Adaptive Multiscale Discretizations for Vision

V. Estellers\textsuperscript{1}, G. Graber\textsuperscript{2}, S. Soatto\textsuperscript{1}

\textsuperscript{1}UCLA Vision Lab, \textsuperscript{2}Institute for Computer Graphics and Vision TU Graz

Abstract. Variational problems in vision are solved numerically on the pixel lattice because it provides the simplest computational grid to discretize the input images even though a uniform grid is seldom well-matched to the complexity of the solution or the resolution power of the model. To address this issue, we introduce multiresolution discretizations that locally adapt the resolution of a piecewise polynomial solution to the resolving power of the variational model or complexity of its solution. Besides classic triangular finite elements, we investigate quad- and octree element grids that efficiently locate pixels and produce multiresolution representations of the solution. We combine our discretization with the optimization algorithms of finite differences to solve the non-differentiable functionals that characterize vision and develop algorithms easy to parallelize. Our 2 and 3D experiments in image segmentation, optical flow, stereo, and depth fusion validate our method as achieving significant computational savings with a minimal loss of accuracy.

1 Introduction: Vision Needs Adaptive Discretizations

Many inference tasks in vision can be formulated as variational optimization problems that estimate a smooth function over the image plane. This variational approach is more robust and accurate than sparse feature techniques because it exploits all the information available in the input images to estimate a solution at pixel resolution. With standard discretizations, however, this produces optimization problems with as many variables as input image pixels and limits the development of variational models for real-time or resource-constrained systems.

Variational techniques solve vision problems by (i) designing cost functionals that model the properties of the solution and input data, (ii) discretizing the continuous model into the digital domain, and (iii) developing numerical algorithms to minimize it. While there is a vast literature on modeling and optimization for vision, the choice of discretization has been largely overlooked. In particular, most algorithms are implemented on the pixel lattice because it is the simplest discretization of the problem, even though it is an extremely inefficient representation of the solution because it is not adapted to the resolution power of the model or the complexity of the images. In Sections 4-5, we introduce adaptive discretizations that match the complexity of the unknown solution and reduce the computational cost of variational models in vision. We draw inspiration from the refinement techniques of finite-element (FE) methods to construct a discretization adapted to the data, multi-resolution in nature, and designed for systems requiring best estimates in real time or within a computational budget.
Conversely, we bring the optimization algorithms of vision to the domain of FE techniques. FE solvers are not useful for vision because they solve variational problems through their Euler-Lagrange PDE and cannot handle non-differentiable functionals like total variation regularizers or robust $\ell_1$ penalties. For this reason, we combine the efficient discretizations of FE methods with the optimization algorithms developed in variational vision. The algorithms of Section 6 are efficient, easy to parallelize and export to GPU.

Finally, we exploit quad- and octree data structures to define our computational grid and efficiently locate pixels into finite elements and active basis. Ours contributions are thus three: (i) we introduce adaptive multi-resolution discretizations for a wide class of vision problems, (ii) we adapt the optimization algorithms of vision to finite-element discretizations and (iii) we use of efficient data structures with search and parallelization compatible with the pixel lattice to define the computational grid and locate pixels into finite elements.

2 Related Methods

The straightforward discretization of the pixel lattice explains its implicit nature in low-level vision and the dominion of finite-differences (FD) discretizations for variational models. The resulting optimizations pose a challenge because they have as many variables as pixels and require algorithms purposefully designed for vision functionals and GPU parallelization [1–4]. These techniques generalize well to FE discretizations [5] as long as the basis are continuous and fine elements tile the support of discontinuities, as our multiresolution discretization proposes.

FE techniques [6] are common in graphics [7–9] and computational sciences [10, 11] because they parametrize surfaces of arbitrary topology and accurately model discontinuities, cracks and shocks on them. Their accuracy results from refinement procedures that match the resolution of the discretization with the complexity of the model [12, 13]. The inverse nature of vision poses a challenge to these techniques because the topology, geometry, and differentiability of the functions are unknown and must be inferred from data. As a result FE techniques have been largely neglected in vision, even though surface reconstruction resorts to similar principles to handle large volumes. In particular [14–19] construct polynomial functions over octrees at multiple resolutions, but their optimizations are limited to least-squares techniques and their discretizations only adapted to the input pointcloud, not the reconstructed surface, and thus susceptible to reproducing its noise and artifacts. In a discrete setting, [20, 21] reconstruct surfaces by solving a labelling problem over non-uniform tetralizations, but their graph-cut formulation is prone to metrification errors and difficult to parallelize. We take a more principled approach and propose an adaptive FE discretization for both 2D and 3D problems that adapts to the complexity of the unknown solution, not the input data, and lead to efficient and easy-to-parallelize algorithms.

Only a few authors explicitly use FE techniques to solve variational problems in vision: [22, 23] first proposed non-uniform B-splines in vision, but their bases cannot be locally refined and their FE solvers are restricted to differen-
Adaptive Multiscale Discretizations for Vision

Tiable functionals; [24, 25] overcome these limitations with bases that suffer from expensive evaluation and refinement. Our work offers a simpler FE discretization well-suited to vision: it allows for variational models with non-differentiable functionals and FE grids with efficient search data structures for the pixel lattice.

Farther away are kernel and parametric models designed for specific applications. In image reconstruction, bases like wavelets [26] or curvelets [27] lead to similar coefficient-based parametrizations but depend on uniform discretizations of the pixel grid to solve the problem. Similarly, piecewise parametric flow and stereo models [28–35] estimate multiple parametric models on the pixel grid or a super-pixelization [36, 37] of it that is only adapted to the image, not to the solution. In image segmentation, narrow-band implementations of active contours implicitly create a non-uniform discretization of the domain but are limited to PDE models, slow descent algorithms, and good initializations. Compared to these techniques, we handle a broader range of applications and optimizations.

3 A Wide Range of Target Problems

Our discretization technique can be applied to any vision task that is formulated as a variational problem like (1). To illustrate its applicability, we choose 4 sample problems with different model and optimization complexities: image segmentation, optical flow, stereo, and depth fusion. These can all be framed as

$$\min_u \int_\Omega [\alpha f(u) + g(\nabla u)] dx,$$

where $\alpha$ is a scalar parameter, $f$ is a data-dependent functional, and $g$ is a regularization functional. The unknown function $u: \Omega \to \mathbb{R}$ is defined over the domain $\Omega \subset \mathbb{R}^d$ and restricted to the space of bounded variation $BV(\Omega)$.

Image segmentation partitions the domain of an image $I$ into homogeneous regions by finding the contours that border those regions. Equivalently, we can use an indicator function $u$ defined on the image domain $\Omega$ to label the regions. A popular model [38, 39] for binary segmentation partitions an image into homogeneous regions $R_1, R_2$ with mean intensities $\mu_1$ and $\mu_2$ by solving

$$\min_{0 \leq u \leq 1} \int_\Omega \alpha[(I-\mu_1)^2-(I-\mu_2)^2]u + |\nabla u| \quad R_1 = \{x|u(x) > \frac{1}{2}\}, \quad R_2 = \Omega \setminus R_1. \quad (2)$$

The optimization problem is convex but not differentiable, and the resolution necessary for the indicator function $u$ is lower than the resolution of the image. Indeed, the indicator function $u$ actually describes the 1-dimensional interface between the two image regions and, consequently, only needs a fine spatial resolution around this interface with a coarse discretization for the rest of $\Omega$.

In optical flow estimation, the unknown of the problem is a vector field $u = (u, v)$ that describes the apparent motion of pixels between two consecutive image frames $I_1, I_2$. A common formulation [40, 41] as an optimization problem

$$\min_u \int_\Omega \alpha|I_1(x) - I_2(x + u)|_\epsilon + |\nabla u| + |\nabla v| \quad |z|_\epsilon = \begin{cases} \frac{1}{2\epsilon^2}|z|^2 & \text{if } |z| \leq \epsilon \\ |z| - \frac{\epsilon}{2} & \text{otherwise} \end{cases},$$

The optimization problem is convex but not differentiable, and the resolution necessary for the indicator function $u$ is lower than the resolution of the image. Indeed, the indicator function $u$ actually describes the 1-dimensional interface between the two image regions and, consequently, only needs a fine spatial resolution around this interface with a coarse discretization for the rest of $\Omega$.
contains a data term $|I_1(x) - I_2(x + u)|_\epsilon$ that measures pixel correspondences with the Huber-loss function $|\cdot|_\epsilon$ and a regularizer $|\nabla u| + |\nabla v|$ that penalizes large deviations of the flow. The data term is usually substituted by a convex approximation $f_i(u) = |b + a \cdot u|_\epsilon$ that linearizes the image around the current flow estimate $u_l$ and the optimization is solved as a sequence of convex problems

$$u_{l+1} \leftarrow \min_u \int \alpha f_i(u) + |\nabla u| + |\nabla v| \quad f_i(u) = |I_1(x) - I_2(x + u_l) + \nabla I_2(x + u_l)(u - u_l)|_\epsilon$$

that halts when the non-linearized energy stops decreasing. The regularizer is a critical part of the model because it resolves the indetermination of the data term in flat regions. In other words, a flow estimate at pixel resolution is only possible assuming a smooth flow because the resolving power of the flow correspondence is lower than the resolution of the images. The pixel discretizations of FD methods thus misuse computational resources for the sake of simple implementations.

A similar model can be applied in stereo reconstruction to estimate the 3D geometry of a scene from a pair of images $I_1, I_2$ captured from different vantage points. To describe the scene geometry visible from the image pair, we use the image plane of $I_1$ to define a chart on the scene and adopt a depth parametrization of its visible surface. Such a depth parametrization conveniently confines the image data and the optimization variable to the image domain $\Omega$ and allows us to formulate the problem as the estimation of the depth map $u$

$$\min_u \int \alpha |I_2(\omega(u)) - I_1|_\epsilon + |\nabla u| \quad \omega = \pi \circ g_r \circ \pi^{-1}, \quad (4)$$

where the domain warping $\omega$ back-projects image pixels onto the surface and projects them back into the second camera displaced by the pose change $g_r$. The objective functional contains a data term that measures the photo-consistency between the reference image $I_1$ and the observed $I_2$ warped into the reference view by the depth estimate. The dependency on $u$ in the data term is again substituted by a convex approximation $f_i(u) = |b + au|_\epsilon$ that linearizes the warping around the current estimate $u_l$ and the original problem by the sequence

$$u_{l+1} \leftarrow \min_u \int \alpha f_i(u) + |\nabla u| \quad f_i(u) = |I_2(\omega(u_l)) - I_1 + \frac{\partial I_2(\omega(u))}{\partial u}(u - u_l)|_\epsilon.$$
Adaptive Multiscale Discretizations for Vision

over the volume containing the scene. The reconstruction is formulated as

$$\min_u \int_{\Omega} \alpha \sum_{i=1}^{L} h_i |u - b_i| + |\nabla u|,$$

(5)

where $b_i, h_i$ are the center and count of $i$-th histogram bin, see [42] for details. The total variation regularizer is necessary to resolve indeterminations in the histograms caused by either overlapping noisy depth maps or undersampled depth areas or holes. Our technique can be applied to other depth-fusion models [43–45, 21, 46] that solve large volumetric optimizations to reconstruct complex topologies and are computationally extremely expensive in uniform-grid discretizations that offer the same voxel resolution close to the the surface and far from it. Similar to image segmentation, a correct discretization only needs fine resolutions close to the surface with coarse discretizations of free space.

4 Non-uniform Piecewise Polynomial Bases for Vision

FE methods approximate a PDE solution by a linear combination

$$u(x) = \sum_{i}^{n} c_i \phi_i(x)$$

of $n$ basis functions, $\phi_i: \Omega \rightarrow \mathbb{R}$, to substitute the original PDE with a system of algebraic equations in the basis coefficients $c = (c_1, \ldots, c_n) \in \mathbb{R}^n$. To apply this paradigm to our minimization, we parametrize the solution with a linear combination of basis functions and approximate the minimization over the infinite-dimensional function space by a minimization over the space of coefficients $\mathbb{R}^n$.

The resolution and smoothness of the discretization is thus determined by the shape of the basis functions, while its computational cost by the evaluation of these functions. These two criteria guide our choice of discretization: first, the basis functions must have non-uniform resolution to represent both sharp image edges and smooth regions at a minimal cost; second, they must have analytic and integrable derivatives to solve (1); finally, they they must be compactly supported with few active functions per point for fast evaluation of $u$. We thus adopt a piecewise polynomial basis continuous over the elements of a polygonal domain tessellation and with minimal support and analytic integrable derivatives.

We introduce the main concepts and principles of linear FE discretizations with triangular elements, with the constructions over quad- and octrees sketched to avoid repetition. We refer the reader to [?] for detailed descriptions and proofs.

4.1 Triangular Linear Finite Elements

A linear function of two variables, $a_1 x_1 + a_2 x_2 + a_0$, is uniquely determined by its values on the three vertices of a non-degenerate triangle $K$. The set of these functions thus defines a functional space $\mathbb{P}(K)$ of dimension three over the triangle. Given two triangles $K_1, K_2$ with a common edge, we can build a piecewise linear function $u \in \mathbb{P}^1(K_i)$ that is uniquely determined by its values at the 4 vertices of $K_1, K_2$ and is continuous across triangles because its value along the shared edge depends only on its values on the shared vertices; the set of these functions
is functional space of dimension 4. Repeating this procedure for a triangulation
of \( \Omega \) with \( n \) vertices \( K \), we obtain a function that is linear on each triangle and
continuous in \( \Omega \). The set of these functions \( V_T = \{ u \in C(\Omega)|u|_K \in \mathbb{P}^1(K) \, \forall K \in \Omega \} \)
defines a space of continuous piecewise linear functions of dimension \( n \) that are
uniquely determined by their values on the triangulation vertices \( v_1, \ldots, v_n \).

A basis of \( V_T \) is thus formed by the set of unique continuous piecewise linear
function that verify \( \phi_i(v_j) = \delta_{ij} \), are compactly supported, and have integrable
derivatives discontinuous at edges. It is useful to construct these basis from a
reference element \( \hat{K} \) to decouple the analytic properties of the basis from the
geometric properties of the triangulation. The reference element is the triangle
with vertices \((0,0), (1,0), (0,1)\) in variables \((\xi, \eta)\) and basis functions
\[
\hat{\phi}_1 = 1 - \xi - \eta \quad \hat{\phi}_2 = \xi \quad \hat{\phi}_3 = \eta. \quad (6)
\]

We can map any triangle \( K \in \Omega \) with vertices \((x_1, y_1), (x_2, y_2), (x_3, y_3)\) bjectively
into \( \hat{K} \) with the affine transform
\[
\begin{pmatrix}
x \\
y
\end{pmatrix} = F_K(\xi, \eta) = \begin{pmatrix}
x_2 - x_1 & x_3 - x_1 \\
y_2 - y_1 & y_3 - y_1
\end{pmatrix} \begin{pmatrix}
\xi \\
y
\end{pmatrix} = B_K \begin{pmatrix}
\xi \\
y
\end{pmatrix} + \begin{pmatrix}
x_1 \\
y_1
\end{pmatrix}. \quad (7)
\]

and evaluate the \( i \)-th basis function of triangle \( K \), \( \phi^K_i \), and its derivatives by
\[
\phi^K_i = \hat{\phi}_i \circ F^{-1}_K \quad \nabla \phi^K_i = B_K^{-T}(\hat{\nabla} \hat{\phi}_i \circ F^{-1}_K), \quad (8)
\]
where \( \nabla \phi^K_i = B_K^{-T} \hat{\nabla} \hat{\phi}_i \) as \( \hat{\nabla} \hat{\phi}_i = \left[ \frac{\partial \hat{\phi}_i}{\partial \xi}, \frac{\partial \hat{\phi}_i}{\partial \eta} \right] \) is constant for linear elements.

### 4.2 Finite Elements Over Quad- and Octrees

When the elements of the tessellation are the rectangular cells of a quadtree, the
same argument shows that \( \mathbb{Q}^1 = \{ a_0 + a_1 \xi + a_2 \eta + a_3 \xi \eta | a_0, \ldots, a_3 \in \mathbb{R} \} \) defines a
polynomial space of dimension 4 where each element is uniquely determined by its
values on the vertices of the cell \( K \). In this case, we define a square reference element \( \hat{K} \) with vertices \((-1, -1), (1, -1), (1, 1), (-1, 1)\) and basis functions
\[
\hat{\phi}_1 = \frac{1}{4}(1 - \xi)(1 - \eta) \quad \hat{\phi}_2 = \frac{1}{4}(1 + \xi)(1 - \eta) \quad \hat{\phi}_3 = \frac{1}{4}(1 + \xi)(1 + \eta) \quad \hat{\phi}_4 = \frac{1}{4}(1 - \xi)(1 + \eta).
\]

We then map the reference space \( \mathbb{Q}^1 \) into any quadtree cell \( K \) by the affine transform
\[
\begin{pmatrix}
x \\
y
\end{pmatrix} = F_K(\xi, \eta) = \frac{1}{2} \begin{pmatrix}
\Delta_x & 0 \\
0 & \Delta_y
\end{pmatrix} \begin{pmatrix}
\xi \\
y
\end{pmatrix} + \begin{pmatrix}
x_c \\
y_c
\end{pmatrix} = B_K \begin{pmatrix}
\xi \\
y
\end{pmatrix} + \begin{pmatrix}
x_c \\
y_c
\end{pmatrix} \quad (9)
\]

that defines a polynomial space \( \mathbb{Q}^1(K) = \{ \hat{q} \circ F^{-1}_K | \hat{q} \in \mathbb{Q}^1 \} \) of dimension 4 over this cell and basis functions \( \phi^K_i = \hat{\phi}_i \circ F^{-1}_K \) satisfying (8). Gluing together the spaces over each
cell, we obtain a space of continuous piecewise polynomial functions \( V_Q \).

The same construction can be extended to three-dimensional functions by defining
a polynomial space with three reference variables \((\xi, \nu, \eta) \in [-1, 1]^3\), extending the
quadtree reference basis with terms \((1 \pm \nu)\), and mapping the resulting 8-dimensional
polynomial space to each cell in an octree tessellation of \( \Omega \) with the affine map
\[
\begin{pmatrix}
x \\
y \\
z
\end{pmatrix} = F_K(\xi, \eta, \nu) = \frac{1}{2} \begin{pmatrix}
\Delta_x & 0 & 0 \\
0 & \Delta_y & 0 \\
0 & 0 & \Delta_z
\end{pmatrix} \begin{pmatrix}
\xi \\
y \\
z
\end{pmatrix} + \begin{pmatrix}
x_c \\
y_c \\
z_c
\end{pmatrix} \quad (10)
\]
to construct the space of piecewise continuous polynomials over an octree \( V_O \).
4.3 Properties of Different Computational Grids

We propose different tessellations of the image plane, a Delaunay triangulation and a quadtree, to investigate their effects on the quality of the discretization. Delaunay triangulations are well suited to FE discretizations because they maximize the minimum angle of a triangulation and avoid the skinny triangles that make FE discretizations unstable. Triangular elements are more flexible representations because we can match image, depth or flow discontinuities with the triangulation’s edges where the solution is allowed discontinuous derivatives. Tessellations by quadtrees offer three computational advantages: first, the cost of evaluating the basis functions is cheap as matrix $B_K$ is diagonal; second, locating a point in a quadtree grid is trivial, and third, they are easy to parallelize. Their main drawback is the alignment of its edges with the image axis, which constrains the discontinuities in the derivatives of the solution to be axis-aligned but is no worse than the axis-alignment of discontinuities in FD discretizations.

5 Accurate and Fast Adaptive Multiresolution

Many problems in vision cannot be directly formulated as convex optimizations, like the optical flow (3) or stereo (4), and the minimization algorithms cannot guarantee convergence to a global minimum. Multiresolution then makes the optimization robust to irrelevant local minima by solving the problem at increasing resolutions and finding a minimum consistent with large scales. In FD discretizations, multiresolution resorts to a pyramid of uniform grid representations to initialize the minimization at finer grids with the solution from coarser grids and speed up convergence to a local minimum detected at larger scales; that is, the computational costs of the representation and minimization grow exponentially with the resolution of the grid regardless of the complexity of the solution. We propose a multiresolution representation where the computational costs of discretization and minimization only grows with resolution when the solution requires.

To this purpose, we locally increase the spatial resolution of our discretizations with a simple procedure: (i) subdivide the elements in regions that require higher spatial resolution, (ii) add the basis functions associated with the new elements to the representation and remove from it the functions from elements that have been refined. Following this principle, we initialize the discretization with a uniform FE tessellation $K^0$, basis function $V^0 = \{\phi_0, \ldots, \phi_n\}$ associated with its vertices and $c^0 = 0$, and alternate:

1. Solve $\min_{u \in \text{span} V^l} \int_\Omega \alpha f(u) + g(\nabla u)$ as in Section ?? initializing $u = \sum_{i \in V^{l-1}} c_i^{l-1} \phi_i^{l-1}$.  
2. Refine the elements in $K^l$ where the objective function evaluated at $u^l$ exceeds threshold $\delta$. This defines a new tessellation $K^{l+1}$ with basis functions $V^{l+1}$.

The process stops when $l = D$ or $K^{l+1} = K^l$ to prevent refinement when the resolution power of the model is exhausted. At each level, the optimization converges with only a few iterations because the algorithm is initialized close to the optimum and produces an estimate of the solution at the scale of the smallest FE cell. This procedure is more efficient than the multiresolution techniques of FD discretizations because the number of minimization variables – the number of basis functions – only increases if the representation of the solution at finner scales requires it: flat areas keep a coarse-element discretization while depth or flow discontinuities are resolved at fine scales by element refinement. Figure 3(q)-3(r) shows this approach.
There are two relevant refinement criteria for inverse problems: minimizing discrepancy between the observed and predicted images with the data term in (1), and accounting for smoothness priors by minimizing the objective functional (with regularizers, Bayesian approach). We adopt the Bayesian approach because it is more robust to image noise and model uncertainty, e.g., refinement based on the stereo or flow data terms would over-tesselate occluded regions unaccounted by the model.

The choice of threshold \( \delta \) and depth \( D \) defines the accuracy of the final solution and the complexity of the final discretization. We propose a simple technique to limit this complexity by refining a fixed percentage \( p \) of elements at each level and bound the number of elements in the final discretization, \( N \). In quadtrees \( N = 2^D (1 + 3p)^{D-D_0} \) is a function of the refinement percentage \( p \) and the depths \( D_0, D \) of the quadtree that match resolution of the uniform grid \( K^0 \) and image pixels. At each level, we then refine the \( p\% \) of elements with largest non-negative objective by setting \( \delta = \max(0, q) \), where \( q \) is the \( 1 - p \) percentile of the objective over the elements. This technique also applies to triangulations and octrees. In image segmentation, it is possible to define the discretization from statistics of image gradients or intensities.

6 Efficient FE Minimization for Vision Models

Finite-element discretizations are designed to solve PDEs by approximating them with an algebraic system of equations in the basis coefficients. Applied to our variational problem (1), FE solvers must first derive the Euler-Lagrange PDE that characterizes the minimum and are limited to differentiable functionals incompatible with most vision models. For this reason, we resort to the algorithms developed in vision for non-differentiable functionals and finite-difference discretizations.

To this purpose, we approximate the integral over \( \Omega \) in (1) with quadrature rules,

\[
\int_\Omega [\alpha f(u(x)) + g(\nabla u(x))] \, dx \approx \sum_{k=1}^m w_k [\alpha f(u(x_k)) + g(\nabla u(x_k))],
\]

where \( x_k \in \mathbb{R}^d \) is a quadrature point and \( w_k > 0 \) its associated quadrature weight. The accuracy of quadrature approximations is well studied for polynomials bases; experimentally, we have found that standard 3-, 4, and 8-point quadrature rules for triangular, quad- and octree cells produce the best trade-off between accuracy and speed.

By restricting \( u \) to the span of the basis \( \phi_1, \ldots, \phi_n \), the minimization (1) becomes

\[
\min_c \sum_{k=1}^m w_k [\alpha f(u(x_k)) + g(\nabla u(x_k))] \quad \text{s.t.}\ \left\{ \begin{array}{l} u(x_k) = \sum_i c_i \phi(x) \quad 1 \leq k \leq m \\
 \quad \nabla u(x_k) = \sum_i c_i \nabla \phi(x) \quad 1 \leq k \leq m \end{array} \right., \quad (11)
\]

The constraints in (11) are affine and can be written in matrix form by defining \( P_k = (\phi_1(x_k), \ldots, \phi_n(x_k)) \in \mathbb{R}^{1 \times n} \) and \( N_k = (\nabla \phi_1(x_k), \ldots, \nabla \phi_n(x_k)) \in \mathbb{R}^{d \times n} \) for each quadrature point and stacking them into sparse matrices \( P \in \mathbb{R}^{m \times n}, N \in \mathbb{R}^{dm \times n} \):}

\[
\min_{c, z} \sum_{k=1}^m \alpha w_k f(y_k) + \sum_{k=1}^m w_k g(z_k) \quad \text{s.t.}\ \left\{ \begin{array}{l} y = Pc \\
 z = Nc \end{array} \right., \quad (12)
\]

The problem now has the standard form of many convex minimization problems solved with splitting techniques. Among them, we adopt a primal-dual formulation and rewrite
solve (13) with algorithm [4] as the sequence of proximal problems and updates

\[ \lambda^{n+1} \leftarrow \min_{\lambda} \sigma F^*(\lambda) + \frac{1}{2} \| \lambda - \lambda^n - \sigma Pe^n \|^2 \]

(14)

\[ \nu^{n+1} \leftarrow \min_{\nu} \sigma G^*(\nu) + \frac{1}{2} \| \nu - \nu^{n+1} - \sigma N\hat{c}^n \|^2 \]

(15)

\[ c^{n+1} = c^n - \tau (N^* \nu^{n+1} + P^* \lambda^{n+1}) \]

(16)

\[ \hat{c}^n = 2c^n - c^{n-1}. \]

(17)

The minimization is efficient because we find simple closed-form solutions for (14)–(15), as detailed next. Our implementation will be publicly available on our webpage.

**Minimization in** \( \nu \) Let \( \hat{\nu} = \nu^n - \sigma N\hat{c}^n \) and recall that the conjugate of a norm is the indicator of its unit ball, the minimization in \( \nu \) for segmentation (2) and depth problems (4)-(5) is decoupled in each quadrature point and simplifies to

\[ \min_{\nu} \sum_{k=1}^{m} \sigma g^* (\nu_k) + \frac{1}{2} |\nu_k - \hat{\nu}_k|^2 \]

\[ = \min_{|\nu_k| < \hat{\nu}_k} \sum_{k=1}^{m} \frac{1}{2} |\nu_k - \hat{\nu}_k|^2 \Rightarrow \nu_k = \frac{w_k}{\max(\hat{\nu}_k, w_k)} \hat{\nu}_k. \]

The minimization in \( \nu \) for the flow problem (10) has this same form for each gradient of a flow component and is solved with this same update for each component.

**Minimization in** \( \lambda \): Let \( \hat{\lambda} = \lambda^n - \sigma Pe^n \), we solve the minimization in \( \lambda \) (14) through Moreau’s identity [47]: we set \( \hat{\lambda} = \lambda - \sigma y^* \) and find \( y^* \) solving the minimization

\[ \min_y F(y) + \frac{\sigma}{2} \| y - \frac{\hat{\lambda}}{\sigma} \|^2 = \min_y \sum_{k=1}^{m} \alpha w_k f(y_k) + 0.5 \sigma (y_k - \sigma^{-1} \hat{\lambda}_k)^2. \]

(18)

With this strategy, the minimization in \( y \) is equivalent to the proximal \( F \)-problem of the primal-dual algorithm encountered in finite-difference discretizations. As a result, we inherit their closed-form solutions and update the dual variable with:

**segmentation** \( \lambda_k = \alpha w_k [\mu_1^2 + \mu_2^2 + (\mu_2 - \mu_1) I(x_k)] \)

(19)

**stereo** \( \lambda_k = \begin{cases} 
\alpha w_k a_k & \text{if } a_k \hat{\lambda}_k + \sigma b_k > \alpha a_k^2 + \sigma \epsilon \\
-\alpha w_k a_k & \text{if } a_k \hat{\lambda}_k + \sigma b_k < -\alpha a_k^2 + \sigma \epsilon \\
\frac{a_k}{\rho_k} [a_k \hat{\lambda}_k + w_k \sigma b_k] & \text{otherwise}
\end{cases} \)

(20)

**optical flow** \( \lambda_k = \begin{cases} 
\alpha w_k a_k & \text{if } a_k^T \hat{\lambda}_k + \sigma b_k > \alpha |a_k|^2 + \sigma \epsilon \\
-\alpha w_k a_k & \text{if } a_k^T \hat{\lambda}_k + \sigma b_k < -\alpha |a_k|^2 + \sigma \epsilon \\
\hat{\lambda}_k - \sigma M_k (\hat{\lambda}_k - \frac{\alpha w_k b_k}{\epsilon} a_k) & \text{otherwise}
\end{cases} \)

(21)

**depth fusion** \( \lambda_k = \text{median}(\hat{\lambda}_k - \sigma b_1, ..., \hat{\lambda}_k - \sigma b_L, W_1 k, ..., W_L k) \)

(22)

where \( W_{ik} = \alpha w_k \sum_{j=1}^{i} h_i(x_k) - \sum_{j=i+1}^{L} h_i(x_k) \), the sub-index \( k \) indicates the components associated with the quadrature point \( x_k \), \( a \) and \( b \) are the variables in the linearization of stereo and flow problems, and the \( 2 \times 2 \) matrix \( M_k = [\sigma I_2 + \alpha w_k a_k a_k^T]^{-1} \) is inverted analytically in our implementation. We refer the reader to the supplementary material for the details of these standard derivations.
7 Experimental Results

To compare discretizations instead of models or optimization algorithms, we fix the models for segmentation, stereo, flow and depth fusion to the state-of-the-art models of Section 3 and the optimization technique to the popular algorithm [4]. We then investigate how our technique compares to the usual finite-difference discretization of the pixel grid with its multiresolution pyramid. This is the natural comparison because the pixel grid defines the finest resolution at which we can do inference from the input images and defines the gold standard in terms of accuracy, that we improve in speed.

Our discretization can only benefit from better models, optimization algorithms, and model parameters. We avoid fine-tuning $\alpha, \epsilon$ to achieve the best state-of-the-art results in each particular application because it is irrelevant for the comparison of the discretizations. We implement all the algorithms in python and run them on an Intel i7 at 2.6 GHz to directly compare speed, accuracy, and memory of the discretizations.

**Speed:** Our discretization with quadtree FEs is 3 times faster than a finite-differences for image segmentation, 2 – 3 times faster for stereo matching, and 3 – 4 times faster for optical-flow. With FE triangulations, the speed gains are similar, with a 2-, 3- and 3-fold speed increase for segmentation, stereo, and flow. The gain in speed from our discretization depends on two factors, the complexity of the minimization problem and the complexity of its solution. 1) In terms of minimization, the speed-up is larger in stereo and optical flow because we benefit from both an adaptive discretization and an an adaptive multiresolution –necessary to overcome the non-convexity of the problem independently of the discretization– while in image segmentation the speed gain is the result of only the adaptivity of the discretization. 2) In terms of the solution, the speed up is larger for solutions with large uniform areas that can be represented with large cells and smaller tessellations. For example, comparing the segmentations of the flower and the bird in Figures 3(m)-3(n), we observe how the complexity of the image is transferred into the tessellations and, as a result, on the number optimization variables and speed: $\times 3$ speedup for the flower with its uniform background and simple contours and $\times 2$ speedup for the bird in the tree branches.

**Accuracy** The gains in speed come with a minimal accuracy loss 0.01 –0.02 degrees of angular error in optical flow and 5 – 15% of the root-mean-square error in stereo for quadtree and triangular tessellations. Quantitative metrics, however, do not capture the quality of the solution as an image or depth map, which exhibit very particular spatial and smoothness properties, see for instance how our discretization of Tsukuba in Figure 5 has lower error than the FD one even though it is visually less appealing. Qualitatively the results of our discretizations are visually very similar to the FD gold standard: they capture all the image content and reproduce the discontinuities of the solution, with the only drawback that sharp transitions appear slightly shakier as they coincide with the edges of the FE tessellation at the finest resolution. Compare to lower resolution with same number of FE as pixels or time budget. The pyramid representation used in FD discretizations subsamples the images by 2 for a direct comparison with the FE quadtree representations. For consistency, the same number of scales is used in the FE refinement of triangulations and multiresolution solvers.

---

1 The optimization with FD remove the constraint $y = P c$ and solve the minimization in $F$ in primal variables; the complexity of FD and FE algorithms thus differ only in the linear operator $K$ defining the minimization constraints and the variables size. This is accounted for in the choice of the speed parameters $\tau = \sigma = \|K\|^{-0.5}$ and stopping when the relative error between consecutive iterates falls below $10^{-4}$. 
Memory and Tesselations The gain in memory depends on the tessellation, as our representation needs to store both the computational FE grid and the spline coefficients. FE Delaunay triangulations need to store the 2D points that define the vertices of the triangulation, while quadtrees can encode its simple tree structure with binary codes. Experimentally, quadtree and triangular tessellations result in comparable gains of speed because the refinement process of triangles is unable to align the discontinuities in depth, flow or segmentation with the edges of the triangles and reduce the number of FE cells. Indeed, aligning the edges of the triangles with discontinuities in u during refinement can lead to finer skewed triangles that compromise the stability of the FE discretization and, consequently, the accuracy of the estimated function. We have experimentally obtained the best performance by incrementally updating the Delaunay triangulation with the barycenter of each triangle that needs refinement and, as a result, we can only align a priori the estimated discontinuities with the edges of the initial tessellation $K^0$. While this is possible to some extent in image segmentation, it is unfeasible in stereo or optical-flow estimation, and quadtree tessellations in terms of memory and ease of parallelization.

Fig. 1: Optical flow performance 1(a)-1(b), time 1(c), and memory 1(d) as a function the number of elements $N$ (1 – 30% of the number of pixels). Dashed lines compare to the FD discretization.

(a) 2.91 RMS err. (b) quadtree 2(a) (c) 2.87 RMS err. (d) quadtree 2(c)

Fig. 2: Depth and flow with quadtree FE discretizations. Refining elements with large data terms (left) produces smaller tiles in flat and occluded regions than refining by objective values (right).

8 Conclusions

Of the basic components needed to solve a variational optimization problem on a computer, discretization has received the least attention despite its important role in achieving computationally efficiency. Discretizations that are adapted to the data (images) rather than the solution (disparity, range, or segmentation) fail to exploit the regularities of the latter and waste computational resources. By allocating resources adaptively, our discretization saves memory and time, which are critical to the adoption of variational methods in resource-constrained settings.

Our experiments in image segmentation, stereo, optical-flow, and depth-fusion illustrate the wide range of applicability of our technique its the computational benefits of estimating an adaptive discretization of the problem at the same time as its solution.
Fig. 3: Comparison of image segmentation with the standard finite-difference (FD) discretization (row 1) and our proposed finite-element (FE) discretizations with triangular (row 2) and quadrilateral elements (row 4). The contours of the segmentation are in blue and Figures 3(k)- 3(l) and 3(s)- 3(t) show how the tessellations of our FE grid adapt to the image structures and are able to achieve a speed-up of $2 - 3$. 
Fig. 4: Comparison of optical flow estimation with the standard finite-difference (FD) discretization (row 3) and our proposed finite-element (FE) discretizations with triangular (row 4) and quadrilateral elements (row 5). Our FE discretizations are 2 – 3 time faster than the FD approach with a minimal loss in angular-error (ae) accuracy.
Fig. 5: Depth estimation from rectified stereo pairs. Row 1-2: reference image and ground-truth displacement. Row 3: displacement estimated with standard finite-difference (FD) discretization. Row 4-5: displacement estimated with the proposed finite-element (FD) discretization with triangular (row 4) and quadrilateral (row 5) elements. Our FE discretizations are $3 - 6$ time faster than the FD approach for a small loss in accuracy of 5% in the average relative error $e_r$ with respect to the ground truth.
References


22. Elguedj, T., Rethore, J., Butari, A.: Isogeometric analysis for strain field measure-

23. Balzer, J., Morwald, T.: Isogeometric finite-elements methods and variational re-

Multiview Stereo. Int. Conf. 3D Vis. (2014)

25. Estellers, V., Scott, M.A., Tew, K., Soatto, S.: Robust Poisson Surface Recon-


27. Candes, E.J., Donoho, D.L.: Curvelets: A surprisingly effective nonadaptive rep-
resentation for objects with edges. DTIC Document (1999)

variable-order parametric models with local deformations (1996)

29. Cremers, D., Soatto, S.: Motion Competition: A Variational Approach to Piecewise

30. Smith, P., Drummond, T., Cipolla, R.: Layered motion segmentation and depth
2004) 479–494

31. Lei, C., Yang, Y.H.: Optical flow estimation on coarse-to-fine region-trees using
discrete optimization (2009)


33. Sun, D., Sudderth, E.B., Black, M.J.: Layered segmentation and optical flow esti-
1775

34. Hornácek, M., Besse, F., Kautz, J., Fitzgibbon, A., Rother, C.: Highly Overparam-
eterized Optical Flow Using PatchMatch Belief Propagation. In Fleet, D., Pajdla,
8691 of Lecture Notes in Computer Science. Springer International Publishing
(2014) 220–234

35. Yang, J., Li, H.: Dense , Accurate Optical Flow Estimation with Piecewise Para-


1632–1648

2009) 272–293

B (2009) 23–45

41. Sun, D., Roth, S., Black, M.J.: Secrets of optical flow estimation and their princi-