:

$$- \left( Re \left[ v^* (F^D)^* (F^D v - y) \right] + \alpha\kappa \right). \quad (22)$$

However, since the image estimate  $v$  is restricted to  $D$ , the first term in (22) is only defined on  $D$ .

In a binary image reconstruction problem [8], the derivation of the velocity extension is also based on the form of the steepest descent direction boundary flow at  $\Gamma$ . In this problem, the image is assumed to take the known *constant* values  $v_{in}$  and  $v_{out}$  on  $D$  and  $\Omega \setminus D$ , respectively. Santosa

showed that in this binary problem, the required velocity field at the boundary (without including the curvature flow) is given by

$$F' = -(v_{in} - v_{out})Re[(F^D)^*(F^D v_{in} - y)]\chi_\Gamma \quad (23)$$

where  $A^*$  denotes the adjoint of an operator  $A$ , and  $\chi_\Gamma$  denotes the indicator function of the boundary  $\Gamma$ . By removing the indicator function from (23) and replacing  $(F^D)^*$  by  $(F^\Omega)^* = (F^D)^* + (F^{\Omega \setminus D})^*$ , he obtained the velocity extension:

$$F'_e = -(v_{in} - v_{out}) \left( Re \left[ (F^D)^* (F^D v_{in} - y) \right] + Re \left[ (F^{\Omega \setminus D})^* (F^D v_{in} - y) \right] \right) \quad (24)$$

where the first and the second terms are defined on  $D$  and on  $\Omega \setminus D$ , respectively. A straightforward analogy of (24) applicable to our problem is

$$F_e = -(E_e + \alpha\kappa) \quad (25)$$

$$E_e = Re \left[ v^* (F^D)^* (F^D v - y) \right] + Re \left[ w^* (F^{\Omega \setminus D})^* (F^D v - y) \right], \quad (26)$$

for some nonzero complex-valued function  $w$  defined on  $\Omega \setminus D$ . Using (26), the level set evolution is now given by

$$\frac{\partial \phi}{\partial t} = E_e |\nabla \phi| + \alpha\kappa |\nabla \phi|. \quad (27)$$

In order to avoid technical difficulties such as scaling and phase mismatch, a good candidate for  $w$  is an estimate of  $v$  defined on  $\Omega \setminus D$  or  $\Omega$ . What functions  $w(\cdot)$  can be used? The following sections answer this question.

#### 5.4. A Self-Referencing Level-Set Method

In many Fourier imaging problem, the inverse Fourier transform image

$$\hat{u}_0 = (F^\Omega)^{-1} y = (F^\Omega)^* y, \quad (28)$$

is used as a reconstructed image or at least as an initial guess for image enhancement algorithms. However, in the level set algorithm for binary image reconstruction problems [8], such information has not been utilized explicitly.

How can we inject the information (28) into the level set algorithm? Recall from our earlier discussion, that we have freedom in the choice of a level-set function. One idea is to use (28) to initialize the level function. Indeed in an inverse scattering problem of permittivity inversion, Dorn, Miller and Rappaport [13] already used the approximate solution from the nonlinear inverse scattering solver as a level function initialization.

Such a formulation of the level function has several advantages over the signed distance function. First, the new level function has approximate informations about the image intensity. Hence, the level function has multiple valleys at the desired locations, which allows fast evolution of the level set. Second, since the term  $|\nabla \phi|$  of (21) works as an edge map of the level function, we can expect more rapid evolution of the level function in the edge area of the approximate images.

Based on these observation, we define an image magnitude function  $m(t, \mathbf{x})$ , and initialize it to the magnitude of the inverse Fourier reconstruction of (28),  $m(0, \mathbf{x}) = |\hat{u}_0(\mathbf{x})|$ . We then pick the level-set function to be

$$\phi(t, x) = g[m(t, \mathbf{x})] \quad \forall t \geq 0 \quad (29)$$

where  $g: \mathbb{R}^+ \rightarrow \mathbb{R}$  is a decreasing invertible continuous differentiable function. For example, an adequate choice might be  $g(z) = T_H - z$ , where  $T_H$  is an appropriate threshold value, which is less than the maximum value of  $m(0, \mathbf{x}) = |\hat{u}_0(\mathbf{x})|$ . This choice then initializes the level set to

$$\phi(0, \mathbf{x}) = g[m(0, \mathbf{x})] \quad , \quad m(0, \mathbf{x}) = |\hat{u}_0(\mathbf{x})|. \quad (30)$$

Importantly, Eq. (29) can now be used to provide a new evolution equation for  $w(t, \mathbf{x})$ , which can be used for our velocity extension (26).

**Proposition 2** *Suppose the level set  $\phi(t, \mathbf{x})$  is defined as in (29), and  $\phi$  is a solution to the evolution equation (27) with initial conditions (30). Then the magnitude function  $m = g^{-1}(\phi)$  is a solution to the following evolution equation:*

$$\frac{\partial m}{\partial t} = -E_e |\nabla m| + \alpha\kappa_m |\nabla m|, \quad (31)$$

$$m(0, \mathbf{x}) = |\hat{u}_0(\mathbf{x})| \quad (32)$$

where  $\kappa_m$  denotes the curvature of  $m(t, \mathbf{x})$ .

*Proof:* See Appendix.  $\square$

Note that the flow (31) and (32) is an anisotropic diffusion of the magnitude image  $|\hat{u}_0(\mathbf{x})|$  produced by direct Fourier inversion. Related anisotropic diffusions have been used for edge enhancement and noise removal. For example, if  $E_e = 0$ , then the flow (31) and (32) becomes a curvature flow [11], where sharp boundaries are preserved and smoothing takes place inside a region, but not across region boundaries. Based on this observation, we propose to use  $m(t, \mathbf{x})$  as an estimate of the image magnitude  $|v(t, \mathbf{x})|$  on  $\Omega \setminus D$ . Now, by assuming that the phase of the image on  $\Omega \setminus D$  does not change significantly during the evolution, we obtain the following evolution for  $w(t, \mathbf{x})$ :

$$w(t, \mathbf{x}) = m(t; \mathbf{x}) e^{j\angle \hat{u}_0(\mathbf{x})}, \quad (33)$$

where  $\angle$  denotes the argument of a complex quantity. Note that with this definition of  $w(t, \mathbf{x})$ , we have  $w(0, \mathbf{x}) = \hat{u}_0(\mathbf{x})$ , so that  $w(0, \mathbf{x})$  agrees with unconstrained Fourier reconstruction on  $\Omega \setminus D$ .

Because the level function is initialized as a function of the image estimate (rather than e.g., a signed distance function), we will call this level-set formulation a *self-referencing level set method*.

#### 5.5. Stability of Level Function

Recall that for the choice of the signed distance function as a level function, periodic re-initialization is performed to maintain the signed distance, and hence stabilize the evolution [11]. Likewise, in order to stabilize their evolution, Dorn, Miller and Rappaport [13] periodically solve the heat equation for  $\phi(t, \mathbf{x})$ , smoothing out irregular behavior of the level function during the evolution. (Such an approach tends to drive the zero level curve away from the expected

position.) In contrast, the proposed self-referencing level set method has no explicit formula for stabilizing the level-function evolution. This raises some questions about its stability.

The main contribution of our algorithm is that such re-initialization steps are not necessary. Note that by substituting (4) into (26), the extension velocity is simplified as:

$$\begin{aligned} E_e &= \operatorname{Re} \left[ v^* (F^D)^* P_D^\perp y \right] + \operatorname{Re} \left[ w^* (F^{\Omega \setminus D})^* P_D^\perp y \right] \\ &= \operatorname{Re} \left[ w^* (F^{\Omega \setminus D})^* P_D^\perp y \right], \end{aligned} \quad (34)$$

where  $P_D^\perp$  is the orthogonal projection onto the range space of  $F^D$ , or, equivalently, onto the null space of  $(F^D)^*$ , and the second equality follows because  $(F^D)^* P_D^\perp = 0$ .

Note that  $E_e$  in (34) is nonzero only in  $\Omega \setminus D$ . Therefore, the level-set evolution is given by

$$\frac{\partial \phi}{\partial t}(t, \mathbf{x}) = \begin{cases} \alpha \kappa |\nabla \phi(t, \mathbf{x})|, & \mathbf{x} \in D \\ (E_e + \alpha \kappa) |\nabla \phi(t, \mathbf{x})|, & \mathbf{x} \notin D \end{cases}. \quad (35)$$

Eq. (35) tells us that if  $\mathbf{x} \notin D$  the level-set evolution is driven by the data-driven force  $E_e$  and the roughness of the level-set  $|\nabla \phi|$ . If the estimate of  $D$  is incorrect, there will be data mismatch (i.e.,  $P_D^\perp y \neq 0$ ) resulting in a nonzero data-driven force  $E_e$  and allowing the level-set to keep evolving. However, for  $\mathbf{x} \in D$  the curvature flow will stabilize the level-set evolution. Thus, the evolution exhibits a switching behavior between the interior and exterior of the domain  $D$ . Whatever the estimated domain  $D$ , within  $D$   $\phi$  only undergoes smoothing by non-isotropic diffusion, and outside  $D$  it evolves under data influence.<sup>1</sup>

This “switching” controls the minimum value of the level function. More specifically, consider the adequate choice  $g(z) = T_H - z$  with  $T_H$  constant throughout the evolution iteration. If  $\phi(t, \mathbf{x}) < 0$ , then  $\mathbf{x}$  becomes an element of  $D$ , and the value of  $\phi(t, \mathbf{x})$  is now suppressed to zero by the curvature flow. Similarly, if  $\phi(t, \mathbf{x}) \geq 0$ , then  $\mathbf{x} \in \Omega \setminus D$  hence it is evolving by the data driven flow. Owing to this feedback cycle, we can show that asymptotically the minimum value of the level function is zero, hence

$$\max_{\mathbf{x} \in \Omega} \lim_{t \rightarrow \infty} |w(t, \mathbf{x})| = T_H. \quad (36)$$

Without such “switching” behavior, the contribution of  $E_e$  on  $D$  may not be negligible and the evolution of  $\phi$  may not converge even for the correct domain  $D$  if the measurement  $y$  is noisy. For example, in the velocity extension method (24) for binary image reconstruction problem [8], if the measurement  $y$  is noisy,  $F \neq 0$  even for the correct domain  $D$ , hence there is no explicit way to stop the evolution.

Another advantage of our algorithm is that the “switching” effect can be easily controlled by the number of CG steps. For examples, if a small number of CG steps are used,  $\hat{v} \neq (F^D)^\dagger y$  and the contribution of  $F$  on  $D$  may

<sup>1</sup>As revealed by the derivation leading to (35), this feature is obtained owing to the problem formulation and the choice of steepest descent – and has little to do with the particular velocity extension. The latter only affects the evolution on  $\Omega \setminus D$ .

be significant. This tends to allow rapid evolution of the level function. On the contrary, for large number of CG steps, the contribution of  $F$  becomes negligible, hence the level function is stabilized. Therefore, the number of CG steps will be used to control the evolution speed and the stability of evolution. In practice, we will use small number of CG iterations in the early phase of evolution in order to allow rapid evolutions, and increases the number of CG steps according to the evolution in order to keep the level set stable.

## 6. NUMERICAL IMPLEMENTATION

Once the velocity is chosen, all coefficients that are involved in the Hamilton-Jacobi equation are defined. Now we describe the numerical algorithm for Equation (27). We assume that the image domain  $\Omega$  is rectangular and discretized into  $N \times N$  pixels of size  $\Delta x$  and  $\Delta y$ . Define  $\phi_{ij}^t = \phi(t, i\Delta x, j\Delta y)$ . The Hamilton-Jacobi equation is replaced by

$$\frac{1}{\Delta t} (\phi_{ij}^{t+1} - \phi_{ij}^t) = E_{ij}^t H(\phi_{ij}^t) + \alpha \kappa_{ij}^t H(\phi_{ij}^t), \quad (37)$$

where  $w_{ij}^t$  denotes the value taken by the function  $w$  at the point  $(i\Delta x, j\Delta y) \in \Omega$  at time  $t$  and where  $H$  corresponds to the numerical Hamiltonian, which is in fact an approximation of the gradient  $|\nabla \phi|$ , and  $\kappa_{ij}^t$  denotes an approximation of the curvature.

Three issues need to be addressed in the implementation of (37). The first is how to define the numerical Hamiltonian and curvature. In order to ensure the stability of the numerical evolution, we follow Sethian’s proposal for the *upwinding* finite difference scheme [11] for  $H$ , which is determined by the sign of the velocity  $E_{ij}$  in (37). The choice ensures that the domain expands when the velocity is positive:

- if  $E_{ij} \leq 0$ , the numerical Hamiltonian is given by

$$\begin{aligned} H(\phi_{ij}^t)^2 &= \max(D_-^x \phi_{ij}^t, 0)^2 + \min(D_+^x \phi_{ij}^t, 0)^2 \\ &\quad + \max(D_-^y \phi_{ij}^t, 0)^2 + \min(D_+^y \phi_{ij}^t, 0)^2 \end{aligned} \quad (38)$$

- if  $E_{ij} > 0$ , then

$$\begin{aligned} H(\phi_{ij}^t)^2 &= \min(D_-^x \phi_{ij}^t, 0)^2 + \max(D_+^x \phi_{ij}^t, 0)^2 \\ &\quad + \min(D_-^y \phi_{ij}^t, 0)^2 + \max(D_+^y \phi_{ij}^t, 0)^2 \end{aligned} \quad (39)$$

where  $D_+^x \phi_{ij}^t = (\phi_{i+1,j}^t - \phi_{i,j}^t)/\Delta x$  and  $D_-^y \phi_{ij}^t = (\phi_{i,j}^t - \phi_{i,j-1}^t)/\Delta y$ . However, the numerical Hamiltonian in the last term of (37) is implemented using the following central difference approximation:

$$H(\phi_{ij}^t)^2 = (D^x \phi_{ij}^t)^2 + (D^y \phi_{ij}^t)^2 \quad (40)$$

where  $D^x \phi_{ij}^t = \frac{\phi_{i+1,j}^t - \phi_{i-1,j}^t}{2\Delta x}$  and  $D^y \phi_{ij}^t = \frac{\phi_{i,j+1}^t - \phi_{i,j-1}^t}{2\Delta y}$ . Since the expression for the curvature of the zero level set assigned to the interface itself is given by

$$\kappa = \nabla \cdot \left( \frac{\nabla \phi}{|\nabla \phi|} \right) = \frac{\phi_{xx} \phi_y^2 - 2\phi_y \phi_x \phi_{xy} + \phi_{yy} \phi_x^2}{(\phi_x^2 + \phi_y^2)^{3/2}}, \quad (41)$$

its numerical approximation  $\kappa_{ij}^t$  is implemented using the central difference formula.

The second issue concerns the boundary conditions to impose on the level-set. Since the velocity is such that the boundary of the domain  $\Omega$  should remain fixed, the level-set function satisfies a Neumann boundary condition:

$$\nabla\phi(t, \mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) = 0 \quad , \quad \mathbf{x} \in \partial\Omega . \quad (42)$$

This condition too is implemented using a finite difference approximation to  $\nabla\phi(t, \mathbf{x})$ . For the rectangular  $\Omega$  in the examples we consider, the normal vector  $\mathbf{n}(\mathbf{x})$  is parallel to the  $x$  or  $y$ -axis. Therefore, the boundary condition  $\nabla\phi \cdot \mathbf{n} = 0$  is discretized to one of the four conditions  $D_-^y = 0$ ,  $D_+^y = 0$ ,  $D_-^x = 0$ , or  $D_+^x = 0$ , depending on the location of the boundary. In each case, the choice between  $D_-$  or  $D_+$  is made so that only pixels in  $\Omega$  are involved.

The third implementation issue concerns the time step  $\Delta t$ . There are inherent time step requirements in the above discrete implementation. We have a Courant-Friedrich-Leroy condition [11] which requires the front to cross no more than one grid cell each time step. Thus, we require that

$$\max_{\Omega} |F| \Delta t \leq \min\{\Delta x, \Delta y\} \quad (43)$$

where the maximum is taken over values for  $F$  for all possible grid points. We use the following time step size:

$$\Delta t = \frac{0.99 \min\{\Delta x, \Delta y\}}{\max_{i,j} |E_{ij}^t + \alpha \kappa_{ij}^t|} . \quad (44)$$

In practice,  $\kappa_{ij}^t$  in (44) becomes very large in a flat region<sup>2</sup>, so  $\Delta t$  becomes very small. In order to avoid this situation, we compute  $\Delta t$  in (44) assuming  $\alpha = 0$ .

## 7. COMPUTATIONAL COST

The main computational burden of the algorithm are the CG steps for the pixel-value update and the computation of the extension velocity (26). First consider the CG step. Specifically, we use the Polak-Ribiere version of the conjugate gradient algorithm.

Initialize:

$$q = (F^D)^* y, \quad \hat{v}^0 = 0 \quad d^0 = q \quad (45)$$

For  $k = 0, 1, \dots$ ,

$$\hat{v}^{k+1} = \hat{v}^k + \alpha_k d^k \quad (46)$$

$$r^{k+1} = (F^D)^* F^D \hat{v}^{k+1} - q \quad (47)$$

$$d^{k+1} = -r^{k+1} + \beta_k d^k \quad (48)$$

where

$$\beta_k = \frac{\langle r^{k+1}, r^{k+1} \rangle}{\langle r^k, r^k \rangle} \quad (49)$$

$$\alpha_k = -\frac{\langle r^k, d^k \rangle}{d_k^* (F^D)^* F^D d_k} \quad (50)$$

For a regular grid, the computation of  $q$  in (45) can be implemented using the fast Fourier transform (FFT) after zero padding  $y$  on  $\Omega \setminus \Phi$ . In (47), the computation of  $(F^D)^* F^D \hat{v}^{k+1}$  can be implemented by computing the FFT

<sup>2</sup>However,  $\kappa_{ij}^t H(\phi_{ij}^t)$  is well defined.

of  $\hat{v}^{k+1}$ , zero-padding on  $\Omega \setminus \Phi$ , and finally performing the inverse FFT (IFFT). Computation of  $\alpha_k$  can be done similarly using one FFT of  $d_k$ . In total, for  $k \geq 1$  one iteration of CG algorithm requires two FFT and one IFFT.

To compute the extension velocity  $E_e$  of (26), we need to compute

$$v^* (F^D)^* (y - F^D v) , \quad (51)$$

which can be again implemented using one FFT and additional IFFT. Therefore, overall complexity of the algorithm per evolution iteration is given by  $O(N^2 \log N)$  for  $N \times N$  images.

## 8. NUMERICAL RESULTS

We provide four examples to evaluate the performance of our algorithm. We use the function  $g(z) = T_H - z$  for all our simulation where  $T_H$  is a threshold value, which is set depending on each problem. The regularization parameter  $\alpha$  in (27) is hand-picked for best performance. To describe the experiments, we introduce the following definitions. For  $N \times N$ -pixel image we measure region sizes,  $|D|$ ,  $|\Phi|$  and  $|\Omega| = N^2$  by the number of pixels they contain. The fraction  $\eta = |D|/N^2$  of the image occupied by the supports of the objects is called the *occupancy rate*. Likewise, because the total number of Fourier samples is  $|\Phi|$ , we call  $\rho = |\Phi|/N^2$  the *sampling rate*. Theoretical analysis indicates that reliable reconstruction requires  $\rho > \eta$ , often by some factor [14].

**Example 1.** The original piecewise constant  $128 \times 128$  image (Fig. 1(a)) consists of two rectangles and a circle with different pixel values (2.353, 1.568, and 0.784). The occupancy rate is  $\eta = 25\%$ . In this experiment, we reconstruct the pixel values and the domain from randomly chosen samples in the 2-D Fourier domain, with sampling rate  $\rho = 50\% = 2\eta$ . Additive complex Gaussian noise is added to the samples at an rms level of 5%. The direct Fourier inversion result  $\hat{u}_0(\mathbf{x})$  after zero-padding (Fig. 1(b)) is quite blurred.

In order to control the evolution speed and stability, we use the following rule for the number of the CG steps in the subsequent simulations:  $N_{CG} = 5 + \lfloor \frac{N_{EV}}{20} \rfloor$  where  $N_{CG}$  and  $N_{EV}$  denote the number of CG steps and the evolution number, respectively, and  $\lfloor x \rfloor$  denotes the largest integer less than or equal to  $x$ . Asymptotically, the choice of  $N_{CG}$  guarantees that the level set becomes stable.

The magnitude of the pixel values and the domain estimates are compared to ground truth in Figs. 2(a)-(b). Accurate pixel value estimates and nearly perfect domain estimate are obtained. The resulting level-function (Fig. 2(c),  $T_H - \phi(t, \mathbf{x})$  shown) appears to have a well-behaved form, free of spurious excursions. Its value is clipped at  $\phi = T_H$ , as predicted by the discussion in Sec. 5.5. Here, the threshold  $T_H$  and the regularization parameter  $\alpha$  are set to 60% of the maximum value of  $m(0, \mathbf{x})$  and 0.3, respectively.

**Example 2.** The test image is generated by cutting the Lena image as shown in Fig. 3. The pixel values are normalized to have maximum value of 2, and  $T_H$  and  $\alpha$  are set to 60% of the maximum value of  $m(0, \mathbf{x})$  and 0.1, respectively. The domain  $D$ , occupancy, and the noise level are all the same as in Example 1. We reduce the sampling

rate down to  $\rho = 37.5\% = 1.5\eta$ . Again, while the direct Fourier inversion (Fig. 3(b)) is blurry, the reconstruction by our algorithm after 200 iteration (Fig. 3(c)) is accurate. The domain estimate (Fig. 3(d)) are also quite accurate.

**Example 3.** This example is drawn from a passive radar imaging problem [1], where the sampling pattern in Fourier space is determined by the locations of the radar and commercial TV/FM stations, and the trajectory of the airborne targets. The pixel values are normalized to have the maximum magnitude of 2, and additive complex Gaussian noise is added to the samples at an rms level of 5%. The occupancy rate is now  $\eta = 37\%$ , but the sampling rate is only  $\rho = 0.33\eta$ . This is a severely undersampled case, which according to theory [15] does not suffice for reconstruction of even one third of the object support. A true VFY aircraft reflectance image and the direct Fourier inversion result are given in Figs. 4(a)(b). The threshold  $T_H$  and the regularization parameter  $\alpha$  are set to 30% of the maximum value of  $m(0, \mathbf{x})$  and 0.001, respectively. The reconstruction by our algorithm after 100 iterations (Fig. 4(c)) clearly shows the structure of the VFY aircraft.

**Example 4.** This example is drawn from a magnetic resonance imaging problem. The goal is “change detection” in a complex background, which is however known, from previous reference scans. Consider the original “Sheep-Logan” phantom given in Fig. 5(a). The maximum value of the phantom is normalized to the value of 5. Suppose we add three “tumors” as shown in Fig. 5(b), which have *unknown* additional absorption values of 0.5, 0.6, and 1.0 and support. Assuming that we know the original healthy phantom of Fig. 5(a), our goal is now to estimate the image in Fig. 5(b) from its noisy Fourier samples. Again, additive complex Gaussian noise is added to the samples at an rms level of 5%. The occupancy rate of the difference image - which corresponds to three tumors - is now  $\eta = 2.3\%$ , and the sampling rate is only  $\rho = 2\eta = 4.6\%$ . A Fourier reconstruction in Fig. 6(b) is very poor and we cannot locate any tumors, because of the poor point spread function of the system. However, the reconstruction after 100 iterations using the new method (Figs. 6(c)) is still good. Here, the threshold  $T_H$  and the regularization parameter  $\alpha$  are set to 70% of the maximum value of  $m(0, \mathbf{x})$  and 0.0005, respectively.

## 9. CONCLUSION

We introduced a new algorithm for image estimation from sparse Fourier samples. The problem was formulated as joint estimation of the supports of unknown sparse objects in the image, and pixel values on these support. The domain and the pixel values are alternately estimated using what we call the self-referencing level-set method and the conjugate gradient method, respectively. Our level function was initialized as a function of the magnitude of the direct Fourier inversion results. This choice also provides an evolution equation for the magnitude image. Furthermore, the number of the conjugate gradient step controls the speed and the stability of evolution, which totally removes the re-initialization step in conventional level set methods. Simulation results suggest that our algorithm provides accurate image and domain estimates from sparse Fourier samples (close to the theoretical minimum) and is quite robust to noise.

## 10. APPENDIX

Note that  $\phi(t, \mathbf{x}) = g(m(t, \mathbf{x}))$ , where  $g : \mathbb{R}^+ \rightarrow \mathbb{R}$  is decreasing invertible continuously differentiable function. Using the chain rule, each term in the evolution equation (27) can be converted as:

$$\frac{\partial \phi(t, \mathbf{x})}{\partial t} = g'(m(t, \mathbf{x})) \frac{\partial m(t, \mathbf{x})}{\partial t} \quad (52)$$

$$|\nabla \phi(t, \mathbf{x})| = |\nabla m(t, \mathbf{x})| |g'(m(t, \mathbf{x}))| \quad (53)$$

$$\begin{aligned} \kappa &= \nabla \cdot \left( \frac{\nabla \phi(t, \mathbf{x})}{|\nabla \phi(t, \mathbf{x})|} \right) \\ &= \nabla \cdot \left( \frac{g'(m(t, \mathbf{x})) \nabla m(t, \mathbf{x})}{|g'(m(t, \mathbf{x}))| |\nabla m(t, \mathbf{x})|} \right) \\ &= \text{sgn}(g'(m(t, \mathbf{x}))) \nabla \cdot \left( \frac{\nabla m(t, \mathbf{x})}{|\nabla m(t, \mathbf{x})|} \right) \\ &= \text{sgn}(g'(m(t, \mathbf{x}))) \kappa_m, \end{aligned} \quad (54)$$

where  $\kappa$  and  $\kappa_m$  denotes a curvature of the function  $\phi(t, \mathbf{x})$  and  $m(t, \mathbf{x})$ , respectively. This tells us that

$$g' \frac{\partial m}{\partial t} = E_c |\nabla m| |g'| + \alpha g' \kappa_m |\nabla m|. \quad (55)$$

Because, by assumption,  $g'(m(t, \mathbf{x})) < 0$ , it can be cancelled out, yielding (31). This concludes the proof.

## REFERENCES

- [1] A. D. Lanterman, “Tracking and recognition of airborne targets via commercial television and FM radio signals,” *Acquisition, Tracking and Pointing XIII*, vol. 3692, April 1999, Orlando, FL.
- [2] Y. Bresler and P. Feng, “Spectrum-blind minimum-rate sampling and reconstruction of 2-D multiband signals,” *Proc. 3rd IEEE Int. Conf. on Image Processing, ICIP'96*, vol. I, Sept. 1996, Lausanne, Switzerland, pp. 701–704.
- [3] R. Venkataramani and Y. Bresler, “Further results on spectrum blind sampling of 2D signals,” *Proc. IEEE Int. Conf. Image Proc., ICIP*, vol. 2, Oct. 1998, Chicago, pp. 752–756.
- [4] Y. Bresler, M. Gastpar, and R. Venkataramani, “Image compression on-the-fly by universal sampling in Fourier imaging systems,” *Proc. 1999 IEEE Information Theory Workshop on Detection, Estimation, Classification, and Imaging*, Feb. 1999, Santa-Fe, NM, pp. 48–48.
- [5] C. A. Bouman and K. Sauer, “A generalized Gaussian image model for edge-preserving map estimation,” *IEEE Trans. on Image Processing*, vol. 2, no. 3, pp. 296–310, July 1993.
- [6] P. Charbonnier, L. Blanc-Feraud, and M. Barlaud, “An adaptive reconstruction method involving discontinuities,” *Proc. of IEEE Int'l Conf. on Acoust., Speech and Sig. Proc.*, 1993.
- [7] A. H. Delaney and Y. Bresler, “Globally convergent edge-preserving regularized reconstruction: An application to limited-angle tomography,” *IEEE Trans. on Image Processing*, no. 2, pp. 204–221, February 1998.

- [8] F. Santosa, “A level-set approach for inverse problems involving obstacles,” *ESAIM: Control, Optimisation and Calculus of Variations*, pp. 17–33, January 1996.
- [9] Y. G. LecLerc, “Constructing simple stable descriptions for image partitioning,” *International Journal of Computer Vision*, pp. 73–102, 1989.
- [10] J. Sokolowski and J. Zolesio, *Introduction to Shape Optimization: Shape Sensitivity Analysis*. New York: Springer-Verlag, 1991.
- [11] J. A. Sethian, *Level Set Methods and Fast Marching Methods*. United Kingdom: Cambridge University Press, 1996.
- [12] D. Adalsteinsson and J. A. Sethian, “The fast construction of extension velocities in level set methods,” *J. Computational Physics*, no. 148, pp. 2–22, 1999.
- [13] O. Dorn, E. Millier, and C. Rappaport, “A shape reconstruction method for electromagnetic tomography using adjoint fields and level sets,” *Inverse Problems*, no. 5, pp. 1119–1156, October 2000.
- [14] M. Gastpar and Y. Bresler, “On the optimum density for nonuniform sampling,” *Proc. ISIT, International Symposium on Information Theory*, June 2000, Sorrento Italy.
- [15] M. Gastpar and Y. Bresler, “On the necessary density for spectrum-blind nonuniform sampling subject to quantization,” *Proc. of IEEE Int’l Conf. on Acoust., Speech and Sig. Proc.*, June 2000, Istanbul.

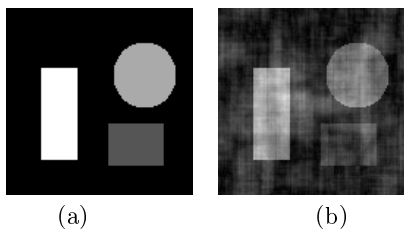


Figure 1: (a) Magnitude image of the original ground truth, and (b) Fourier inversion result with zero-padding.

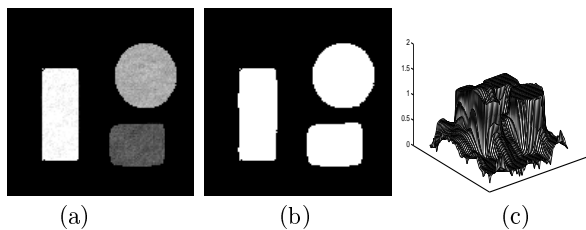


Figure 2: (a) Magnitude of the estimated image, (b) estimated domain, and (c) level function.

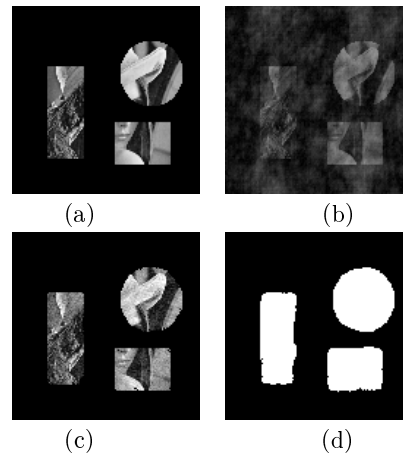


Figure 3: Example 2 (a) partial lena image, (b) direct Fourier inversion, (c) reconstruction by our algorithm, and (d) the estimated domain.

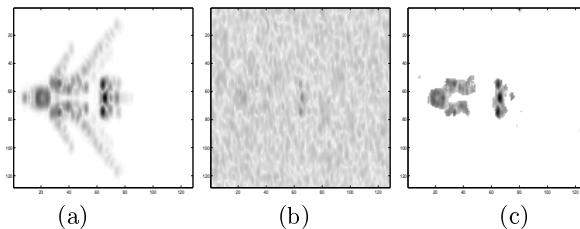


Figure 4: Example 3 (a) original VFY aircraft image, (b) direct Fourier reconstruction, and (c) reconstruction by our algorithm.

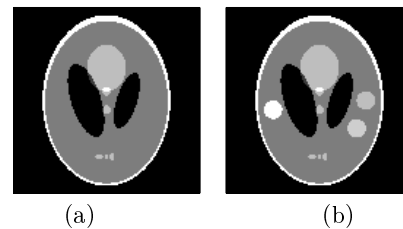


Figure 5: Example 4 (a) original Sheep-Logan phantom, (b) modified phantom with three additional tumors.

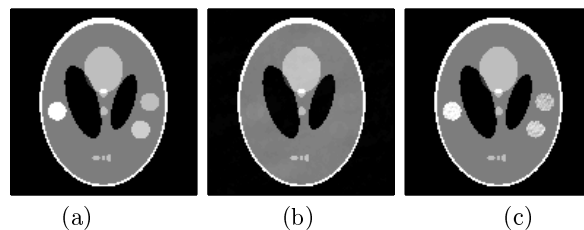


Figure 6: Example 4 (a) original phantom image with tumors, (b) direct Fourier reconstruction, and (c) reconstruction by our algorithm.