TriangleFlow: Optical Flow with Triangulation-based Higher-Order Likelihoods

Ben Glocker[1], T. Hauke Heibel[1], Nassir Navab[1], Pushmeet Kohli[2], Carsten Rother[2]

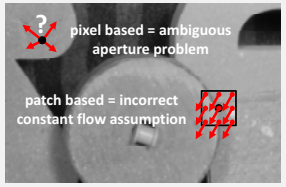
[1]CAMP, Technische Universität München, Germany
[2]Microsoft Research, Cambridge, UK

Properties of Matching Costs for Optical Flow

* **Patch based:** Discriminative, Constant flow assumption
  - considers multiple pixels simultaneously
  - incorrect costs at flow boundaries

* **Pixel based:** Ambiguous, Valid
  - ill-posed problem
  - assigns correct cost to the label assignment

* **Higher order:** Discriminative, Valid
  - considers multiple pixels simultaneously
  - assigns correct cost to the label assignment

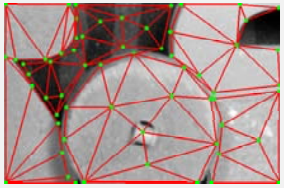


Higher-Order Likelihoods for Optical Flow

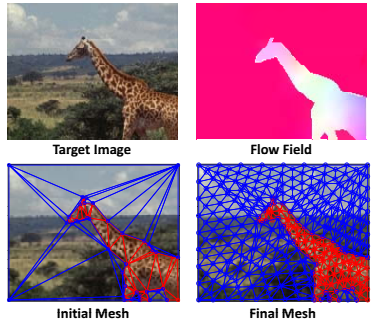
* **Tractable?**
  - Clique of K pixels has a label space of size $|L|^K$

* **Our assumption:** Optical flow is piecewise affine
  - mainly solid objects which translate, rotate, or scale
  - non-solids (e.g. textiles) can be modeled by local affine motions

* **Our solution:** Complexity reduction via triangulation
  - triangulations are adaptive and flexible
  - each triangle constitutes a local affine warp on a sub-image

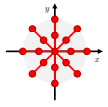


Giraffe Sequence



Random Field Energy: $E(\mathbf{x}) = \sum_{c \in C} \psi_c(\mathbf{x}_c)$

* **Optimization via hierarchical fusion moves**
  - based on QPBO and higher-order clique reduction
  - each MAP labeling is an update on the triangle configuration
  - nodes correspond to triangle control points
  - labels correspond to 2D displacement vectors
  - iterative mesh and label space refinement
  - sparse star-like search space discretization



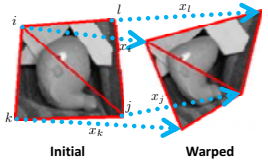
Algorithm: TriangleFlow

```
input: Images I,J
output: Flow field F

F := estimateInitialFlow(I,J);
M := estimateMultiLayerMesh(I,F);
L := initializeLabelSpace();
while continue
  ΔF := fusionMoves(I,J,F,M,L);
  if( E(F+ΔF) < E(F) )
    F := F+ΔF;
  else if( L_count < L_max )
    L := refineLabelSpace();
    L_count++;
  else if( M_count < M_max )
    M := refineMesh();
    L_count:=0;
    M_count++;
  else continue := false;
end
```

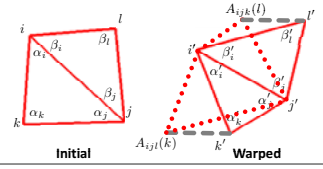
* **Likelihood encoded on triple cliques**
  - each clique labeling is a local affine warp
  - one energy term per triangle
  - evaluates the cost between sub-image I_{ijk} and J_{ijk} exactly
  - photo-consistent matching cost: correlation coefficient

$$\psi_{ijk}(x_i, x_j, x_k) = 1 - \frac{\mathrm{cov}(I_{ijk}, J_{ijk})}{\sigma_{I_{ijk}} \sigma_{J_{ijk}}}$$



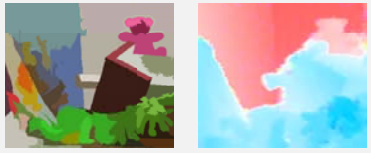
* **Motion prior on triple cliques: ADP**
  - triangle-based: *angle deviation penalty* (ADP)
  - penalizes difference between initial angles $(\alpha_i, \alpha_j, \alpha_k)$ and warped angles $(\alpha'_i, \alpha'_j, \alpha'_k)$
  - allows translation, rotation, and scaling of triangle (*i*,*j*,*k*)

* **Motion prior on quadruple cliques: NAMP**
  - rectangle-based: *non-affine motion penalty* (NAMP)
  - penalizes warps A_{ijk} and A_{ijl} of neighboring triangles
  - allows affine motions for the rectangle (*i*,*j*,*k*,*l*)



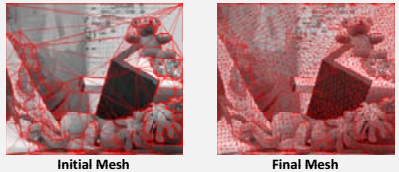
Layered Representation

* **Goal:** Determine layers of consistent motion
* **Input:** Over-segmentation + Initial flow field

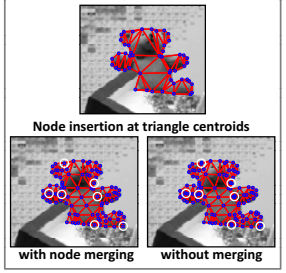


* **Spectral clustering:** Fixed to 15 clusters
* **Output:** Segments with similar affine motion

* **Multi-Layer:** One triangulation per segment

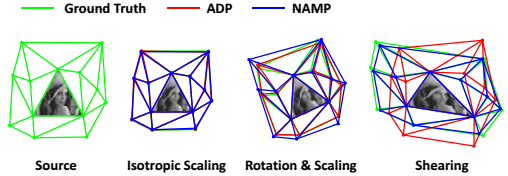


Mesh Refinement



Motion Penalty: ADP versus NAMP

* **Evaluation:** Comparison of lowest energy configurations for affine transformations under different priors



Evaluation on the Middlebury Dataset

* **TriangleFlow:** Sharp motion boundaries, high-quality flow fields, takes up to 4200 seconds
* **Multi-Layer:** 15 motion layers, explicit occlusion handling in overlap areas
* **Mesh Refinement:** coarse-to-fine strategy, 4 levels, > 10,000 triangles on finest level