Non-oriented MLS Gradient Fields

Jiazhou Chen, Gael Guennebaud, Pascal Barla, Xavier Granier

HAL Id: hal-00857265
https://hal.inria.fr/hal-00857265

Non-oriented MLS Gradient Fields

Jiazhou Chen[1,2]   Gaël Guennebaud [2]   Pascal Barla[2]   Xavier Granier[2]

[1] College of Computer Science - Zhejiang University of Technology
[2] Inria - LaBRI (CNRS, Univ. Bordeaux) - LP2N (CNRS, Univ. Bordeaux, IOGS)

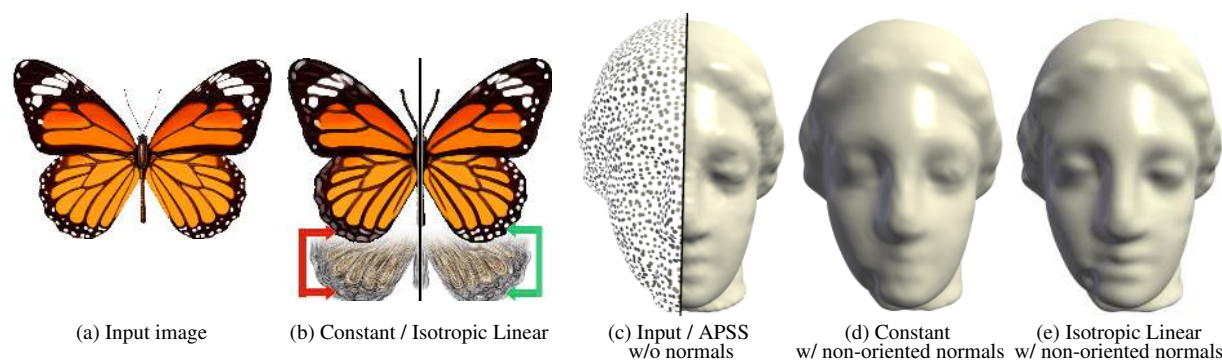|				
(a) Input image	(b) Constant / Isotropic Linear	(c) Input / APSS w/o normals	(d) Constant w/ non-oriented normals	(e) Isotropic Linear w/ non-oriented normals

Figure 1: *For images (a), an MLS constant approximation (similar to classical structure tensor smoothing [KKD09]) tends to oversmooth details (b-left). For 3D point sets (c-left), not using a normal field leads to poor quality reconstructions when using spherical MLS approximations [GG07] (c-right). The use of non-oriented normals leads to easy-to-implement improvements (d-e) without the burden of having to coherently orient them. For both kind of applications, constant approximations (b-left,d) are outperformed by our novel isotropic linear approximations (b-right,e) that better preserve the image and surface features, while exhibiting much smoother and stable surface reconstructions.*

Abstract

We introduce a new approach for defining continuous non-oriented gradient fields from discrete inputs, a fundamental stage for a variety of computer graphics applications such as surface or curve reconstruction, and image stylization. Our approach builds on a moving least square formalism that computes higher-order local approximations of non-oriented *input gradients. In particular, we show that our novel isotropic linear approximation outperforms its lower-order alternative: surface or image structures are much better preserved, and instabilities are significantly reduced. Thanks to its ease of implementation (on both CPU and GPU) and small performance overhead, we believe our approach will find a widespread use in graphics applications, as demonstrated by the variety of our results.*

Categories and Subject Descriptors (according to ACM CCS):   I.3.3 [Computer Graphics]: Picture/Image Generation—Display algorithms I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems

1. Motivation and previous work

The use of gradient information for both surface reconstruction and image processing has been proven to increase accuracy. For surface reconstruction, normals have been widely used to regularize ill-posed optimization problems [HDD*92, SOS04, KBH06], or to improve the robustness of local implicit fitting techniques [AK04, AA04b, GG07]. When stylizing images and videos, the use of a gradient field permits to orient filters along image features for a variety of applications going from line drawing [KLC07], to color abstraction [KLC09] or painterly rendering [Her98, HE04, PP09]. In either case, this results in reconstructed surfaces or stylized images that better preserve the original features.
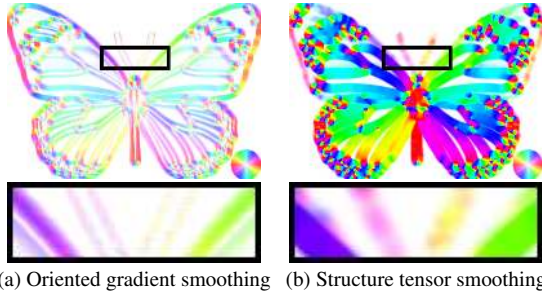
(a) Oriented gradient smoothing (b) Structure tensor smoothing

Figure 2: *Illustration of pair cancellation issues [BVL*06]: averaging opposite gradients on both sides of thin lines produces artificial zero gradients on the features (a), while filtering based on the structure tensor yields coherent directions (b). The input gradients are computed by a 5×5 Gaussian derivative. Note that we employ a symmetric HSV color code (see colored disc) to visualize gradients: hue represents non-oriented direction and saturation represents magnitude.*

However, directly dealing with gradient vectors requires them to be consistently oriented. For surface reconstruction, even though several options exist to robustly estimate *non-oriented* normals from raw point clouds [MGG04, AC-STD07], several authors have recognized that consistently orienting them is considerably harder, and as difficult as reconstructing the surface itself [HLZ*09, MDGD*10]. The same issue arises in image processing: even though differential image operators yield oriented gradients, their orientations are most of the time too incoherent for smoothing purpose. For instance, a thin line feature is expected to yield a unique tangent direction while its sides define opposite gradients. As illustrated in Fig 2a, directly smoothing oriented gradients introduces pair cancellation issues [BVL*06]. As with surface reconstruction, attempting to re-orient such gradients (either locally [KLC09] or globally [XCOJ*09]) is bound to fail if the reference direction is not reliable and/or when considering large neighborhoods.

In order to overcome these difficulties, the most common approach consists in encoding the *non-oriented* vector information into a *structure tensor* [BG87]. This representation has been employed to interpolate very sparse non-oriented vector data for interactive tensor field design [ZHT07] or image abstraction [KK11] purposes. Filtering structure tensors regularizes noise and/or merges gradients coming from multiple image channels [KD08]. In the context of surface reconstruction, it has been used by Alliez et al. [ACSTD07] through an expensive global eigenvalue problem, and suggested by Amenta et al. [AK04] for local plane fitting.

Unfortunately, a structure tensor can only encode a locally *constant* gradient field. This constitutes a major limitation when used to average information over curved neighborhoods. In this case, it quickly fails to preserve the original structure and tends to smooth out important surface or im-

age details (Fig 1b-left & d). This problem is amplified with increasing neighborhood sizes and/or sparsity. Medium-to-large filter sizes are not only required to regularize noise found in 3D point sets or images, but also for simplification or abstraction purposes. We also emphasize that in image processing, sparsity arises when resampling low resolution information, and/or in places where the input gradient has a close-to-zero magnitude. This limitation of low-order approximations has already been pointed out in the *Algebraic Point Set Surfaces* (APSS) framework [GG07], where moving from planar to spherical local approximations leads to great improvements both in terms of quality and robustness to sparse data and large scales. However, this Moving Least Squares (MLS) method requires consistently oriented normals and performs poorly without normals (Fig 1c-right).

In this paper, we introduce a new local approximation method of discrete *non-oriented* gradient fields that better preserves surface or image features (Fig 1b-right & 1e). The key ingredient of our approach is an extension of the structure tensor to a *higher-order* basis. In particular, we show that an isotropic linear basis provides the best trade-off between accuracy and smoothness. Using a MLS formulation, we define a *continuous* non-oriented gradient field at arbitrary scales. Surface and curve approximation from a raw point cloud is achieved by a MLS integration of the local gradient fields, leading to a continuous implicit reconstruction. Our results show that our approach better preserves image and manifold structures (in particular curved patterns), and we demonstrate real-time performances on the GPU.

2. Preliminaries on MLS

Our approach builds on the moving least squares (MLS) formalism, which we quickly explain here. Given a set of positions $\mathbf{x}_i \in \mathbb{R}^d$ with associated values $\mathbf{f}_i \in \mathbb{R}^m$, the goal is to reconstruct a continuous function $\mathbf{F} : \mathbb{R}^d \to \mathbb{R}^m$ that approximates the specified input. The unknown function $\mathbf{F}$ is locally approximated around the evaluation point $\mathbf{x}$ through a weighted least-squares minimization:

$$\mathbf{f}_{\mathbf{x}} = \arg\min_{\mathbf{p}} \sum_i w_i(\mathbf{x}) \|\mathbf{p}(\mathbf{x}) - \mathbf{f}_i\|^2 , \qquad (1)$$

where $\mathbf{p}$ is a low-degree multi-variate and multi-valued polynomial. The value of $\mathbf{F}$ is then set to the value of $\mathbf{f}_{\mathbf{x}}$ at the evaluation point, that is: $\mathbf{F}(\mathbf{x}) = \mathbf{f}_{\mathbf{x}}(\mathbf{x})$.

In Eq 1, w_i is a smooth weight function, decreasing with respect to the distance $\|\mathbf{x}_i - \mathbf{x}\|$, which plays the role of a low-pass filter. A convenient choice is to use $w_i = \phi(\|\mathbf{x}_i - \mathbf{x}\|/s_i)$, where s_i is the radius of influence (or size) of the i-th primitive, and ϕ is a compactly supported function:

$$\phi(t) = \begin{cases} (1-t^2)^2 & \text{if } t < 1 \\ 0 & \text{otherwise.} \end{cases} \qquad (2)$$

The size s_i should typically be chosen proportional to the local sampling size. An important property of MLS is that the

continuity of the reconstruction depends on the continuity of the weight functions w_i [Lev98]. For instance, the aforementioned choice yields C^1 reconstructions.

This formalism is directly amenable to the reconstruction of 3D images [LGM*08], or oriented vector fields such as deformation fields [SMW06]. For the reconstruction of $d-1$ manifolds, a common approach is to seek an implicit scalar field such that its zero iso-contour approximates the input positions $\mathbf{x}_i$. In the MLS formalism, this means we are seeking local approximations such that $f_{\mathbf{x}}(\mathbf{x_i}) \approx 0$. To avoid the unwanted trivial solution $f_{\mathbf{x}} = 0$, additional constraints based on oriented input normals are commonly employed.

The MLS approach contrasts with variational global methods such as [ACSTD07] in that it does not require any preprocess and involves only local computations, making it suitable to the handle real-time settings and time-varying inputs [GGG08]. It produces continuous and analytically differentiable reconstructions, and does not require to define a somewhat arbitrary discretization. The weight function permits to elegantly control the amount of smoothing in a local fashion. The price to pay for these benefits is that since regressions are performed locally, a MLS approach is potentially less robust to disambiguate tricky configurations (e.g., close sheets) than techniques working globally.

3. Non-oriented MLS gradient fields

The MLS formalism presented in the previous section is not directly amenable to the reconstruction of non-oriented gradient fields since it is sensitive to the sign of input values $\mathbf{f}_i$. As motivated in Section 1, this is necessary for a number of applications including filtering, stylization, simplification and reconstruction. The main contribution of this paper is to explain how to adapt the MLS formalism to handle non-oriented input normals. Our approach starts by reconstructing a non-oriented gradient field matching the input normals, and then integrates it to recover a local implicit field if required by the application. We first describe the general approach (Sec 3.1), then discuss the choices of appropriate regularization (Sec 3.2), higher-order basis (Sec 3.3) and how to compute differential properties (Sec 3.4).

3.1. General approach

Our algorithm takes as input a set of non-oriented and unit vectors $\mathbf{g}_i \in \mathbb{R}^d$ specified at sample positions $\mathbf{x}_i \in \mathbb{R}^d$, with d the dimension of the ambient space (see Fig 3a for an example in $\mathbb{R}^2$). The samples are either pixels of an image or come from an arbitrary scattered point cloud. In this work, we further assume the directions $\mathbf{g}_i$ come from the normalized gradient of an unknown continuous scalar potential. Their orientation (i.e., signs) are either unknown and/or irrelevant.

Given an arbitrary evaluation point $\mathbf{x} \in \mathbb{R}^d$ and an Euclidean neighborhood $N_s(\mathbf{x}) = \{\mathbf{y} \in \mathbb{R}^d; ||\mathbf{x} - \mathbf{y}|| \leq s\}$ of size
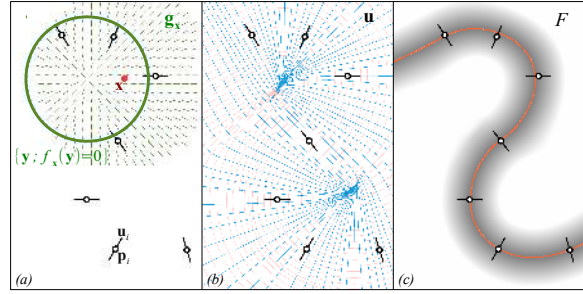

Figure 3: *Overview of our MLS approach. (a) In green, the local approximation $\mathbf{g_x}$ of the input vectors $\mathbf{u}_i$ around the evaluation point $\mathbf{x}$. (b) Streamlines of the global reconstruction of the continuous non-oriented vector field $\mathbf{u}$ (in blue), and of its respective 2D tangent field (in light red). (c) The reconstructed continuous and unsigned scalar potential F, and its 0-isovalue (in orange). Note that this iso-contour **does not** correspond to the tangent field streamlines. This figure has been made using the isotropic linear basis.*

s, we assume all gradients $\mathbf{g}_i$ (with $\mathbf{x}_i \in N_s$) may be well approximated by a low degree polynomial gradient field $\mathbf{g_x}$ defined in matrix form as (Fig 3a):

$$\mathbf{g_x}(\cdot) = \mathbf{B}(\cdot)^T \mathbf{u}(\mathbf{x}), \qquad (3)$$

where $\mathbf{B}$ is a polynomial basis matrix with d columns, and $\mathbf{u}$ is the vector of unknown polynomial coefficients. For local gradient approximation, $\mathbf{B}$ must represent an integrable basis, and its particular choice will be discussed in section 3.3.

The key idea to determine the polynomial coefficients $\mathbf{u}$ is to consider the sum of the squared dot products between the local gradient field and the prescribed input gradients $\mathbf{g}_i$, which is maximized when they are aligned, independently of their orientation:

$$\mathbf{u}(\mathbf{x}) = \arg\max_{\mathbf{u}} \sum_i w_i(\mathbf{x}) \left(\mathbf{g}_i^T \left(\mathbf{B}(\mathbf{x}_i - \mathbf{x})^T \mathbf{u} \right) \right)^2. \qquad (4)$$

Taken together, Eq 3 and 4 provide an analog to Eq 1 that is invariant to the sign of $\mathbf{g}_i$.

In matrix form, the maximization becomes:

$$\mathbf{u}(\mathbf{x}) = \arg\max_{\mathbf{u}} \mathbf{u}^T \mathbf{A}(\mathbf{x})\mathbf{u}, \qquad (5)$$

with the covariance matrix $\mathbf{A}$ defined as:

$$\mathbf{A}(\mathbf{x}) = \sum_i w_i(\mathbf{x}) \mathbf{B}(\mathbf{x}_i - \mathbf{x}) \mathbf{g}_i \mathbf{g}_i^T \mathbf{B}(\mathbf{x}_i - \mathbf{x})^T. \qquad (6)$$

From these local approximations, we reconstruct a continuous, but non-oriented, vector field $\mathbf{G} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ that globally approximates the discrete vectors $\mathbf{g}_i$ (Fig 3b). Since we are using a centered basis, $\mathbf{G}$ is classically defined as $\mathbf{G}(\mathbf{x}) = \mathbf{g_x}(0)$. Although the $\mathbf{g_x}$ are proper gradient fields (i.e., curl-free vector fields), this might not be the case of $\mathbf{G}$ in general (see Figure 17). This MLS reconstruction of the gradient field is used directly for image stylization.

When the input samples are supposed to lie on an unknown manifold, the goal is to recover a local scalar potential $f_{\mathbf{x}} = c + h_{\mathbf{x}}$. Since $\nabla f_{\mathbf{x}} = \mathbf{g}_{\mathbf{x}}$, $h_{\mathbf{x}}$ is directly obtained by integrating $\mathbf{g}_{\mathbf{x}}$. The constant term c is recovered such that the 0-isosurface of $f_{\mathbf{x}}$ best approximates the input sample positions $\mathbf{x}_i$ nearby $\mathbf{x}$ (Fig 3a). To this end, we minimize the potentials $f_{\mathbf{x}}(\mathbf{x}_i)$ in a moving least square sense, yielding:

$$c = -\frac{\sum_i w_i(\mathbf{x}) h_{\mathbf{x}}(\mathbf{x}_i - \mathbf{x})}{\sum_i w_i(\mathbf{x})}. \qquad (7)$$

As before, the sought-for manifold is implicitly defined as the 0-isosurface of a continuous unsigned scalar potential $F : \mathbb{R}^d \to \mathbb{R}$. It is computed by evaluating the local approximation $f_{\mathbf{x}}$ at $\mathbf{x}$: $F(\mathbf{x}) = f_{\mathbf{x}}(0)$, as illustrated in Fig 3c.

Two questions remain open though. Indeed, the maximization of Eq 5 is an under-constrained problem which has to be regularized in order to avoid the trivial solution of a vector field with infinite magnitude. This question will be addressed in the next sub-section, after which we will discuss the critical choice of a higher order polynomial basis for the local gradient fields.

3.2. Problem regularization

Before presenting our regularization constraint for a higher-order basis, we briefly study the case of a constant local approximation.

Regularization for constant approximations

In the case where $\mathbf{g}_{\mathbf{x}}$ is a constant function, $\mathbf{B}$ is the identity matrix and $\mathbf{g}_{\mathbf{x}}(\cdot) = \mathbf{u}(\mathbf{x})$. Since we assume unit input vectors, a natural regularization is then to constrain $\mathbf{u}(\mathbf{x})$ to be a unit vector. Under such a constraint, the solution $\mathbf{u}(\mathbf{x})$ of Eq 5 is directly found as the eigenvector $\mathbf{v}$ of the maximal eigenvalue λ of the following eigenvalue problem [All07]:

$$\mathbf{A}(\mathbf{x})\mathbf{v} = \lambda \mathbf{v}. \qquad (8)$$

This formulation corresponds to a continuous definition of the *structure tensor* built from discrete and possibly scattered inputs. In the context of MLS surface reconstruction, it has already been suggested by Amenta et al. [AK04], though to our knowledge no result has been shown yet.

Generic regularization

In order to overcome the limitations of constant approximations, we study the possibility to locally approximate the vector field by higher order polynomials. In this case, the previous regularization term ($\|\mathbf{g}_{\mathbf{x}}(\mathbf{y})\|^2 = 1, \forall \mathbf{y} \in \mathbb{R}^d$) does not apply: if $\mathbf{g}_{\mathbf{x}}$ is a polynomial of arbitrary degree, then its magnitude is not constant anymore. Nevertheless, one could be tempted to enforce this in a least squares sense by minimizing $\sum_i w_i(\mathbf{x})(\|\mathbf{g}_{\mathbf{x}}(\mathbf{x}_i - \mathbf{x})\| - 1)^2$. To our knowledge, there is no direct method to solve for Eq 5 with such additional
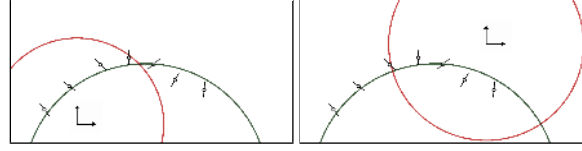

Figure 4: *Constraint comparisons. Fitted curves obtained with the naive constraint $\|\mathbf{u}(\mathbf{x})\| = 1$ (in red), and our constraint given in Eq 9 (in green). The only difference between left and right images is the choice of origin.*

least squares constraints. Moreover, this strategy would exhibit the undesirable effect to favor local solutions close to a constant gradient field because this is the only solution that can fully satisfy such a constraint in general. Naively constraining the Euclidean norm of the solution vector (i.e., $\|\mathbf{u}(\mathbf{x})\| = 1$) is not an option either. Indeed, as depicted in Fig 4, for a general basis, such a normalization is clearly not invariant to similarity transformations, and introduces a huge bias in the optimization process [Pra87].

Alliez et al. [ACSTD07] addresses a similar issue by solving over the solution space of unit biharmonic energy. In their approach, this regularization is applied to the final solution (i.e., $\int \|\Delta F\|^2 = 1$), thus leading to an expensive global problem that we strive to avoid here. Applying the same regularization to individual fits (i.e., $\int \|\Delta f_{\mathbf{x}}\|^2 = 1$) enforces the solutions to have non-zero high-order terms, thus preventing the approximation of constant gradient fields, and areas nearby inflection points.

Our key observation here is that, **in average**, the norm of $\mathbf{g}_{\mathbf{x}}$ should be equal to 1 nearby the considered samples. This is achieved by a quadratic normalization constraint:

$$\mathbf{u}(\mathbf{x})^T \mathbf{Q} \mathbf{u}(\mathbf{x}) = 1, \qquad (9)$$

with $\mathbf{Q}$ a symmetric positive-definite matrix:

$$\mathbf{Q}(\mathbf{x}) = \frac{\sum_i w_i(\mathbf{x}) \mathbf{B}(\mathbf{x}_i - \mathbf{x}) \mathbf{B}(\mathbf{x}_i - \mathbf{x})^T}{\sum_i w_i(\mathbf{x})}. \qquad (10)$$

Equation 9 reduces the space of solutions to the set of unitary vectors with respect to our *data-dependent* quadratic norm defined by the matrix $\mathbf{Q}$. Note that the choice of the target magnitude, here 1, is totally arbitrary and does not affect the directions of the fitted gradient. Strictly speaking, our norm $\mathbf{Q}$ is obviously not affine invariant. However, by construction it is both invariant to the choice of a basis frame, and to similarity transformations of the input data. The solution $\mathbf{u}(\mathbf{x})$ of Eq 5 is now directly found as the eigenvector $\mathbf{v}$ of the maximal eigenvalue λ of the following *generalized* eigenvalue problem [Tau91]:

$$\mathbf{A}(\mathbf{x})\mathbf{v} = \lambda \mathbf{Q}(\mathbf{x})\mathbf{v}. \qquad (11)$$

When $\mathbf{g}_{\mathbf{x}}$ represents a constant vector field, then $\mathbf{Q}$ is the identity matrix, and our generalized approach gently boils down to the previous structure tensor.
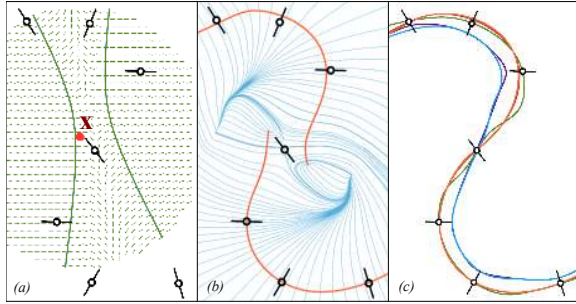
Figure 5: *Illustration of the overfitting issue of the fully integrable linear basis* $(\mathbf{g}^1, f^1)$, *with (a) the local gradient field and best fitted isoline for the given red point, and (b) the global MLS reconstruction of the gradient field and isocurve. (c) Comparison of various MLS variants: planar fit without normals (magenta) and with non-oriented normals (***our*** f^0, blue), spherical fit without normals (green) and with non-oriented normals (***our*** f^*, orange), and for comparison purpose, spherical fit with consistently oriented normals (red). Note that these last two curves match almost exactly.*

3.3. Choice of basis

The choice of basis for $\mathbf{g_x}$ is critical since it characterizes the quality of the approximation. For the sake of clarity, we explicitly develop Eq 3 for constant and linear basis. More precisely, we consider constant $\mathbf{g}^0$, linear isotropic $\mathbf{g}^*$, and general linear $\mathbf{g}^1$ gradient fields whose expression and respective scalar potential are given below:

$$
\begin{array}{rclcrcl}
\mathbf{g}^0(\mathbf{y}) & = & \mathbf{r} & \rightarrow & f^0(\mathbf{y}) & = & c + \mathbf{r}^T \mathbf{y} \\
\mathbf{g}^*(\mathbf{y}) & = & \mathbf{r} + l\mathbf{y} & \rightarrow & f^*(\mathbf{y}) & = & c + \mathbf{r}^T \mathbf{y} + \frac{1}{2} l \mathbf{y}^T \mathbf{y} \\
\mathbf{g}^1(\mathbf{y}) & = & \mathbf{r} + \mathbf{L} \cdot \mathbf{y} & \rightarrow & f^1(\mathbf{y}) & = & c + \mathbf{r}^T \mathbf{y} + \frac{1}{2} \mathbf{y}^T \mathbf{L} \mathbf{y}
\end{array}
$$

where the scalar c is a constant term, and the vector $\mathbf{r}$ stores the coefficients of the linear term. For the quadratic term, we either use a scalar value l, or a symmetric matrix $\mathbf{L}$ to ensure that $\mathbf{g}^1$ is an integrable gradient field. To better see the link with the formulation of Eq 3, observe that $\mathbf{g}^*$ is obtained using $\mathbf{B}^T = (\mathbf{I}, \mathbf{y})$ and $\mathbf{u} = (\mathbf{r}^T, l)^T$. Once integrated, they represent a hyper-plane, a hyper-sphere, and a general quadric respectively. $\mathbf{g}^*$ and f^* are illustrated in Fig 3, compared to others in Fig 5 for sparse data, and in Fig 6 for an image. As already noticed in Section 1, the constant basis cannot approximate well highly-curved regions, yielding over-smoothing of the features and even instabilities (Fig 5c, 6b). On the other hand, the general linear basis of $\mathbf{g}^1$ already presents too many degrees of freedom (d-o-f) (5 in 2D, and 9 in 3D), and leads to over-fitting issues in the form of the generation of oscillations and details which are not present in the input data (Fig 5a, 5b, 6c).

The isotropic basis $\mathbf{g}^*$ clearly appears to be the best trade-off. Compared to a constant basis it only increases by one the number of d-o-f while offering a much richer space of solutions. Keeping the number of d-o-f as low as possible is highly desirable, not only for performance reasons, but
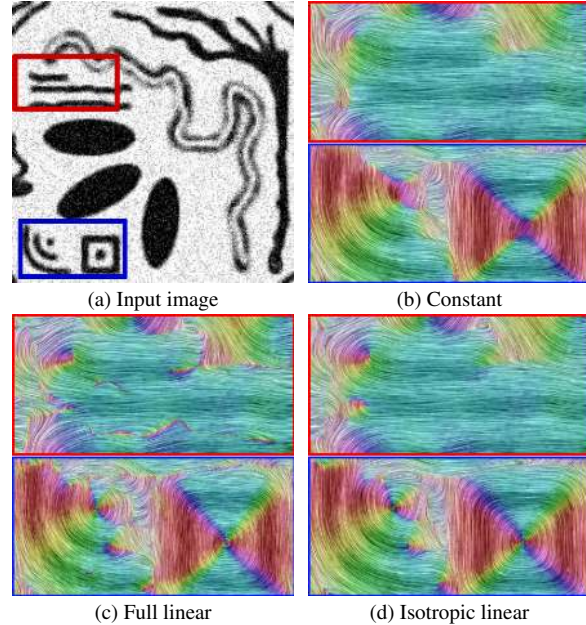

(a) Input image   (b) Constant

(c) Full linear   (d) Isotropic linear

Figure 6: *Comparisons using a noisy image. The input noisy image (400×400) in (a) requires a large smoothing neighborhood ($s = 16$). The gradient field is visualized using line-integral convolution, and the color code of Fig. 2. The constant approximation (b) tends to oversmooth the data at line extremities and dot centers. The full linear approximation (c) leads to over-fitting issues. Our isotropic linear solution provides the best trade-off (d).*

also to keep high stability with small neighborhoods. Enlarging the neighborhood increases both the computation costs and the risk to take into account different and/or inconsistent parts of the input data. Moreover, it should be noted that in the case of manifold reconstruction, our approach works best if the norm of the gradient along a given isovalue of the respective local scalar potential is constant. Since this is not the case of the full linear gradient, this basis also suffers from a slight bias in the fitting process.

Figure 5c shows that the non-oriented normal information has a greater impact on spherical MLS (isotropic basis) than on planar MLS. Our intuition is that the non-oriented normal information is required to avoid the fitted spheres to *sink* in-between the input samples, thus producing a high quality smooth surface that well preserves the features. Finally, it should be noted that in this example, our MLS reconstruction with non-oriented normals works very closely to the APSS method which can fit spheres with oriented normals only.

3.4. Differential quantities

Assuming differentiable weight functions w_i, our global gradient and scalar fields $\mathbf{G}$ and F can be analytically differentiated. In particular, the computation of surface or curve

normals requires to first compute the gradient ∇F of our reconstructed scalar field, which is given by:

$$\nabla F(\mathbf{x}) = \nabla c(\mathbf{x}) + \left(1, \mathbf{x}^T, \mathbf{x}^T \mathbf{x}\right) \nabla \mathbf{u}(\mathbf{x}) + \mathbf{B}(\mathbf{x})^T \mathbf{u}(\mathbf{x}) . \quad (12)$$

The computation of ∇c is straightforward, as well as $\mathbf{B}^T \mathbf{u}$ since it corresponds to our global gradient $\mathbf{G}$. In contrast, the computation of $\nabla \mathbf{u}$ is more involved since it requires the differentiation of a generalized eigenvalue problem:

$$\nabla \mathbf{u} = \mathbf{V}(\Sigma - \lambda \mathbf{I})^+ \mathbf{V}^T \left(\nabla \mathbf{A} - \lambda \mathbf{Q}\right) \mathbf{u} - \frac{1}{2} \left(\mathbf{u}^T \nabla \mathbf{Q} \mathbf{u}\right) \mathbf{u} ,$$

where $\mathbf{V}$ and Σ are the eigenvectors and eigenvalues of equation 11, and $(.)^+$ denote the Moore-Penrose inverse of the singular matrix $\Sigma - \lambda \mathbf{I}$ (see Magnus [Mag85] for details).

Another useful differential quantity is the tangential curvature κ_t of the tangential field. This requires to compute the Jacobian of our reconstructed gradient field, which could be obtained as above. However, in our applications we have found sufficient to neglect the variations of $\mathbf{g_x}$ with respect to $\mathbf{x}$. κ_t thus boils down to the curvature of the local gradient field $\mathbf{g_x}$, that is: $\kappa_t \approx l/||\mathbf{G}||$.

4. Applications

4.1. Image stylization

This section demonstrates our non-oriented MLS gradient fields on a variety of image stylization techniques. We show how our approach improves them by systematically comparing the results obtained with our Constant Approximations (CA) and Isotropic Linear Approximations (ILA).

Implementation details

In this paper, we employed a 5×5 Gaussian derivative filter to compute the image gradient ∇I from which we get input unit vectors $\mathbf{u}_i$. For fast implementation, we use the following alternative *separable* and compactly supported weight function:

$$w_i(x,y) = \phi\left(\frac{||x - x_i||}{s}\right) \phi\left(\frac{||y - y_i||}{s}\right) ||\nabla I(x_i, y_i)||^2 .$$

where the last factor is employed to favor gradients with large magnitudes.

Multi-channel (e.g., color) images yield multiple gradient vectors per pixels which are all taken into account and approximated during the construction of the covariance matrix $\mathbf{A}$ (Eq 6). Our image processing system is entirely implemented on the GPU, and our results are obtained using an NVIDIA GeForce GTX 460 with 1GB of memory.

Thanks to the separability and GPU implementation, our system easily achieves real-time performance. For instance, on a 512×512 image, our non-optimized implementation of the ILA costs 5.1 ms (resp. 11.5 ms) for a neighborhood of size $s = 5$ (resp. 20) pixels. Compared to a structure tensor, we observe a slowdown factor of about 1.5 to 2 because of the need to solve a slightly larger eigenvalue problem (3×3 instead of 2×2). Our ILA implementation extracts the

maximal eigenvalue using the iterative Power method, while we use a direct solver for CA. Better performance may be achieved using a direct 3×3 eigensolver.

Results

We first provide in Fig 7 a formal comparison between the three basis functions we have considered in this paper. To this end, we have generated three configurations of an analytic height field $h(x,y) = \exp\left(-\frac{x^2}{r_x^2} - \frac{(y + \alpha \exp(-x^2/r_x^2))^2}{r_y^2}\right)$: spherical ($r_x = r_y = 1, \alpha = 0$), elliptical ($r_x = 2, r_y = 1, \alpha = 0$) and croissant-shaped ($r_x = 2, r_y = 1, \alpha = 1$). We have then created images of their corresponding gradients, corrupted them with additive white noise and reconstructed non-oriented gradients with each of the three basis. We introduce the following local error measure for comparison: $err = |sin(\theta^* - \theta)|$, where θ^* and θ are the angles of the ground truth and reconstructed gradient directions respectively (this formula is used to discard differences in orientation). The total error measure is simply obtained by summing up local error measurements. This comparison shows the superiority of the isotropic linear basis compared to the other two: it produces the lowest errors without introducing any spurious artifact.

Figure 8 illustrates our approach applied to the flow-based color abstraction (F-Abs) filter [KLC09]. This filter employs a 1D bilateral filter [TM98] along the tangential field (orthogonal to the gradient field) to preserve image structure through abstraction. In comparison to CA, it can be seen that ILA much better preserves important structures of the input image. This filter can also be seen in the supplemental video, where our method naturally exhibits good temporal coherence even when processing each frame independently. The advantage of our ILA is further illustrated in Fig 9, where continuous glass patterns [PP09] are guided by our gradient fields to mimic brush painting. Smoothing directly oriented directions completely loses the original image structure, our CA tends to over-smooth the round shape of the eye, while our ILA preserves well its roundness.

As already noted, our CA method can be seen as a continuous variant of classical *structure tensor* smoothing [KD08]. This is a key advantage for enabling high-quality dynamic zooms without any particular overhead since the cost of our approach is linear in term of number of evaluations, i.e., in term of the number of pixels of the output image. This contrasts with approaches based on a global formulation (e.g., harmonic smoothing) for which the whole input image always has to be considered. This ability is visible in Fig 6, but also in the insets of Fig 10, and in the video where the lines produced by coherence-enhancing filtering [KK11] naturally separate or merge depending on the zooming level. This filter applies a 1D shock filter [OR90] along the gradient field to sharpen edges and a 1D bilateral filter along the tangent field to abstract colors. Note that the line-drawing in Fig 10a provides a sparse information since most of the
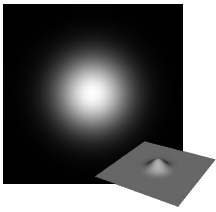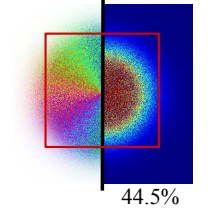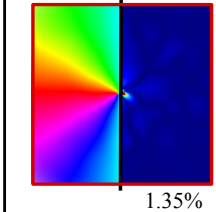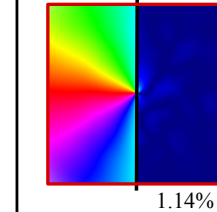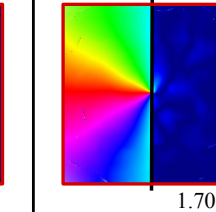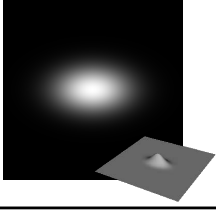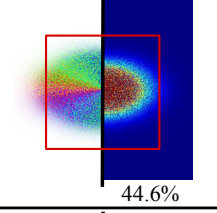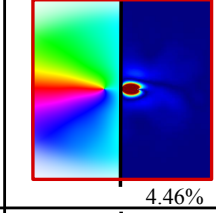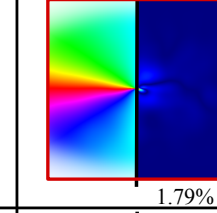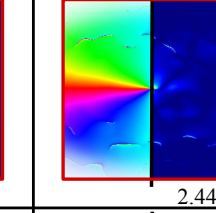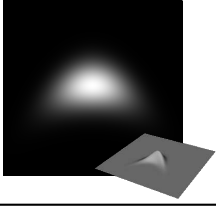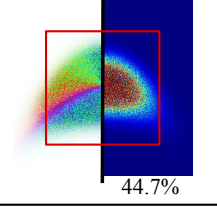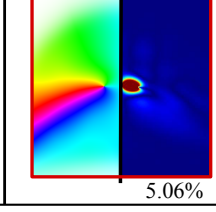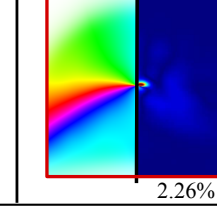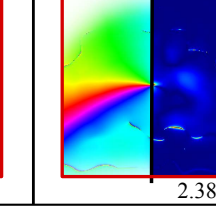
Figure 7: *Comparisons using an analytical function. We use three configurations of an analytical height field (see text) to justify our choice of basis. The corresponding input gradient fields are corrupted with noise, with gradients displayed on the left half with the color code of Fig 2, and an error measure displayed with a blue-to-red gradient on the right half. The total error is given at the bottom of each image. The same visualizations are used to compare non-oriented gradient fields obtained with the constant, isotropic linear and full linear basis. Observe in particular how the isotropic linear basis produces significantly lower local and total errors compared to the constant basis, while avoiding spurious artifacts as opposed to the full linear basis.*

pixels present a null gradient. Figure 10d depicts the ability of our higher-order ILA to provide a tangential curvature information κ for free. In this example, we extend the original coherence-enhancing filter to automatically adjust the shock filter size with respect to κ. As a last image stylization example, we compare CA and ILA in Fig 11 at multiple scales using the eXtended Difference of Gaussians (X-DoG) filter [Win11], which converts a color image into a black-and-white style. Here we show that ILA is much more stable than CA across scales.

4.2. Surface and curve reconstruction

When applied to surface reconstruction, our novel approach essentially extends the APSS method [GG07] to support non-oriented input normals, while APSS is limited to consistently oriented inputs.

Implementation details

As a fast preprocess, we employed the sphere fitting without normal technique of the APSS framework to estimate the input normal directions $\mathbf{u}_i$ since it better preserves the surface details than simpler covariance analysis. An alternative would be to use the *natural normal* field definition

of Alexa et al. [AA09]. The reconstructed unsigned implicit functions have been meshed using the advancing front algorithm implemented in MeshLab. Oriented normals may be re-estimated after this step, although we have not found this necessary in practice. Our approach is also well suited for direct raytracing [AA04a], or for the GPU accelerated resampling framework of Guennebaud et al. [GGG08] to enable real-time visualization. We have relied on such approximate reconstruction in our sketching application, which explains why reconstructed curves exhibit slight thickness variations.

Results

Figure 1 illustrates the ability of our approach to reconstruct very sparse data with high fidelity. Our approach performs also remarkably well in the presence of noise as illustrated in Fig 12. As can be seen, even with a small filter size, a constant gradient field basis already looses many traits of the underlying shape while exhibiting a lot of noise. On the other hand, our approach is able to smooth out the noise while retaining most surface details. Figure 13 shows our surface reconstruction applied to an object with sharp features. Near sharp edges and corners, the solution of Eq 8 becomes poorly defined (the highest eigenvalue is not uniquely
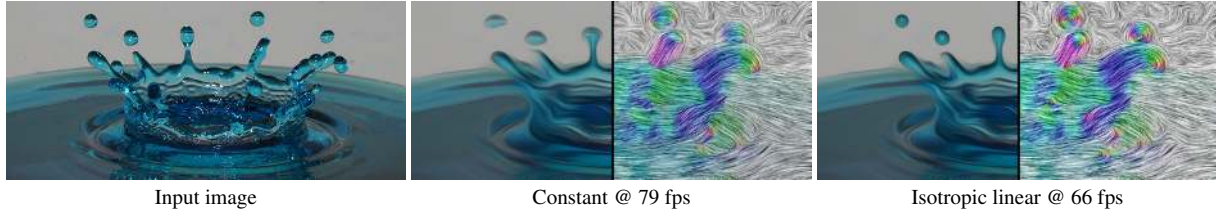
Input image   Constant @ 79 fps   Isotropic linear @ 66 fps

Figure 8: *Flow-based abstraction on a 500×327 image using a neighborhood of size $s = 16$. Note the general improvement in shape preservation using our isotropic linear approximation, especially for water droplets.*
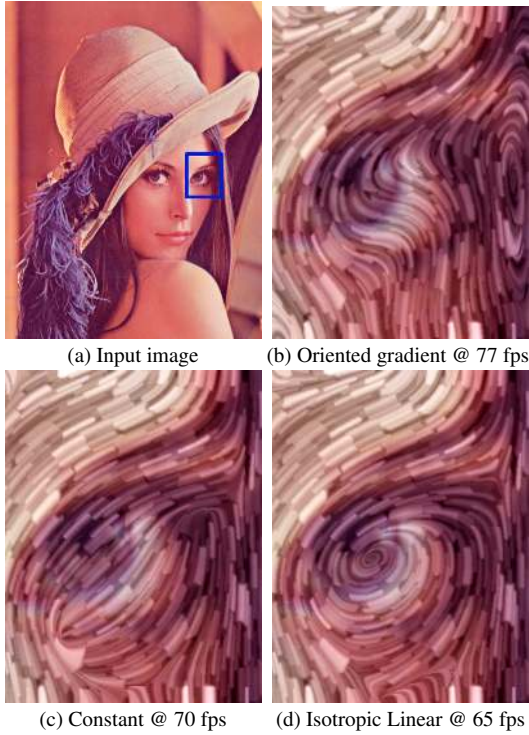

(a) Input image   (b) Oriented gradient @ 77 fps

(c) Constant @ 70 fps   (d) Isotropic Linear @ 65 fps

Figure 9: *Continuous glass pattern on the eye of Lena image (394×510) using a neighborhood of size $s = 12$ for the MLS gradient field. We use a factor of 2.0 for tangential smoothing, and a 10 pixel-wide brush thickness. Directly smoothing oriented gradients quickly loses the original shape (b). In contrast to CA (c), our ILA (d) better preserves the structures in the original image.*
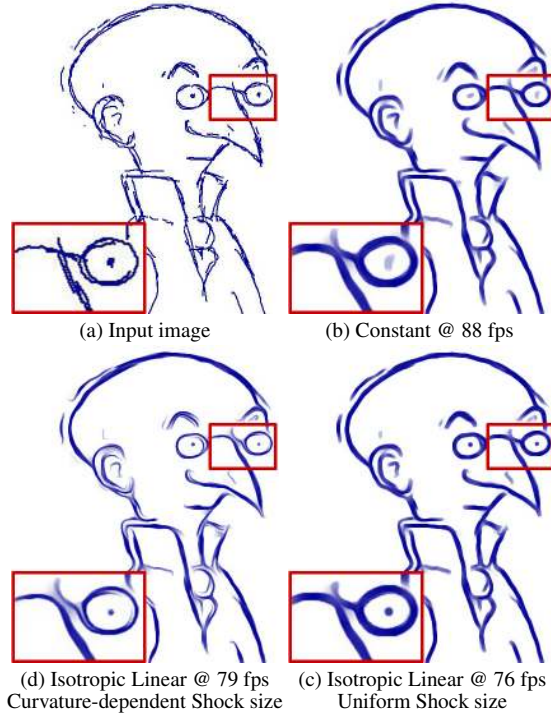

(a) Input image   (b) Constant @ 88 fps

(d) Isotropic Linear @ 79 fps   (c) Isotropic Linear @ 76 fps
Curvature-dependent Shock size   Uniform Shock size

Figure 10: *Coherence-enhancing filtering for line drawing simplification on a 422×499 image using a neighborhood of size $s = 11$ for the MLS gradient field. We use a factor of 2.0 for tangential smoothing, and a 10 pixel-wide shock filter size. Compared to CA (b), our ILA (c) improves the overall sharpness. Furthermore, it permits to locally adjust the shock size based on curvature to vary stylization (d).*
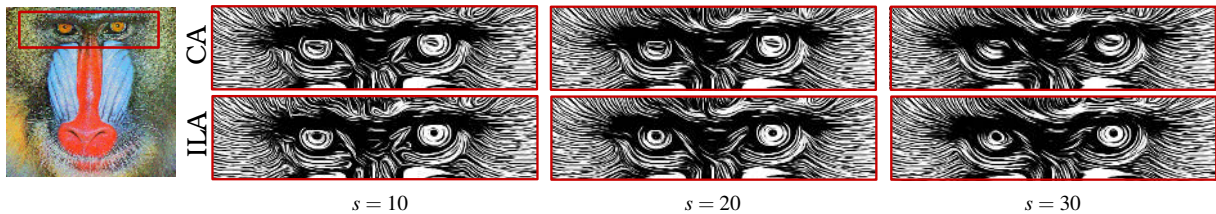

$s = 10$ $s = 20$ $s = 30$

Figure 11: *X-DoG at multiple scales. We compare CA and ILA on a region around the eyes of the Mandril image at three scales. Observe how round features are better preserved through increasing scales with our approach.*
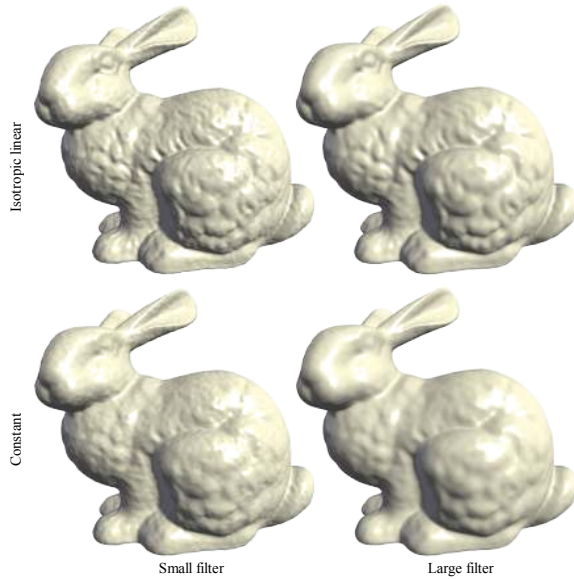
Figure 12: *Noisy surface reconstruction. Reconstruction of the Stanford bunny model with 1% additional noise using our non-oriented isotropic linear (top) and constant (bottom) gradient field approximations, and using two different filter sizes (1.7, and 2.3%).*
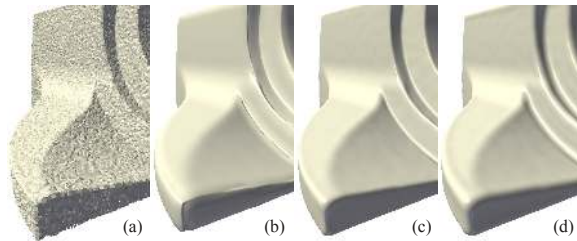

Figure 13: *The fandisk model corrupted with noise (a), and reconstructed using non-oriented constant gradients (b), our isotropic linear basis (c), and the APSS method (d). Our solution (c) is free of artifacts, and sharper than (d).*

Figure 14: *A highly non-uniformly sampled model is challenging for purely local methods, yet we obtain a satisfying reconstruction using our method.*

Sketched strokes   Reconstructed curves

Sketched strokes   Reconstructed curves   Varying thickness

Figure 15: *Curve from sketched strokes. Our approach for reconstructing of a 2D curve from drawn strokes is quite robust to the drawing style, like straight lines (top row) or scribbling (middle row). In the bottom row, the reconstructed curves are further re-stylized with varying thickness depending on the approximate tangential curvature.*

defined), and so using a constant basis to reconstruct the gradient field leads to jagged edges. This issue is addressed by our approach that even exhibits sharper edges compared to the fitting of spheres using oriented normals (APSS). Figure 14 illustrates our approach on a very challenging sampling obtained using a Kreon laser scanning device. Mullen et al. [MDGD*10] present a reconstruction of this model with their global non-oriented approach. As discussed in section 5, their solution is generally more robust to topological ambiguities. On the other hand, it is also one or two orders of magnitude slower compared to our purely local solution.

Figure 15 illustrates the natural stroke simplification ability of our approach. Here the 2D real-time reconstruction takes place in the background while the input gradient information directly comes from the stroke directions. Coherently
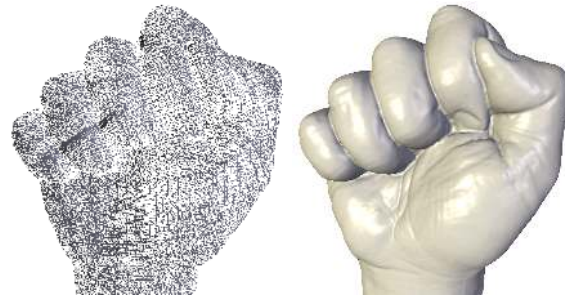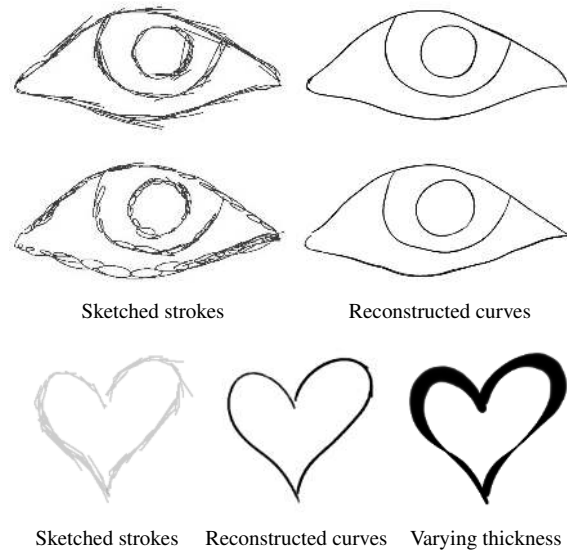
orienting the gradients between the different strokes would be particularly challenging on these examples. We further show at the bottom of Fig 15 that the approximate tangential curvature κ can be used to vary line thickness, hence re-stylizing the input line drawing.

5. Discussion

Joint interpolation. Both our constant and higher-order gradient fields are defined at any location, thus allowing for infinite zoom at a cost which is independent of the zoom level. On the other hand, our implementations of the presented stylization filters currently make use of simple bilinear interpolation of the color information. We believe sharper results
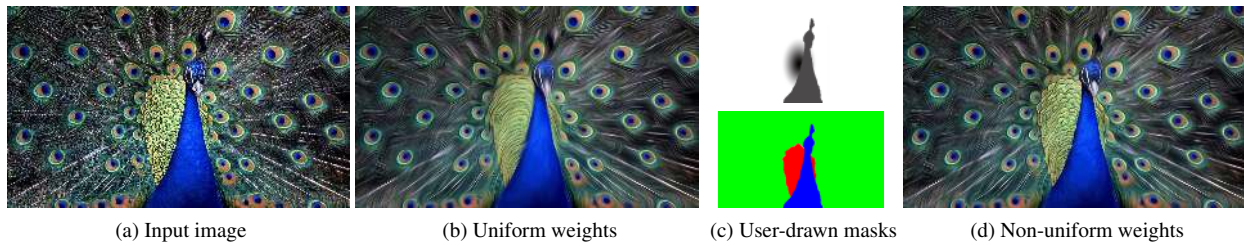
| (a) Input image | (b) Uniform weights | (c) User-drawn masks | (d) Non-uniform weights |

Figure 16: *Non-uniform weights for adaptive flow-based abstraction By adjusting the neighborhood size using a hand-drawn scale image (c-top) and hand-selected segmentation (c-bottom), we locally improve the preservation of small features (e.g., on the peacock body and head) and the preservation of original discontinuities (e.g., between the body and tail feathers), as seen when comparing (b) and (d). A comparison with constant approximations is given in supplemental materials.*

could be obtained by using more elaborated color interpolation techniques. Our gradient fields could certainly help here for better edge preservation.

Extensions. In this paper we focused on the approximation of *smooth* gradient fields mainly ignoring the junctions, occlusions and sharp edges that may arises in images or 3D point clouds. This is however an orthogonal problem. Indeed, to deal with discontinuities, many extensions of MLS have already been proposed (e.g., [FCOS05, OGG09]), and many of them could be directly applied to our approach without major difficulties. In the same vein, we consider the choice of the neighborhood size s as a separate and more general problem. For instance, Fig 16 illustrates the use of masks to clip the neighborhoods at discontinuities, and locally adjust their size. In future work, we would like to investigate the adaptation of the Growing Least-Squares approach [MBG*12] to identify pertinent scales of non-oriented gradient fields automatically. For videos, we think that the use of motion flow could increase temporal smoothing, and help deal more properly with occluding contours.

Residual curl. As shown in Figure 17, the reconstructed MLS field **G** might not be curl-free. Strictly speaking, it is not a pure gradient field. Nevertheless, in practice, regions with non-zero curls are restricted to singularities (in red) and to locations where there is no meaningful information, i.e., regions where the magnitude of the input gradient is very small (in blue).

Sheet separation. It is worth noting that if the input normals are known to be consistently oriented, then our novel MLS surface approach has no real advantages compared to the APSS method. In particular, oriented normals combined with sphere fitting performs remarkably well at *sheet separation* under very sparse sampling (Fig 18d), whereas our approach will collapse the two sheets more like any planar approximations would do (Fig 18b). Of course, this remark only applies if the normal orientations can be known in advance without any ambiguity (Fig 18c). We believe that a global integration of our MLS gradient field could probably overcome this difficulty at the cost of significantly more expensive computations [ACSTD07].
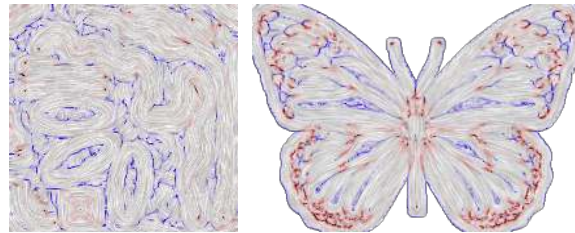

Figure 17: *Our reconstructed MLS fields locally have non-zero curl (red/blue pixels), as illustrated on the noisy and butterfly images. Locations where the input gradient magnitude is smaller (resp. larger) than a threshold value (we use 0.0005) are shown in blue (resp. red).*

Extra zeros. As usual with MLS surface reconstruction, the global scalar field F might present some *unwanted extra zeros* [AA06]. Since, by definition, they happen far away from the input points, they are not a problem for smoothing, resampling, or meshing with an advancing front. If evaluations happen in the whole space (e.g., for raytracing) then solution such as the one proposed by Guennebaud and Gross [GG07] may be used.

6. Conclusion

We have demonstrated how the classical structure tensor can be advantageously generalized to achieve higher-order approximations of non-oriented gradient fields. In particular, we have shown that the best trade-off between stability and reconstruction fidelity lies in local isotropic linear approximations, provided our appropriate regularization is employed. The presented MLS formalism provides continuous solutions at arbitrary scales with a low computational complexity. Our approach is particularly effective for local surface and curve reconstructions from sparse and/or noisy data sets, as well as for image abstraction and stylization. They constitute two kinds of applications for which it is seldom possible to orient gradients consistently. A potential alternative would be to consider non-oriented fields as oriented vector fields on the covering space [NP09]. Such an approach deserves further investigations, and would present
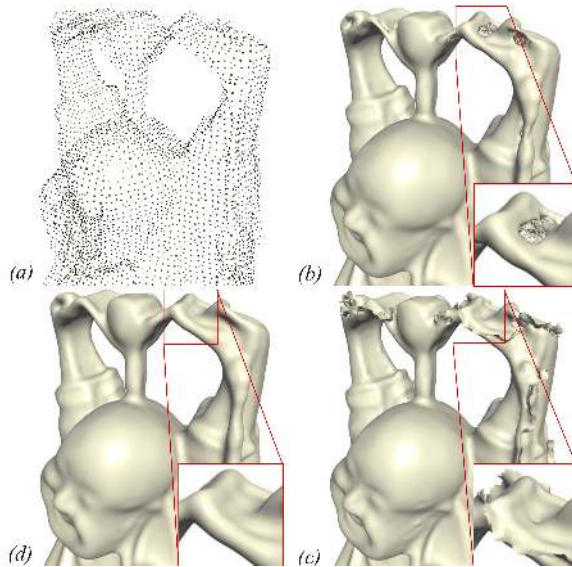
Figure 18: *A highly downsampled version of the Happy Buddha presents some very thin undersampled layers (a), and yields to* collapsing *artefacts with our approach (b). On this difficult test case, the normal orienter of MeshLab (based on [HDD*92]) fails to correctly orient the computed normals, thus making a correct reconstruction impossible for APSS (c), unless using the normals of the original high resolution model (d).*

the benefit of enabling classic vector field processing onto non-oriented fields.

We believe that our approach could benefit to many other types of data and applications. Since our MLS formulation works for arbitrary dimensions, it could be used for scientific visualization (e.g., 3D medical imaging). It may serve as a basis for multi-scale image decomposition, using increasing MLS support sizes [MBG*12]. For videos, optic flows may be used to combine corresponding gradients across subsequent frames, for instance by advecting the MLS weight functions. Finally, global surface reconstruction methods that rely on analytic gradient fields [KBH06, ACSTD07] could also benefit from our approach.

Acknowledgments

References

[AA04a] ADAMSON A., ALEXA M.: Approximating bounded, non-orientable surfaces from points. In *Shape Modeling International* (2004), pp. 243–252. 7

[AA04b] ALEXA M., ADAMSON A.: On normals and projection operators for surfaces defined by point sets. In *Proc. Eurographics Symposium on Point-Based Graphics* (2004), pp. 149–156. 1

[AA06] ADAMSON A., ALEXA M.: Anisotropic point set surfaces. In *Afrigraph '06: Proc. international conference on Computer graphics, virtual reality, visualisation and interaction in Africa* (2006), ACM, pp. 7–13. 10

[AA09] ALEXA M., ADAMSON A.: Interpolatory point set surfaces-convexity and hermite data. *ACM Transactions on Graphics (TOG) 28*, 2 (2009), 20. 7

[ACSTD07] ALLIEZ P., COHEN-STEINER D., TONG Y., DESBRUN M.: Voronoi-based variational reconstruction of unoriented point sets. In *Proc. Eurographics Symposium on Geometry Processing 2007* (2007), pp. 39–48. 2, 3, 4, 10, 11

[AK04] AMENTA N., KIL Y. J.: Defining point-set surfaces. *ACM Trans. Graph. 23*, 3 (2004), 264–270. 1, 2, 4

[All07] ALLAIRE G.: *Numerical Analysis and Optimization: An Introduction to Mathematical Modelling and Numerical Simulation*. 2007. 4

[BG87] BIGÜN J., GRANLUND G. H.: Optimal orientation detection of linear symmetry. In *Proc. IEEE International Conference on Computer Vision* (1987), pp. 433–438. 2

[BVL*06] BROX T., VAN DEN BOOMGAARD R., LAUZE F., VAN DE WEIJER J., WEICKERT J., KORNPROBST P.: Adaptive structure tensors and their applications. In *Visualization and image processing of tensor fields*, Weickert J., Hagen H., (Eds.), vol. 1. Springer, Jan. 2006, ch. 2, pp. 17–47. 2

[FCOS05] FLEISHMAN S., COHEN-OR D., SILVA C. T.: Robust moving least-squares fitting with sharp features. *ACM Trans. Graph. 24*, 3 (2005), 544–552. 10

[GG07] GUENNEBAUD G., GROSS M.: Algebraic point set surfaces. *ACM Trans. Graph. 26*, 3 (2007). 1, 2, 7, 10

[GGG08] GUENNEBAUD G., GERMANN M., GROSS M.: Dynamic sampling and rendering of algebraic point set surfaces. *Computer Graphics Forum 27*, 2 (2008). 3, 7

[HDD*92] HOPPE H., DEROSE T., DUCHAMP T., MCDONALD J., STUETZLE W.: Surface reconstruction from unorganized points. In *Proc. SIGGRAPH '92* (1992), ACM, pp. 71–78. 1, 11

[HE04] HAYS J., ESSA I.: Image and video based painterly animation. In *NPAR'04: Proc. symposium on Non-photorealistic animation and rendering* (2004), ACM/Eurographics, pp. 113–120. 1

[Her98] HERTZMANN A.: Painterly rendering with curved brush strokes of multiple sizes. In *Proc. SIGGRAPH '98* (1998), ACM, pp. 453–460. 1

[HLZ*09] HUANG H., LI D., ZHANG H., ASCHER U., COHEN-OR D.: Consolidation of unorganized point clouds for surface reconstruction. *ACM Trans. Graph. 28*, 5 (2009). 2

[KBH06] KAZHDAN M., BOLITHO M., HOPPE H.: Poisson surface reconstruction. In *Proc. Eurographics Symposium on Geometry Processing 2006* (2006), pp. 43–52. 1, 11

[KD08] KYPRIANIDIS J., DÖLLNER J.: Image abstraction by structure adaptive filtering. *Proc. EG UK Theory and Practice of Computer Graphics* (2008), 51–58. 2, 6

[KK11] KYPRIANIDIS J., KANG H.: Image and video abstraction by coherence-enhancing filtering. *Computer Graphics Forum 30*, 2 (2011), 593–602. 2, 6

[KKD09] KYPRIANIDIS J., KANG H., DÖLLNER J.: Image and video abstraction by anisotropic kuwahara filtering. *Computer Graphics Forum 28*, 7 (2009), 1955–1963. 1

[KLC07] KANG H., LEE S., CHUI C.: Coherent line drawing. In *NPAR'07: Proc. symposium on Non-photorealistic animation and rendering* (2007), ACM/Eurographics, pp. 43–50. 1

[KLC09] KANG H., LEE S., CHUI C. K.: Flow-based image abstraction. *IEEE Trans. Visualization and Comput. Graph. 15*, 1 (2009), 62–76. 1, 2, 6

[Lev98] LEVIN D.: The approximation power of moving least-squares. *Mathematics of Computation 67* (1998), 1517–1531. 2

[LGM*08] LEDERGERBER C., GUENNEBAUD G., MEYER M., BACHER M., PFISTER H.: Volume MLS Ray Casting. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1372–1379. 3

[Mag85] MAGNUS J. R.: On differentiating eigenvalues and eigenvectors. *Econometric Theory 24*, 1 (1985), 179–191. 6

[MBG*12] MELLADO N., BARLA P., GUENNEBAUD G., REUTER P., SCHLICK C.: Growing Least Squares for the Continuous Analysis of Manifolds in Scale-Space. *Computer Graphics Forum* (July 2012). 10, 11

[MDGD*10] MULLEN P., DE GOES F., DESBRUN M., COHEN-STEINER D., ALLIEZ P.: Signing the Unsigned: Robust Surface Reconstruction from Raw Pointsets. *Computer Graphics Forum 29*, 5 (2010), 1733–1741. 2, 9

[MGG04] MITRA N., GUYEN N., GUIBAS: Estimating surface normals in noisy point cloud data. *International Journal of Computational Geometry and Applications* (2004). 2

[NP09] NIESER M., POLTHIER K.: Parameterizing singularities of positive integral index. In *13th IMA International Conference on Mathematics of Surfaces XIII* (2009), pp. 265–277. 10

[OGG09] OZTIRELI C., GUENNEBAUD G., GROSS M.: Feature preserving point set surfaces based on non-linear kernel regression. *Computer Graphics Forum 28*, 2 (2009), 493–501. 10

[OR90] OSHER S., RUDIN L. I.: Feature-oriented image enhancement using shock filters. *SIAM J. Numer. Anal. 27*, 4 (Aug. 1990), 919–940. 6

[PP09] PAPARI G., PETKOV N.: Continuous glass patterns for painterly rendering. *Image Processing, IEEE Transactions on 18*, 3 (2009), 652–664. 1, 6

[Pra87] PRATT V.: Direct least-squares fitting of algebraic surfaces. In *Proc. SIGGRAPH '87* (1987), ACM, pp. 145–152. 4

[SMW06] SCHAEFER S., MCPHAIL T., WARREN J.: Image deformation using moving least squares. *ACM Trans. Graph. 25*, 3 (2006), 533–540. 3

[SOS04] SHEN C., O'BRIEN J. F., SHEWCHUK J. R.: Interpolating and approximating implicit surfaces from polygon soup. *ACM Trans. Graph. 23*, 3 (2004), 896–904. 1

[Tau91] TAUBIN G.: Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell. 13*, 11 (Nov. 1991), 1115–1138. 4

[TM98] TOMASI C., MANDUCHI R.: Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on* (1998), IEEE, pp. 839–846. 6

[Win11] WINNEMÖLLER H.: XDoG: advanced image stylization with eXtended Difference-of-Gaussians. In *NPAR '11 :Proc. symposium on Non-Photorealistic Animation and Rendering* (2011), ACM/Eurographics, pp. 147–156. 7

[XCOJ*09] XU K., COHEN-OR D., JU T., LIU L., ZHANG H., ZHOU S., XIONG Y.: Feature-aligned shape texturing. *ACM Trans. Graph. 28*, 5 (Dec. 2009), 108:1–108:7. 2

[ZHT07] ZHANG E., HAYS J., TURK G.: Interactive tensor field design and visualization on surfaces. *IEEE Trans. Visualization and Computer Graphics 13*, 1 (2007), 94–107. 2