Surface-From-Gradients Without Discrete Integrability Enforcement: a Gaussian Kernel Approach

Heung-Sun Ng, Tai-Pang Wu, Member, IEEE Computer Society and Chi-Keung Tang, Senior Member, IEEE Computer Society

Abstract—Representative surface reconstruction algorithms taking a gradient field as input enforces the integrability constraint in a discrete manner. While enforcing integrability allows the subsequent integration to produce surface heights, existing algorithms have one or more of the following disadvantages: they can only handle dense per-pixel gradient fields, smooth out sharp features in a partially integrable field, or produce surface distortion in the results. In this paper, we present a method which does not enforce discrete integrability, and reconstructs a 3D continuous surface from a gradient or a height field, or a combination of both, which can be dense or sparse. The key of our approach is the use of kernel basis functions, which transfers the continuous surface reconstruction problem into high dimensional space where a closed-form solution exists. By using the Gaussian kernel, we can derive a straightforward implementation which is able produce results better than traditional techniques. In general, an important advantage of our kernel based method is that the method does not suffer discretization and finite approximation, both of which leads to surface distortion, which is typical of Fourier or wavelet bases widely adopted by previous representative approaches. We perform comparison with classical and recent methods, on benchmark as well as challenging data sets, to demonstrate that our method produces accurate surface reconstruction that preserves salient and sharp features. The source code and executable of the system is available for downloading.

Index Terms—Surface from gradients, integrability, kernel methods, basis functions.

I. INTRODUCTION

Shape-from-shading and photometric stereo methods usually output a dense and noisy gradient field from which surface heights are estimated by integration. To integrate a noisy gradient field for obtaining a continuous surface, many methods [8], [28], [24], [16], [2] uniformly enforce the integrability constraint over the entire gradient field. While such enforcement is effective in curbing noise for smooth surface integration, sharp features will be smoothed out and surface distortion will be produced. They are mainly caused by perturbing the input normals to satisfy the integrability constraint.

Recently, the work by Karacali and Snyder [13], [14] assumed the locations of discontinuity edges are known, and relaxed the uniform integrability enforcement by modifying the feasible subspace, and applied partial integrability on the discrete input gradient field. To handle noise, a noise reduction scheme was proposed in [14]. Other approaches handle a partially integrable field by altering input gradients or normals to make the entire field integrable. Typical examples include: Agrawal et al [1], which perturbed input gradients in order to make the curl zero everywhere in the field. A more general framework was presented in [2], in which multiple approaches (the Poisson method [28], Fourier basis functions [8], alpha-surface, M-estimator, regularization, and anisotropic diffusion) are generalized by the same objective function. The solution to the corresponding optimization simultaneously produces an integrable field while fractional/binary discontinuity labels are estimated.

A general observation among typical approaches that deal with partially integrable fields is that, as long as the output gradient field has a sufficiently small curl magnitude everywhere, that is, the global integrability constraint is satisfied, surface reconstruction is considered done. Note, however, that since a continuous surface is not necessarily differentiable everywhere, a surface computed by integrating such integrable field may not be the desired output. Consider for example a simple ramp surface shown in Fig. 1, which is a $C^0$ continuous surface with a $C^1$ orientation discontinuity. A continuous ramp surface cannot be obtained by integrating the normal map, which is shown in Fig. 1 as a needle map (produced by projecting the 3D unit normals orthogonally onto the 2D screen). Any methods that enforce integrability by perturbing normal orientations along the orientation discontinuity in the middle will render distortion in the resulting surface, as shown in the figure.

Unlike most previous approaches, we do not enforce the integrability constraint over the discrete domain. In this paper, we propose to reconstruct a continuous surface from the input gradients or normals, where fine details are preserved by treating the input gradients as normal constraints. Our reconstruction algorithm, which uses kernel basis functions, operates in the continuous, higher dimensional domain, and
outputs an accurate, continuous 3D representation not limited to a height field. The kernel-based formulation allows a linear and closed-form solution, which leads to a straightforward implementation where initialization and convergence are not issues in the program execution. Our algorithm can handle sparse input consisting of surface normals or gradients, 3D locations, or a combination of both. This sets our work apart from traditional approaches in continuous surface reconstruction with discontinuity consideration, where a complex formulation was usually adopted, initial heights were usually given to guarantee convergence, with their results less complex than ours. As we shall demonstrate, an important advantage of our kernel-based approach is that it does not suffer discretization and finite approximation, both of which will lead to surface distortion typical of Fourier or wavelet approaches widely used by previous methods. As a kernel-based formalism, we shall discuss in this paper the properties of one particular instance, the Gaussian kernel, and defer the detail study on the other kernels to previous literatures such as [3], [33].

The organization of this paper is as follows: Section II reviews related works. Section III discusses important issues in conventional surface-from-gradients algorithms to motivate our new approach and method. The mathematical framework is detailed in section IV, starting from our surface representation to the use of kernels in formulating the linear system embedded with linear constraints. We provide in section V results and comparison on surface reconstruction on a variety of input data. Finally, we conclude our paper in section VI. The source codes and executable are available from the contact author’s website.

II. RELATED WORK

In this section, we review related work on previous methods integrability constraint are applied to a gradient field. Works on surface reconstruction with discontinuity consideration, which is also related to our algorithm, are also reviewed.

A. Enforcing Integrability

Projection onto an Integrable Space. One classical approach that enforces integrability using such projection approach was proposed by Frankot and Chellappa [8]. In this method a dense non-integrable field of gradients is projected onto a set of orthogonal integrable basis (Fourier basis functions). Integrability is enforced in the Fourier space. Karacali and Snyder [13] defined a uniformly integrable surface in terms of an orthonormal set of gradient fields, which is a subset of the gradient space obtained by applying an orthonormalization procedure. The gradient field is partially projected onto this space to obtain a partially integrable field. This method was extended in [14] to handle noise explicitly by using multiscale orthonormal transformation. Shape from shapelets was proposed in [16]. This method accepts dense normals as input, and correlates the measured surface gradients with the gradients of a redundant set of wavelet basis functions (referred to as shapelets) which satisfy a set of constraints. The gradient field can be regarded as being projected onto the shapelet space. Goldman et al. [10] focused on photometric stereo with spatially-varying BRDFs. The method reconstructs surfaces by projecting the recovered gradient field onto the subspace of feasible gradient fields. A method for surface reconstruction from gradients was proposed in [12]: by minimizing a least square cost function, it is possible to exactly reconstruct a surface of polynomial of degree two or higher without suffering from low-frequency bias. However, the method is still limited in its use of Fourier transform in the discrete domain, which we will argue against in the paper.

Enforcing the Zero Curl Constraint. A method was proposed in [1] by enforcing integrability using the zero curl constraint. The input gradient field is decomposed into a sum of an integrable gradient field and a residual gradient field. The integrable field is then obtained by subtracting the residual field directly from the input field. This results in an under-determined system, requiring the reduction of the number of unknowns by disconnecting the gradient field (similarly done in [13], [14]) to preserve partial integrability on a partially connected gradient field. In doing so, the output field is guaranteed to be integrable, but direct perturbation of the field will distort the resulting surface.

Weighted Gradient Field. The method proposed in [2] focused on discontinuity and introduced anisotropic weights for better surface estimation. In the work binary or continuous weights were used to weigh the input gradient field. Affine transformation can be applied to the gradient field, where the input gradient field is attenuated or even disconnected according to the degree of integrability. However, important features may be lost at positions where the integrability constraint is not satisfied.

B. Surface Reconstruction

Classical works in surface reconstruction assumed surface gradients, surface heights and other constraints as input, and many of them were able to output a continuous surface and preserve underlying orientation discontinuities. Blake and Zisserman [4] developed a functional optimization framework using a weak membrane model that explicitly deals with discontinuities. Szeliski [29] treated the surface reconstruction problem as a two-dimensional interpolation problem using regularization theory [32]. Using a similar energy functional,Terzopoulos et al. [31] inflated surfaces using thin plate energy given sparse geometric cues. Later, the method in [30] provided a compact framework for visible surface reconstruction.

A non-linear and complex energy functional was derived, where initialization and convergence rate are issues. Moreover, initial heights are required to constrain the solution and guarantee good convergence rate. However, good initial heights are often unavailable in surface-from-gradients algorithms. In [7], a method for fitting multi-dimensional curves was proposed which uses non-linear energy minimization. However, this method primarily deals with unorganized point sets where gradients or surface normals are not considered. In [20], good 3D surfaces are produced by separating high- and low-frequency components, which preserves high-frequency details and avoid low-frequency bias using an efficient linear system. This method however does not handle the problem of surface-from-gradients where only gradients are available, but aims
at alternately improving 3D locations and orientations where initial estimations are available as input.

While most surface-from-gradients methods produce a discrete height field as output, some methods are able to produce a continuous surface. In [37], the proposed method takes as input a combination of user-specified constraints, such as surface positions, normals, silhouettes, and creases to generate a continuous surface that satisfies the supplied constraints. The associated energy functional is formulated based on thin-plate-spline. In [25], a closed-form solution was proposed for single-view reconstruction of curved 3D surfaces. However, the user needs to supply the inflation constraints, by guessing the surface heights at sample image locations. The input normals are specified in the parameter space, which is less intuitive than directly specifying normals in the spatial domain.

Our method is closely related to radial basis and kernel based methods. In [33], the theoretical aspects of spline models were studied. Based on the general framework, a theory was developed for the estimation of the underlying functions from noisy data using radial basis and kernel functions. However, the issues on surface-from-normals were not studied. In [9], Generalized Regularization Network was proposed as an approximation scheme for a broad class of smoothness functionals or regularizers. This work can be regarded as a study of the relationship between basis functions and regularization.

Readers may refer to [22], [6], [26] for implicit surface estimation. In these works, radial basis function, spline or Green’s function were used for data interpolation. The input to the methods consists of volumetric data or a point cloud which may not be available in our case. However, these methods can be used in our work for generating the final 3D mesh for visualization. On the other hand, the main difference of our work when compared with these methods is the mapping function: our method maps an image location to a 3D position $(\mathbb{R}^2 \rightarrow \mathbb{R}^3)$, while the implicit surface approaches map a 3D position to an iso-level $(\mathbb{R}^3 \rightarrow \mathbb{R})$.

A preliminary version of this paper appeared in [21] where our kernel-based, closed-form solution was first introduced. The present coverage completes the description by providing an in-depth study of the inherent problems of previous methods which motivate our new continuous kernel-based formulation. We present a detailed analysis of our algorithm, and perform extensive experimental comparison.

III. ISSUES IN CONVENTIONAL SURFACE FROM GRADIENTS ALGORITHMS

Despite the advances in surface-from-gradients algorithms, their surface results suffer distortion in different degrees on simple examples (Fig. 1) as well as complex ones (Fig. 16). In this section, we will analyze the main cause which we believe is largely due to the integrability enforcement:

\begin{itemize}
  \item[a)] in the \textit{discrete} image domain [28], [24], [13], [14], [1], [2], and/or
  \item[b)] the use of basis functions in the frequency domain in a \textit{finite} setting [8], [16].
\end{itemize}

**Discrete vs Continuous** Existing algorithms of surface-from-gradients use a discrete configuration to perform surface reconstruction. This leads to a number of disadvantages. First, integrability needs to be explicitly enforced by changing input gradients in order to compute a surface in the discrete setting. However, this is not required in the continuous case. Refer to Fig. 2, which demonstrates a case where a single height value is unable to satisfy all the gradient/normal constraints, hence introducing inconsistency which is referred to as violation in integrability. Methods such as direct Poisson solver will simply spread out the discrete non-integrable residual across the entire gradient field to ameliorate the inconsistency, whereas the field is actually integrable when considered in the continuous domain, as shown in Fig. 2.

Second, discrete algorithms usually use a discrete formula such as:

$$\frac{\partial z(x, y)}{\partial x} \approx z(x + 1, y) - z(x - 1, y)$$

(1)

to approximate partial derivatives. Note that when the normal direction is lying on the image plane, the corresponding gradient values $\frac{\partial f(x, y)}{\partial x}$ and/or $\frac{\partial f(x, y)}{\partial y}$ will be undefined, which will introduce severe numerical instability. Discrete formulation fails to capture normals lying on the image plane. Such “problematic” normals are required to be perturbed before they are input to algorithms using discrete gradient operators thus introducing surface distortion. These normals are common in practice, which occur along an object’s silhouette in the image which is the projection of the object’s occlusion boundary in 3D.

**Limitation of Fourier/Wavelet Basis Functions** Projection of a gradient field onto a set of basis functions is one common way to enforce integrability. Both orthogonal set (Fourier basis in [8]), and non-orthogonal redundant set (wavelet basis in [16]) have been proposed. The underlying idea of using Fourier function or wavelet is to decompose a function into a set of different frequency components. However, the frequency domain analysis is not appropriate when a finite set of basis functions is considered in a discrete manner, especially when the desired output surface is continuous and connected, which is typical of shape-from-shading and photometric stereo. This can be illustrated using a simple example of a rectangular function as shown in Fig. 3. Consider the Fourier transform of the rectangular function:

$$\text{rect}(x) \quad \leftrightarrow \quad \text{sinc}(f) = \frac{\sin \pi f}{\pi f}$$

(2)
We can see that representing a spatially sharp feature involves all frequency values \( f \) where \( \text{sinc}(f) \neq 0 \), which indicates an infinite number of frequency components in the continuous domain is necessary. When the number of components is assumed to be finite as in [8], where the number of components is set to be the same as the image size, ripple-like artifacts (known as ringing artifacts in digital image processing; more details in [11]) will be resulted in the output surface as shown in the figure. In other words, a smooth surface satisfying the gradient constraint simply does not exist in the solution space. The experimental results using [8] and [16] in Fig. 3 confirm the above arguments.

From the above analysis, we therefore conclude that a continuous formulation for surface-from-gradients is preferred: discrete enforcement of the integrability constraint and the use of a finite number of frequency components are no longer issues. Because of the shortcomings in existing continuous formulations (as reviewed in section II-B), in this paper, we propose a new solution that operates in the continuous domain and does not enforce integrability in a discrete manner. Our algorithm is based on kernel basis functions and has a closed-form solution.

IV. A KERNEL-BASED, CLOSED-FORM SOLUTION WITHOUT DISCRETE ENFORCEMENT OF INTEGRABILITY

In this section, we describe how we can generate a 3D continuous surface given a complete or incomplete 3D gradient/normal map, within a unified framework implemented as a closed-form solution based on kernel basis functions. Our algorithm accepts as input any combination of:

- Normals/Gradients They can be either sparse or dense, residing on a 2D regular (image) or irregular grid space. The relationship between the normal and gradient is given by:
  \[
  n = \frac{1}{\sqrt{p_x^2 + p_y^2 + 1}} \begin{pmatrix} -p_x \\ -p_y \\ 1 \end{pmatrix},
  \]
  where \( n \) is a surface normal and \( p_x \) and \( p_y \) are the gradients along \( x \) and \( y \) direction respectively.

- 3D points They can be either sparse or dense, scattering in a finite space where discontinuities can also be specified (e.g., occlusion boundaries or orientation discontinuities between surface regions).

Previous methods can either handle a subset of the situations listed above, or produce results suffering severe distortion, or use a somewhat complex problem formulation such as [31], [30].

In the following, we will first describe the surface representation using a general, continuous non-linear function. Then by using the “kernel trick” typical of machine learning, the function is transformed into a linear representation. Next, we will incorporate the available information as constraints into the resulting system, and finally derive a closed-form solution from the linear system. Different choices of basis functions and regularizers will also be analyzed, which is an issue for a kernel-based formulation.

A. Representation

Rather than producing a Monge patch (height field), we define a mapping function \( f : \mathbb{R}^2 \rightarrow \mathbb{R}^3 \) that maps a 2D point in the continuous image domain to the corresponding 3D location. The form of the function can be defined as:

\[
  f(t) = (f_x(t), f_y(t), f_z(t))^T
\]

where \( t = (u, v)^T \in \mathbb{R}^2 \) are the image coordinates, \( f_a : \mathbb{R}^2 \rightarrow \mathbb{R} \) and \( a \in \{x, y, z\} \). Therefore, the 3D surface is parameterized by \( t \).

B. The kernel method – non-linear to linear transformation

Since \( f_a \) can be any continuous function, which in general can be non-linear and complex, our representation should be sufficiently versatile and yet linear (in high dimensional space) in nature for numerical stability. We propose to model \( f_a \) by using kernel functions [27]. Using kernel functions, a non-linear problem can be cast into a high-dimensional feature space where linear techniques can be used to solve the problem. In our case, \( f_a \) is modeled as a high-dimensional hyperplane in the reproducing kernel Hilbert space (RKHS) \( \mathcal{H} \), a Hilbert space of functions which can be treated as a special feature space [3], [5]. Using a feature map \( \phi \) mapping \( t \) into the RKHS, \( f_a \) can be modeled as the following form:

\[
  f_a(t) = \langle w, \phi(t) \rangle
\]

where \( \langle \cdot, \cdot \rangle \) is the dot operator and \( w \in \mathcal{H} \) is a weighting vector in the feature space. It provides a linear map from the Hilbert space back to the Euclidean space. Our desired surface is thus connected to the high-dimensional space using this function.

Following the Representer’s Theorem [3], we first decompose the high-dimensional vector \( w \) into two components: one component in the span \( \phi(t_i) \) and the other as an orthogonal component:

\[
  w = \sum_{i=1}^{N} \alpha_i \phi(t_i) + w_\perp.
\]

Note that Eqn. 5 can be regarded as a hyperplane in a high dimensional feature space where \( w_\perp \) is the normal to the
C. Data Constraints

After describing the form of \( f \), we now describe how the input information is encoded to constrain the output \( f \). We will translate the available information (complete or incomplete) in a single view into data constraints for the estimation of \( \{ \alpha_{ai} \} \). Let

\[
\hat{\alpha}_a = (\alpha_{a1}, \alpha_{a2}, \cdots, \alpha_{aN})^T \tag{12}
\]

\[
\Lambda = \begin{pmatrix} \hat{\alpha}_x^T & \hat{\alpha}_y^T & \hat{\alpha}_z^T \end{pmatrix}^T \tag{13}
\]

We will arrange the data constraints into the following form:

\[
A\Lambda = b \tag{14}
\]

where \( A \) is the Gram matrix and \( b \) is the observation vector that will be derived in the upcoming sections (Section IV-C1 – IV-C2). All the known information is thus encapsulated in Eqn. 14, which consists of a set of linear equations. Note that we can incorporate additional constraints into Eqn. 14, as long as the constraints are linear.

1) Projection Term: Let us first assume that the input image is formed by orthographic projection. The \( u \) and \( v \) components of the image points should coincide with the \( x \) and \( y \) components of the corresponding 3D points:

\[
f_x(u,v) = u \tag{15}
\]

\[
f_y(u,v) = v \tag{16}
\]

Note that the final 3D surface estimated is not necessarily a Monge patch, because the projection term operates as a soft constraint in our solution. Although the above equations look trivial and redundant, recall that we are using a parametric surface representation, where the role of \( (u,v) \) will be changed from an image location to a parameter.

Suppose that we have some incomplete heights (which can be obtained along the object silhouette where we can assume \( z = 0 \), or derived from disparity values, or given by users), the \( z \) components of the corresponding image points can be set to the known values.

Then, we let \( \mathcal{P} = \{ s_j \in \mathbb{R}^2 \mid j = 1, 2, \ldots, m \} \) be the set of image locations where the heights are known. Note that \( \mathcal{P} \) can be empty when no heights are available. When \( \mathcal{P} \neq \emptyset \), the estimated surface should satisfy:

\[
f(s_j) = \begin{pmatrix} s_j \\ h_j \end{pmatrix} \tag{17}
\]

where \( h_j \) is the known height value at \( s_j \). A more general form of the projection constraint can be written as:

\[
f(s_j) = v_j \tag{18}
\]

where \( v_j = (\nu_{jx}, \nu_{jy}, \nu_{jz})^T \) is the 3D location for \( s_j \). Note that Eqn. (18) can be used to encode known 3D locations without resorting to the assumption of orthographic projection. Observe that each component in Eqn. (18) is in the same form:

\[
\hat{f}_a(s_j) = \nu_{ja} \tag{19}
\]

\[
\sum_{i}^{N} \alpha_{ai} k(t_i, s_j) = \nu_{ja} \tag{20}
\]

where \( a \in \{ x, y, z \} \). Writing in terms of the linear vector dot product to suit the form in Eqn. 14, we have:

\[
\hat{k}_{s_j}^T \alpha_a = \nu_{ja} \tag{21}
\]

where

\[
\hat{k}_{s_j} = (k(t_1, s_j), k(t_2, s_j), \cdots, k(t_N, s_j))^T \tag{22}
\]

Writing all the three channels explicitly, we obtain:

\[
\begin{pmatrix}
\hat{k}_{s_j}^T \\
\hat{k}_{s_j}^T \\
\hat{k}_{s_j}^T
\end{pmatrix} \Lambda = \begin{pmatrix} \nu_{jx} \\ \nu_{jy} \\ \nu_{jz} \end{pmatrix} \tag{23}
\]
which can readily be represented by Eqn. 14.

2) Normal Term: Let $\mathcal{V} = \{ c_j \in \mathbb{R}^2 | j = 1, \ldots, n \}$ be the set of image locations with known normal orientation. Note that $\mathcal{P} \cap \mathcal{V}$ can be empty or non-empty, because $\mathcal{P}$ may or may not equal to $\mathcal{V}$. Direct incorporation of the normal constraint using squared distances or angular errors will lead to complex and non-linear equations, for example, $\|1 - (\hat{n}^T \mathbf{n})^2\|^2$, where $\hat{n}$ is the known unit normal, and $\mathbf{n}$ is the normal to be estimated. Rather, we optimize the estimated surface by aligning it with the known normals given by $\mathcal{V}$, by making use of the fact that the tangents along surface are aligned with the known normals given by $\mathcal{V}$.

Such preferential treatment should be allowed in the solution. This is because a larger curvature will induce a larger estimated normal will be made as perpendicular as possible to the given normals.

Note that the norms of the surface normals are not restricted to one. This is because a larger curvature will induce a larger norm, which should impose higher weight in the computation. Such preferential treatment should be allowed in the solution.

Let $\mathbf{n}(t) \in \mathbb{R}^3$ be the surface normal at $t$ and $\mathbf{q}_p(t) = (\partial f_p(t)/\partial u, \partial f_p(t)/\partial v)^T$ be the tangent along direction $p \in \{u, v\}$ at $t$, where

$$\frac{\partial f_u(t)}{\partial u} = \sum_i^N \alpha_{u,i} k_u(t_i, t) \quad \quad (24)$$

$$\frac{\partial f_v(t)}{\partial v} = \sum_i^N \alpha_{v,i} k_v(t_i, t) \quad \quad (25)$$

Here, $k_u(t_i, t)$ and $k_v(t_i, t)$ are the derivatives of the kernel function along the directions $u$ and $v$ respectively. Note that a compact expression of the derivatives is important to make it easy and practical for the optimization in the implementation. Consider the set $\mathcal{V}$, the estimated surface should satisfy:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} f(c_j) = c_j \quad \quad (26)$$

which represents the orthographic projection along the $x$ and $y$ directions. When $c_j$ is available in $\mathcal{P}$, Eqn. 26 can be ignored to avoid duplication. In addition, we have the following constraint for tangent and normal:

$$\mathbf{q}_p(c_j)^T \mathbf{n}(c_j) = 0 \quad \quad (27)$$

$$n_x(c_j) \frac{\partial f_u(c_j)}{\partial p} + n_y(c_j) \frac{\partial f_v(c_j)}{\partial p} + n_z(c_j) \frac{\partial f_z(c_j)}{\partial p} = 0 \quad \quad (28)$$

where $p \in \{u, v\}$ and the above is the same for the two directions. Substituting Eqn. 24 and Eqn. 25 into Eqn. 27:

$$n_x(c_j) \sum_i^N \alpha_{x,i} k_p(t_i, c_j) + n_y(c_j) \sum_i^N \alpha_{y,i} k_p(t_i, c_j) + n_z(c_j) \sum_i^N \alpha_{z,i} k_p(t_i, c_j) = 0 \quad \quad (29)$$

which can be expressed in the linear vector dot product form:

$$n_x(c_j) \hat{k}_{pc}^T \alpha_x + n_y(c_j) \hat{k}_{pc}^T \alpha_y + n_z(c_j) \hat{k}_{pc}^T \alpha_z = 0$$

where

$$\hat{k}_{pc} = (k_p(t_1, c_j), k_p(t_2, c_j), \cdots, k_p(t_N, c_j))^T \quad \quad (30)$$

By stacking the sets of equations obtained from Eqn. 23, 26 and 30, we obtain Eqn. 14.

In the following two sections we will explain the smoothness term and regularization term respectively which will be incorporated in the final solution. Typical of kernel-based approach, a regularization term is required to stabilize the solution computed in the high-dimensional space [27]. Because geometric surface smoothness in the 3D space may not be adequately imposed by a regularizer, a geometric smoothness term is needed especially in our case of surface reconstruction. The Laplacian regularizer (section IV-E) is chosen which minimizes second-order derivatives. As we will demonstrate, a flat surface will be produced without using the smoothness term proposed in the following section.

D. Smoothness Term

Since the available information can be incomplete/sparse gradients or heights or a combination of both, we need to use a smoothness term to incorporate the necessary interpolation ability into the algorithm. The idea is to minimize the smoothness over the estimated surface in the gradient domain.

To propagate the shape information from one image location to another, we consider the first-order neighborhood and minimize their difference. To produce a smooth surface for non-input sites with no information, we minimize the surface gradients or tangents between kernel sites while preserving sharp features. Let $(s, t) \in \mathcal{N}$ be the set of first order neighbor locations, we minimize the following:

$$w(t_x, t_t) ||\mathbf{q}_p(t_x) - \mathbf{q}_p(t_t)||^2 \quad \quad (33)$$

where $w(t_x, t_t) \in [0, 1]$ is a soft label to indicate the degree of discontinuity, if such information is available. If normal or depth discontinuity exists between $t_x$ and $t_t$, $w(t_x, t_t) = 0$. Otherwise, $w(t_x, t_t) = 1$ (default value). It can be user-supplied or computed by using, for instance, [35].

Next, we expand Eqn. 33 to make it amenable to our linear system, observing that:

$$\mathbf{q}_p(t_x) - \mathbf{q}_p(t_t) = \left( \begin{array}{c} \frac{\partial f_x(t_x)}{\partial p} - \frac{\partial f_x(t_t)}{\partial p} \\ \frac{\partial f_y(t_x)}{\partial p} - \frac{\partial f_y(t_t)}{\partial p} \\ \frac{\partial f_z(t_x)}{\partial p} - \frac{\partial f_z(t_t)}{\partial p} \end{array} \right) \quad \quad (34)$$
By considering one channel, \( a \in \{x, y, z\} \):

\[
\left( \frac{\partial f_a(t_s)}{\partial p} - \frac{\partial f_a(t_t)}{\partial p} \right)
= \left( \sum_{i}^{N} \alpha_i k_p(t_i, t_s) - \sum_{i}^{N} \alpha_i k_p(t_i, t_t) \right)
= \sum_{i}^{N} \alpha_i (k_p(t_i, t_s) - k_p(t_i, t_t))
= k_{st}^T \alpha_a
\]

(35)

where

\[
k_{st}^T = \begin{pmatrix}
    k_p(t_1, t_s) - k_p(t_1, t_t) \\
    k_p(t_2, t_s) - k_p(t_2, t_t) \\
    \vdots \\
    k_p(t_N, t_s) - k_p(t_N, t_t)
\end{pmatrix}
\]

(36)

Putting Eqn. 36 back into Eqn. 34 and rearranging:

\[
q_p(t_s) - q_p(t_t) = \left( k_{st}^T \alpha_x, k_{st}^T \alpha_y, k_{st}^T \alpha_z \right) \Lambda
\]

(37)

(38)

By gathering all channels \( a \in \{x, y, z\} \), \( p \in \{u, v\} \), \( (s, t) \in \mathcal{N} \) and adding back the weighting term, we can rewrite the set of Eqn. 33 into matrix form:

\[
||S\Lambda||^2
\]

(39)

and this term will be incorporated in the final solution (see Eqn. 53 to be introduced shortly) and thus 3D location can be estimated at each site.

E. Regularizer

Regularizers are typically used in kernel-based methods to limit the solution space of the system and thus improve the algorithmic stability.

In this section, we will discuss two regularizers: minimizing the squared-norm \( ||w||^2 \) and minimizing the Laplacian of the function \( f_a \).

Minimizing \( ||w||^2 \) This is a common choice in machine learning, and was shown in [27] that a small value of \( ||w||^2 \) is effective in restricting the solution space in the high-dimensional feature space.

Recall that decomposing \( w \) leads to:

\[
w = \sum_{i}^{N} \alpha_i \phi(t_i) + w_{\perp}
\]

(40)

Note that the orthogonal component \( w_{\perp} \) is ignored in the following due to its independence of \( \alpha \). Using the Mercer’s Theorem (Eqn. 10), the squared-norm becomes:

\[
||w||^2 = \left( \sum_{i}^{N} \alpha_i \phi(t_i) \right)^T \left( \sum_{j}^{N} \alpha_j \phi(t_j) \right)
\]

(41)

(42)

(43)

(44)

where each \( K_{ij} = k(t_i, t_j) \). Combining the equations given by the three coordinates \( (x, y, z) \) of our surface representation, the regularization term becomes:

\[
\left( \alpha_x^T, \alpha_y^T, \alpha_z^T \right) \left( \begin{array}{c}
\mathbf{K} \\
\alpha_x \\
\alpha_y \\
\alpha_z
\end{array} \right) = \Lambda^T \mathbf{K} \Lambda
\]

(45)

(46)

Minimizing the Laplacian While minimizing the squared-norm above is a typical choice of producing a stable solution in the high dimensional feature space, it however lacks of a geometric meaning. We propose to minimize the Laplacian of the function \( f_a \).

Using our representation, the Laplacian of function \( f_a \) is \( \Delta \) is the Laplacian operator:

\[
\Delta f_a(t) = \frac{\partial^2 f_a(t)}{\partial u^2} + \frac{\partial^2 f_a(t)}{\partial v^2}
\]

(47)

Here, we observe that taking the Laplacian of \( f_a \) is the same as taking the Laplacian of the kernel function \( k \). For instance, we choose the Gaussian kernel (reasons to be discussed), and hence the above equation actually applies the Laplacian of Gaussian (LoG) filter on the \( \alpha \) map.

Considering all kernel sites and all the three components \( (x, y, z) \), the Laplacian regularization term \( \Omega(f) \) is:

\[
\Omega(f) = \begin{pmatrix}
\mathbf{K}_L \\
\mathbf{K}_L \\
\mathbf{K}_L
\end{pmatrix}
\begin{pmatrix}
\alpha_x \\
\alpha_y \\
\alpha_z
\end{pmatrix}
\]

(48)

(49)

where

\[
\mathbf{K}_{Lij} = \frac{\partial^2 k(t_i, t_j)}{\partial u^2} + \frac{\partial^2 k(t_i, t_j)}{\partial v^2}
\]

(50)

(51)
Comparison Here, we compare the two regularizers described above using a similar synthetic example in [2] and rope, as shown in Fig. 4. While both regularizers are able to preserve fine details of the object, in the result produced using the $\|w\|^2$ regularizer, unsightly rippling artifacts are observed on the surface due to the lack of geometric consideration, where regularization is performed in the high-dimensional feature space. In contrast, the results obtained using our Laplacian regularizer demonstrates a better surface without significant rippling artifact. We also demonstrate the significance of the smoothness term, where the rippling artifact can be largely removed when both the smoothness term (which minimizes the first derivative) and the Laplacian regularization term (which minimizes the second derivative) are considered.

F. Final Solution

Combining the data constraint, smoothness constraint and the Laplacian regularizer (i.e., Eqns 14, 39 and 49), the overall energy function is:

$$E(f) = \|A\Lambda - b\|^2 + \lambda\|S\Lambda\|^2 + \mu\|\Omega(f)\|^2$$  \hspace{1cm} (52)

where $\lambda$ and $\mu$ are the weights to control the influence of the smoothness and the regularization term. Typical values of $\lambda$ and $\mu$ are 0.1 and 0.01. We can see that the surface estimation problem is now translated into a regression problem. While previous methods such as [2] minimize a similar form of energy function, there are significant differences. The cost function in [2], defined by the $L_2$ norm between the surface gradient and the given gradient, is the upper bound of the zero curl constraint [35]. Also, recall from Figure 2 that a discrete height-field is computed using these methods. Suppose we compute the output normal map from the resultant height-field and subtract from it the input normal map, errors are inevitable because the output is optimized subject to the zero curl constraint, while the input is partially integrable and therefore does not have a zero curl (see also Table II

![Fig. 4. Comparison of regularizers. The frontal view and the zoom-in view of the results using the $\|w\|^2$ regularizer and the Laplacian regularizer are respectively shown. Note the rippling artifacts in the results obtained using the squared-norm regularizer, and also the flatter surface shape in the results when the smoothness term is absent.](image1)

![Fig. 5. Visualization of the matrix $A^T A + \lambda S^T S + \mu K_L^T K_L$, where $\lambda = 0.1, \mu = 0.01$. The value in each entry is normalized to $[0, 1]$, and is displayed as pixel intensity.](image2)
where quantitative comparison is available). On the other hand, our method fits a continuous surface that satisfies the input normals and 3D positions, if available. As shown in Figure 2, discrete approximations and input normal perturbation are not needed.

To minimize our energy function, we set the first derivative of the energy function w.r.t. \( \lambda \) to zero, and obtain a set of linear equations:

\[
(A^T A + \lambda S^T S + \mu K_L^T K_L)\lambda = A^T b
\]  

(53)

To solve Eqn. (53), the Gauss-Seidel method can be applied with successive over-relaxation to obtain \( \lambda \), so we can find the 3D location at each image point.

In order to generate a 3D surface mesh for visualization, we use the solved \( f \) (Eqn. 3) to find the 3D point for every image location, and then generate the surface by applying [23], for instance.

At this point, one may notice that the sizes of matrices \( A, S \) and \( K_L \) (dimension is \( 3N \times 3N \)) can be very large for a dense input, for example, a normal map produced by photometric stereo. However, the matrices are extremely sparse when an appropriate kernel function is chosen. So, to make the program execution practical, we have to choose a kernel function \( k \) such that while it possesses the necessary properties required by a kernel function (that is, Eqns 10 and 11), it can produce sparse matrices \( A, S \) and \( K_L \). The choice of kernels will be described shortly.

G. Choice of Kernels

While we have yet to define the kernel to be used, it should be noted that the advantages brought by our kernel-based formalism (that is, natural incorporation of geometric constraints and continuous surface representation) are not affected by the choice of kernel as long as it satisfies the reproducing property described by Eqn. 10.

To represent the surface, a mixture of radial basis functions (Eqn. 11) is used to model \( f_a \), and hence an appropriate basis function has to be chosen as the kernel. In the following, we will discuss our choice of kernel: the Gaussian function. We will also compare the Gaussian kernel with Fourier bases which operate in the frequency domain. Fourier bases are typically used in representative surface-from-gradients algorithms based on basis functions, such as the classical method [8] and the more recent solution [16].

Note that the choice of kernel will only change the geometric properties of the surface (e.g., the density of the final matrix in Eqn. 53). For example, \( k(t, t') = \|t - t'\| \) will produce a 3D thin-plate spline surface [33]. More details can be found in [3].

1) Gaussian Function: The Gaussian function has the following form:

\[
k(t, t') = \exp\left(-\frac{\|t - t'\|^2}{2\sigma^2}\right)
\]  

(54)

Note that the Gaussian kernel matrix \( K \) where each \( K_{ij} = k(t_i, t_j) \), in general is semi positive definite. However, it is strictly positive definite (hence with full rank) in our case where all kernel sites are unique, as proved in [18].

Differentiability is important for our closed-form solution which incorporates the Laplacian regularizer. The kernel function should to be at least twice differentiable to make the Laplacian regularizer effective. The Gaussian function whose first and second derivatives for \( p \in \{u, v\} \) are respectively:

\[
\frac{\partial}{\partial p} k(t, t') = \frac{-(t_p - t'_p)}{\sigma^2} \exp\left(-\frac{\|t - t'\|^2}{2\sigma^2}\right)
\]  

(55)

\[
\frac{\partial^2}{\partial p^2} k(t, t') = \frac{(t_p - t'_p)^2 - \sigma^2}{\sigma^4} \exp\left(-\frac{\|t - t'\|^2}{2\sigma^2}\right)
\]  

(56)

The Gaussian function is simple and conducive to a sparse system (Fig. 5), which is a desirable property to reduce the computational and memory requirements for making our system practical. Based on our experiments, a Gaussian with a small width (hence a small \( \sigma \)) covering the first-order neighborhood in the image grid is sufficient, where the long tail of the Gaussian will produce a lot of negligible values close to zero which are numerically safe to remove. Table I summarizes the number of kernel sites used, the memory usage, the density of the final matrix (defined as the number of non-zero entries in the matrix) and the computation time. The implementation was coded using C++, and the experiments were performed on a computer with 8GB RAM and a 1.8GHz AMD CPU.

Another advantage of the Gaussian function is that it has a well understood geometric meaning, which allows us to further take depth discontinuity information, if available, into account, by simply multiplying a weighting function the Gaussian kernel to obtain our final kernel function:

\[
k(t, t') = d(t, t') \exp\left(-\frac{\|t - t'\|^2}{\sigma}\right)
\]  

(59)

where \( d(t, t') \in \{0, 1\} \). If depth discontinuity exists between the straight line joining \( t \) and \( t' \), \( d(t, t') = 0 \). Otherwise, \( d(t, t') = 1 \). So, \( d \) can be optionally obtained by image snapping tools or from an occlusion map in stereo analysis, which can be used to further indicate and control the influence between kernel sites.

2) Fourier Basis: We perform comparison with a typical choice of basis function, Fourier basis function, which is expressed as

\[
\varphi(u, v, \omega_x, \omega_y) = \exp(j\omega_x u + j\omega_y v)
\]  

(60)

Putting this into the surface representation \( f_a \) (Eqn. 11) and replacing the index \( i \) by the set of frequency values \( \Omega = (\omega_x, \omega_y) \), we obtain:

\[
f_a(t) = \sum_{\Omega} C_{\Omega} \exp(j\omega_x u + j\omega_y v)
\]  

(61)

Interestingly, Eqn. 61 shares the same form of the surface representation found in [8], the classical surface-from-gradients approach using Fourier Basis, where \( \{C_i\} \) is the set of coefficients of the Fourier series expansion of \( f_a \). This equation can be viewed as a summation of periodic functions
(sine/cosine functions) with different frequencies, and hence the function \( f_\omega \) can be regarded as being transformed into the frequency domain. However, this transformation may produce undesirable effect, as discussed in section III, when a finite set of frequencies \( \omega \) is used in the discrete approximation.

A finite approximation in the frequency domain is unwise due to the global nature of periodic functions. For instance, consider the removal of a simple sine wave component (or a single frequency component) which propagates from \(-\infty\) to \(+\infty\). Removing this single wave will have a widespread effect on the whole spatial domain, which explains the surface distortion suffered in [8] obtained using Fourier Basis (or [16] obtained using wavelet, which shares the same underlying spirit) where discrete and finite approximation was used. On the other hand, a finite set of kernel sites used in a continuous formulation does not have this problem.

In contrast with digital image processing, where the discrete version of Fourier analysis works fine due to the discrete nature of digital images, our goal is to reconstruct a continuous surface. Also, while the Gaussian function has a well-understood geometric meaning allowing spatial kernels defined by Gaussians to be disconnected easily, it is difficult to impose discontinuity constraint in the frequency domain. Together they justify the design of our continuous kernel-based algorithm and the choice of Gaussian kernel, which is also conducive to a straightforward and practical closed-form solution for implementation given the closed-form solution derived above.

### V. Results

**Synthetic case.** We first use a synthetic case to demonstrate the robustness of our approach, by adding to the input different amount of Gaussian noise. Fig. 6 shows a Torus, a genus-one object, where the orientations of the normals are perturbed. The results show that our kernel-based, closed-form solution is quite robust and degrades gracefully with increasing amount of noise. Another synthetic example shown in Fig. 7 demonstrates the ability of our approach in incorporating and preserving discontinuities. The sparse normal and discontinuity maps for a single view are typically available in single-view modeling [37], [25], [36]. Rather than guessing and manually assigning heights as done in [25], we specify normals and discontinuities easily via a 2D sketching interface [36]. Because the input is very sparse, the integration approach cannot be applied here. Our closed-form solution, on the other hand, produces a surface faithful to the input gradients where the specified discontinuity is preserved well.

**Surface from dense surface gradients.** We compare our kernel-based, closed-form solution with other surface-from-gradients method, most of which can handle only dense gradients. The Poisson method [28] reconstructs the surfaces by solving a Poisson equation. Regularization [2] uses the term \( r(s) = \sqrt{1 + s^2} \) to regularize the Poisson solver. M-Estimator [2] uses iterative reweighed least square method to weaken the effect of outlier. Affine transformation [2] is also compared, which estimates surface and generalizes the Poisson solver with edge-preserving diffusion tensor. Figs 8 – 10 show several input dense normal maps obtained by photometric stereo [34], and the output surface reconstruction results obtained using various methods. In particular, for Shapelet [16] and affine transformation [2], we follow the suggested parameters available in the authors’ Matlab implementation.

**Rope** is a complex object with a lot of mesostructures. While all the surfaces were generated using images of the same resolution, our method suffers less distortion due to quantization error. For **Hair**, the normal map is very noisy because some hairs of the wig were always under shadows regardless the illumination direction when the images were

<table>
<thead>
<tr>
<th>Data set</th>
<th>size</th>
<th>#kernel sites</th>
<th>memory used</th>
<th>density</th>
<th>running time</th>
</tr>
</thead>
<tbody>
<tr>
<td>rope</td>
<td>176x170</td>
<td>29920</td>
<td>56MB</td>
<td>0.921%</td>
<td>4852.92sec</td>
</tr>
<tr>
<td>hair</td>
<td>240x314</td>
<td>75360</td>
<td>1517MB</td>
<td>0.389%</td>
<td>11856.20sec</td>
</tr>
<tr>
<td>face</td>
<td>150x151</td>
<td>23556</td>
<td>358MB</td>
<td>0.914%</td>
<td>3712.01sec</td>
</tr>
<tr>
<td>male torso</td>
<td>169x281</td>
<td>47489</td>
<td>801MB</td>
<td>0.512%</td>
<td>5337.90sec</td>
</tr>
<tr>
<td>female torso</td>
<td>148x282</td>
<td>41736</td>
<td>688MB</td>
<td>0.565%</td>
<td>5045.55sec</td>
</tr>
<tr>
<td>vase</td>
<td>128x128</td>
<td>16384</td>
<td>223MB</td>
<td>1.151%</td>
<td>2868.21sec</td>
</tr>
<tr>
<td>Mozart</td>
<td>256x256</td>
<td>65536</td>
<td>1445MB</td>
<td>0.486%</td>
<td>8821.53sec</td>
</tr>
</tbody>
</table>

**Fig. 6.** Noise robustness. Top: color-coded dense normal maps. Bottom: the reconstructed surfaces. From left to right: the normals are perturbed with additive Gaussian noise with standard deviations of 0, 0.3, 0.6, 0.9 radians respectively.

**Fig. 7.** From left to right: the sparse needle map, the discontinuity map, our kernel-based, closed-form solution with/without discontinuity consideration.
Fig. 8. Results on dense gradients: Rope. From left to right in the top: the image, the color-coded normal map obtained using photometric stereo, and two views of the respective reconstructed surfaces. The rest of the figure shows the comparison of the corresponding views generated by other algorithms.

captured for photometric stereo computation. Note that our method produces less distortion on the upper right part of the wig where a lot of outliers exist. Face is a complex structure with typical facial features. The subject had a few pimples on the face, and wore eye glasses which introduce surface orientation discontinuities to the resulting dense normal map. Note the faithfulness of the surface results generated by our kernel-based, closed-form solution, while other methods produce a relatively flatter surfaces.

If the given dense normal map contains a lot of noise and sharp discontinuity, our kernel-based method may fail. Note that in our method, a 3D surface is modeled by kernel functions which are chosen to be Gaussian. The Hair example can be regarded as a failure case. The input normal map contains a lot of noise and discontinuity. None of the tested methods produce satisfactory results.

Surface from very sparse surface gradients. Figs 11 and 12 show respectively the front and back views of the sparse normals assigned to the Torso images. For each view, the normals along each pair of closely-spaced curves are made pointing away from each other. They serve to model crease curves which incorporate surface details on the torsos. Note that without any user-input height, our surface results are good

Fig. 9. Results on dense gradients: Hair. From left to right in the top: the image, the color-coded normal map obtained using photometric stereo, and one view of the respective reconstructed surfaces. The rest of the figure shows the comparison of corresponding views generated by other algorithms.
considering only sparse normals are available. We tested the Male Torso using previous methods and show the results in Fig. 13. The resultant surfaces are unsatisfactory since the input normals are sparse. For [16], we tested the input with different input parameters. Better results with less distortion are produced, but the visual quality of the surfaces is still quite poor.

Surface from sparse surface gradients and sparse heights. We propose an application on disparity editing, by assigning sparse normals to the disparity map based on a monocular view. State-of-the-art stereo algorithms already produce very good disparity map for modeling the global structure of a real scene. However, local structures are usually lost during stereo reconstruction due to limited image resolution. For the Tsukuba dataset (Fig. 14), for instance, while the global structure of the scene is excellent, the details on the bust and the curved shape of the lamp-shade are both lost.

We propose to improve the disparity map by taking as input the disparity along the object silhouettes (sparse heights without normals), and assign in the interior of each chosen disparity region sparse normals (sparse normals without heights), based on a given stereo image. The principle behind the specification of local structures using normals is justified in the seminar work Pictorial Relief by Koenderink [15], which concludes that humans are very good at assigning local surface normals for specifying local shape. To specify the normals, we make use of intelligent scissor [19] to follow the salient curves on the image, followed by interactive normal transfer using a sketching interface [36]. Given the incomplete heights and incomplete normals, our kernel-based, closed-form solution generates a highly-detailed disparity surface as shown in Fig. 14, where the bust suffers significantly less distortion after rotation.

Benchmark data. Finally, we present the results of Vase in Fig. 15 and the results of Mozart in Fig. 16. Both data sets are benchmark data used by existing methods. We calculate the surface normals from the data points by finite differences. Only surface normals are used in the subsequent surface estimation. For the Vase example, the surface produced by our method has a maximum height closer to the maximum height of the ground truth (note that it is required to tune the scale parameter in the shapelet method [16]). For the Mozart example, the surface reconstructed by our method and M-Estimator are nearly identical to the ground truth. Note however that M-Estimator [2] requires a dense normal map as input.

Table II compares the mean squared error (MSE) for various surface-from-gradients algorithms. The error is computed as the mean squared error between the ground-truth gradient map and the reconstructed surfaces in the gradient domain.

VI. CONCLUSION

This paper presents a new surface-from-gradients algorithm: our kernel-based algorithm computes a continuous surface that preserves fine details from a noisy gradient field without enforcing integrability in the discrete image domain. The choice of Gaussian kernel is justified by its simplicity in
Fig. 10. Results on dense gradients: Face. From left to right on the top: the image, the color-coded normal map obtained using photometric stereo, and the frontal views of the respective reconstructed surfaces. The rest of the figure shows the comparison of corresponding views generated by other algorithms.

Fig. 13. Surfaces from sparse gradients generated by various approaches. The missing normals are assumed to be and thus filled with \([0 \ 0 \ 1]^T\) which was done in [16]. In particular, the method in [16] reconstructed the surface using correlating multiple shapelet scales, while an over-smoothed surface is reconstructed using a large minimum \(\sigma\). Increasing the number of scales used will produce an over-flattened surface.

TABLE II

<table>
<thead>
<tr>
<th></th>
<th>Vase</th>
<th>Mozart</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poisson</td>
<td>0.3445</td>
<td>1.1170</td>
</tr>
<tr>
<td>Fourier</td>
<td>0.4698</td>
<td>1.6696</td>
</tr>
<tr>
<td>Wavelet</td>
<td>0.3335</td>
<td>2.1018</td>
</tr>
<tr>
<td>Regularization</td>
<td>0.3815</td>
<td>1.7794</td>
</tr>
<tr>
<td>M-estimator</td>
<td>0.3445</td>
<td>1.0925</td>
</tr>
<tr>
<td>Affine transformation</td>
<td>0.3565</td>
<td>1.3173</td>
</tr>
<tr>
<td>Curl=0</td>
<td>0.3435</td>
<td>1.7990</td>
</tr>
<tr>
<td>Kernel-based</td>
<td>0.3202</td>
<td>1.0683</td>
</tr>
</tbody>
</table>

Fig. 14. Surface reconstruction from sparse heights and sparse normals. Top: from left to right, the input image, incomplete height map with depth values represented in gray scale, incomplete normal map where blue curves are used to indicate the loci where normals are available. Middle and Bottom: the left shows the original disparity map. The right shows the disparity surface generated using our closed-form solution.

Fig. 15. Results on Vase.
implementation, its efficiency as it leads to a sparse system, and the quality of results which does not suffer from distortion due to discrete and finite approximation. Our kernel-based formulation has a closed-form solution, which provides a general framework to handle input consisting of normals/gradadients without corresponding 3D location, or 3D locations without corresponding normals/gradients, while the input can be dense or sparse. We have performed extensive experiments to demonstrate the effectiveness of our new algorithm on different combinations of inputs, and compared our method with others on synthetic, complex, and benchmark data to show quantitatively and qualitatively that our results are better in terms of result quality. The source codes and executable of our system can be downloaded at www.cs.ust.hk/~pang.

ACKNOWLEDGMENT

We gratefully thank Kovesi [16] and Agrawal et al [2] for their Matlab implementations of the tested methods available in their respective project websites. This research is supported by the Hong Kong Research Grant Council (grant number 620207).

REFERENCES


Heung-Sun Ng received the BEng and MPhil degrees in computer science, the Hong Kong University of Science and Technology (HKUST). His research interests include kernel-based methods in computer vision.

Tai-Pang Wu received the PhD degree in Computer Science from the Hong Kong University of Science and Technology (HKUST) in 2007. He was awarded the Microsoft Fellowship in 2006. After graduation, he spent a year (2007 - 2008) at Microsoft Research Asia (MSRA) to continue his research work. He is currently a Postdoctoral Researcher at the CSE Department of the Chinese University of Hong Kong (CUHK). His research interests include Computer Vision and Computer Graphics.

Chi-Keung Tang received the MSc and PhD degrees in computer science from the University of Southern California (USC), Los Angeles, in 1999 and 2000, respectively. Since 2000, he has been with the Department of Computer Science at the Hong Kong University of Science and Technology (HKUST) where he is currently a full professor. He is an adjunct researcher at the Visual Computing Group of Microsoft Research Asia. His research areas are computer vision, computer graphics, and human-computer interaction. He has served on the program committees of ICCV, CVPR, ECCV, and ICPR. He served as an area chair for ACCV 2006 (Hyderabad), ICCV 2007 (Rio de Janeiro), ICCV 2009 (Kyoto), and as a technical papers committee member for the inaugural SIGGRAPH Asia 2008 (Singapore). He will serve as an area chair for ICCV 2009 (Kyoto). He is a senior member of the IEEE Computer Society.