Thormas Brox, Jitendra Malik

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE 2010

Presenter：Haw-Shiuan Chang

LARGE DISPLACEMENT OPTICAL FLOW:

Descriptor Matching in Variational Motion Estimation

Outline

- Introduction
  - Problem
  - Application
  - Challenge
- Motivation
  - Traditional method
  - Encounter problem
- Method
  - Descriptor match
  - Combine descriptor match and continuous method
  - Optimization
- Experiments
- Conclusion
  - Advantage/disadvantage

Outline

- Introduction
  - Problem
  - Application
  - Challenge
- Motivation
  - Traditional method
  - Encounter problem
- Method
  - Descriptor match
  - Combine descriptor match and continuous method
  - Optimization
- Experiments
- Conclusion
  - Advantage/disadvantage

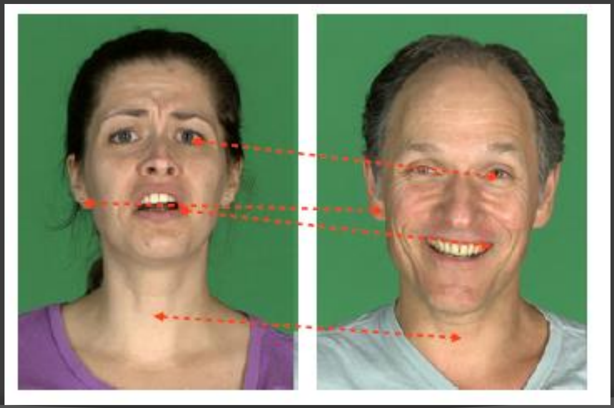
- **The three most important problems in computer vision are:**

1. Registration

2. Registration

3. Registration

- **By Takeo Kanade, Carnegie Mellon University.**

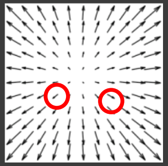


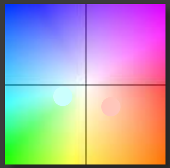
This slide comes from Artificial Intelligence course (15381) at CMU in Fall 2010

Registration type

- Images
  - Scene to scene
  - Object to object
    - Classification /clustering /detection methods
  - Space to space
    - Geometry-based methods
  - Point to point
    - Descriptor-based methods
- Video
  - Region to region
    - tracking methods
  - Point to point
    - Optical flow /stereo matching

Optical flow problem



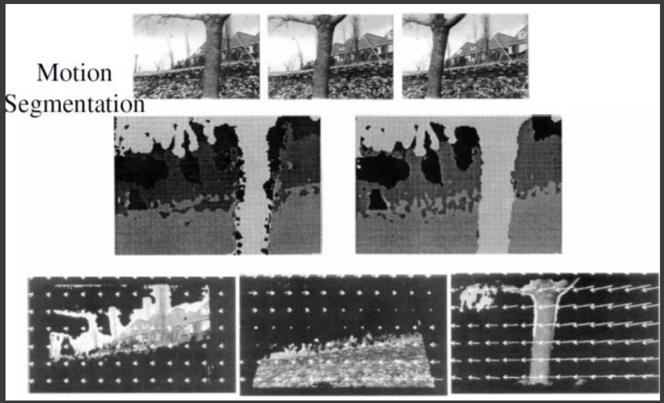


- Middlebury Benchmark [Baker et al. 07]
- Dominant Scheme: Coarse-to-Fine Warping

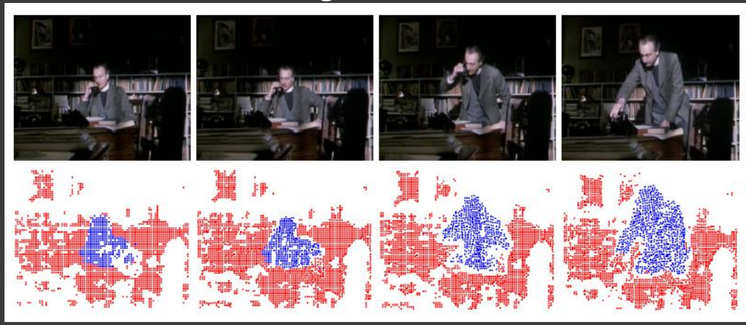
This example comes from Motion Detail Preserving Optical Flow Estimation CVPR talk

Application - Segmentation

- Motion segmentation
- Video segmentation



Layered Representation of Motion Video using Robust Maximum-Likelihood Estimation of Mixture Models and MDL Encoding, ICCV 1995



Object Segmentation by Long Term Analysis of Point Trajectories, ECCV 2010

Efficient Hierarchical Graph-Based Video Segmentation, CVPR 2010

Layered Segmentation and Optical Flow Estimation Over Time, CVPR 2012

Application - Retrieval

Auto morphing

Motion synthesis via moving objects transfer

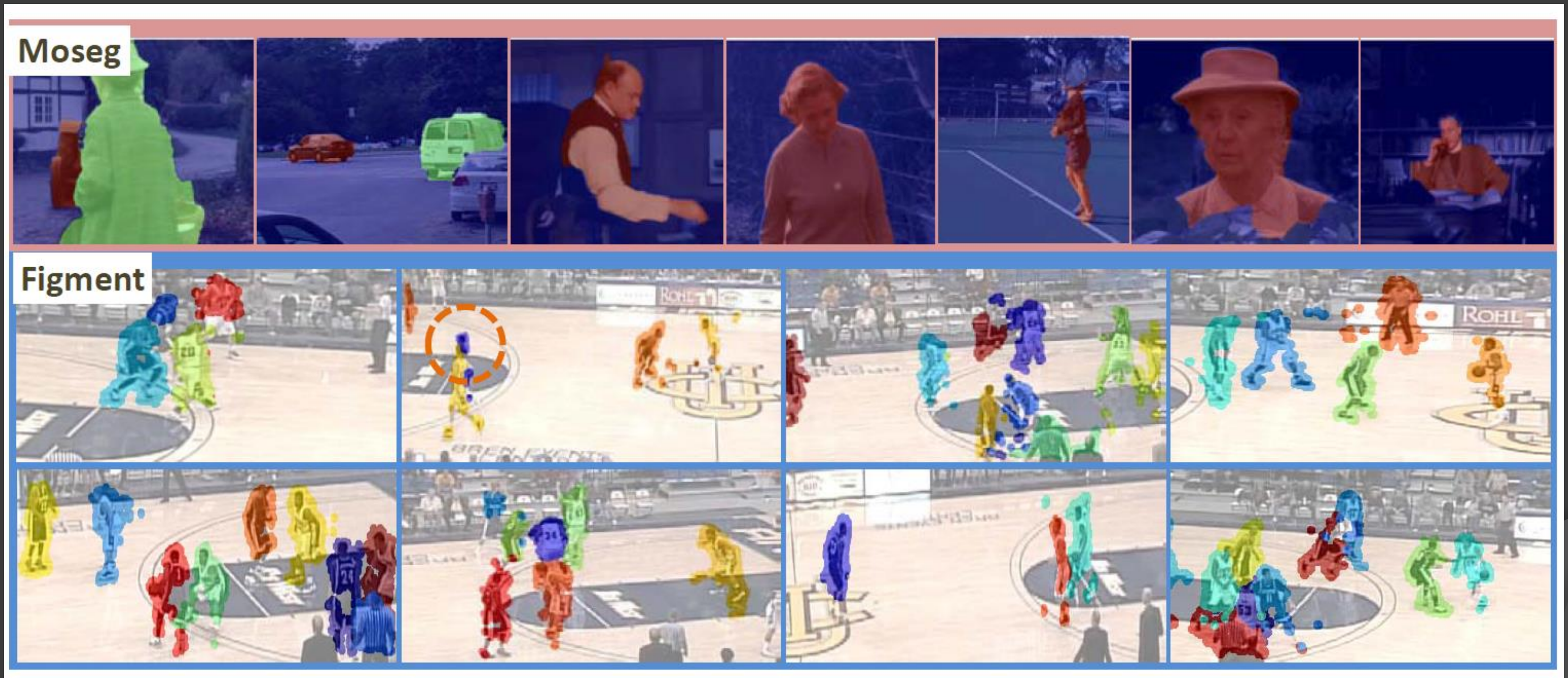
Novelty/background separation

SIFT Flow: Dense Correspondence across Scenes and its Applications
PAMI 2011

Multi View Registration for Novelty/Background Separation
CVPR 2012

Application – Detection-Free Multiple Object Tracking

Optical Flow -> point trajectory -> motion cue
Motion cue + saliency cue + spatial cue = tracking

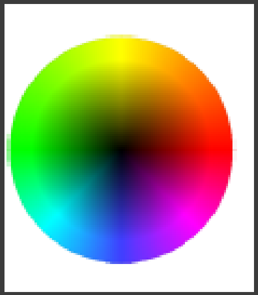


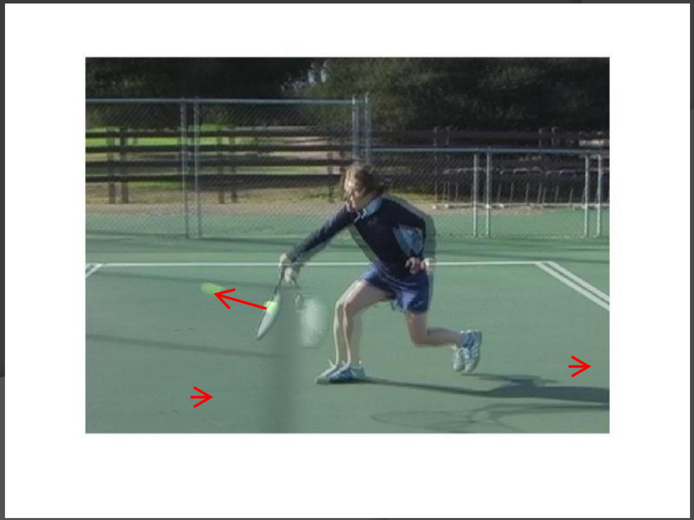
Video Segmentation by Tracing Discontinuities in a Trajectory Embedding CVPR 2012

Challenge - Optimization

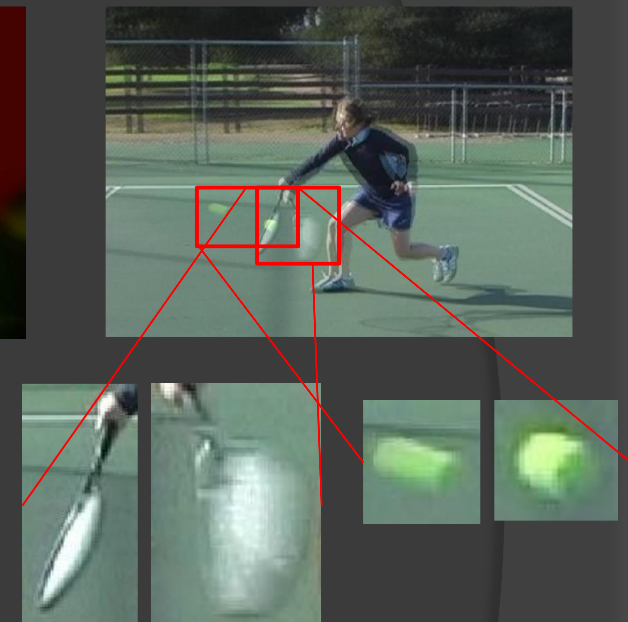
- Dense match
  - High dimension optimization problem
- Local ambiguity / aperture problem
  - Highly non-convex minimization
- Large displacement
  - Large search space
  - Non-linear optimization problem

This example comes from Freiburg-Berkeley Motion Segmentation Dataset (FBMS-59)





Challenge – Changing features



- Non-rigid motion
- Scale change
- Occlusion
- Illumination change
- Motion blur
- Noise
- …
- Unlike tracking, the state-of-the-art optical flow methods still couldn't deal with too complex cases.
- However, it improves a lot recently.

Optical flow development track

- Determining Optical Flow
- The Robust Estimation of Multiple Motions: Parametric and Piecewise-Smooth Flow Fields (Discontinuity)
- A PDE Model for Computing the Optical Flow (Continuous domain)
- Segmentation-Based Motion with Occlusions Using Graph-Cut Optimization
- Illumination-Robust Variational Optical Flow with Photometric Invariants
- Efficient MRF Deformation Model for Non-Rigid Image Matching
- Fusionflow: Discrete-Continuous Optimization for Optical Flow Estimation
- SIFT Flow: Dense Correspondence across Difference Scenes (across images)
- Optical Flow Estimation on Coarse-to-Fine Region-Trees using Discrete Optimization
- Large Displacement Optical Flow
- Large Displacement Optical Flow Computation without Warping
- Motion Detail Preserving Optical Flow Estimation (Large/small Displacement)

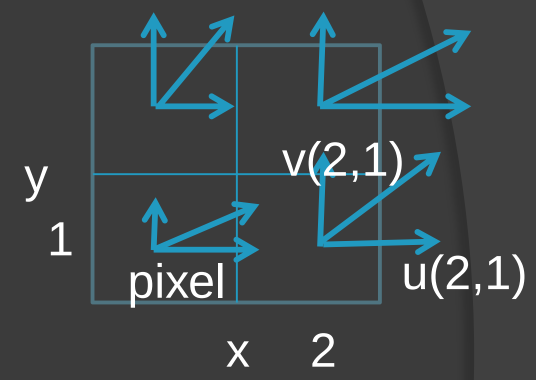
1989
1996
1999
2006
2007
2008
2009
2010

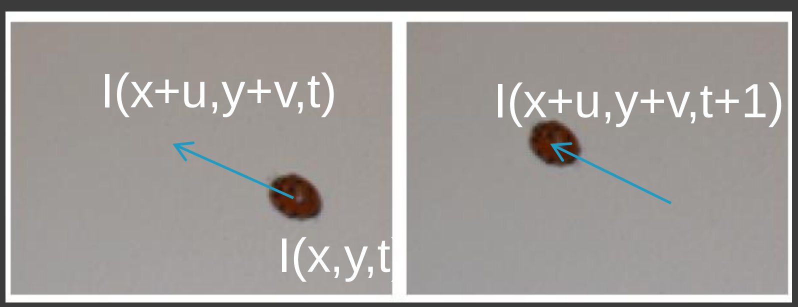
Outline

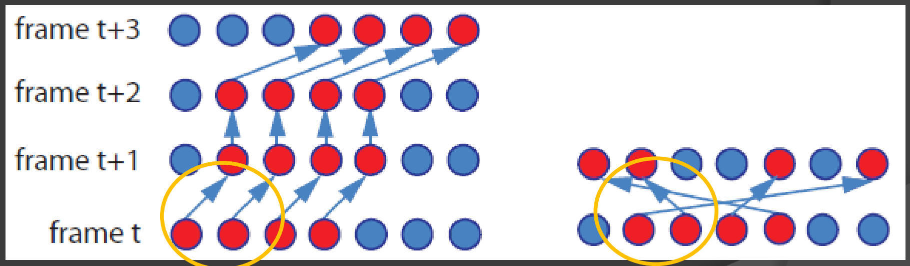
- Introduction
  - Problem
  - Application
  - Challenge
- Motivation
  - Traditional method
  - Encounter problem
- Method
  - Descriptor match
  - Combine descriptor match and continuous method
  - Optimization
- Experiments
- Conclusion
  - Advantage/disadvantage

Traditional optical flow objective function

- Define
  - $\mathbf{W}(x,y,t)=(\ \mathbf{u}(x,y,t)\ ,\ \mathbf{v}(x,y,t)\ ,\mathbf{1})$
  - $X(x,y,t)=(\ x,y,t\ )$
- $E(\mathbf{W})=E_D(\mathbf{W})+\alpha E_S(\mathbf{W})$







$$=\sum_{x_0,y_0}\Big(I(x_0+u(x_0,y_0,t_0),y+v(x_0,y_0,t_0),t_0+1)$$

This figure comes from “Motion coherent tracking with multi-label MRF optimization”, BMVC 2010

Data term linearization

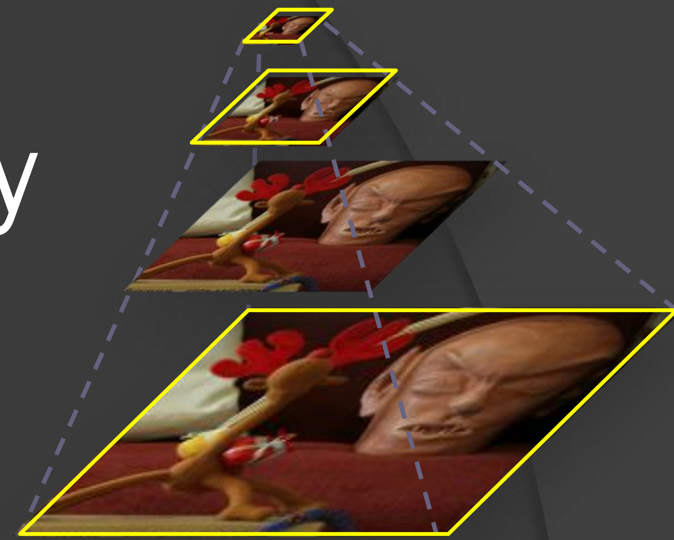
- In order to make objective function become convex, we can linearize the data term.
- However, this approximation only hold when u and v are small.

$$
\begin{aligned}
&\mathrm{E_D}(W) = \sum_{x_0,y_0} \left(\mathrm{I(X+W)} - \mathrm{I}(X)\right)^2 \\
&= \sum_{x_0,y_0} \left(\mathrm{I(x_0+u(x_0,y_0,t_0), y+v(x_0,y_0,t_0), t_0+1) - I(x_0, y_0, t_0)}\right)^2 \\
&\approx \sum_{x_0,y_0} \left(\frac{\partial I}{\partial x}(X)u(\mathrm{x_0, y_0, t_0}) + \frac{\partial I}{\partial y}(X)v(\mathrm{x_0, y_0, t_0}) + \mathrm{I}(X + (0,0,1)) - \mathrm{I}(X) \right)^2 \\
&\textit{when } u(\mathrm{x_0, y_0, t_0}), v(\mathrm{x_0, y_0, t_0}) \text{ is small}
\end{aligned}
$$

$$
\begin{aligned}
&\text{For fix } \mathrm{t{=}t_0{+}1,\ I(x_0{+}u(x_0,y_0), y_0{+}v(x_0,y_0))} = \\
&I(\mathrm{x_0, y_0}) + \left[\frac{\partial I}{\partial x}(\mathrm{x_0, y_0}) \quad \frac{\partial I}{\partial y}(\mathrm{x_0, y_0})\right]\left[u(\mathrm{x_0, y_0}) \quad v(\mathrm{x_0, y_0})\right]^T + \\
&\frac{1}{2!}\left[u(\mathrm{x_0, y_0}) \quad v(\mathrm{x_0, y_0})\right] \begin{bmatrix} \frac{\partial I}{\partial x \partial x}(\mathrm{x_0, y_0}) & \frac{\partial I}{\partial x \partial y}(\mathrm{x_0, y_0}) \\ \frac{\partial I}{\partial y \partial x}(\mathrm{x_0, y_0}) & \frac{\partial I}{\partial y \partial y}(\mathrm{x_0, y_0}) \end{bmatrix} \begin{bmatrix} u(\mathrm{x_0, y_0}) \\ v(\mathrm{x_0, y_0}) \end{bmatrix} + \ldots
\end{aligned}
$$

- How about when u and v are large?

Coarse-to fine strategy



The Multi-scale problem

Ground truth

Ground truth

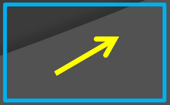
Ground truth

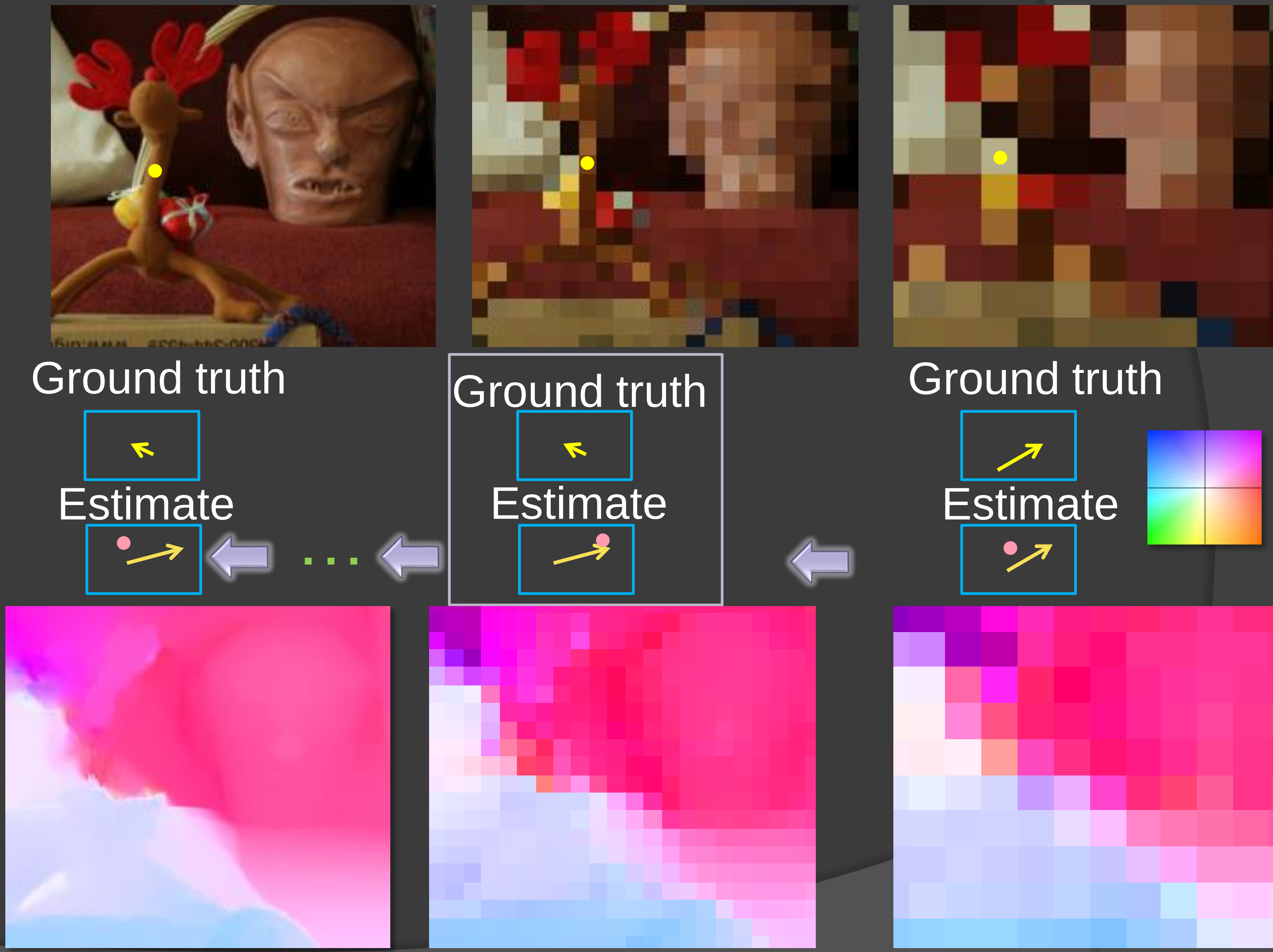
The Multi-scale problem

Ground truth

Ground truth

Ground truth

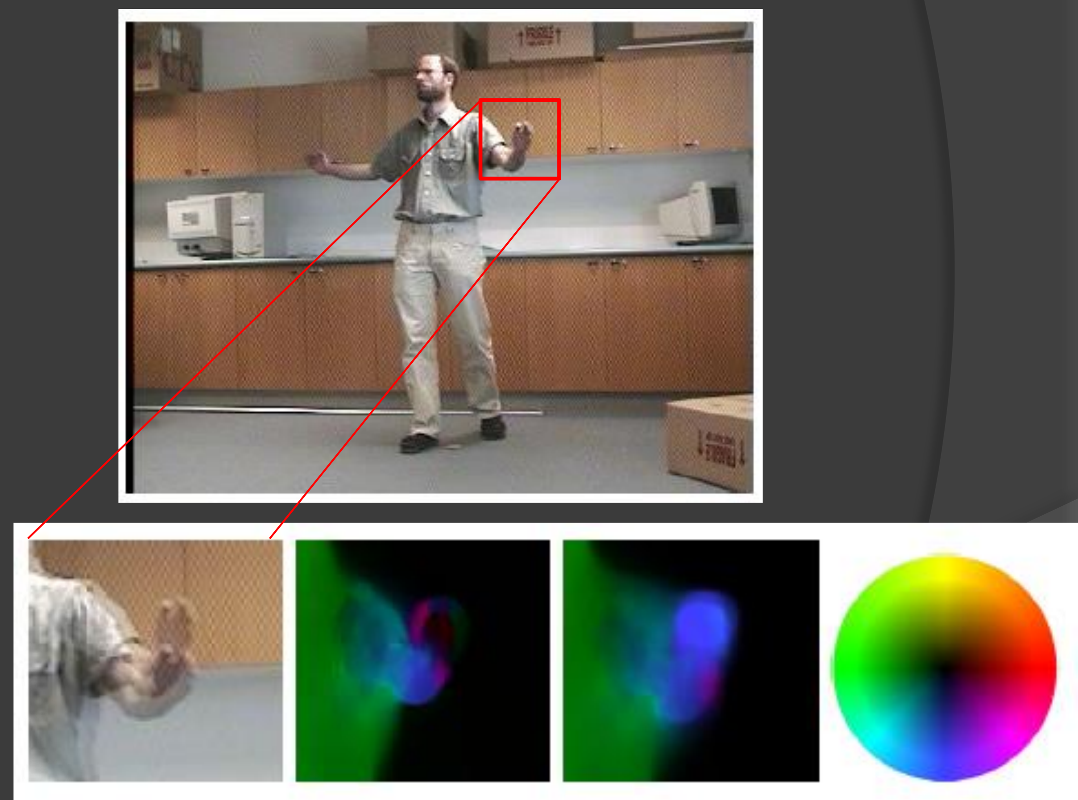




This example comes from Motion Detail Preserving Optical Flow Estimation CVPR talk

Motivation

- Unfortunately, objects moving fast are often small in this word, and when
  - Local motion > own structure
  - The moving part disappears due to the coarse-to-fine matching.

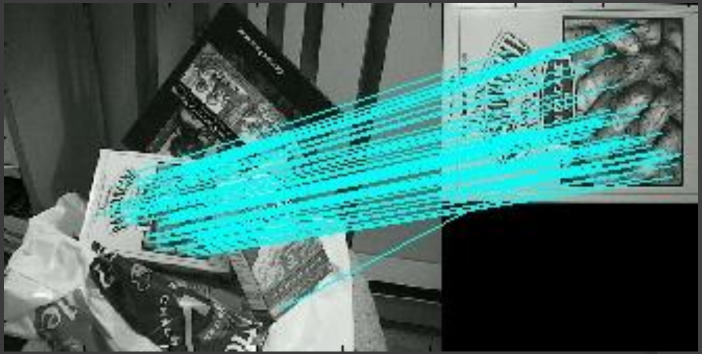


Overlap input | coarse-to-fine | LDOF

Outline

- Introduction
  - Problem
  - Application
  - Challenge
- Motivation
  - Traditional method
  - Encounter problem
- Method
  - Descriptor match
  - Combine descriptor match and continuous method
  - Optimization
- Experiments
- Conclusion
  - Advantage/disadvantage

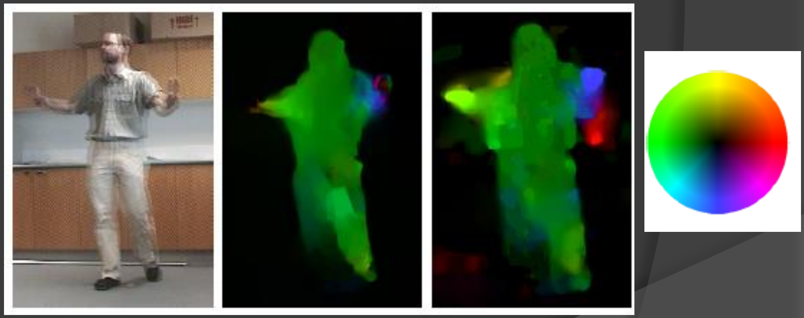
Descriptor match



- Globally optimize:

$$E_{\text{desc}}(\mathbf{w}_1) = \int \delta(\mathbf{x})\, |\mathbf{f}_2(\mathbf{x} + \mathbf{w}_1(\mathbf{x})) - \mathbf{f}_1(\mathbf{x})|^2 \, d\mathbf{x}$$

- Type
  - SIFT?
  - Region?
  - HOG(histogram of oriented gradients) or GB(geometric blur)?
- Pros:
  - No distance constraint
- Cons:
  - Outlier from complex scene or occlusion
  - Sparse discrete match
  - Limited for changing features



Baseline HOG

Can we combine the (sparse) descriptor match with the (dense) coarse-to-fine strategy?

Objective function

- Conventional: E(**W**)=E_D(**W**)+ α E_S(**W**)
- LDOF: E(**W**)=E_{color}(**W**)+γ$E_{gradient}$(**W**)+αE_{smooth}(**W**)+βE_{match}(**W**,$\mathbf{W}_1$)+E_{desc}($\mathbf{W}_1$)

E_D(**W**)

$$E_{\mathrm{color}}(\mathbf{w}) = \int_\Omega \Psi\left(|\mathbf{I}_2(\mathbf{x}+\mathbf{w}(\mathbf{x})) - \mathbf{I}_1(\mathbf{x})|^2\right)\, d\mathbf{x}$$

$$E_{\mathrm{grad}}(\mathbf{w}) = \int_\Omega \Psi\left(|\nabla\mathbf{I}_2(\mathbf{x}+\mathbf{w}(\mathbf{x})) - \nabla\mathbf{I}_1(\mathbf{x})|^2\right)\, d\mathbf{x}$$

E_s(**W**)

$$E_{\mathrm{smooth}}(\mathbf{w}) = \int_\Omega \Psi\left(|\nabla u(\mathbf{x})|^2 + |\nabla v(\mathbf{x})|^2\right)\, d\mathbf{x}$$

Descriptor match constraint

$$E_{\mathrm{desc}}(\mathbf{w}_1) = \int \delta(\mathbf{x})\, |\mathbf{f}_2(\mathbf{x}+\mathbf{w}_1(\mathbf{x})) - \mathbf{f}_1(\mathbf{x})|^2\, d\mathbf{x}$$

Loss to decide $\mathbf{W}_1$(x)
$\mathbf{W}_1$(x): descriptor match result

$$E_{\mathrm{match}}(\mathbf{w}) = \int \delta(\mathbf{x})\rho(\mathbf{x})\Psi\left(|\mathbf{w}(\mathbf{x}) - \mathbf{w}_1(\mathbf{x})|^2\right)\, d\mathbf{x}$$

Loss to make **W** close to $\mathbf{W}_1$

$$\Psi(s^2) = \sqrt{s^2+\epsilon^2}$$

robust penalty function
(s ↓ quadratic, s ↑ linear)

f(x): the feature space of descriptor

δ(x): indicate match function
ρ(x): confidence of descriptor match

Optimization process

- Descriptor match to globally optimize $\mathbf{W}_1$
- Graduated non-convex optimization
  - One hard problem (non-convex, non-linear)

    $\boxed{\text{Coarse-to-fine + Linearization}}$

    Many easier sub-problems(convex, linear)
- Discretize

Why still could be “linearize”?

Why still using “coarse-to-fine”?

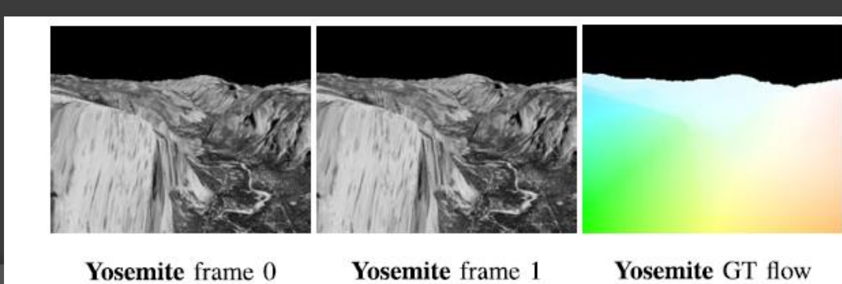
Why continuous model?

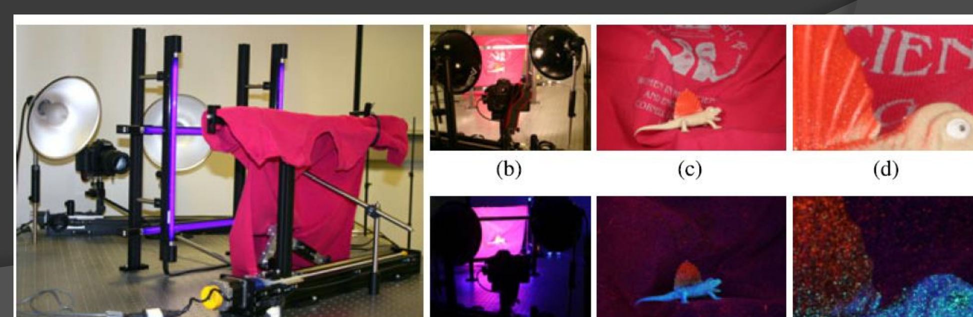
Outline

- Introduction
  - Problem
  - Application
  - Challenge
- Motivation
  - Traditional method
  - Encounter problem
- Method
  - Descriptor match
  - Combine descriptor match and continuous method
  - Optimization
- Experiments
- Conclusion
  - Advantage/disadvantage

Middlebury benchmark (IJCV2011)

- In optical flow problem, ground truth is hard to get.
- This benchmark uses
  - High speed camera
  - Fluorescent
  - Synthetic scene



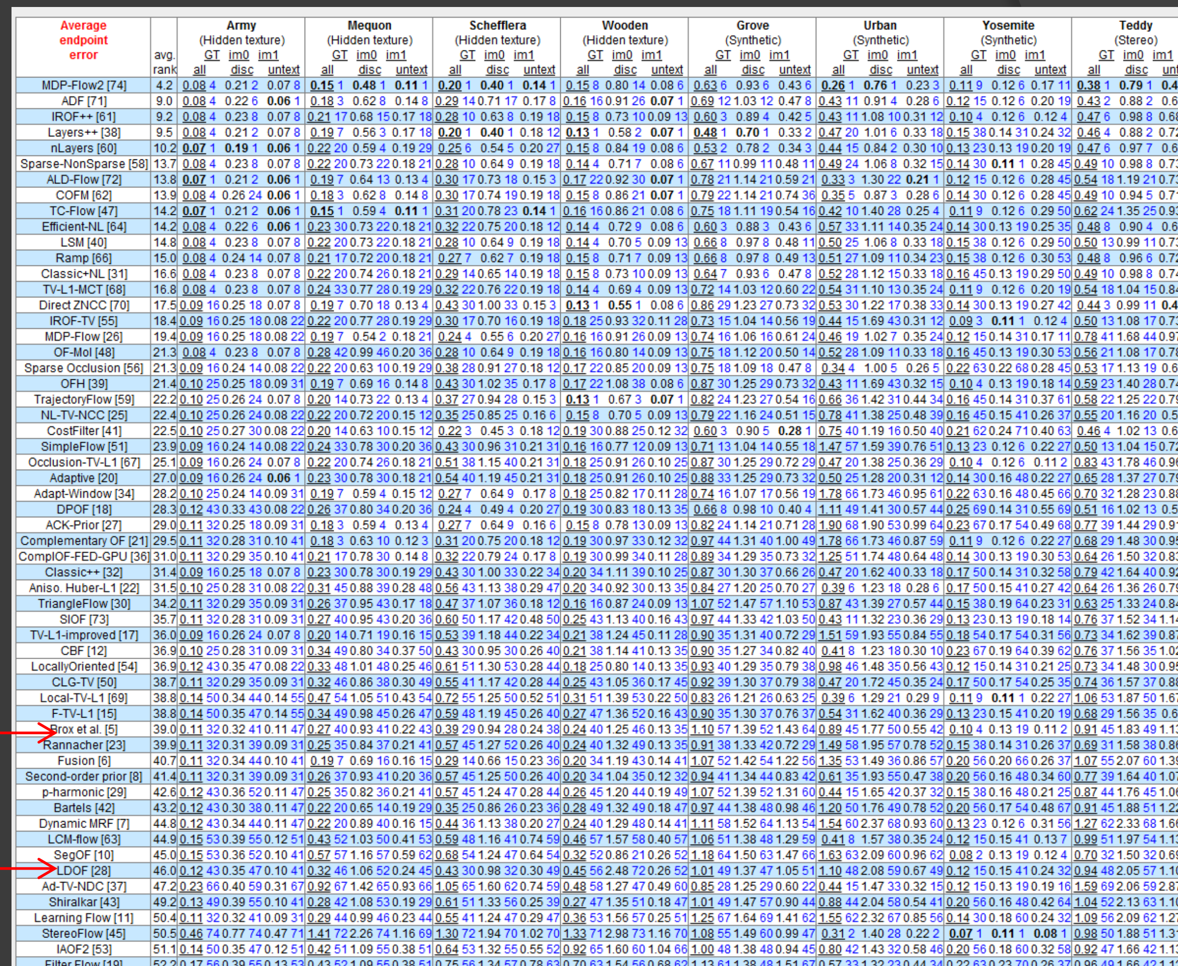


Price of large displacement

- There is "No" large displacement in Middlebury dataset.

Average endpoint error

Conventional baseline(2004)	0.501
LDOF(2009)	0.561



Average endpoint error	avg. rank	Army (Hidden texture) GT im0 im1 all			Mequon (Hidden texture) GT im0 im1 all			Schefflera (Hidden texture) GT im0 im1 all			Wooden (Hidden texture) GT im0 im1 all			Grove (Synthetic) GT im0 im1 all			Urban (Synthetic) GT im0 im1 all			Yosemite (Synthetic) GT im0 im1 all			Teddy (Stereo) GT im0 im1 all		
		all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
MDP-Flow2 [74]	4.2	0.08 4	0.21 2	0.07 8	0.15 1	0.48 1	0.11 1	0.20 1	0.40 1	0.14 1	0.15 8	0.80 14	0.08 6	0.63 6	0.93 6	0.43 6	0.26 1	0.76 1	0.23 3	0.11 9	0.12 6	0.17 11	0.38 1	0.79 1	0.44 1
ADF [71]	9.0	0.08 4	0.22 6	0.06 1	0.18 3	0.62 8	0.14 8	0.29 14	0.71 17	0.17 8	0.16 16	0.91 26	0.07 1	0.69 12	1.03 12	0.47 8	0.43 11	0.91 4	0.28 6	0.12 15	0.12 6	0.20 19	0.43 2	0.88 2	0.63 6
IROF++ [61]	9.2	0.08 4	0.23 8	0.07 8	0.21 17	0.68 15	0.17 18	0.28 10	0.63 8	0.19 18	0.15 8	0.73 10	0.09 13	0.60 3	0.89 4	0.42 5	0.43 11	1.08 10	0.31 12	0.10 4	0.12 6	0.12 4	0.47 6	0.98 8	0.68 1
Layers++ [38]	9.5	0.08 4	0.21 2	0.07 8	0.19 7	0.56 3	0.17 18	0.20 1	0.40 1	0.18 12	0.13 1	0.58 2	0.07 1	0.48 1	0.70 1	0.33 2	0.47 20	1.01 6	0.33 18	0.15 38	0.14 31	0.24 32	0.46 4	0.88 2	0.72 1
nLayers [60]	10.2	0.07 1	0.19 1	0.06 1	0.22 20	0.59 4	0.19 29	0.25 6	0.54 5	0.20 27	0.15 8	0.84 19	0.08 6	0.53 2	0.78 2	0.34 3	0.44 15	0.84 2	0.30 10	0.13 23	0.13 19	0.20 19	0.47 6	0.97 7	0.67 9
Sparse-NonSparse [58]	13.7	0.08 4	0.23 8	0.07 8	0.22 20	0.73 22	0.18 21	0.28 10	0.64 9	0.19 18	0.14 4	0.71 7	0.08 6	0.67 11	0.99 11	0.48 11	0.49 24	1.06 8	0.32 15	0.14 30	0.11 1	0.28 45	0.49 10	0.98 8	0.73 1
ALD-Flow [72]	13.8	0.07 1	0.21 2	0.06 1	0.19 7	0.64 13	0.13 4	0.30 17	0.73 18	0.15 3	0.17 22	0.92 30	0.07 1	0.78 21	1.14 21	0.59 21	0.33 3	1.30 22	0.21 1	0.12 15	0.12 6	0.28 45	0.54 18	1.19 21	0.73 1
COFM [62]	13.9	0.08 4	0.26 24	0.06 1	0.18 3	0.62 8	0.14 8	0.30 17	0.74 19	0.19 18	0.15 8	0.86 21	0.07 1	0.79 22	1.14 21	0.74 36	0.35 5	0.87 3	0.28 6	0.14 30	0.12 6	0.28 45	0.49 10	0.94 5	0.71 1
TC-Flow [47]	14.2	0.07 1	0.21 2	0.06 1	0.15 1	0.59 4	0.11 1	0.31 20	0.78 23	0.14 1	0.16 16	0.86 21	0.08 6	0.75 18	1.11 19	0.54 16	0.42 10	1.40 28	0.25 4	0.11 9	0.12 6	0.29 50	0.62 24	1.35 25	0.93 3
Efficient-NL [64]	14.2	0.08 4	0.22 6	0.06 1	0.23 30	0.73 22	0.18 21	0.32 22	0.75 20	0.18 12	0.14 4	0.72 9	0.08 6	0.60 3	0.88 3	0.43 6	0.57 33	1.11 14	0.35 24	0.14 30	0.13 19	0.25 35	0.48 8	0.90 4	0.63 6
LSM [40]	14.8	0.08 4	0.23 8	0.07 8	0.22 20	0.73 22	0.18 21	0.28 10	0.64 9	0.19 18	0.14 4	0.70 5	0.09 13	0.66 8	0.97 8	0.48 11	0.50 25	1.06 8	0.33 18	0.15 38	0.12 6	0.29 50	0.50 13	0.99 11	0.73 1
Ramp [66]	15.0	0.08 4	0.24 14	0.07 8	0.21 17	0.72 20	0.18 21	0.27 7	0.62 7	0.19 18	0.15 8	0.71 7	0.09 13	0.66 8	0.97 8	0.49 13	0.51 27	1.09 11	0.34 23	0.15 38	0.12 6	0.30 53	0.48 8	0.96 6	0.72 1
Classic+NL [31]	16.6	0.08 4	0.23 8	0.07 8	0.22 20	0.74 26	0.18 21	0.29 14	0.65 14	0.19 18	0.15 8	0.73 10	0.09 13	0.64 7	0.93 6	0.47 8	0.52 28	1.12 15	0.33 18	0.16 45	0.13 19	0.29 50	0.49 10	0.98 8	0.74 2
TV-L1-MCT [68]	16.8	0.08 4	0.23 8	0.07 8	0.24 33	0.77 28	0.19 29	0.32 22	0.76 22	0.19 18	0.14 4	0.69 4	0.09 13	0.72 14	1.03 12	0.60 22	0.54 31	1.10 13	0.35 24	0.11 9	0.12 6	0.20 19	0.54 18	1.04 15	0.84 2
Direct ZNCC [70]	17.5	0.09 16	0.25 18	0.07 8	0.19 7	0.70 18	0.13 4	0.43 30	1.00 33	0.15 3	0.13 1	0.55 1	0.08 6	0.86 29	1.23 27	0.73 32	0.53 30	1.22 17	0.38 33	0.14 30	0.13 19	0.27 42	0.44 3	0.99 11	0.44 1
IROF-TV [55]	18.4	0.09 16	0.25 18	0.08 22	0.22 20	0.77 28	0.19 29	0.30 17	0.70 16	0.19 18	0.18 25	0.93 32	0.11 28	0.73 15	1.04 14	0.56 19	0.44 15	1.69 43	0.31 12	0.09 3	0.11 1	0.12 4	0.50 13	1.08 17	0.73 1
MDP-Flow [26]	19.4	0.09 16	0.25 18	0.08 22	0.19 7	0.54 2	0.18 21	0.24 4	0.55 6	0.20 27	0.16 16	0.91 26	0.09 13	0.74 16	1.06 16	0.61 24	0.46 19	1.02 7	0.35 24	0.12 15	0.14 31	0.17 11	0.78 41	1.68 44	0.97 4
OF-Mol [48]	21.3	0.08 4	0.23 8	0.07 8	0.28 42	0.99 46	0.20 36	0.28 10	0.64 9	0.19 18	0.16 16	0.80 14	0.09 13	0.75 18	1.12 20	0.50 14	0.52 28	1.09 11	0.33 18	0.16 45	0.13 19	0.30 53	0.56 21	1.08 17	0.78 2
Sparse Occlusion [56]	21.3	0.09 16	0.24 14	0.08 22	0.22 20	0.63 10	0.19 29	0.38 28	0.91 27	0.18 12	0.17 22	0.85 20	0.09 13	0.75 18	1.09 18	0.47 8	0.34 4	1.00 5	0.26 5	0.22 63	0.22 68	0.28 45	0.53 17	1.13 19	0.67 9
OFH [39]	21.4	0.10 25	0.25 18	0.09 31	0.19 7	0.69 16	0.14 8	0.43 30	1.02 35	0.17 8	0.17 22	1.08 38	0.08 6	0.87 30	1.25 29	0.73 32	0.43 11	1.69 43	0.32 15	0.10 4	0.13 19	0.18 14	0.59 23	1.40 28	0.74 2
TrajectoryFlow [59]	22.2	0.10 25	0.26 24	0.07 8	0.20 14	0.73 22	0.13 4	0.37 27	0.94 28	0.15 3	0.13 1	0.67 3	0.07 1	0.82 24	1.23 27	0.54 16	0.66 36	1.42 31	0.44 34	0.16 45	0.14 31	0.37 61	0.58 22	1.25 22	0.79 2
NL-TV-NCC [25]	22.4	0.10 25	0.26 24	0.08 22	0.22 20	0.72 20	0.15 12	0.35 25	0.85 25	0.16 6	0.15 8	0.70 5	0.09 13	0.79 22	1.16 24	0.51 15	0.78 41	1.78 49	0.58 46	0.16 45	0.15 41	0.26 37	0.55 20	1.16 20	0.55 4
CostFilter [41]	22.5	0.10 25	0.27 30	0.08 22	0.20 14	0.63 10	0.15 12	0.22 3	0.45 3	0.18 12	0.19 30	0.88 25	0.12 32	0.60 3	0.90 5	0.28 1	0.75 40	1.19 16	0.50 40	0.21 62	0.24 71	0.40 63	0.46 4	1.02 13	0.62 5
SimpleFlow [51]	23.9	0.09 16	0.24 14	0.08 22	0.24 33	0.78 30	0.20 36	0.43 30	0.96 31	0.21 31	0.16 16	0.77 12	0.09 13	0.71 13	1.04 14	0.55 18	1.47 57	1.59 39	0.76 51	0.13 23	0.12 6	0.22 27	0.50 13	1.04 15	0.72 1
Occlusion-TV-L1 [67]	25.1	0.09 16	0.26 24	0.07 8	0.22 20	0.74 26	0.18 21	0.51 38	1.15 40	0.21 31	0.18 25	0.91 26	0.10 25	0.87 30	1.25 29	0.72 29	0.47 20	1.38 25	0.36 29	0.10 4	0.12 6	0.11 2	0.83 43	1.78 46	0.96 3
Adaptive [20]	27.0	0.09 16	0.26 24	0.06 1	0.23 30	0.78 30	0.18 21	0.54 40	1.19 45	0.21 31	0.18 25	0.91 26	0.10 25	0.88 33	1.25 29	0.73 32	0.50 25	1.28 20	0.31 12	0.14 30	0.16 48	0.22 27	0.65 28	1.37 27	0.79 2
Adapt-Window [34]	28.2	0.10 25	0.24 14	0.09 31	0.19 7	0.59 4	0.15 12	0.27 7	0.64 9	0.17 8	0.18 25	0.82 17	0.11 28	0.74 16	1.07 17	0.56 19	1.78 66	1.73 46	0.95 61	0.22 63	0.16 48	0.45 66	0.70 32	1.28 23	0.88 3
DPOF [18]	28.3	0.12 43	0.33 43	0.08 22	0.26 37	0.80 34	0.20 36	0.24 4	0.49 4	0.20 27	0.19 30	0.83 18	0.13 35	0.66 8	0.98 10	0.40 4	1.11 49	1.41 30	0.57 44	0.25 69	0.14 31	0.55 69	0.51 16	1.02 13	0.54 3
ACK-Prior [27]	29.0	0.11 32	0.25 18	0.09 31	0.18 3	0.59 4	0.13 4	0.27 7	0.64 9	0.16 6	0.15 8	0.78 13	0.09 13	0.82 24	1.14 21	0.71 28	1.90 68	1.90 53	0.99 64	0.23 67	0.17 54	0.49 68	0.77 39	1.44 29	0.91 3
Complementary OF [21]	29.5	0.11 32	0.28 31	0.10 41	0.18 3	0.63 10	0.12 3	0.31 20	0.75 20	0.18 12	0.19 30	0.97 33	0.12 32	0.97 44	1.31 40	1.00 49	1.78 66	1.73 46	0.87 59	0.11 9	0.12 6	0.22 27	0.68 29	1.48 30	0.95 3
ComplOF-FED-GPU [36]	31.0	0.11 32	0.29 35	0.10 41	0.21 17	0.78 30	0.14 8	0.32 22	0.79 24	0.17 8	0.19 30	0.99 34	0.11 28	0.89 34	1.29 35	0.73 32	1.25 51	1.74 48	0.64 48	0.14 30	0.13 19	0.30 53	0.64 26	1.50 32	0.83 2
Classic++ [32]	31.4	0.09 16	0.25 18	0.07 8	0.23 30	0.78 30	0.19 29	0.43 30	1.00 33	0.22 34	0.20 34	1.11 39	0.10 25	0.87 30	1.30 37	0.66 26	0.47 20	1.62 40	0.33 18	0.17 50	0.14 31	0.32 58	0.79 42	1.64 40	0.92 3
Aniso. Huber-L1 [22]	31.5	0.10 25	0.28 31	0.08 22	0.31 45	0.88 39	0.28 48	0.56 43	1.13 38	0.29 47	0.20 34	0.92 30	0.13 35	0.84 27	1.20 25	0.70 27	0.39 6	1.23 18	0.28 6	0.17 50	0.15 41	0.27 42	0.64 26	1.36 26	0.79 2
TriangleFlow [30]	34.2	0.11 32	0.29 35	0.09 31	0.26 37	0.95 43	0.17 18	0.47 37	1.07 36	0.18 12	0.16 16	0.87 24	0.09 13	1.07 52	1.47 57	1.10 53	0.87 43	1.39 27	0.57 44	0.15 38	0.19 64	0.23 31	0.63 25	1.33 24	0.84 2
SIOF [73]	35.7	0.11 32	0.28 31	0.09 31	0.27 40	0.95 43	0.20 36	0.60 50	1.17 42	0.48 50	0.25 43	1.13 40	0.16 43	0.97 44	1.33 42	1.03 50	0.43 11	1.32 23	0.36 29	0.13 23	0.13 19	0.18 14	0.76 37	1.52 34	1.14 5
TV-L1-improved [17]	36.0	0.09 16	0.26 24	0.07 8	0.20 14	0.71 19	0.16 15	0.53 39	1.18 44	0.22 34	0.21 38	1.24 45	0.11 28	0.90 35	1.31 40	0.72 29	1.51 59	1.93 55	0.84 55	0.18 54	0.17 54	0.31 56	0.73 34	1.62 39	0.87 3
CBF [12]	36.9	0.10 25	0.28 31	0.09 31	0.34 49	0.80 34	0.37 50	0.43 30	0.95 30	0.26 40	0.21 38	1.14 41	0.13 35	0.90 35	1.27 34	0.82 40	0.41 8	1.23 18	0.30 10	0.23 67	0.19 64	0.39 62	0.76 37	1.56 35	1.02 4
LocallyOriented [54]	36.9	0.12 43	0.35 47	0.08 22	0.33 48	1.01 48	0.25 46	0.61 51	1.30 53	0.28 44	0.18 25	0.80 14	0.13 35	0.93 40	1.29 35	0.79 38	0.98 46	1.48 35	0.56 43	0.12 15	0.14 31	0.21 25	0.73 34	1.48 30	0.95 3
CLG-TV [50]	38.7	0.11 32	0.29 35	0.09 31	0.32 46	0.86 38	0.30 49	0.55 41	1.17 42	0.28 44	0.25 43	1.05 36	0.17 45	0.92 39	1.30 37	0.79 38	0.47 20	1.72 45	0.35 24	0.17 50	0.17 54	0.25 35	0.74 36	1.57 37	0.88 3
Local-TV-L1 [69]	38.8	0.14 50	0.34 44	0.14 55	0.47 54	1.05 51	0.43 54	0.72 55	1.25 50	0.52 51	0.31 51	1.39 53	0.22 50	0.83 26	1.21 26	0.63 25	0.39 6	1.29 21	0.29 9	0.11 9	0.11 1	0.22 27	1.06 53	1.87 50	1.67 5
F-TV-L1 [15]	38.8	0.14 50	0.35 47	0.14 55	0.34 49	0.98 45	0.26 47	0.59 48	1.19 45	0.26 40	0.27 47	1.36 52	0.16 43	0.90 35	1.30 37	0.76 37	0.54 31	1.62 40	0.36 29	0.13 23	0.15 41	0.20 19	0.68 29	1.56 35	0.66 8
Brox et al. [5]	39.0	0.11 32	0.32 41	0.11 47	0.27 40	0.93 41	0.22 43	0.39 29	0.94 28	0.24 38	0.24 40	1.25 46	0.13 35	1.10 57	1.39 52	1.43 64	0.89 45	1.77 50	0.55 42	0.10 4	0.13 19	0.11 2	0.91 45	1.83 49	1.13 4
Rannacher [23]	39.9	0.11 32	0.31 39	0.09 31	0.25 35	0.84 37	0.21 41	0.57 45	1.27 52	0.26 40	0.24 40	1.32 49	0.13 35	0.91 38	1.33 42	0.72 29	1.49 58	1.95 57	0.78 52	0.15 38	0.14 31	0.26 37	0.69 31	1.58 38	0.86 3
Fusion [6]	40.7	0.11 32	0.34 44	0.10 41	0.19 7	0.69 16	0.16 15	0.29 14	0.66 15	0.23 36	0.20 34	1.19 43	0.14 41	1.07 52	1.42 54	1.22 56	1.35 53	1.49 36	0.86 57	0.20 56	0.20 66	0.26 37	1.07 55	2.07 60	1.39 5
Second-order prior [8]	41.4	0.11 32	0.31 39	0.09 31	0.26 37	0.93 41	0.20 36	0.57 45	1.25 50	0.26 40	0.20 34	1.04 35	0.12 32	0.94 41	1.34 44	0.83 42	0.61 35	1.93 55	0.47 38	0.20 56	0.16 48	0.34 60	0.77 39	1.64 40	1.07 4
p-harmonic [29]	42.6	0.12 43	0.36 52	0.11 47	0.25 35	0.82 36	0.21 41	0.57 45	1.24 47	0.28 44	0.26 45	1.20 44	0.19 49	1.07 52	1.39 52	1.31 60	0.44 15	1.65 42	0.37 32	0.15 38	0.16 48	0.21 25	0.87 44	1.76 45	1.06 4
Bartels [42]	43.2	0.12 43	0.30 38	0.11 47	0.22 20	0.65 14	0.19 29	0.35 25	0.86 26	0.23 36	0.28 49	1.32 49	0.18 47	0.97 44	1.38 48	0.98 46	1.20 50	1.76 49	0.78 52	0.20 56	0.17 54	0.48 67	0.91 45	1.88 51	1.22 5
Dynamic MRF [7]	44.8	0.12 43	0.34 44	0.11 47	0.22 20	0.89 40	0.16 15	0.44 36	1.13 38	0.20 27	0.24 40	1.29 48	0.14 41	1.11 58	1.52 64	1.13 54	1.54 60	2.37 68	0.93 60	0.13 23	0.12 6	0.31 56	1.27 62	2.33 68	1.66 5
LCM-flow [63]	44.9	0.15 53	0.39 55	0.12 51	0.43 52	1.03 50	0.41 53	0.59 48	1.16 41	0.74 59	0.46 57	1.57 58	0.40 57	1.06 51	1.38 48	1.29 59	0.41 8	1.57 38	0.35 24	0.12 15	0.15 41	0.13 7	0.99 51	1.97 54	1.13 4
SegOF [10]	45.0	0.15 53	0.36 52	0.10 41	0.57 57	1.16 57	0.59 62	0.68 54	1.24 47	0.64 54	0.32 52	0.86 21	0.26 52	1.18 64	1.50 63	1.47 66	1.63 63	2.09 60	0.96 62	0.08 2	0.13 19	0.12 4	0.70 32	1.50 32	0.69 1
LDOF [28]	46.0	0.12 43	0.35 47	0.10 41	0.32 46	1.06 52	0.24 45	0.43 30	0.98 32	0.30 49	0.45 56	2.48 72	0.26 52	1.01 49	1.37 47	1.05 51	1.10 48	2.08 59	0.67 49	0.12 15	0.15 41	0.24 32	0.94 48	2.05 57	1.10 4
Ad-TV-NDC [37]	47.2	0.23 66	0.40 59	0.31 67	0.92 67	1.42 65	0.93 66	1.05 65	1.60 62	0.74 59	0.48 58	1.27 47	0.49 60	0.85 28	1.25 29	0.60 22	0.44 15	1.47 33	0.32 15	0.12 15	0.13 19	0.19 16	1.59 69	2.06 59	2.87 7
Shiralkar [43]	49.2	0.13 49	0.39 55	0.10 41	0.28 42	1.08 53	0.19 29	0.61 51	1.33 56	0.25 39	0.27 47	1.35 51	0.18 47	1.01 49	1.47 57	0.90 44	0.88 44	2.04 58	0.54 41	0.20 56	0.16 48	0.42 64	1.04 52	2.13 63	1.10 4
Learning Flow [11]	50.4	0.11 32	0.32 41	0.09 31	0.29 44	0.99 46	0.23 44	0.55 41	1.24 47	0.29 47	0.36 53	1.56 57	0.25 51	1.25 67	1.64 69	1.41 62	1.55 62	2.32 67	0.85 56	0.14 30	0.18 60	0.24 32	1.09 56	2.09 62	1.27 5
StereoFlow [45]	50.5	0.46 74	0.77 74	0.47 71	1.41 72	2.26 74	1.16 69	1.30 72	1.94 70	1.02 70	1.33 71	2.98 73	1.16 70	1.08 55	1.49 60	0.99 47	0.31 2	1.40 28	0.22 2	0.07 1	0.11 1	0.08 1	0.98 50	1.88 51	1.31 5
IAOF2 [53]	51.1	0.14 50	0.35 47	0.12 51	0.42 51	1.09 55	0.38 51	0.64 53	1.32 55	0.55 52	0.92 65	1.60 60	1.04 66	1.00 48	1.38 48	0.94 45	0.80 42	1.43 32	0.58 46	0.20 56	0.18 60	0.32 58	0.92 47	1.66 42	1.13 4
Filter Flow [19]	52.2	0.17 56	0.39 55	0.13 53	0.43 52	1.09 55	0.38 51	0.75 56	1.34 57	0.78 63	0.70 63	1.54 56	0.68 62	1.13 61	1.38 48	1.51 67	0.57 33	1.32 23	0.44 34	0.22 63	0.23 70	0.26 37	0.96 49	1.66 42	1.12 4

Comparison between chosen descriptor

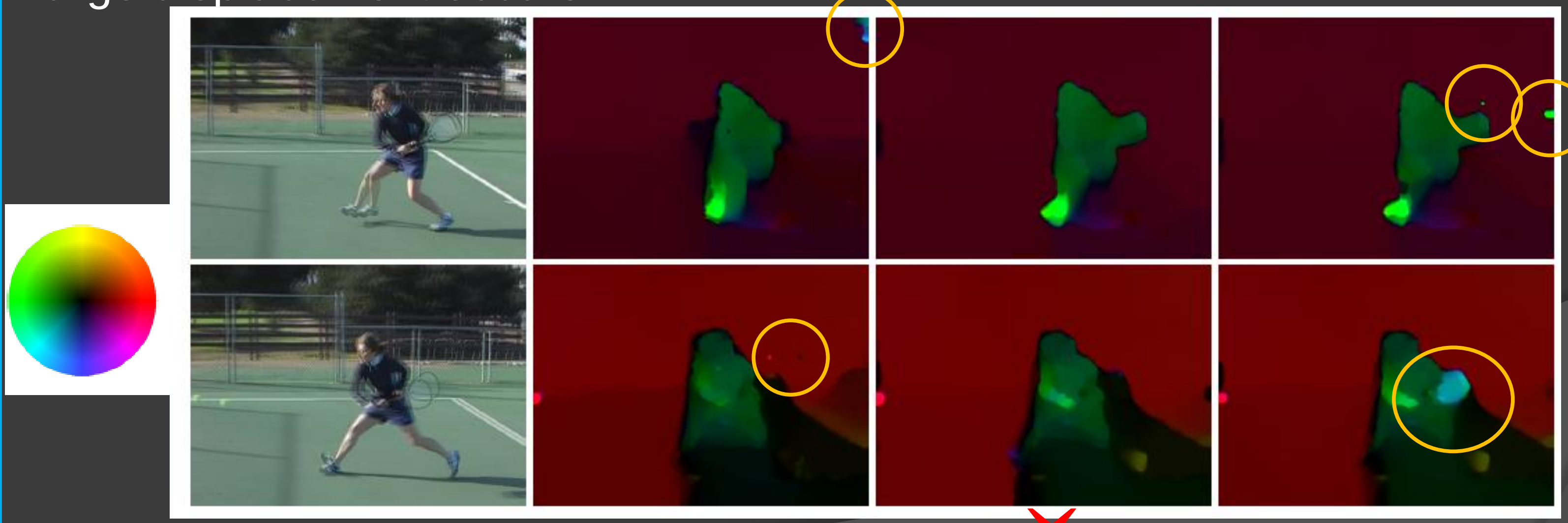
- Region match is too sparse.
- GB has more correct and wrong match.
- HOG is more efficient.

No large displacement

	Warping only ($\beta = 0$)	Regions	HOG	GB
Dimetrodon	1.82	1.74	1.85	1.95
Grove2	2.09	2.25	2.68	2.79
Grove3	5.59	6.55	6.38	6.35
Urban2	2.28	3.05	2.64	3.15
Urban3	3.99	5.76	5.07	5.19
RubberWhale	3.77	3.84	3.94	4.14
Hydrangea	2.32	2.36	2.44	2.54
Venus	5.19	7.37	6.45	6.52
Average	**3.38**	**4.11**	**3.93**	**4.08**

TABLE 1

Large displacement occurs

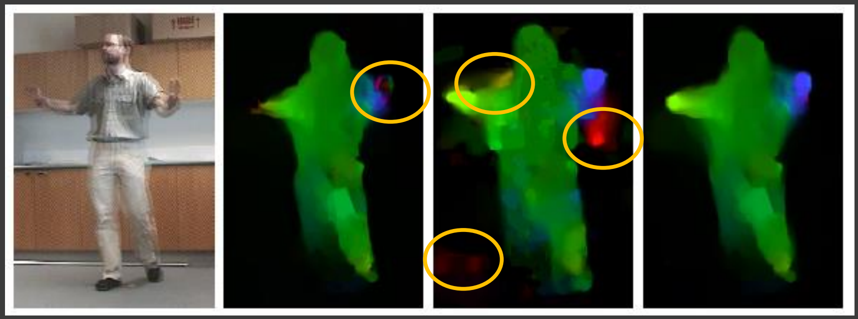


Region HOG GB

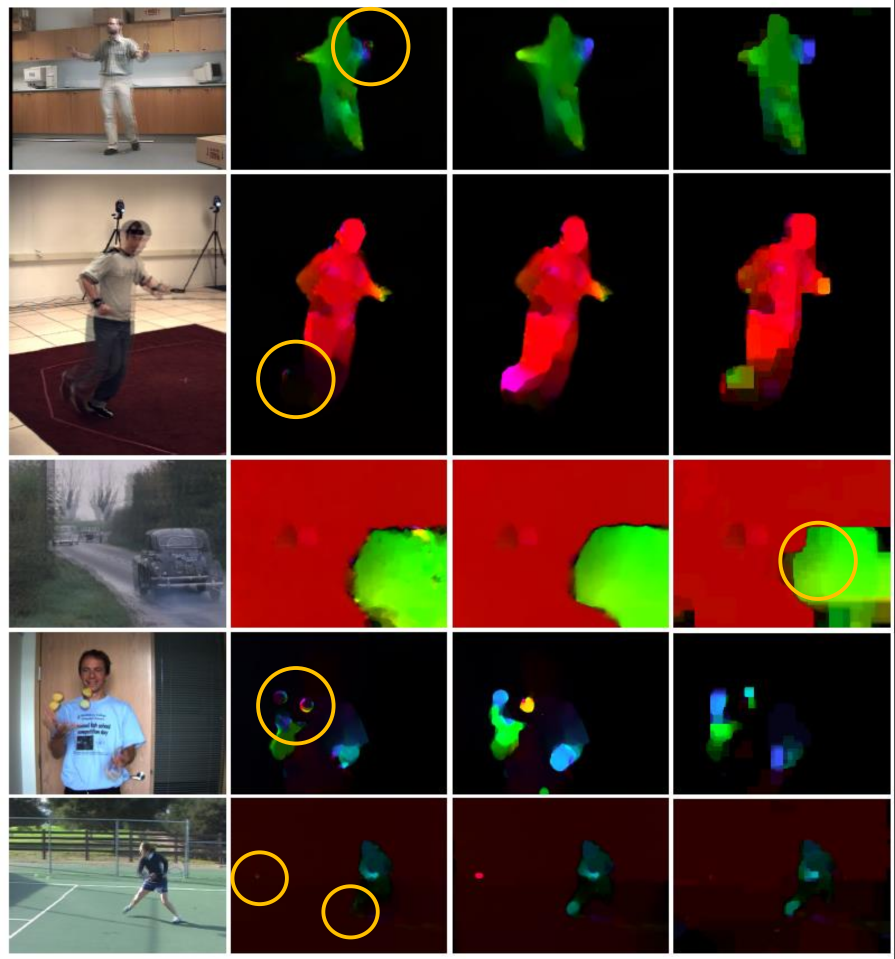
Comparison with only use descriptor match

- LDOF is more accurate and can deal with occlusion and ambiguity in smoothly textural areas.

	Warping [11]	LDOF	SIFT flow [23]
Fig.6, row 1, 384 × 288	7s	18s	99s
Fig.6, row 2, 373 × 485	12s	29s	167s
Fig.6, row 3, 450 × 350	10s	44s	144s
Fig.6, row 4, 640 × 480	21s	80s	67s*
Fig.6, row 5, 530 × 380	13s	39s	177s



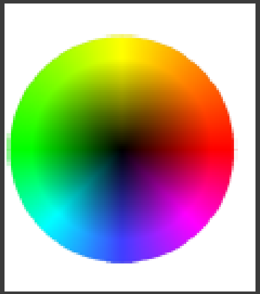
Baseline HOG LDOF



Baseline LDOF SIFT flow

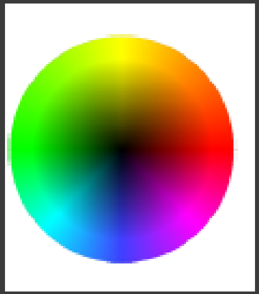
Experiments on video with large displacement-1

- http://lmb.informatik.uni-freiburg.de/research/opticalflow/



Experiments on video with large displacement-2

- http://lmb.informatik.uni-freiburg.de/research/opticalflow/



Outline

- Introduction
  - Problem
  - Application
  - Challenge
- Motivation
  - Traditional method
  - Encounter problem
- Method
  - Descriptor match
  - Combine descriptor match and continuous method
  - Optimization
- Experiments
- Conclusion
  - Advantage/disadvantage

Conclusion

- Investigates how to combine
  - Descriptor match
  - Classic coarse-to-fine method to optimize continuous energy function. (Variantional model)
- LDOF becomes the first successful algorithm which deals with small scale objects with large movements.

Reference

- Thomas Brox, Andres Bruhn, Nils Papenberg, Joachim Weickert, High Accuracy Optical Flow Estimation Based on a Theory for Warping ECCV 2004
- Large Displacement Optical Flow CVPR 2009
- Narayanan Sundaram, Thomas Brox, and Kurt Keutzer, Dense Point Trajectories by GPU-accelerated Large Displacement Optical Flow ECCV 2010
- Thomas Brox, Jitendra Malik,Large Displacement Optical Flow: Descriptor Matching in Variational Motion Estimation PAMI 2010
- Li Xu, Jiaya Jia, Yasuyuki Matsushita, Motion Detail Preserving Optical Flow Estimation PAMI 2012

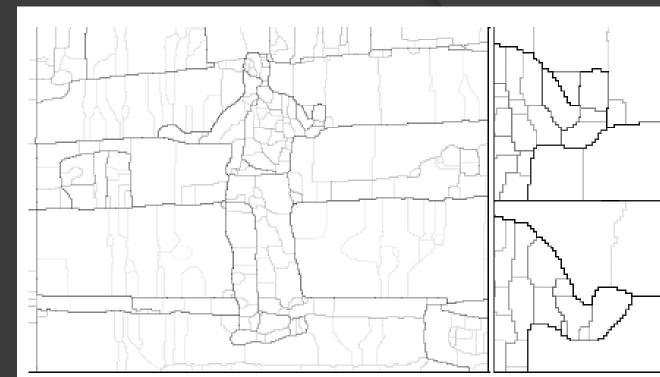
APPENDIX

Descriptor match-Region match

- Segmentation method: gPb-owt-ucm
- Normalize to 32x32 patch
- Find 10 nearest patch
  - SIFT + corresponding color feature
  - Filter out:
    - Too large displacement
    - Too large scale change
- Perform local deformation by conventional optical flow and select 5 nearest patch (or only best match)
- Backward check

$$\rho(\mathbf{x}_i) := \frac{d_2 - d_1}{d_1}$$

$$\frac{\bar{d}^2(i) - d^2(i,j)}{d^2(i,j)}$$



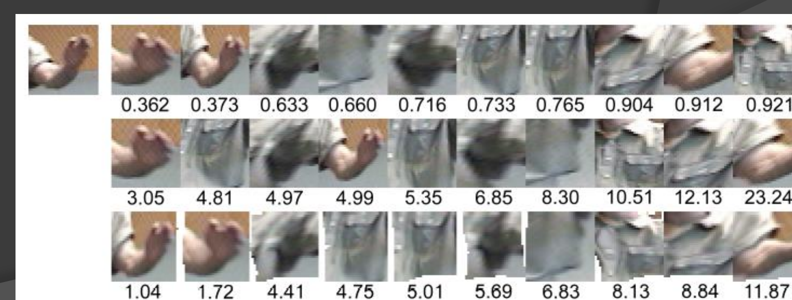
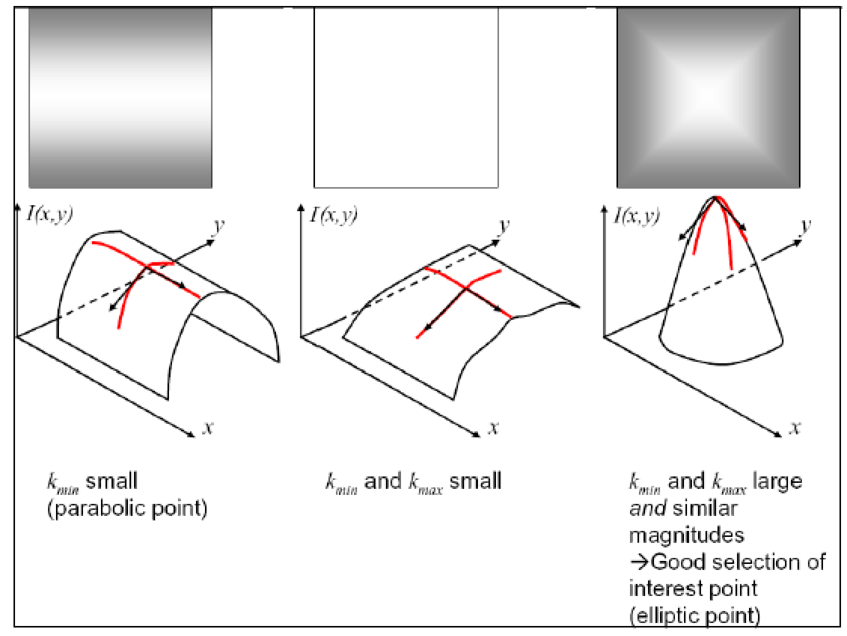


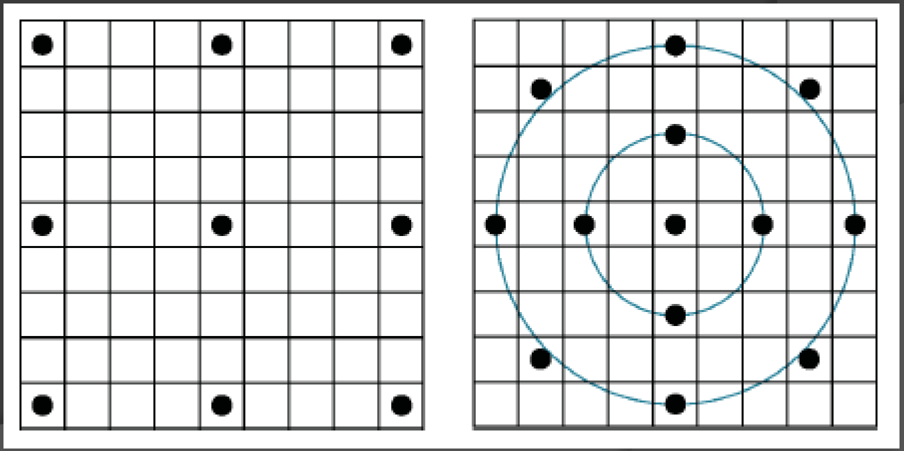
Figure 4. Nearest neighbors and their distances using different descriptors. **Top:** SIFT and color. **Center:** Patch within region. **Bottom:** Patch within region after distortion correction.

Descriptor match-HOG & GB

- efficient when using integral images
- HOG
  - 15 bins, 7x7 neighborhood
  - 135 dimension for each point
  - match from 1/16 points to all points
  - Filter out points with no structure (smaller $\lambda < 1/8\ \lambda_{avg}$ of $\nabla I\ \nabla I^T$)
  - Backward check
- GB
  - 15 bins, gaussian neighborhood with $\sigma = 0,1,2$
  - 195 dimension for each point

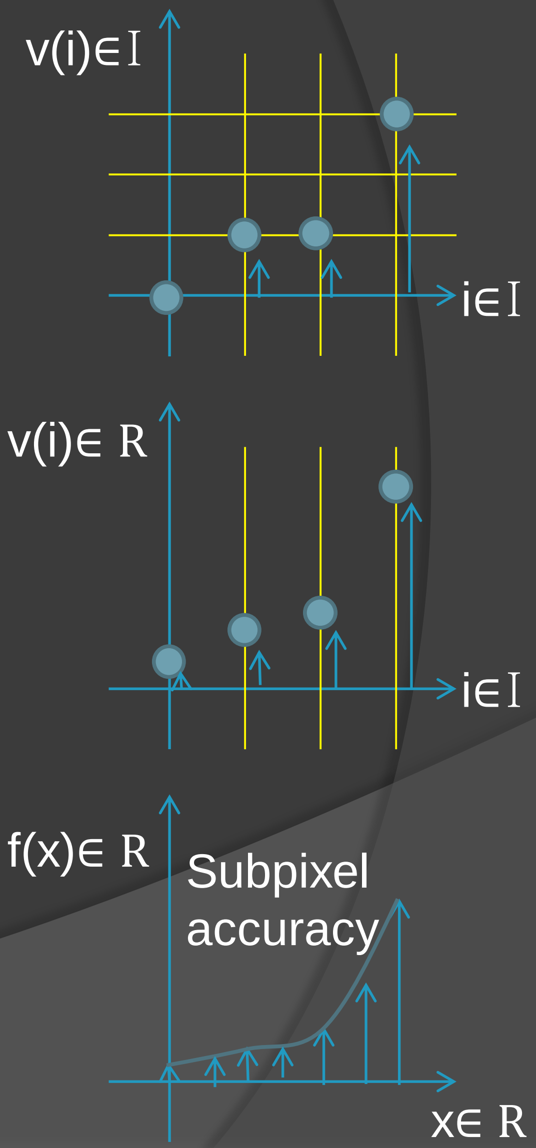


This figure comes from Computer Vision course (16720) at CMU in Fall 2010



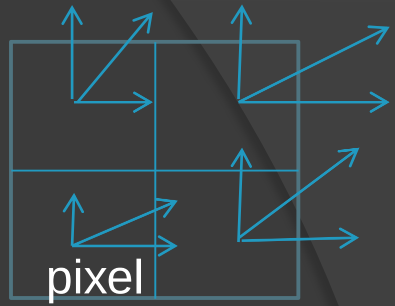
Domain and range

- Vector
  - Discrete to Discrete optimize problem
    - MRF, CRF, Belief propagation, dynamic programming, Greedy search
  - Discrete to Continuous optimize problem
    - Closed-form solution
    - EM algorithm
    - Gradient descent
    - …
- Function
  - Continuous to Continuous optimize problem
    - Euler-Lagrange equation
- In optical flow case,
  different types of variable: different advantage
  - The vector one: sharp edge
  - The function one: subpixel accuracy

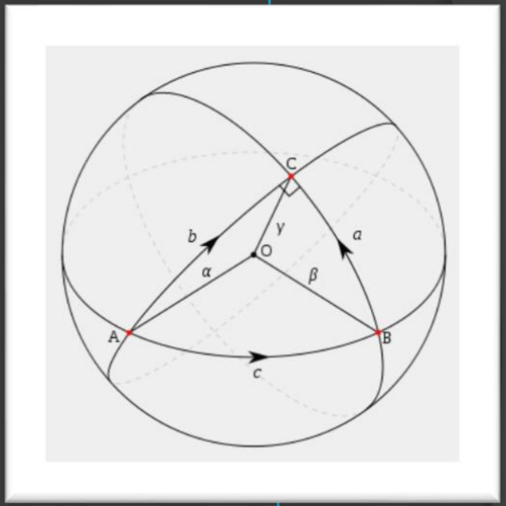


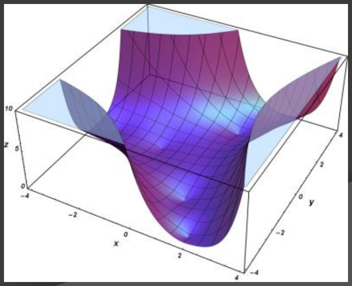
Variational model

- Calculus of variation(變分學)
  - Finding a function which maximizes or minimizes a functional (objective function)
    - solving a partial difference equation VS
find maximum of functional
    - ↕
    - solving the equation VS
maximize a function
  - For example, finding the function of a line with smallest length on one surface.
- Optical flow case
  - Find the function from continuous Image coordinate domain to continuous shift range which fit two frames best

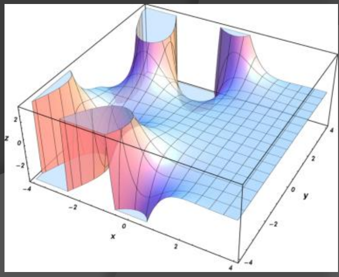


Flow array





x direction flow



y direction flow

Flow field

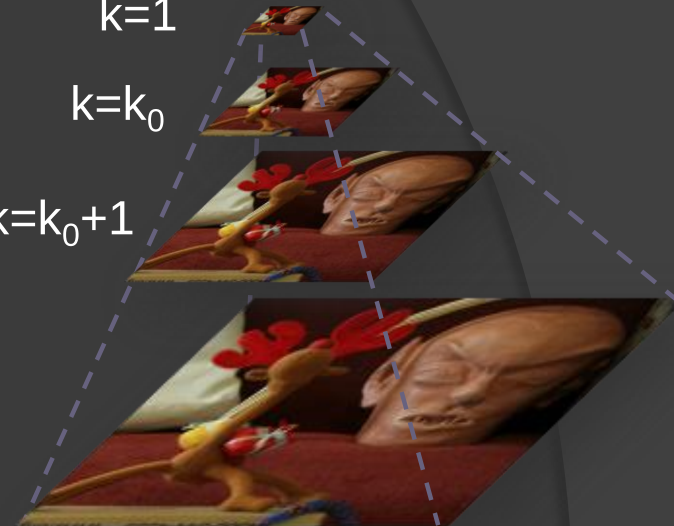
Why still could be "Linearize"?

Assume k=layer number

k=1

k=k_0

k=k_0+1



k=k_0+1  W^{k_0+1}

k=k_0  W^{k_0}

$$W^{k_0+1} = (u^{k_0+1}, v^{k_0+1})$$

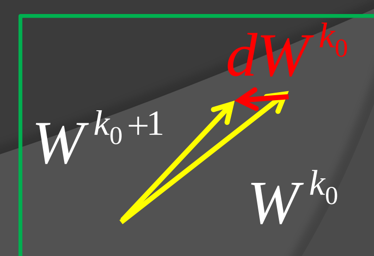
$$= W^{k_0} + dW^{k_0}$$

$$= (u^{k_0}, v^{k_0}) + (du^{k_0}, dv^{k_0})$$

- Fine pyramid scale (0.95 in this paper)
  - du^k, dv^k small enough.
- Linearize on du,dv instead of u, v:

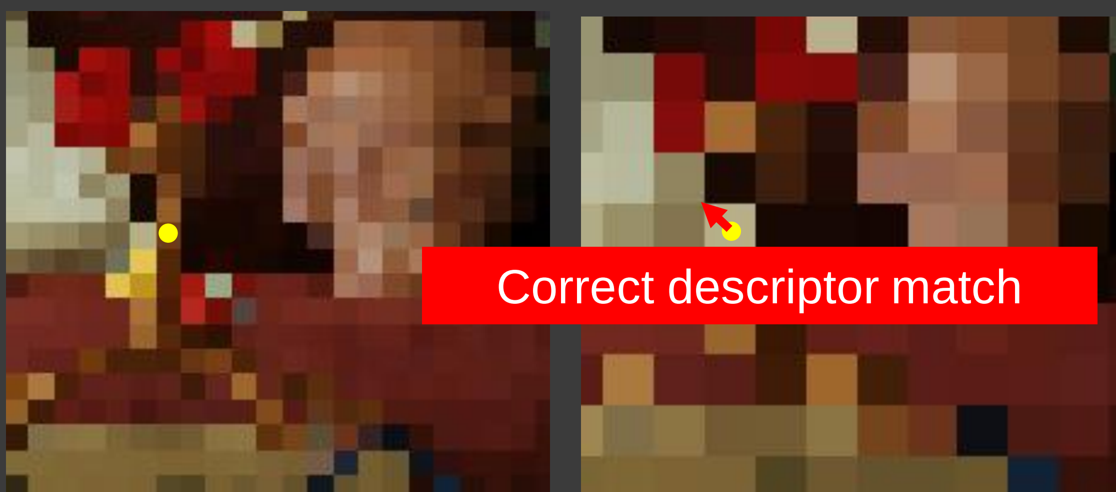
$$\mathrm{I}(\mathrm{X}+\mathrm{W}^{k+1}) - \mathrm{I}(\mathrm{X}) = \mathrm{I}(\mathrm{X}+\mathrm{W}^{k}+d\mathrm{W}^{k}) - \mathrm{I}(\mathrm{X})$$

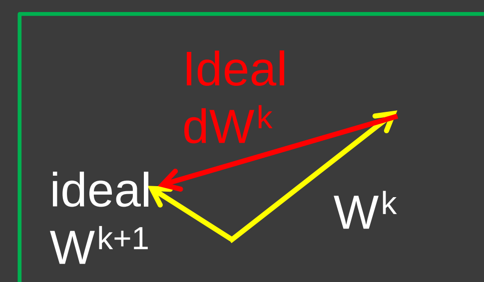
$$= \mathrm{I}(\mathrm{X}+\mathrm{W}^{k}) - \mathrm{I}(\mathrm{X}) + \mathrm{I}_x^k du^k + \mathrm{I}_y^k dv^k$$



Why still using "coarse-to-fine"?

- For large structure, the coarse-to-fine technique is more accurate than descriptor match.
- For small structure:



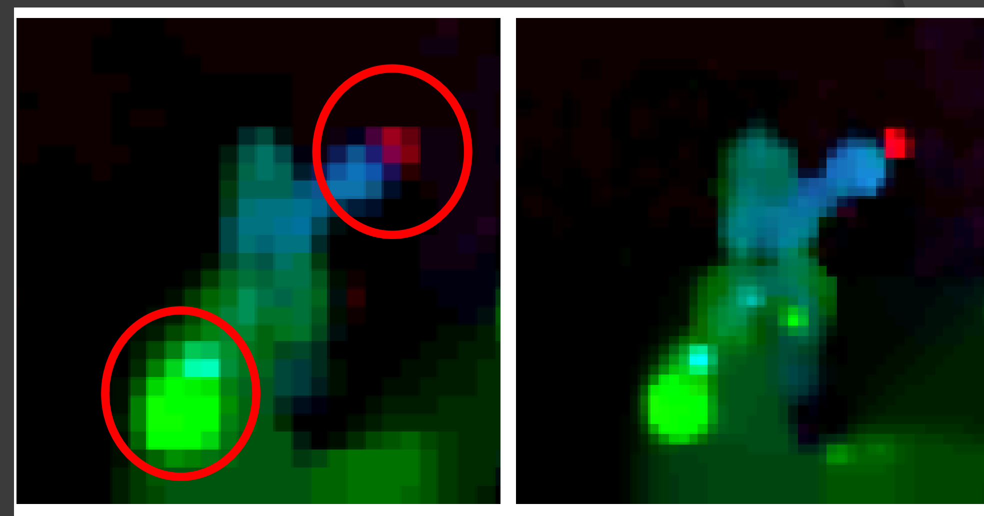


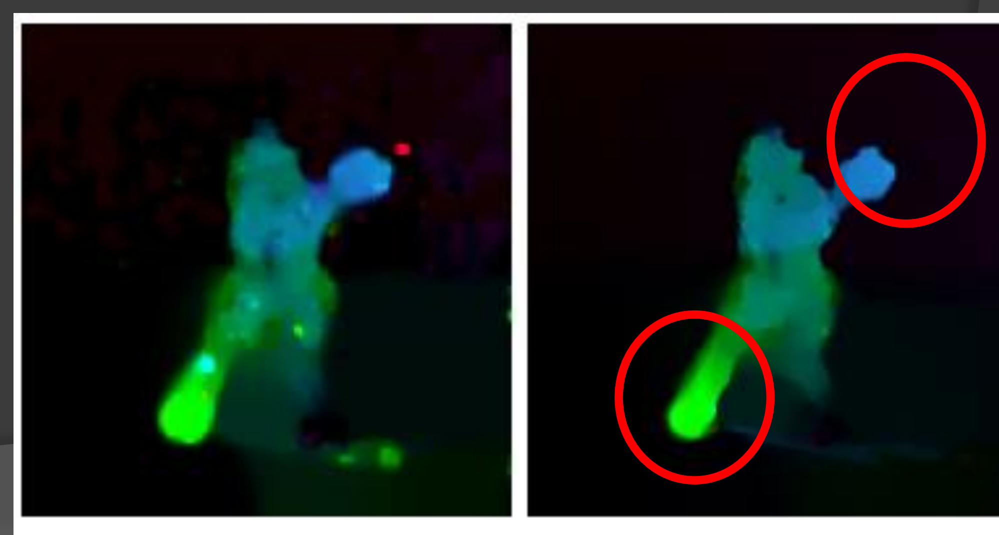
move distance > size scale
Ideal dW^k is too large to be found

descriptor match W_1
Ideal dW^k become much smaller

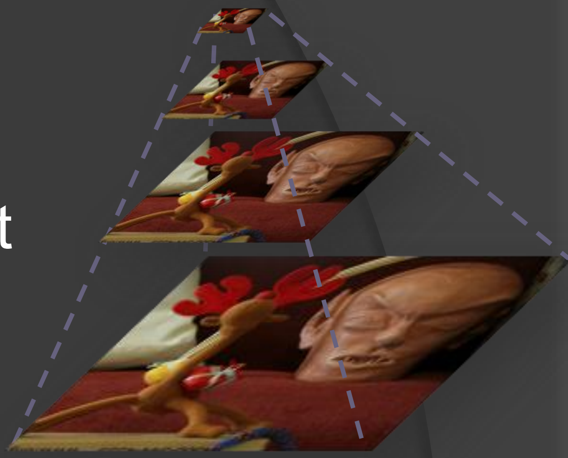
coarse-to-fine example with large displacement

- In coarse layer, the detail has been smoothed away.
- So it shows the result of descriptor matches on the feet, racket and some outlier on background.





Why continuous model?

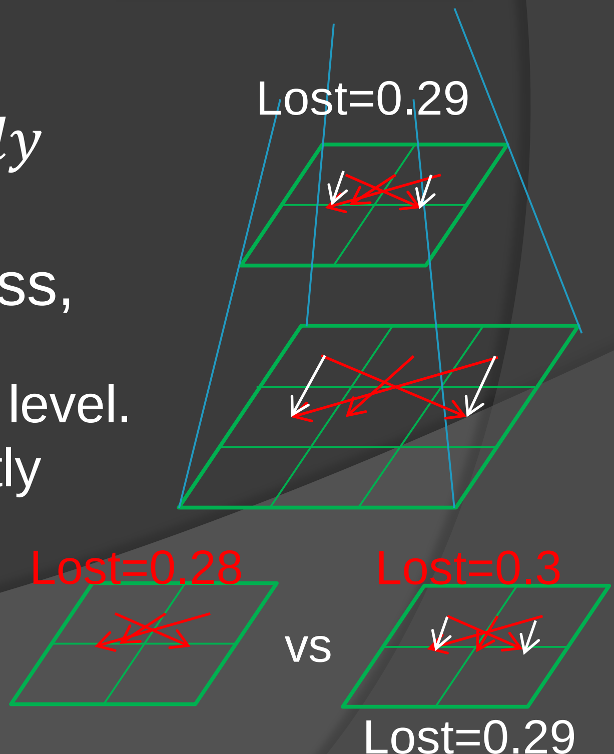


- We have to discretize in the end, why don't we discretize it at the first place?
- traditional discrete objective function:
  $\mathrm{E}(\boldsymbol{W}) = \sum_{x,y} E(u(x,y,t_0), v(x,y,t_0))$
  continuous objective function (variational model):

$$\mathrm{E}(\boldsymbol{W}) = \iint E(u(x,y,t_0), v(x,y,t_0))dx, dy$$

- In the coarse-to-fine optimization process,
  - Discrete objective function can't see the subpixel accuracy difference until the finer level.
  - The decision of coarse level will significantly affect the fine level.

Lost are different
But the discrete objective function can't see the difference



Optimize process

$$E(\mathbf{w}) = E_{\text{color}}(\mathbf{w}) + \gamma E_{\text{gradient}}(\mathbf{w}) + \alpha E_{\text{smooth}}(\mathbf{w}) + \beta E_{\text{match}}(\mathbf{w}, \mathbf{w}_1) + E_{\text{desc}}(\mathbf{w}_1),$$

- Euler-Lagrange equation: $L_x(t, q(t), q'(t)) - \frac{\mathrm{d}}{\mathrm{d}t} L_v(t, q(t), q'(t)) = 0.$

$$\begin{aligned}
&\Psi'\left(I_z^2\right) I_z I_x + \gamma \Psi'\left(I_{xz}^2 + I_{yz}^2\right)\left(I_{xx} I_{xz} + I_{xy} I_{yz}\right) \\
&\quad + \beta\, \rho\, \Psi'\left((u-u_1)^2 + (v-v_1)^2\right)(u - u_1) \\
&\quad - \alpha \operatorname{div}\left(\Psi'\left(|\nabla u|^2 + |\nabla v|^2\right) \nabla u\right) = 0 \\
&\Psi'\left(I_z^2\right) I_z I_y + \gamma \Psi'\left(I_{xz}^2 + I_{yz}^2\right)\left(I_{xy} I_{xz} + I_{yy} I_{yz}\right) \\
&\quad + \beta\, \rho\, \Psi'\left((u-u_1)^2 + (v-v_1)^2\right)(v - v_1) \\
&\quad - \alpha \operatorname{div}\left(\Psi'\left(|\nabla u|^2 + |\nabla v|^2\right) \nabla v\right) = 0,
\end{aligned}$$

$$\begin{array}{ll}
I_x := \partial_x I_2(\mathbf{x}+\mathbf{w}) & I_{xy} := \partial_{xy} I_2(\mathbf{x}+\mathbf{w}) \\
I_y := \partial_y I_2(\mathbf{x}+\mathbf{w}) & I_{yy} := \partial_{yy} I_2(\mathbf{x}+\mathbf{w}) \\
I_z := I_2(\mathbf{x}+\mathbf{w}) - I_1(\mathbf{x}) & I_{xz} := \partial_x I_z \\
I_{xx} := \partial_{xx} I_2(\mathbf{x}+\mathbf{w}) & I_{yz} := \partial_y I_z .
\end{array}$$

$$
\mathrm{I}_z^{k+1} = \mathrm{I}(\mathrm{X}+\mathrm{W}^{k+1}) - \mathrm{I}(\mathrm{X}) = \mathrm{I}(\mathrm{X}+\mathrm{W}^{k}+d\mathrm{W}^{k}) - \mathrm{I}(\mathrm{X})
$$

$$
= \mathrm{I}(\mathrm{X}+\mathrm{W}^{k}) + \mathrm{I}_x^k du^k + \mathrm{I}_y^k dv^k - \mathrm{I}(\mathrm{X}) = \mathrm{I}_z^k + \mathrm{I}_x^k du^k + \mathrm{I}_y^k dv^k
$$

- It is like to warp image2 according to the W^k at every scale, and compute flow between wrapped image2 and image1

$$
\begin{aligned}
E^k(du^k, dv^k) = &\int_\Omega \Psi\left((I_x^k du^k + I_y^k dv^k + I_z^k)^2\right)\, d\mathbf{x} \\
&+\gamma \int_\Omega \Psi\left((I_{xx}^k du^k + I_{xy}^k dv^k + I_{xz}^k)^2\right)\, d\mathbf{x} \\
&+\gamma \int_\Omega \Psi\left((I_{xy}^k du^k + I_{yy}^k dv^k + I_{yz}^k)^2\right)\, d\mathbf{x} \\
&+\beta \int_\Omega \Psi\left((u^k + du^k - u_1)^2 + (v^k + dv^k - v_1)^2\right)\, d\mathbf{x} \\
&+\alpha \int_\Omega \Psi\left(|\nabla(u^k + du^k)|^2 + |\nabla(v^k + dv^k)|^2\right).
\end{aligned}
$$

$$
\begin{aligned}
&\Psi_1' I_x^k (I_z^k + I_x^k du^k + I_y^k dv^k) + \beta\rho\Psi_3'(u^k + du^k - u_1) \\
&\quad +\gamma\, \Psi_2' I_{xx}^k (I_{xz}^k + I_{xx}^k du^k + I_{xy}^k dv^k) \\
&\quad +\gamma\, \Psi_2' I_{xy}^k (I_{yz}^k + I_{xy}^k du^k + I_{yy}^k dv^k) \\
&\quad -\alpha\, \mathrm{div}\left(\Psi_4' \nabla(u^k + du^k)\right) = 0 \\
&\Psi_1' I_y^k (I_z^k + I_x^k du^k + I_y^k dv^k) + \beta\rho\Psi_3'(v^k + dv^k - v_1) \\
&\quad +\gamma\, \Psi_2' I_{xy}^k (I_{xz}^k + I_{xx}^k du^k + I_{xy}^k dv^k) \\
&\quad +\gamma\, \Psi_2' I_{yy}^k (I_{yz}^k + I_{xy}^k du^k + I_{yy}^k dv^k) \\
&\quad -\alpha\, \mathrm{div}\left(\Psi_4' \nabla(v^k + dv^k)\right) = 0
\end{aligned}
$$

$$
\begin{aligned}
\Psi_1' &:= \Psi'\left((I_z^k + I_x^k du^k + I_y^k dv^k)^2\right) \\
\Psi_2' &:= \Psi'\left((I_{xz}^k + I_{xx}^k du^k + I_{xy}^k dv^k)^2 \right. \\
&\qquad \left. + (I_{yz}^k + I_{xy}^k du^k + I_{yy}^k dv^k)^2\right) \\
\Psi_3' &:= \Psi'\left((u^k + du^k - u_1)^2 + (v^k + dv^k - v_1)^2\right) \\
\Psi_4' &:= \Psi'\left(|\nabla(u^k + du^k)|^2 + |\nabla(v^k + dv^k)|^2\right).
\end{aligned}
$$

- E^k is convex and can be globally optimize.
  - By making many assumptions and approximations at the coarse-to-fine step.
  - follow the spirit of graduated non-convex optimization.
- If we don't use continuous model
  - How to discretize the equation at each scale without introducing any inconsistent discretizaion artifact?
  - the relationships between flows on each scale are undefined…

Advantage

- Relatively simple, robust and fast
  - Provide source code of optical flow and GPU accelerated version code for finding point trajectory.
  - For 640x480 image:143s (CPU version) -> 1.84s (GPU version) suitable for video usage.
  - Point trajectory derived from this method induces or refines some new motion segment and multi-object tracking techniques.

	Warping [11]	LDOF	SIFT flow [23]
Fig.6, row 1, 384 × 288	7s	18s	99s
Fig.6, row 2, 373 × 485	12s	29s	167s
Fig.6, row 3, 450 × 350	10s	44s	144s
Fig.6, row 4, 640 × 480	21s	80s	67s*
Fig.6, row 5, 530 × 380	13s	39s	177s

Disadvantage

- The most important disadvantage:
  LDOF directly incorporate the match result of descriptor into the energy function.
  - Descriptor match can't be perfect:
    introduce the bias of energy function
  - Descriptor match can provide the possible optimize direction, but we have to choose the best result only according to unbiased energy function.
    - This concept is realized by "Motion Detail Preserving Optical Flow Estimation PAMI 2012"
      - Achieve the best result in Middlebury evaluation set and also with ability to deal with large displacement.

Disadvantage

- The descriptor match process doesn’t merge into optimize process and biased energy function, so in order to
  - 1. suppress the false match result
  - 2. reduce the false flow caused by occlusion near large displacement
  - 3. reduce the Local Ambiguity / Aperture problem
- It needs to increase the weight of smoothness term(α). It brings about two problems:
  - 1. Blur image boundary
  - 2. When background is close to the foreground, the smallest cost will become staying at the same position without any moving.

Disadvantage

- Although trying hard to suppress, there are still some of them can't be compensate.

Motion blur

False match

Smooth surface

Blur boundary

Remain at the same position