Adaptive Discretizations for Vision

V. Estellers and S. Soatto

UCLA Vision Lab

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. To adapt the complexity of the discretization to the solution, it is necessary to adopt finite-element techniques that match the resolution of piecewise polynomial bases to the resolving power of the variational model, but such techniques have been neglected in vision. To address this issue, we investigate the advantages and drawbacks of finite-element discretizations for variational models in vision, their multiresolution properties, and the optimization algorithms to resolve them. Our 2 and 3D experiments in image segmentation, optical flow, stereo, and depth fusion reveal the conditions where finite-element can outperform finite-difference discretizations by achieving significant computational savings with a minimal loss of accuracy.

1 Discretizations for Variational Problems in Vision

Many inference tasks in vision are formulated as variational optimization problems that estimate a function over the image plane. This variational approach is more robust and accurate than sparse feature techniques because it exploits all the information in the input images to estimate a solution at pixel resolution. With standard discretizations this produces optimization problems with as many variables as image pixels, require regularization to resolve undeterminations, and limit the deployment of variational models in resource-constrained systems.

Variational techniques solve vision problems by (i) designing cost functionals that describe the properties of the solution, (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, the choice of discretization has been largely overlooked in vision. Most algorithms are implemented on the pixel lattice, the simplest discretization of the image plane, with finite-difference (FD) approximations of spatial derivatives. This makes the implementation simple but inefficient in flat areas because the discretization is not adapted to the resolution power of the model or the complexity of the solution. To investigate the effects of FD discretizations on variational vision, we introduce alternative finite-element (FE) discretizations that match the complexity of the unknown solution, instead of the input data, and reduce the computational cost the optimization. The discretizations are adapted to the variational model and a multiscale construction that mimics the multiresolution optimiztion of variational models. Finally, we also investigate the integration of FE techniques with the optimization algorithms used in vision for non-differentiable functionals with Total Variation (TV) or ℓ_1 penalties. Our three contributions are: (i) we introduce adaptive multi-resolution discretizations for a wide class of vision problems, (ii) we adapt the optimization algorithms of vision to FE discretizations, (iii) we investigate the advantages and drawbacks of FE discretizations in vision.

2 Related Methods

FE techniques [23] are common in graphics and computational sciences [1, 2] because they parametrize surfaces of arbitrary topology and can model discontinuities. The inverse nature of vision poses a challenge to these techniques because the topology 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 techniques [3–5] resort to similar principles to handle large volumes by constructing basis functions over octrees at multiple resolutions. These surface reconstruction techniques, however, are limited to least-squares optimizations ver octree grids adapted to the input pointcloud, not the reconstructed surface, that reproduce ithe noise and artifacts of the input data. Our FE discretization adapts to the complexity of the unknown solution, not the input data, and seamlessly merge with the optimization algorithms [6, 7] used for variational 2 and 3D vision models.

Only a few authors explicitly use FE techniques in vision: [8] first proposed non-uniform B-splines in vision, but their bases cannot be locally refined and their solvers are restricted to differentiable functionals; [9, 10] overcome these limitations with bases that suffer from expensive evaluation and refinement. We investigate simpler FE discretizations with non-differentiable functionals.

Loosely related are kernel and parametric models designed for specific applications. Wavelets in image reconstruction lead to similar coefficient-based parametrizations but depend on uniform discretizations of the pixel grid. The flow and stereo methods [11–14] estimate multiple parametric models on the pixel grid or its super-pixelization [15] but are only adapted to the input image, not to the solution. In active contours segmentation, narrow-band implementations implicitly create a non-uniform discretization of the domain but are limited to PDE models and slow descent algorithms. Compared to these techniques, we handle a broader range of applications and optimizations.

3 A Wide Range of Target Problems

We target variational models of the form

$$\min_u \int_{\Omega} [\alpha f(u) + g(\nabla u)] dx, \tag{1}$$

where α 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 resctricted to the space of bounded variation $BV(\Omega)$. To illustrate its applicability, we choose 4 sample problems with different model and optimization complexities: segmentation, optical flow, stereo, and depth fusion.

Image segmentation partitions the domain of an image I into homogeneous regions. A popular model [16, 17] for binary segmentation finds the indicator function u that partitions I into two regions with mean intensities μ_1, μ_2 solving

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

and thresholding u to binarize the solution of the convex optimization problem. The resolution necessary for the indicator function u is lower than the resolution of the image because we are only interested on its zero-level set and, where we need a fine spatial resolution with a coarse discretization for the rest of Ω.

In optical flow estimation, the unknown of the problem is a vector field $\boldsymbol{u} = (u, v)$ that describes the apparent motion of pixels between two consecutive image frames I_1, I_2. A common formulation [18, 19] as an optimization problem

$$\min_u \int_\Omega \alpha|I_1(x) - I_2(x+\boldsymbol{u})|_\epsilon + |\nabla u| + |\nabla v| \quad |z|_\epsilon = \begin{cases} \frac{1}{2\epsilon}|z|^2 & \text{if } |z| \leq \epsilon \\ |z| - \frac{\epsilon}{2} & \text{otherwise} \end{cases}, \quad (3)$$

contains a data term $|I_1(x) - I_2(x+\boldsymbol{u})|_\epsilon$ that measures pixel correspondences with the Huber-loss function $|\cdot|_\epsilon$ and a regularizer $|\nabla u| + |\nabla v|$ that penalizes large gradients of the flow. The data term is usually substituted by a convex approximation $f_l(\boldsymbol{u}) = |b + a \cdot \boldsymbol{u}|_\epsilon$ that linearizes the image around the current flow estimate $\boldsymbol{u}_l$. The optimization is then solved as a sequence of convex problems

$$\boldsymbol{u}_{l+1} \leftarrow \min_{\boldsymbol{u}} \int \alpha f_l(\boldsymbol{u}) + |\nabla u| + |\nabla v| \quad f_l(\boldsymbol{u}) = |I_1(x) - I_2(x+\boldsymbol{u}_l) + \nabla I_2(x+\boldsymbol{u}_l)(\boldsymbol{u}-\boldsymbol{u}_l)|_\epsilon$$

The regularizer is a critical part of the model because it resolves the ambiguities of the data term in flat regions, where a flow estimate at pixel resolution relies on the regularizer to resolve the excessive degrees of freedom of pixel discretizations.

A similar model can be applied in stereo reconstruction to estimate the 3D geometry of a scene from a pair of images with different vantage points. A depth parametrization describes the scene geometry visible from the image pair and confines the optimization variable to the image domain Ω to formulates the problem as the estimation of the depth map u by solving the optimization

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

where the domain warping ω back-projects image pixels from I_1 onto the surface and then projects them onto I_2 with the relative camera pose change g_r. The dependency on u in the data term is again substituted by a convex approximation $f_l(u) = |b + au|_\epsilon$ that linearizes the warping around the current u_l and the original problem is solved by the sequence

$$\boldsymbol{u}_{l+1} \leftarrow \min_{\boldsymbol{u}} \int \alpha f_l(\boldsymbol{u}) + |\nabla u| \quad f_l(u) = |I_2(\omega(u_l)) - I_1 + \frac{\partial I_2(\omega(u))}{\partial u}(u - u_l)|_\epsilon.$$

The TV regularizer penalizes large variations in depth and is again the key to the resolution of the undetermined data term in flat regions. The smoothness imposed by the regularizer can also be obtained with an adaptive discretization.

The optimization problem (1) has also been used to estimate a single surface from a collection of partial and noisy measuremets [20–22]. We adopt the model of [22] to make no assumptions on the topology of the scene and estimate a signed-distance representation u of the surface from a collection of depth maps described by truncated distance functions and combined into a histogram of distance functions $h\colon \Omega \to \mathbb{R}^L$ over the volume. The reconstruction solves

$$\min_u \int_\Omega \alpha \sum_{i=1}^{L} h_i |u - b_i| + |\nabla u|, \tag{5}$$

where $b_i, h_{,i}$ are the center and count of i-th histogram bin [22]. The regularizer $|\nabla u|$, is necessary to resolve indeterminations in the histograms caused by either overlapping noisy depth maps or undersampled depth areas or holes. Similar to segmentation, a fine discretization is only necessary close to the surface.

4 Adaptive Piecewise Polynomial Basis

FE methods approximate a PDE solution by a linear combination $u(x) = \sum_i^n c_i \phi_i(x)$ of basis functions, $\phi_i\colon \Omega \to \mathbb{R}$, and substitute the original PDE with a system of algebraic equations in the basis coefficients $\boldsymbol{c} = [c_1, \dots, c_n]$. When we apply this paradigm to our problem, we approximate the minimization over the function space $BV(\Omega)$ by a minimization over the space of coefficients $\boldsymbol{c} \in \mathbb{R}^n$.

The resolution and smoothness of the discretization is determined by the shape of the basis functions, while its computational cost by their evaluation. This guides our choice of discretization: first, the basis functions must have non-uniform resolution to represent both sharp edges and smooth regions at a minimal cost; second, they must have integrable derivatives ∇u; and they must be compactly supported to evaluate u fast. We thus propose a piecewise polynomial basis over elements with minimal support and analytic derivatives.

We review the basic properties of FE discretizations [23] with triangular elements, and sketch their extension to quad- and octrees. Each tessellation offers different properties: Delaunay triangulations avoid the skinny triangles that make FE discretizations unstable and can match discontinuities in ∇u with the edges of the triangulation; while quad- and octree tesselations reduce the cost of evaluation but constrain the discontinuities of ∇u to be axis-aligned.

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 3 vertices of a non-degenerate triangle K. The set of these functions thus defines a functional space $\mathbb{P}^1(K)$ of dimension 3 over the triangle. Given two triangles K_1, K_2 with a common edge, a piecewise linear function

$u \in \mathbb{P}^1(K_i)$ uniquely determined by its values at the 4 vertices of the triangles is continuous across them; these functions thus form a functional space of dimension 4. Repeating this procedure for a triangulation of Ω with n vertices $\mathcal{K}$, we obtain a function that is linear on each triangle, continuous in Ω, and uniquely determined by its values on the triangulation vertices $v_1, \ldots, v_n$. These functions define a vector space V_T of continuous piecewise linear functions known as triangular linear finite elements.

The space V_T has dimension n and a basis formed by the set of unique continuous piecewise linear function that verify $\phi_i(v_j) = \delta_{ij}$. These functions are compactly supported, have integrable derivatives discontinuous at edges, and can be constructed from a reference element $\hat{K}$ that decouples the analytic properties of the basis from the geometry of the triangulation. The reference element is the triangle with vertices $(0,0), (1,0), (0,1)$ and basis functions

$$\hat{\phi}_1(\xi,\eta) = 1 - \xi - \eta \qquad \hat{\phi}_2(\xi,\eta) = \xi \qquad \hat{\phi}_3(\xi,\eta) = \eta. \tag{6}$$

We can map any triangle $K \in \mathcal{K}$ with vertices $(x_1,y_1), (x_2,y_2), (x_3,y_3)$ bijectively into $\hat{K}$ with the affine transform

$$\begin{bmatrix} x \\ y \end{bmatrix} = F_K(\xi,\eta) = \begin{pmatrix} x_2-x_1 & x_3-x_1 \\ y_2-y_1 & y_3-y_1 \end{pmatrix} \begin{bmatrix} \xi \\ \eta \end{bmatrix} + \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = B_K \begin{bmatrix} \xi \\ \eta \end{bmatrix} + \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}. \tag{7}$$

and evaluate the i-th basis function of triangle K, ϕ_i^K, and its derivatives by

$$\phi_i^K = \hat{\phi}_i \circ F_K^{-1} \qquad \nabla\phi_i^K = B_K^{-T}(\hat{\nabla}\hat{\phi}_i \circ F_K^{-1}), \tag{8}$$

where $\nabla\phi_i^K = B_K^{-T}\hat{\nabla}\hat{\phi}_i$ as $\hat{\nabla}\hat{\phi}_i = [\frac{\partial\hat{\phi}_i}{\partial\xi}, \frac{\partial\hat{\phi}_i}{\partial\eta}]$ 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. In this case, the reference element $\hat{K}$ is the square with vertices $(\pm 1, \pm 1)$ and basis functions

$$\hat{\phi}_1 = \frac{1}{4}(1-\xi)(1-\eta) \;\; \hat{\phi}_2 = \frac{1}{4}(1+\xi)(1-\eta) \;\; \hat{\phi}_3 = \frac{1}{4}(1+\xi)(1+\eta) \;\; \hat{\phi}_4 = \frac{1}{4}(1-\xi)(1+\eta).$$

The affine transform (9) maps the reference space $\mathbb{Q}^1$ into any quadtree cell K to define a polynomial space $\mathbb{Q}^1(K) = \{\hat{q} \circ F_K^{-1} | \hat{q} \in \mathbb{Q}^1\}$ of dimension 4 and basis $\phi_i^K = \hat{\phi}_i \circ F_K^{-1}$.

$$\begin{bmatrix} x \\ y \end{bmatrix} = F_K(\xi,\eta) = \frac{1}{2}\begin{pmatrix} \Delta_x & 0 \\ 0 & \Delta_y \end{pmatrix} \begin{bmatrix} \xi \\ \eta \end{bmatrix} + \begin{bmatrix} x_c \\ y_c \end{bmatrix} = B_K \begin{bmatrix} \xi \\ \eta \end{bmatrix} + \begin{bmatrix} x_c \\ y_c \end{bmatrix} \quad \begin{matrix} (x_c, y_c) & \text{cell center} \\ \Delta_x \times \Delta_y & \text{cell size} \end{matrix} \tag{9}$$

Gluing together the spaces over each cell, we obtain the space of continuous piecewise polynomial functions over a quadtree grid V_Q. This construction extends to 3D domains 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 polynomial space to each cell in an octree tessellation.

4.3 Adaptive Multiresolution

Many variational models in vision, e.g., optical flow or stereo, are not convex problems guaranteed to convergence to a global minimum and require multiresolution strategies to guide the optimization algorithm to a local minimum relevant at multiple scales. In FD discretizations, this is achieved by solving the optimization problem over a pyramid of uniform grids of increasing resolutions, initializing the minimization at finer grids with the solution from coarser grids and exponentially increasing the memory and computational costs of the discretization regardless of the complexity of the solution.

In FE discretizations, it is possible to only increase the cost of multiresolution representations with the resolution of the discretization when the solution requires it. To this purpose, we locally increase the spatial resolution of FE discretizations by subdividing only the elements that require higher spatial resolution and updating their basis functions. Following this principle, we initialize the discretization with a uniform FE tessellation $\mathcal{K}^0$, basis function $V^0 = \{\phi_1^0, \ldots, \phi_n^0\}$, $\boldsymbol{c}^0 = 0$, and alternate between:

1. Solve $\min_{u \in \operatorname{span} V^l} \int_\Omega \alpha f(u) + g(\nabla u)$ initializing $u = \sum_{i \in V^{l-1}} c_i^{l-1} \phi_i^{l-1}$.
2. Refine the elements in $\mathcal{K}^l$ where the objective function evaluated at u^l exceeds threshold δ to define a finer tessellation $\mathcal{K}^{l+1}$ and basis functions V^{l+1}.

The process stops when the resolution of the pixels is reached. At each level, the optimization converges with only a few iterations because the algorithm is initialized close to the optimum, as in multiresolution FD pyramids, but is more efficient because the number of minimization variables – basis functions– only increases if the representation of the solution at finner scales requires it: flat areas keep a coarse-element discretization while discontinuities are resolved at finer scales. See Fig. 1(m)-1(n) in our Experiments.

The refinement criteria tries to minimize the objective function by increasing the resolution of the discretization in areas where the data term or the regularizer are large and the estimated solution violates the model assumptions. We set threshold δ to fix the number of elements N in the final discretization when we refine the same percentage p at each level. In quadtrees, for instance, $N = 2^{D_0}(1+3p)^{D-D_0}$ is a function of the depths D_0, D of quadtrees matching the resolution of the initial tessellation and pixel grids, and δ is the $1-p$ percentile of the objective over the elements. This technique also applies to triangulations and octrees and can be improved by application-specific criteria that match statistics of image gradients or voxel occupation in depth fusion.

5 Minimization

FE 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 algorithms developed in vision for non-differentiable functionals.

To this purpose, we use quadrature to approximate the integral with the sum

$$\int_\Omega [\alpha f(u(x)) + g(\nabla u(x))] \, \mathrm{d}x \approx \sum_{k=1}^{m} w_k \left[\alpha f(u(x_k)) + g(\nabla u(x_k))\right] \tag{10}$$

where $x_k \in \mathbb{R}^d$ are quadrature points and $w_k > 0$ quadrature weights. As the objective functional only depends on the value of u at quadrature points, it is convenient to define the variable $\boldsymbol{u} = [u(x_1), \ldots, u(x_m)] = P\boldsymbol{c} \in \mathbb{R}^m$, where the k-th row of matrix P is $[\phi_1(x_k) \ldots \phi_n(x_k)] \in \mathbb{R}^{1\times n}$. Similarly, the gradient of u at the quadrature points satisfies $[\nabla u(x_1), \ldots, \nabla u(x_m)] = N\boldsymbol{c} \in \mathbb{R}^m$, where the k-th row of matrix $N \in \mathbb{R}^{dm\times n}$ is $[\nabla\phi_1(x_k) \ldots \nabla\phi_n(x_k)]$, and the optimization problem

$$\min_u \underbrace{\sum_{k=1}^m \alpha w_k f(u(x_k))}_{F(P\boldsymbol{c})} + \underbrace{\sum_{k=1}^m w_k g(\nabla u(x_k))}_{G(N\boldsymbol{c})} = \min_{\boldsymbol{c}}\ F(P\boldsymbol{c}) + G(N\boldsymbol{c}). \tag{11}$$

has the standard form of many convex minimization problems solved with splitting techniques. Among them, we adopt a primal-dual formulation and rewrite (11) as the saddle-point problem with dual variables $\boldsymbol{\lambda} \in \mathbb{R}^m,\ \boldsymbol{\nu} \in \mathbb{R}^{dm}$

$$\max_{\boldsymbol{\lambda},\ \boldsymbol{\nu}}\ \min_{\boldsymbol{c}}\ -F^*(\boldsymbol{\lambda}) - G^*(\boldsymbol{\nu}) + \langle \boldsymbol{\lambda}, P\boldsymbol{c}\rangle + \langle \boldsymbol{\nu}, N\boldsymbol{c}\rangle, \tag{12}$$

where * denotes the convex conjugate and $\langle\cdot,\cdot\rangle$ the Euclidean scalar product. We then solve (12) with algorithm [7] as the sequence of proximal problems and updates

$$\boldsymbol{\lambda}^{n+1} \leftarrow \min_{\boldsymbol{\lambda}}\ \sigma F^*(\boldsymbol{\lambda}) + \frac{1}{2}\|\boldsymbol{\lambda} - (\boldsymbol{\lambda}^n + \sigma P\bar{\boldsymbol{c}}^n)\|^2 \tag{13}$$

$$\boldsymbol{\nu}^{n+1} \leftarrow \min_{\boldsymbol{\nu}}\ \sigma G^*(\boldsymbol{\nu}) + \frac{1}{2}\|\boldsymbol{\nu} - (\boldsymbol{\nu}^{n+1} + \sigma N\bar{\boldsymbol{c}}^n)\|^2 \tag{14}$$

$$\boldsymbol{c}^{n+1} = \boldsymbol{c}^n - \tau(N^*\boldsymbol{\nu}^{n+1} + P^*\boldsymbol{\lambda}^{n+1}) \tag{15}$$

$$\bar{\boldsymbol{c}}^n = 2\boldsymbol{c}^n - \boldsymbol{c}^{n-1}. \tag{16}$$

The minimization is efficient because we find simple closed-form solutions for (13)–(14) similar to the ones in FD discretizations that now parallelize over quadrature points instead of pixels. Experiments show that 3-, 4, and 8-point quadrature over triangular, quad- and octree elements produce accurate approximations, see Fig. 6.

Minimization in $\boldsymbol{\lambda}$: We solve the proximal problem that updates λ through Moreau's identity, solving the minimization problem

$$\boldsymbol{y}^\star =\ \arg\min_{\boldsymbol{y}}\ F(\boldsymbol{y}) + \frac{\sigma}{2}\|\boldsymbol{y} - \frac{\hat{\boldsymbol{\lambda}}}{\sigma}\|^2 = \min_{\boldsymbol{y}}\ \sum_{k=1}^m \alpha w_k f(y_k) + 0.5\sigma(y_k - \frac{\hat{\lambda}_k}{\sigma})^2. \tag{17}$$

and updating the dual variable $\boldsymbol{\lambda} = \hat{\boldsymbol{\lambda}} - \sigma\boldsymbol{y}^\star$ accordingly. The objective function in the minimization in $\boldsymbol{y}$ is decoupled in of its components y_k with a different term in the sum and, consequently, the minimization is solved by independently minimizing each on of these terms, that is,

$$\min_{y_k}\ \alpha w_k f(y_k) + 0.5\sigma(y_k - \frac{\hat{\lambda}_k}{\sigma})^2. \tag{18}$$

As the function f is convex, the minimizers are the zeros of the subgradient of the objective function with respect to each y_k. The resulting equations, after standard algebraic manipulations, directly provide the closed-form solutions and the following dual updates:

$$\text{segmentation } \lambda_k = \alpha w_k[\mu_1^2 + \mu_2^2 + (\mu_2 - \mu_1)I(x_k)] \tag{19}$$

$$\text{stereo } \lambda_k = \begin{cases} \alpha w_k a_k & \text{if } \ a_k\hat{\lambda}_k + \sigma b_k > \alpha a_k^2 + \sigma\epsilon \\ -\alpha w_k a_k & \text{if } a_k\hat{\lambda}_k + \sigma b_k < -(\alpha a_k^2 + \sigma\epsilon) \\ \frac{\alpha a_k}{\rho_k}[a_k\hat{\lambda}_k + w_k\sigma b_k] & \text{otherwise} \end{cases} \tag{20}$$

$$\text{optical flow } \boldsymbol{\lambda}_k = \begin{cases} \alpha w_k \boldsymbol{a}_k & \text{if } \ \boldsymbol{a}_k^T\hat{\boldsymbol{\lambda}}_k + \sigma b_k > \alpha|\boldsymbol{a}_k|^2 + \sigma\epsilon \\ -\alpha w_k \boldsymbol{a}_k & \text{if } \boldsymbol{a}_k^T\hat{\boldsymbol{\lambda}}_k + \sigma b_k < -(\alpha|\boldsymbol{a}_k|^2 + \sigma\epsilon) \\ \hat{\boldsymbol{\lambda}}_k - \sigma M_k(\hat{\boldsymbol{\lambda}}_k - \frac{\alpha w_k b_k}{\epsilon}\boldsymbol{a}_k) & \text{otherwise} \end{cases} \tag{21}$$

$$\text{depth fusion } \lambda_k = \text{median}(\hat{\lambda}_k - \sigma b_1, ..., \hat{\lambda}_k - \sigma b_L, W_{1k}, ..., W_{Lk}) \tag{22}$$

where $W_{ik} = \alpha w_k[\sum_{j=1}^{i} h_i(x_k) - h_i(x_k)]$, the sub-index k indicates the components associated with the quadrature point x_k, $\boldsymbol{a}$ and b are the variables in the linearization of stereo and flow problems, and the 2×2 matrix $M_k = [\sigma I_2 + \frac{\alpha w_k}{\epsilon}\boldsymbol{a}_k\boldsymbol{a}_k^T]^{-1}$ is inverted analytically in our implementation. These standard derivations of these updates are provided next:

Image segmentation In image segmentation, the minimization in $\boldsymbol{y}$ reads

$$\min_{\boldsymbol{y_k}\in\mathbb{R}} \ \alpha w_k[\mu_1^2 + \mu_2^2 + (\mu_2 - \mu_1)I(x_k)y_k + 0.5\sigma(y_k - \frac{\hat{\lambda}_k}{\sigma})^2. \tag{23}$$

The optimality conditions of (23) are obtained by computing the zeros of the derivative of the objective, that is

$$\boldsymbol{y}_k = \frac{1}{\sigma}[\hat{\lambda}_k - \alpha w_k[\mu_1^2 + \mu_2^2 + (\mu_2 - \mu_1)I(x_k)]. \tag{24}$$

The dual variable $\boldsymbol{\lambda}_k = \hat{\boldsymbol{\lambda}}_k - \sigma\boldsymbol{y}_k^\star$ is thus updated by

$$\lambda_k = \alpha w_k[\mu_1^2 + \mu_2^2 + (\mu_2 - \mu_1)I(x_k) \tag{25}$$

as reported in Equation (19) of the manuscript.

Flow and Stereo In optical flow, the minimization in $\boldsymbol{y}$ reads

$$\min_{\boldsymbol{y_k}\in\mathbb{R}^2} \ \alpha w_k|b_k + \boldsymbol{a}_k^T\boldsymbol{y_k}|_\epsilon + 0.5\sigma(y_k - \frac{\hat{\lambda}_k}{\sigma})^2. \tag{26}$$

The optimality conditions of (26) are obtained by computing the zeros of the derivative of the objective. With the Huber norm, we analyze two cases.

If $|b_k + \boldsymbol{a}_k^T\boldsymbol{y}_k| > \epsilon$, the optimality conditions

$$\alpha w_k \,\text{sign}(b_k + \boldsymbol{a}_k^T\boldsymbol{y}_k)\boldsymbol{a}_k + \sigma\boldsymbol{y}_k - \hat{\lambda}_k = 0 \tag{27}$$

are solved by analyzing the two possible values of $\text{sign}(b_k + \boldsymbol{a}_k^T\boldsymbol{y}_k)$ in $|b_k + \boldsymbol{a}_k^T\boldsymbol{y}_k| > \epsilon$, that is,

$$\boldsymbol{y}_k = \begin{cases} \hat{\lambda}_k - \alpha w_k\boldsymbol{a}_k & \text{if } \ \boldsymbol{a}_k^T\hat{\boldsymbol{\lambda}}_k + \sigma b_k > \alpha|\boldsymbol{a}_k|^2 + \sigma\epsilon \\ \hat{\lambda}_k + \alpha w_k\boldsymbol{a}_k & \text{if } \ \boldsymbol{a}_k^T\hat{\boldsymbol{\lambda}}_k + \sigma b_k < -[\alpha|\boldsymbol{a}_k|^2 + \sigma\epsilon]. \end{cases} \tag{28}$$

If $|b_k + \boldsymbol{a}_k^T \boldsymbol{y}_k| < \epsilon$, the optimality conditions results in a 2×2 system of equations

$$\frac{1}{\epsilon} \alpha w_k \boldsymbol{a}_k (b_k + \boldsymbol{a}_k^T \boldsymbol{y}_k) + \sigma \boldsymbol{y}_k - \hat{\lambda}_k = 0 \tag{29}$$

$$[\frac{1}{\epsilon} \alpha w_k \boldsymbol{a}_k \boldsymbol{a}_k^T + \sigma I_2] \boldsymbol{y}_k = \hat{\lambda}_k - \frac{1}{\epsilon} \alpha w_k b_k \boldsymbol{a}_k \tag{30}$$

and the condition $|b_k + \boldsymbol{a}_k^T \boldsymbol{y}_k| = \epsilon$ becomes $|\boldsymbol{a}_k^T \hat{\boldsymbol{\lambda}}_k + \sigma b_k| < \alpha \boldsymbol{a}_k^T \boldsymbol{a}_k + \sigma \epsilon$.

Combining both cases, we obtain the closed-form solution of (26)

$$\boldsymbol{y}_k^\star = \begin{cases} \hat{\lambda}_k - \alpha w_k \boldsymbol{a}_k & \text{if } \ \boldsymbol{a}_k^T \hat{\boldsymbol{\lambda}}_k + \sigma b_k > \alpha |\boldsymbol{a}_k|^2 + \sigma \epsilon \\ \hat{\lambda}_k + \alpha w_k \boldsymbol{a}_k & \text{if } \ \boldsymbol{a}_k^T \hat{\boldsymbol{\lambda}}_k + \sigma b_k < -[\alpha |\boldsymbol{a}_k|^2 + \sigma \epsilon] \\ \boldsymbol{y}_k = M_k^{-1} \hat{\lambda}_k - \frac{1}{\epsilon} \alpha w_k b_k \boldsymbol{a}_k & \text{otherwise} \end{cases}, \tag{31}$$

where $M_k = \frac{1}{\epsilon} \alpha w_k \boldsymbol{a}_k \boldsymbol{a}_k^T + \sigma I_2$. This results on the following update for the dual variable $\boldsymbol{\lambda}_k = \hat{\boldsymbol{\lambda}}_k - \sigma \boldsymbol{y}_k^\star$

$$\boldsymbol{\lambda}_k = \begin{cases} \alpha w_k \boldsymbol{a}_k & \text{if } \ \boldsymbol{a}_k^T \hat{\boldsymbol{\lambda}}_k + \sigma b_k > \alpha |\boldsymbol{a}_k|^2 + \sigma \epsilon \\ -\alpha w_k \boldsymbol{a}_k & \text{if } \boldsymbol{a}_k^T \hat{\boldsymbol{\lambda}}_k + \sigma b_k < -(\alpha |\boldsymbol{a}_k|^2 + \sigma \epsilon) \\ \hat{\boldsymbol{\lambda}}_k - \sigma M_k (\hat{\boldsymbol{\lambda}}_k - \frac{\alpha w_k b_k}{\epsilon} \boldsymbol{a}_k) & \text{otherwise} \end{cases}, \tag{32}$$

as reported in Equation (21) of the manuscript. The closed-form expressions for stereo are obtained analogously, susbtituting the linearization vector $\boldsymbol{a}_k \in \mathbb{R}^2$ by the corresponding linearization scalar $\boldsymbol{a}_k \in \mathbb{R}$ and the inverse matrix by an inverse scalar value.

Depth fusion In depth fusion, the minimization in $\boldsymbol{y}$ reads

$$\min_{\boldsymbol{y_k} \in \mathbb{R}} \ \alpha w_k \sum_{i=1}^{L} h_i |y_k - b_i| + 0.5 \sigma (y_k - \frac{\hat{\lambda}_k}{\sigma})^2. \tag{33}$$

Its optimality conditions are again obtained by computing the zeros of the sub-derivative of the objective as follows

$$\alpha w_k \sum_{i=1}^{L} h_i \operatorname{sign}(y_k - b_i) + \sigma y_k - \hat{\lambda}_k = 0 \tag{34}$$

$$y_k = \frac{1}{\sigma} (\hat{\lambda}_k - \alpha w_k \sum_{i=1}^{L} h_i \operatorname{sign}(y_k - b_i)). \tag{35}$$

Taking into consideration that the bin centers $b_1, \ldots, b_L$ are sorted, we can analyze the possible values of $\operatorname{sign}(y_k - b_i)$ to obtain the following closed-form expression:

$$y_k = \operatorname{median}(b_1, ..., b_L, W_{1k}, ..., W_{Lk}) \qquad W_{ik} = \frac{\hat{\lambda}_k}{\sigma} - \frac{\alpha w_k}{\sigma} [\sum_{j=1}^{i} h_i(x_k) - h_i(x_k)]. \tag{36}$$

The closed-form update (22) for the dual variable $\boldsymbol{\lambda}_k = \hat{\boldsymbol{\lambda}}_k - \sigma \boldsymbol{y}_k^\star$ follows immediately:

$$\lambda_k = \operatorname{median}(\hat{\lambda}_k - \sigma b_1, ..., \hat{\lambda}_k - \sigma b_L, W_{1k}, ..., W_{Lk}). \tag{37}$$

Minimization in $\boldsymbol{\nu}$ Let $\hat{\boldsymbol{\nu}} = \boldsymbol{\nu}^n + \sigma N \bar{\boldsymbol{c}}^n$ and recall that the conjugate of a norm is the indicator of its dual unit ball, the minimization in ν for (2), (4)-(5) and each flow component in (3) decouples in each quadrature point and simplifies to

$$\min_{\boldsymbol{\nu}} \sum_{k=1}^{m} \sigma g^*(\nu_k) + \frac{1}{2}|\nu_k - \hat{\nu_k}|^2 = \min_{|\nu_k|<w_k} \sum_{k=1}^{m} \frac{1}{2}|\nu_k - \hat{\nu_k}|^2 \;\Rightarrow\; \nu_k = \frac{w_k}{\max(\; w_k, |\hat{\nu}_k|\;)}\hat{\nu}_k.$$

Gradient-based Minimization Algorithms We have also compared the discretizations with gradient-based optimization algorithms. To this purpose, we have smoothed the objective functions with Moreau-Yosida regularization of TV and ℓ_1. This leads to a Huber penalty

$$h(u) = |u|_\epsilon = \begin{cases} \frac{1}{2\epsilon}u^2 & |u| < \epsilon \\ |u| - \frac{\epsilon}{2} & |u| \geq \epsilon \end{cases}, \tag{38}$$

for the ℓ_1 norm of depth-fusion, and avoids the staircase artifacts of TV regularization in flow, stereo, and depth fusion with the regularizer

$$g(\nabla u) = \begin{cases} \frac{1}{2\epsilon}|\nabla u|^2 & |\nabla u| < \epsilon \\ |\nabla u| - \frac{\epsilon}{2} & |\nabla u| \geq \epsilon \end{cases}, \tag{39}$$

where ϵ is the regularization parameter. It is not possible to apply this regularization to the TV regularizer of the segmentation model (2) without loosing the optimality of the thresholded solution [16].

As the objective functional is now differentiable, we can apply gradient-based descent algorithms that only require the gradient of the functionals. We will exemplify it with the stereo problem:

$$\min_u \int \alpha|b + au|_\epsilon + g(\nabla u) \tag{40}$$

$$\tag{41}$$

that we discretize with quadrature approximations into

$$\min_{\boldsymbol{u}} \alpha \underbrace{\sum_{k=1}^{m} w_k h(b_k + a_k u_k)}_{F_1(P\boldsymbol{c})} + \underbrace{\sum_{k=1}^{m} w_k g(\nabla u_k)}_{F_2(N\boldsymbol{c})} \tag{42}$$

$$\min_{\boldsymbol{c}} \;\alpha F_1(P\boldsymbol{c}) + F_2(N\boldsymbol{c}) = E(c), \tag{43}$$

where

$$F_1(\boldsymbol{y}) = \sum_{k=1}^{m} w_k h(b_k + a_k y_k) \qquad k = 1, \ldots, m \tag{44}$$

$$F_2(\boldsymbol{z}) = \sum_{k=1}^{m} w_k g(z_k) \qquad k = 1, \ldots, m, \tag{45}$$

and $z_k \in \mathbb{R}^2$ is the block component of $N\boldsymbol{c}$ that discretizes ∇u at the k-th quadrature point. By the chain rule, we have $\nabla E = \alpha P^T \nabla F_1(P\boldsymbol{c}) + N^T \nabla F_2(N\boldsymbol{c})$ and we only need to compute $\nabla F_1, \nabla F_2$. These are trivial and have k-th components

$$[\nabla F_1(\boldsymbol{y})]_k = w_k h'(b_k + a_k y_k) a_k \quad \text{where} \quad h'(u) = \begin{cases} \frac{u}{\epsilon} & |u| < \epsilon \\ \operatorname{sign}(u) & |u| \geq \epsilon \end{cases} \tag{46}$$

and

$$[\nabla F_2(\boldsymbol{z})]_k = w_k \nabla g(z_k) \quad \text{where} \quad \nabla g(z) = \begin{cases} \frac{1}{\epsilon} z & |z| < \epsilon \\ \frac{z}{|z|} & |z| \geq \epsilon \end{cases}. \tag{47}$$

As a result, the computational complexity of computing the gradient is similar to the computation of each primal-dual iteration: it requires the computation of gradients of F_1, F_2, or the F proximal updates, that decouple in each quadrature point and matrix-vector multiplications with N, P, N^T and P^T. Although gradient descent leads to slower convergence than the primal-dual approach, the quasi Newton L-BFGS [24] has comparable performance to it. We decided to opt for the primal-dual approach in our experiments because it is not restricted to differentiable functionals.

6 Experimental Results

Our experiments are designed to compare discretizations, not variational models or optimization algorithms. To this purpose, we implement the segmentation, stereo, flow and depth-fusion models of Section 3 and the optimization algorithm [7] with our FE and the standard FD discretizations and multiresolution pyramids. Models and algorithm are chosen for their prevalence, not to top the state-of-the-art of each application.

Since the pixel grid defines the finest resolution to do inference from the input images, the performance of FD discretizations at pixel grid defines the gold standard in terms of accuracy. We use standard metrics in each application: angular error (ae) in flow, relative root mean-square (rms) error in stereo, and visual inspection in segmentation and depth fusion. To compare time performance, we implement all the algorithms in python and run them on an Intel *i7* at 2.6 with the same number of warps, iterations, and image pyramids in FD discretizations and FE refinement levels. We investigate the speed, accuracy, and limitations of each discretization in Figures 1-7.

Speed: Our discretization with quadtree FEs is 3 times faster than a FD for image segmentation, $0.6-2$ times faster for stereo matching, and $3-5$ times faster for optical-flow, and $4-5$ faster for depth fusion. With FE triangulations, the speed gains are similar. The gain in speed from our discretization depends on the complexity of both the minimization problem and solution. 1) 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. Compare the segmentations of the flower and the bird in Figures 1(i)-1(j), where the complexity of the image is transferred into the tessellation and computational cost. 2) In terms of minimization, the speed-up is lower for image segmentation and depth fusion because they only benefit from the adaptivity of discretization, while flow and stereo benefit from an adaptive discretization and an adaptive multiresolution to amortize the overheads of constructing FE tessellations.

Accuracy The gains in speed come with a loss in accuracy: $0.01-0.02°$ of angular error in optical flow, below 3% of relative root-mean-square error in stereo, and small perturbations of the contours in image segmentation and fusion along the tesellation edges. The results of FE discretizations are thus qualitatively similar to the FD ones but tend to blurr sharp transitions in u; this causes small artifacts in the estimated flow and stereo results of Fig 2-3 but has minimal impact on the segmentation of Fig. 1 and fusion of depth maps in Fig. 5 as u is post-processed to extract its zero-level set.

Quadrature is accurate for the regularizer because we constrain the solution to the span of a spline basis, but is only a valid approximation for the integration of the data term if large FE cells correspond to flat regions of in the image plane or volume, while discontinuities in u are supported by fine elements. Figure 6 compares the evaluation of the data term over all the image pixels with different quadrature approximations: using quadrature to both compute the integral in the optimization and refinement, using quadrature only on the optimization but not on the refinement, and evaluating the data term over all the pixels for both the optimization and refinement. The experiment shows that quadrature speeds optimization with a small accuracy loss and the refinement process partitions elements with large objective values, caused by either large errors in the forward model (data term) or sharp transitions in u (regularizer).

Trade-Offs FE discretizations reduces the computational cost of optimization as they reduce the number of variables but introduce a data-structure overhead by tessellating of the image domain into elements. When the information in the data term is uniformly distributed, like in flow or stero problems, FE discretizations results in larger accuracy losses or reduced speed ups because the optimal tessellation approaches a uniform grid. For problems where the input information clusters in small areas, like the image gradients in segmentation or surface patches in depth fusion, the speed up is large and compensates for any loss of accuracy that can be corrected in the post-processing that extracts the zero-level set of u. This property is of particular interest for 3D reconstruction where large uniform grids are unfeasible.

The memory requirements of FE discretizations depend on the tessellation. Delaunay triangulations need to store the 2D points and edges of the triangulation, while quadtrees can encode its simple tree structure with binary codes. Experimentally, the refinement process of triangles is unable to align the discontinuities in u to the edges of the triangles without creating skewer triangles that compromise the stability of the FE discretization as we must estimate jointly the optimal u and its discretization.

Finally, Figure 7 shows the effect of the number of final elements in the accuracy and time performance of our discretization. Unsurprisingly, as the number of finite-elements in the final discretization increases, as a percentage of the number of pixels, the accuracy of the discretization improves while its running time degrades. In general, the accuracy of the discretization stops increasing after the number of elements reaches a threshold around 15-20% of the image size because the resolving power of the variational model limits the accuracy of the estimation irrespective of the discretization; as a result, we set 20% of the number of pixels as the upper-bound on the number of elements in our adaptive discretization for all our experiments. In some cases, as illustrated by the *Hydrangea* flow of Figure 7(a), the accuracy of the discretization stops increasing earlier (7% in this case) because the complexity of the solution is relatively low. In both cases, the number of final elements should not exceed these thresholds to avoid increasing the computational cost with no gain in accuracy.

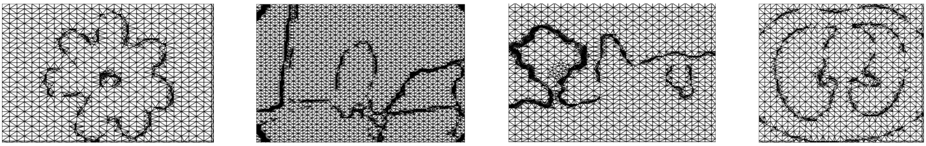
7 Conclusions

The discretization of variational problems has received little attention in vision as the pixel lattice offers an obvious computational grid that, if not efficient, is simple and accurate. As a result, when accuracy and not speed or memory is the bottleneck, finite-difference are better discretizations for variational models. Finite element discretizations suit problems where the information in the input data is spatially sparse as they allow for multiresolution optimizations that allocates resources adaptively to match the complexity of the solution. This is important for 3D problems like surface reconstruction or image segmentation, but offers limited benefits for 2D problems like optical flow or stereo that are described by matching pixels of uniform lattices.

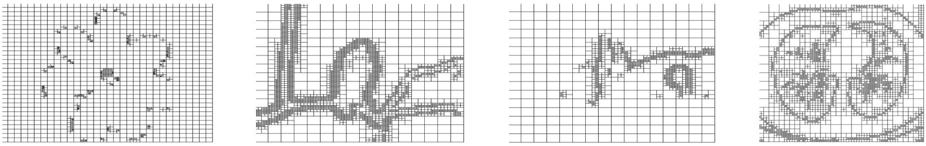
(a) FD 1.9 s (b) FD 3.6 s (c) FD 4.4 s (d) FD 3.0 s

(e) tri-FE 0.4 s (f) tri-FE 2.5 s (g) tri-FE 1.4 s (h) tri-FE 1.0 s

(i) quad-FE 0.6 s (j) quad-FE 1.6 s (k) quad-FE 0.8 s (l) quadFE 1.7

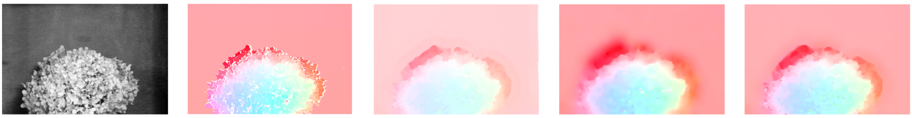


(m) triangulation (n) triangulation (o) triangulation (p) triangulation



(q) quadtree (r) quadtree (s) quadtree (t) quadtree

Fig. 1: Comparison of image segmentation with the FD and FE discretizations with triangular (tri-FE) and rectangular (quad-FE) elements. The contours of the segmentation are in blue. The tessellations of FE discretizations in row 4 and 5 adapt to the image structures and speed-up ×2–×3 the optimization.



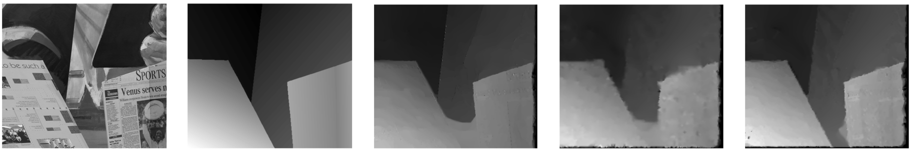
(a) I_1 (b) ground truth (c) 0.055°ae, 71s (d) 0.067°ae, 13s (e) 0.056°ae, 25s

(f) I_1 (g) ground truth (h) 0.11°ae, 77s (i) 0.12°ae, 30s (j) 0.12°ae, 24s

(k) I_1 (l) ground-truth (m) 0.15°ae, 92s (n) 0.16°ae, 17s (o) 0.16°ae, 16s

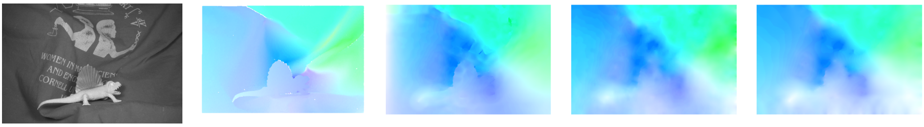
Fig. 2: Optical flow estimation with FD discretization (column 3) and triangular (column 4) and quadrilateral (column 5) FE discretizations. FE discretizations are $2-3$ times faster than the FD approach for a loss of 0.01° of angular error.

(a) I_1 (b) ground truth (c) 9.3 rms, 25s (d) 9.3 rms, 19s (e) 8.7 rms, 12s



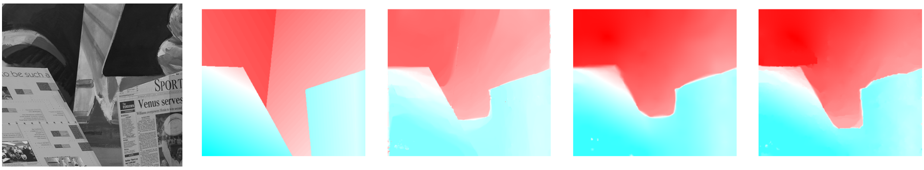
(f) I_1 (g) ground truth (h) 6.6 rms, 46s (i) 9.1 rms, 21s (j) 6.9 rms, 37s

Fig. 3: Depth-from-stereo estimation with FD (column 3) and triangular (column 4) and quadrilateral (column 5) FE discretizations. FE are 2 time faster than the FDs for a loss in accuracy below 3% in the relative RMS error.



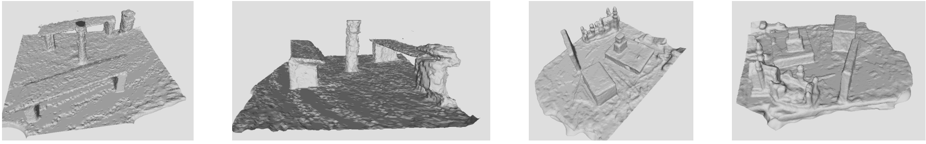
(a) I_1
(b) ground truth
(c) 0.13°ae, 73s
(d) FE 0.10°ae, 13s
(e) 0.11°ae, 25s

(f) I_1
(g) ground truth
(h) 0.24°ae, 1028s
(i) 0.30°ae, 119s
(j) 0.23°ae, 173s

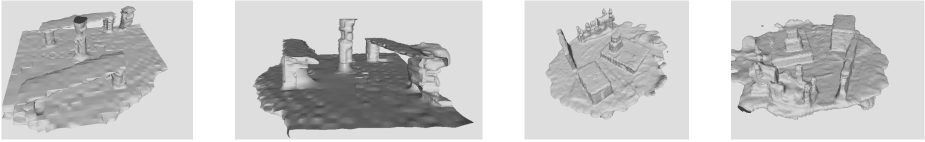


(k) I_1
(l) ground truth
(m) 0.14°ae, 356s
(n) 0.15°ae, 106s
(o) 0.12°ae, 148s

Fig. 4: Optical flow estimation with FD discretization (column 3) and triangular (column 4) and quadrilateral (column 5) FE discretizations. FE discretizations are $2-3$ times faster than the FD approach for a loss of 0.01° of angular error.



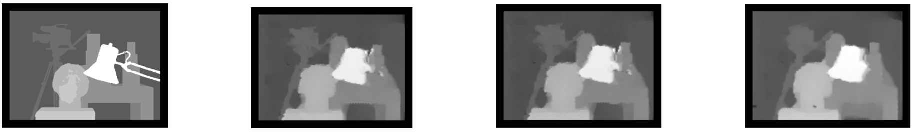
(a) FD 142 s
(b) FD 142 s
(c) FD 9460 s
(d) FD 9460 s



(e) FE 45 s
(f) FE 45 s
(g) FE 1540 s
(h) FE 1540 s

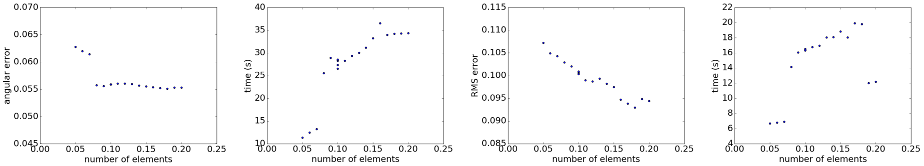
Fig. 5: Surfaces reconstructed from 14 and 24 depth maps of a small outdoor scene (left) and a tabletop model (right). FD and FE reconstructions have comparable qualities, but the FE is 3 and 6 times faster because the optimization problem is smaller and the histograms are only computed at quadrature points.

(a) ground truth (b) 0.056°ae, 25s (c) 0.055°ae, 32s (d) 0.056°ae, 66s

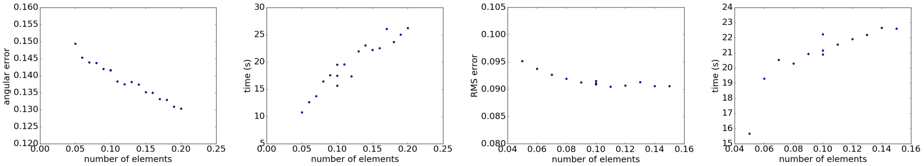


(e) ground truth (f) 8.7%rms, 12s (g) 8.7%rms, 17s (h) 8.6%rms, 32s

Fig. 6: Effects of quadrature approximations in optimization and element selection for refinement. Column 2: quadrature approximation in optimization and refinement. Column 3: quadrature approximation in optimization, pixel integration for refinement. Column 4: pixel integration in optimization and refinement.



(a) quadtree FE (b) quadtree FE (c) triangular FE (d) triangular FE



(e) quadtree FE (f) quadtree FE (g) triangular FE (h) triangular FE

Fig. 7: Effect of the number of elements on the accuracy of the discretization for *Hydrangea*'s and *Rubberwhale*'s optical flow (7(a)-7(b) and 7(e)-7(f)) and *Venus* and *Tsukuba*'s depth from stereo (7(g)-7(h) and 7(c)-7(d)). As expected, the accuracy of the discretization improves while its running time degrades as the number of finite elements (as a percentage of the number of pixels) in the final discretization increases.

References

1. P. Les and T. Wayne, "The Nurbs Book," *Springer Verlag, Berlin*, 1997.
2. T. W. Sederberg, J. Zheng, D. Sewell, and M. Sabin, "Non-uniform recursive subdivision surfaces," *SIGGRAPH*, pp. 387–394, 1998.
3. M. Kazhdan and H. Hoppe, "Screened poisson surface reconstruction," *ACM Trans. Graph.*, vol. 32, pp. 1–13, jun 2013.
4. F. Calakli and G. Taubin, "SSD: Smooth Signed Distance Surface Reconstruction," *Comput. Graph. Forum*, vol. 30, pp. 1993–2002, sep 2011.
5. B. Ummenhofer and T. Brox, "Global , Dense Multiscale Reconstruction for a Billion Points," *CVPR*, 2015.
6. P. L. Combettes and J.-C. Pesquet, "Proximal Splitting Methods in Signal Processing," in *Fixed-Point Algorithms Inverse Probl. Sci. Eng. SE - 10*, Springer Optimization and Its Applications, pp. 185–212, Springer New York, 2011.
7. A. Chambolle and T. Pock, "A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging," *J. Math. Imaging Vis.*, pp. 120–145, 2010.
8. J. Balzer and T. Morwald, "Isogeometric finite-elements methods and variational reconstruction tasks in visionA perfect match," *CVPR*, pp. 1624–1631, 2012.
9. T. Morwald, J. Balzer, and M. Vincze, "Direct optimization of T-splines based on Multiview Stereo," *Int. Conf. 3D Vis.*, 2014.
10. V. Estellers, M. A. Scott, K. Tew, and S. Soatto, "Robust Poisson Surface Reconstruction," *SSVM*, 2015.
11. D. Cremers and S. Soatto, "Motion Competition: A Variational Approach to Piecewise Parametric Motion Segmentation," *IJCV*, vol. 62, pp. 249–265, nov 2004.
12. T. Nir, A. Bruckstein, and R. Kimmel, "Over-Parameterized Variational Optical Flow," *IJCV*, vol. 76, no. 2, pp. 205–216, 2008.
13. D. Sun, E. B. Sudderth, and M. J. Black, "Layered segmentation and optical flow estimation over time," *CVPR*, pp. 1768–1775, jun 2012.
14. J. Yang and H. Li, "Dense , Accurate Optical Flow Estimation with Piecewise Parametric Model," *CVPR*, pp. 1019–1027, 2015.
15. C. Vogel, K. Schindler, and S. Roth, "Piecewise rigid scene flow," in *ICCV*, pp. 1377–1384, 2013.
16. T. F. Chan, S. Esedoglu, and M. Nikolova, "Algorithms for Finding Global Minimizers of Image Segmentation and Denoising Models," *SIAM J. Appl. Math.*, 2006.
17. T. Goldstein, X. Bresson, and S. Osher, "Geometric Applications of the Split Bregman Method: Segmentation and Surface Reconstruction," *J. Sci. Comput.*, vol. 45, pp. 272–293, nov 2009.
18. A. Wedel, T. Pock, C. Zach, H. Bischof, and D. Cremers, "An improved algorithm for TV-L 1 optical flow," *Stat. Geom. Approaches to Vis. Motion Anal.*, vol. Springer B, pp. 23–45, 2009.
19. D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles," *CVPR*, pp. 2432–2439, jun 2010.
20. C. Zach, T. Pock, and H. Bischof, "A Globally Optimal Algorithm for Robust TV-L1 Range Image Integration," *ICCV*, pp. 1–8, 2007.
21. P. Merrell, A. Akbarzadeh, L. Wang, P. Mordohai, J. M. Frahm, R. Yang, D. Nistér, and M. Pollefeys, "Real-time visibility-based fusion of depth maps," *ICCV*, 2007.
22. G. Graber, T. Pock, and H. Bischof, "Online 3D reconstruction using convex optimization," *2011 ICCV Work.*, pp. 708–711, 2011.
23. T. J. R. Hughes, *The finite element method: linear static and dynamic finite element analysis.* Courier Corporation, 2012.
24. J. Nocedal,*Numerical optimization.* Springer, 1999