ECE5554 – Computer Vision
Lecture 5b – Optical Flow

Creed Jones, PhD

Today's Objectives

- Parametric Estimation of Motion (§8.2)
- Robust Estimation
- Optical Flow (§8.4)
  - Image Warping

Parametric motion estimation

- As mentioned in the last lecture, tracking is possible for many geometric transformations between consecutive images
- Consider how to estimate the motion parameters
  - the coefficients of the geometric transformation matrix

Global (parametric) motion models

Motion between frames is more than just rigid translation, <u>but can still be modeled by a parametric equation</u>

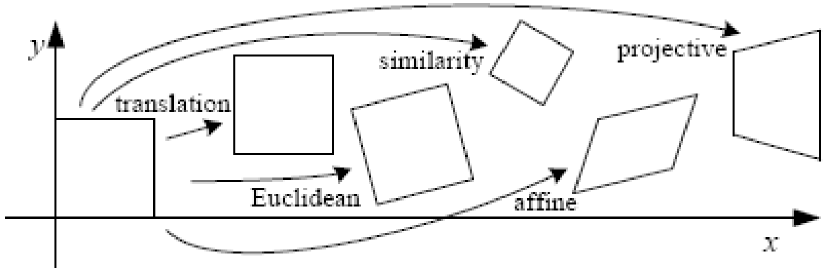
2D Models:

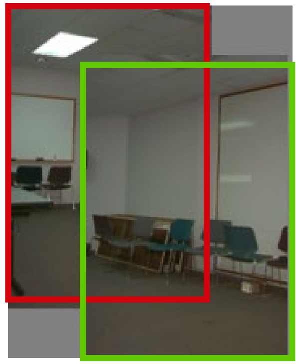
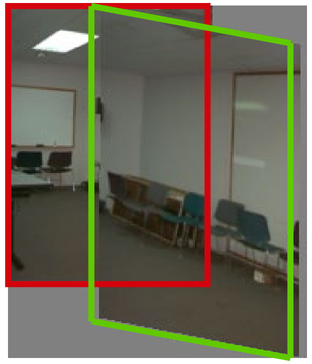
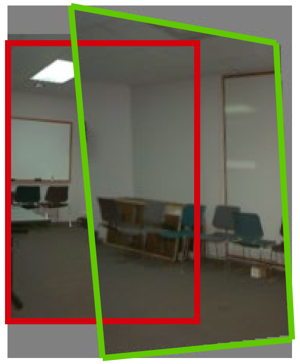
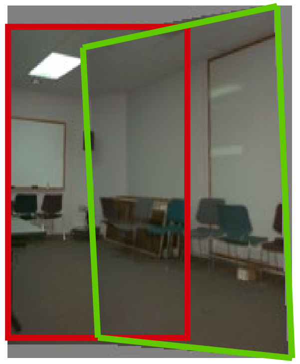
- Affine
- Quadratic
- Planar projective transform (Homography)

3D Models:

- Instantaneous camera motion models
- Homography+epipole
- Plane+Parallax

Motion models



Translation	Affine	Perspective	3D rotation
			
2 unknowns	6 unknowns	8 unknowns	3 unknowns

The Brightness Constraint

- Brightness Constancy Equation:

$$J(x,y) \approx \boxed{I(x+u(x,y),\, y+v(x,y))}$$

Or, equivalently, minimize :

$$E(u,v) = (J(x,y) - I(x+u, y+v))^2$$

Linearizing (assuming small *(u,v)*) using Taylor series expansion:

$$J(x,y) \approx \boxed{I(x,y) + I_x(x,y)\cdot u(x,y) + I_y(x,y)\cdot v(x,y)}$$

Gradient Constraint (or the Optical Flow Constraint)

$$E(u,v) = (I_x \cdot u + I_y \cdot v + I_t)^2$$

Minimizing:

$$\frac{\partial E}{du} = \frac{\partial E}{dv} = 0$$

$$I_x(I_x u + I_y v + I_t) = 0$$

$$I_y(I_x u + I_y v + I_t) = 0$$

In general

$$I_x, I_y \neq 0$$

Hence,

$$I_x \cdot u + I_y \cdot v + I_t \approx 0$$

Example: Affine Motion

$$u(x,y) = a_1 + a_2 x + a_3 y$$
$$v(x,y) = a_4 + a_5 x + a_6 y$$

- Substituting into the Brightness Constancy Equation:

$$\boxed{I_x \cdot u + I_y \cdot v + I_t \approx 0} \quad \boxed{I_x(a_1 + a_2 x + a_3 y) + I_y(a_4 + a_5 x + a_6 y) + I_t \approx 0}$$

Each pixel provides 1 linear constraint in 6 *global* unknowns

Least Square Minimization (over all pixels):

$$\boxed{Err(\vec{a}) = \sum \left[I_x(a_1 + a_2 x + a_3 y) + I_y(a_4 + a_5 x + a_6 y) + I_t \right]^2}$$

Other 2D Motion Models

Quadratic – instantaneous approximation to planar motion

$$u = q_1 + q_2 x + q_3 y + q_7 x^2 + q_8 xy$$
$$v = q_4 + q_5 x + q_6 y + q_7 xy + q_8 y^2$$

Projective – exact planar motion

$$x' = \frac{h_1 + h_2 x + h_3 y}{h_7 + h_8 x + h_9 y}$$

$$y' = \frac{h_4 + h_5 x + h_6 y}{h_7 + h_8 x + h_9 y}$$

and

$$u = x' - x, \quad v = y' - y$$

3D Motion Models

Instantaneous camera motion:

Global parameters: $\boxed{\Omega_X, \Omega_Y, \Omega_Z, T_X, T_Y, T_Z}$

Local Parameter: $\boxed{Z(x, y)}$

$$u = -xy\Omega_X + (1 + x^2)\Omega_Y - y\Omega_Z + (T_X - T_Z x)/Z$$
$$v = -(1 + y^2)\Omega_X + xy\Omega_Y - x\Omega_Z + (T_Y - T_Z x)/Z$$

Homography+Epipole

Global parameters: $\boxed{h_1, \ldots, h_9, t_1, t_2, t_3}$

Local Parameter: $\boxed{\gamma(x, y)}$

$$x' = \frac{h_1 x + h_2 y + h_3 + \gamma t_1}{h_7 x + h_8 y + h_9 + \gamma t_3}$$
$$y' = \frac{h_4 x + h_5 y + h_6 + \gamma t_1}{h_7 x + h_8 y + h_9 + \gamma t_3}$$

and : $u = x' - x, \quad v = y' - y$

Residual Planar Parallax Motion

Global parameters: $\boxed{t_1, t_2, t_3}$

Local Parameter: $\boxed{\gamma(x, y)}$

$$u = x^w - x = \frac{\gamma}{1 + \gamma t_3}(t_3 x - t_1)$$
$$v = y^w - x = \frac{\gamma}{1 + \gamma t_3}(t_3 y - t_2)$$

Patch matching (revisited)

- How do we determine correspondences?
  - *block matching* or *SSD* (sum squared differences)

$$E(x,y;d) = \sum_{(x',y')\in N(x,y)} [I_L(x'+d,y') - I_R(x',y')]^2$$

Correlation and SSD

- For larger displacements, do template matching
  - Define a small area around a pixel as the template
  - Match the template against each pixel within a search area in next image.
  - Use a match measure such as correlation, normalized correlation, or sum-of-squares difference
  - Choose the maximum (or minimum) as the match
  - Sub-pixel estimate (Lucas-Kanade)

Discrete Search vs. Gradient Based

- Consider image I translated by u_0, v_0

$$
\begin{aligned}
I_0(x, y) &= I(x, y) \\
I_1(x + u_0, y + v_0) &= I(x, y) + \eta_1(x, y) \\
E(u, v) &= \sum_{x,y} \left(I(x, y) - I_1(x + u, y + v)\right)^2 \\
&= \sum_{x,y} \left(I(x, y) - I(x - u_0 + u, y - v_0 + v) - \eta_1(x, y)\right)^2
\end{aligned}
$$

- The discrete search method simply searches for the best estimate.
- The gradient method linearizes the intensity function and solves for the estimate

Look more closely at the Shi-Tomasi feature tracker

1. Find good features (min eigenvalue of 2×2 Hessian)
2. Use Lucas-Kanade to track with pure translation
3. Use affine registration with first feature patch
4. Terminate tracks whose dissimilarity gets too large
5. Start new tracks when needed

Tracking results

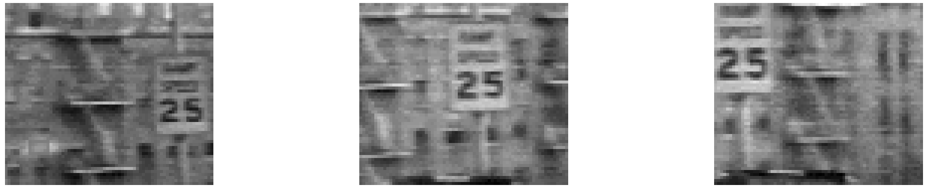


Figure 1: Three frame details from Woody Allen's *Manhattan*. The details are from the 1st, 11th, and 21st frames of a subsequence from the movie.

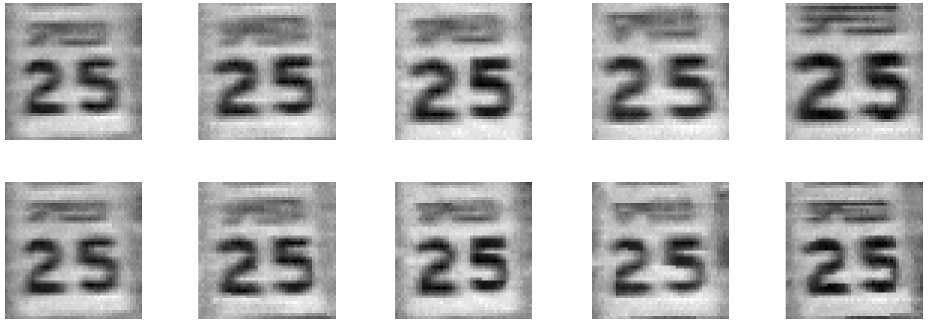


Figure 2: The traffic sign windows from frames 1,6,11,16,21 as tracked (top), and warped by the computed deformation matrices (bottom).

Tracking - dissimilarity

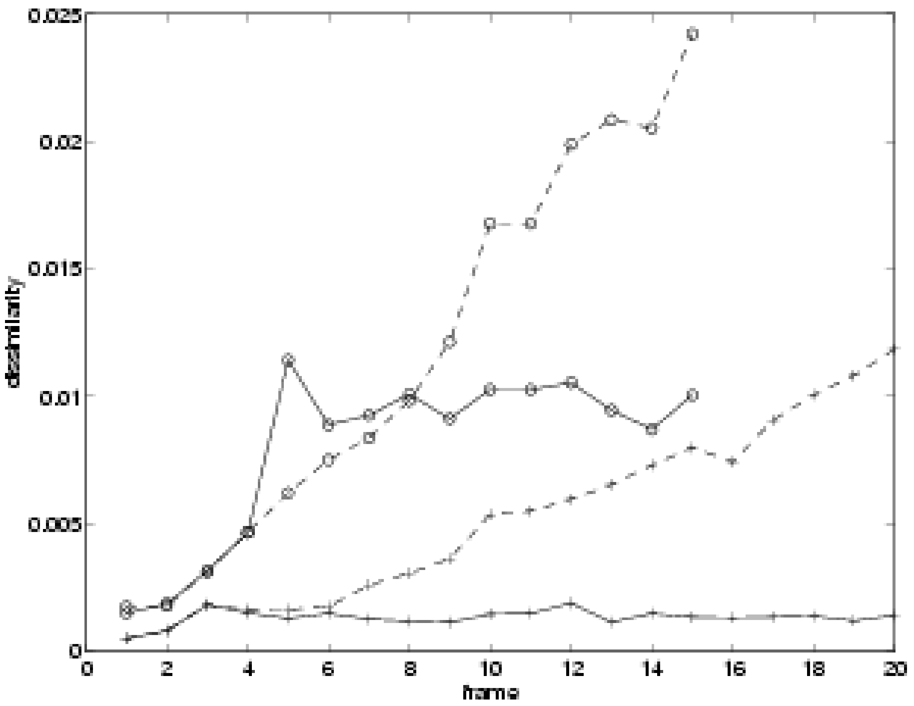


Figure 3: Pure translation (dashed) and affine motion (solid) dissimilarity measures for the window sequence of figure 1 (plusses) and 4 (circles).

Tracking results

Figure 13: Labels of some of the features in figure 11.

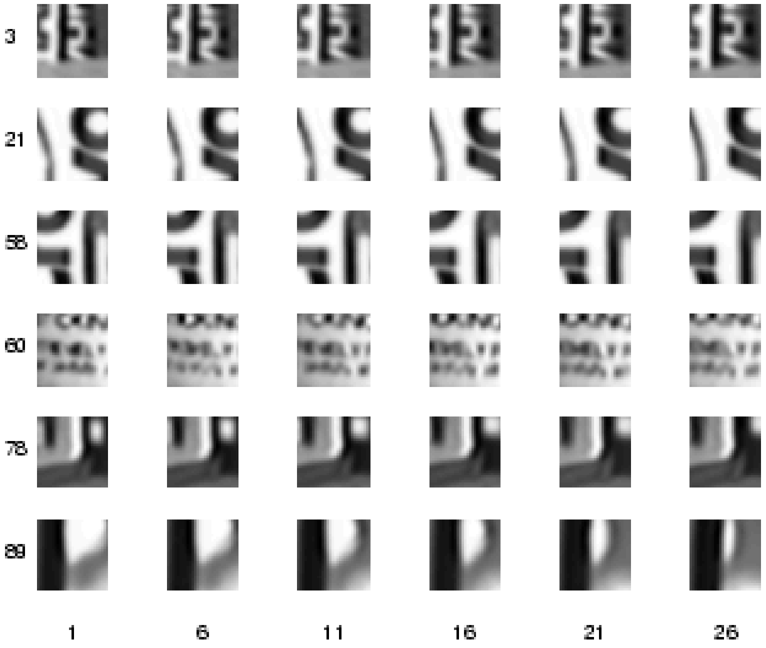


Figure 14: Six sample features through six sample frames.

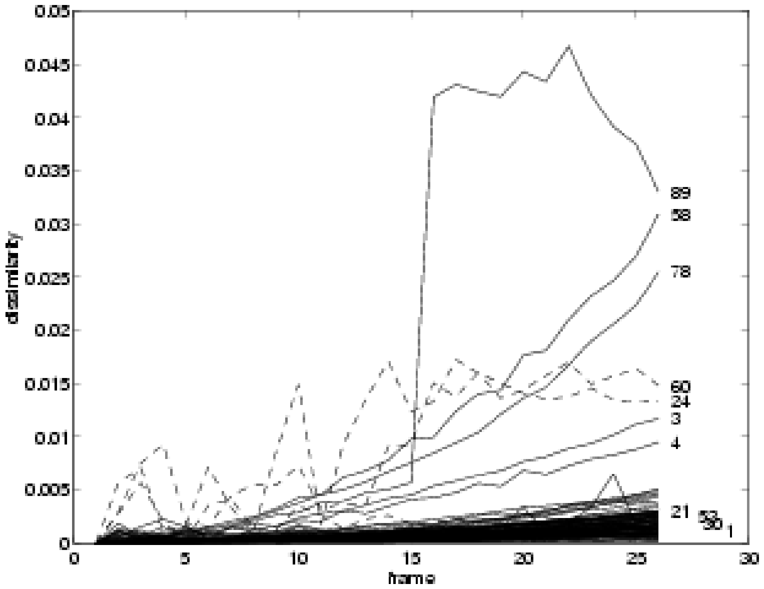


Figure 15: Affine motion dissimilarity for the features in figure 11. Notice the good discrimination between good and bad features. Dashed plots indicate aliasing (see text).

Features 24 and 60 deserve a special discussion, and

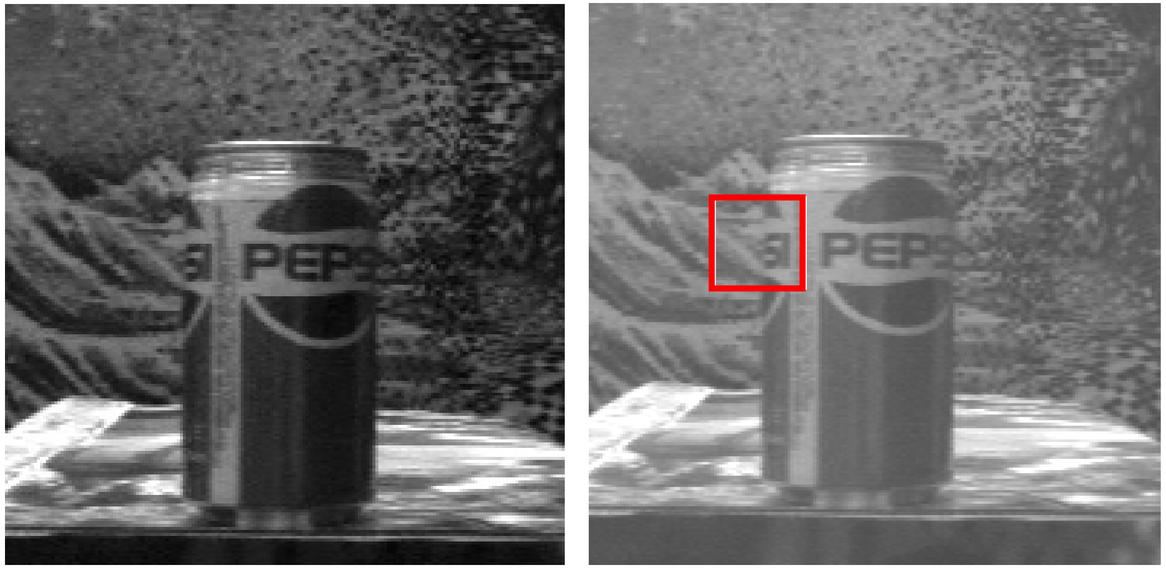
Correlation Window Size

- Small windows lead to more false matches
- Large windows are better this way, but…
  - Neighboring flow vectors will be more correlated (since the template windows have more in common)
  - Flow resolution also lower (same reason)
  - More expensive to compute

- Small windows are good for local search:
  more detailed and less smooth (noisy?)
- Large windows good for global search:
  less detailed and smoother

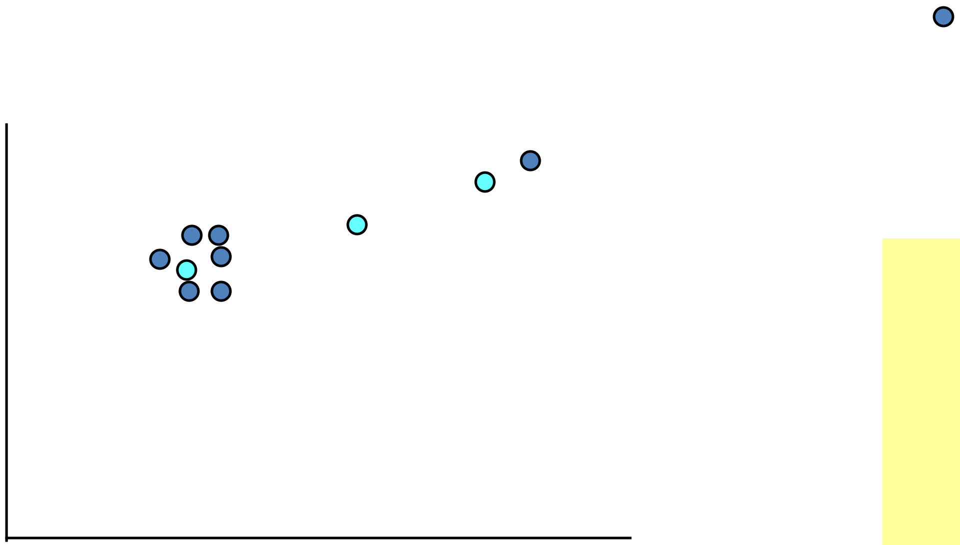
Robust Estimation

- Noise distributions are often non-Gaussian, having much heavier tails. Noise samples from the tails are called outliers.
- Sources of outliers (multiple motions):
  - specularities / highlights
  - jpeg artifacts / interlacing / motion blur
  - multiple motions (occlusion boundaries, transparency)



Robust Estimation

Standard Least Squares Estimation allows too much influence for outlying points



$$E(m) = \sum_i \rho(x_i)$$

$$\rho(x_i) = (x_i - m)^2$$

$$\text{Influence } \psi(x) = \frac{\partial \rho}{\partial x} = (x_i - m)$$

Robust Estimation

$$E_d(u_s, v_s) = \sum \rho\left(I_x u_s + I_y v_s + I_t\right)$$ Robust gradient constraint

$$E_d(u_s, v_s) = \sum \rho\left(I(x, y) - J(x + u_s, y + v_s)\right)$$ Robust SSD

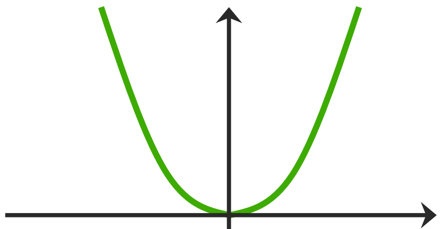
$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial \mathbf{f}}{\partial x_1} & \cdots & \dfrac{\partial \mathbf{f}}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f_1}{\partial x_1} & \cdots & \dfrac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial f_m}{\partial x_1} & \cdots & \dfrac{\partial f_m}{\partial x_n} \end{bmatrix}.$$

Robust Estimation

Problem: Least-squares estimators penalize deviations between data & model with quadratic error f^n (extremely sensitive to outliers)

error penalty function

$$\rho(\epsilon) = \epsilon^2$$



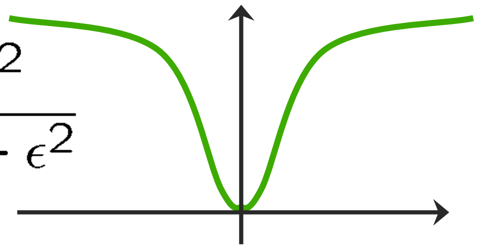
influence function

$$\psi(\epsilon) = \frac{\partial \rho(\epsilon)}{\partial \epsilon} = 2\,\epsilon$$

Redescending error functions (e.g., Geman-McClure) help to reduce the influence of outlying measurements.

error penalty function

$$\rho(\epsilon;\, s) = \frac{\epsilon^2}{s + \epsilon^2}$$



influence function

$$\psi(\epsilon;\, s) = \frac{2\,\epsilon\, s}{(s + \epsilon^2)^2}$$

HOW WELL DO THESE TECHNIQUES WORK?

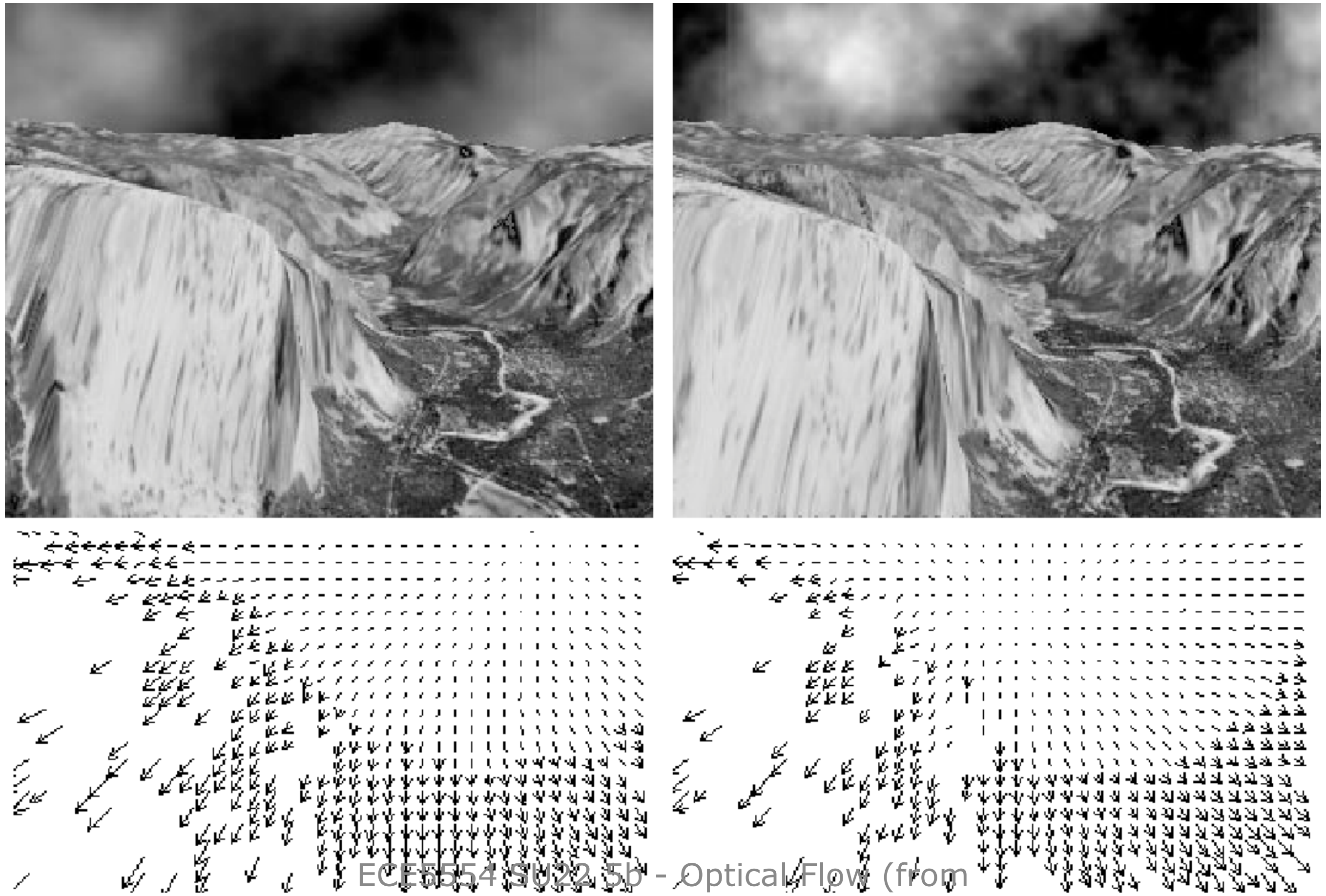
A Database and Evaluation Methodology for Optical Flow

Simon Baker, Daniel Scharstein, J.P Lewis, Stefan Roth, Michael Black, and Richard Szeliski

ICCV 2007

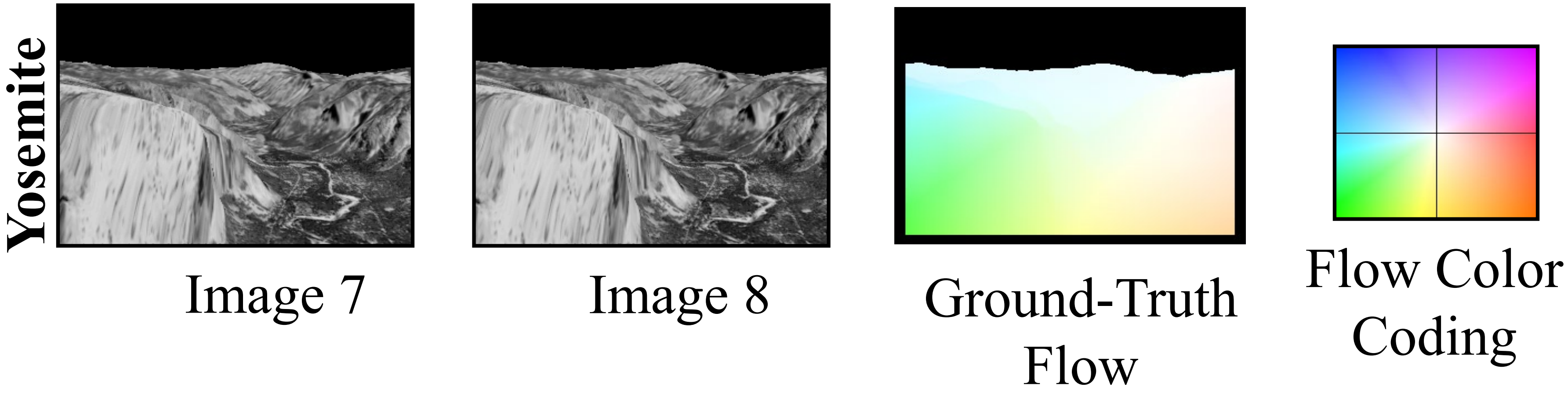
http://vision.middlebury.edu/flow/

Yosemite is a famous synthetic image sequence, widely used for motion estimation research



Limitations of Yosemite

- Only sequence used for quantitative evaluation



- Limitations:
  – Very simple and synthetic
  – Small, rigid motion
  – Minimal motion discontinuities/occlusions

Limitations of Yosemite

- Only sequence used for quantitative evaluation

Yosemite

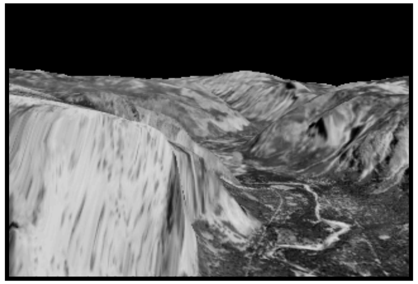


Image 7

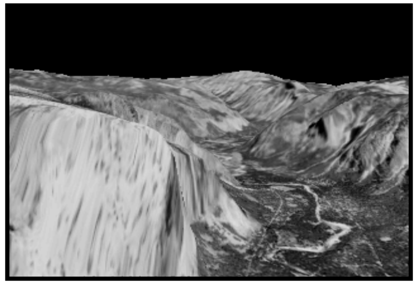
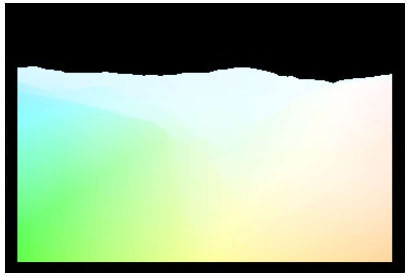
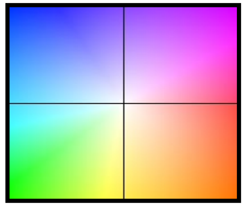


Image 8



Ground-Truth Flow

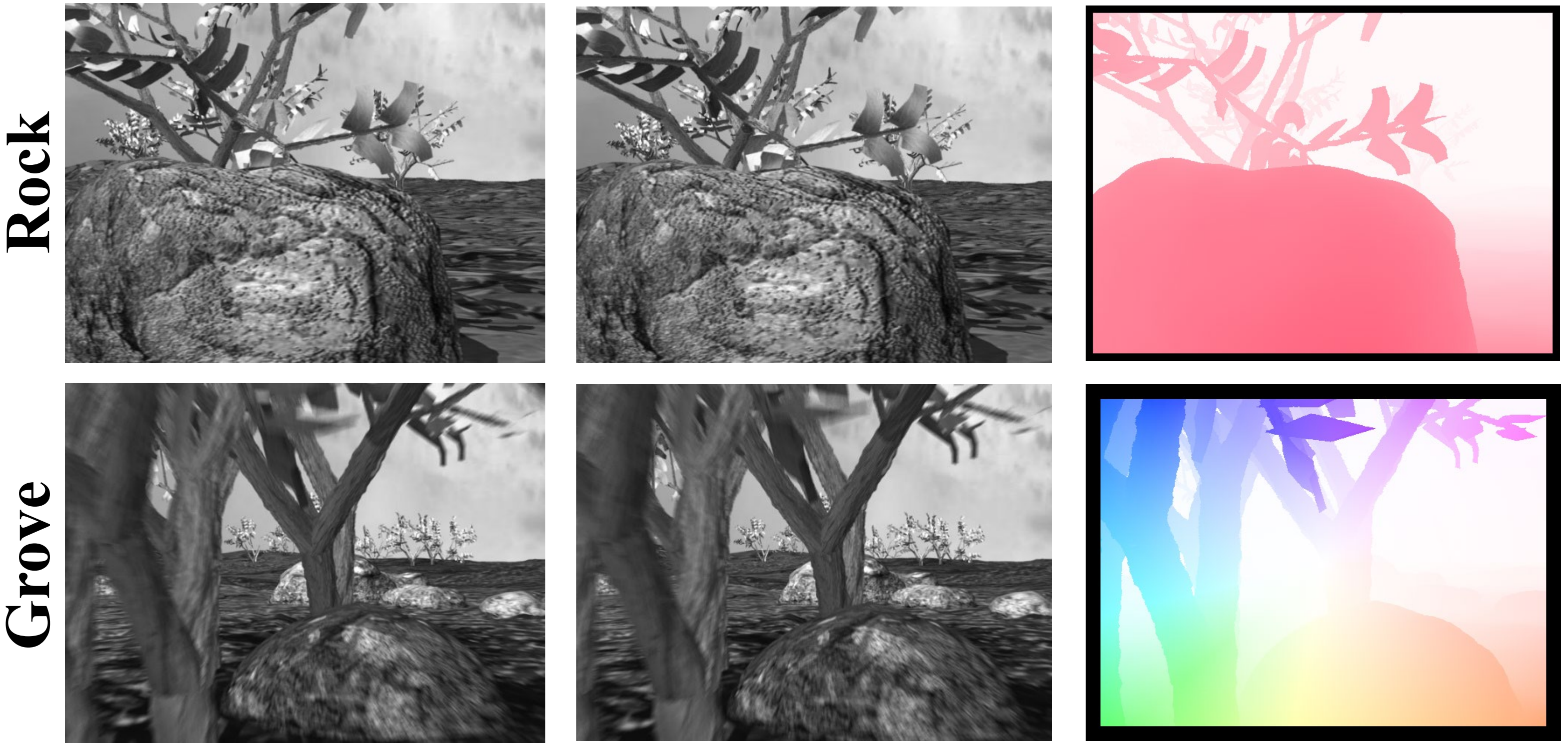


Flow Color Coding

- Current challenges:
  - Non-rigid motion
  - Real sensor noise
  - Complex natural scenes
  - Motion discontinuities
  - Need more challenging and more realistic benchmarks

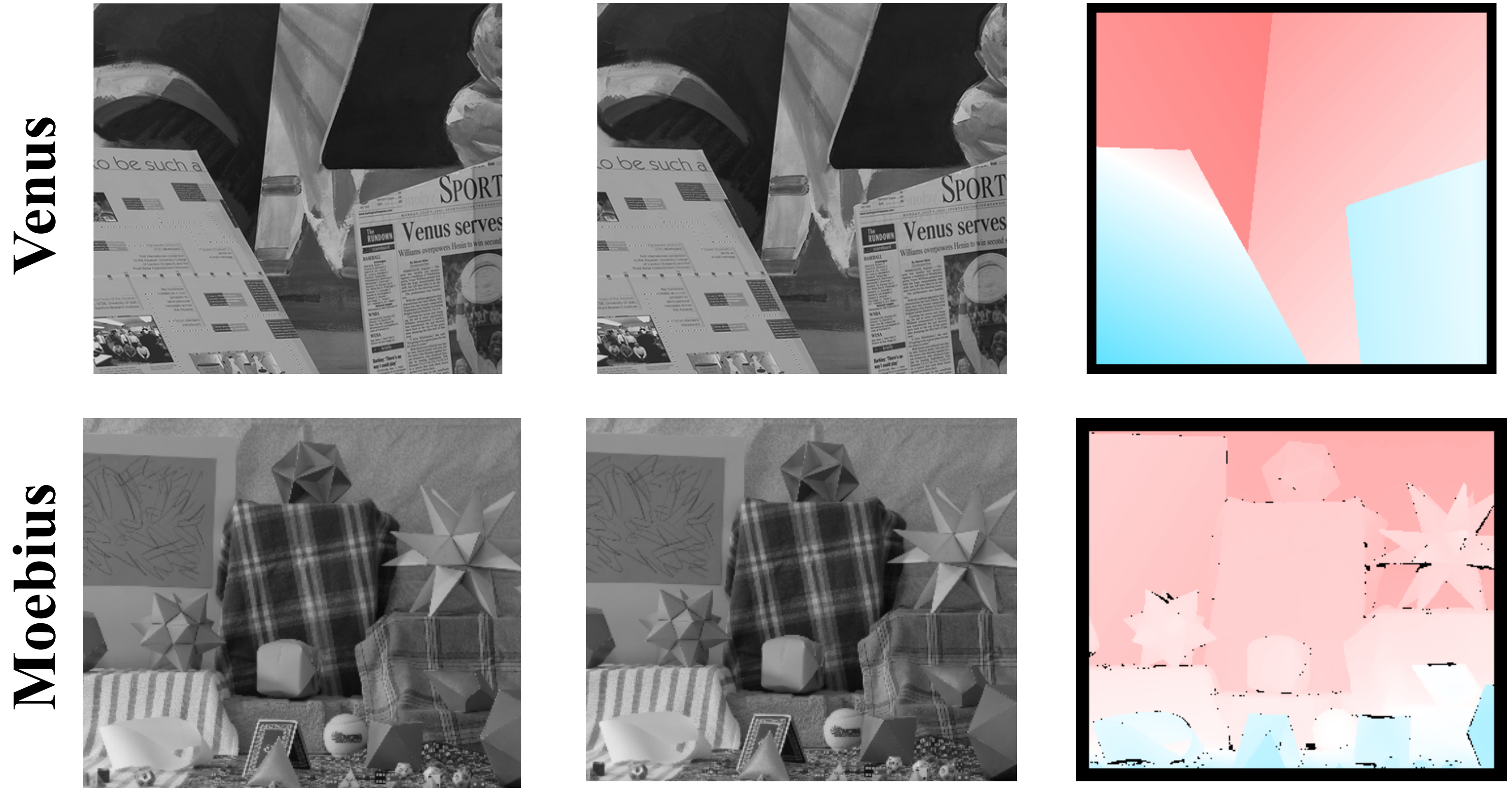
Realistic synthetic imagery

- Randomly generate scenes with “trees” and “rocks”
- Significant occlusions, motion, texture, and blur
- Rendered using Mental Ray and “lens shader” plugin



Modified stereo imagery

- Recrop and resample ground-truth stereo datasets to have appropriate motion for OF



Dense flow with hidden texture

- Paint scene with textured fluorescent paint
- Take 2 images: One in visible light, one in UV light
- Move scene in very small steps using robot
- Generate ground-truth by tracking the UV images

Visible

UV

Setup Lights Image Cropped

Experimental results

Algorithms:

- Pyramid LK: OpenCV-based implementation of Lucas-Kanade on a Gaussian pyramid
- Black and Anandan: Author's implementation
- Bruhn et al.: Our implementation
- MediaPlayerTM: Code used for video frame-rate upsampling in Microsoft MediaPlayer
- Zitnick et al.: Author's implementation

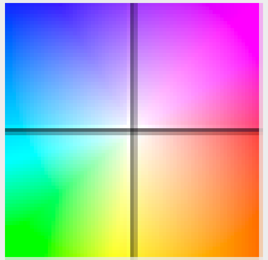
Optical flow evaluation results

Choose error measures: Average SD R1.0 R3.0 R5.0 A50 A75 A95

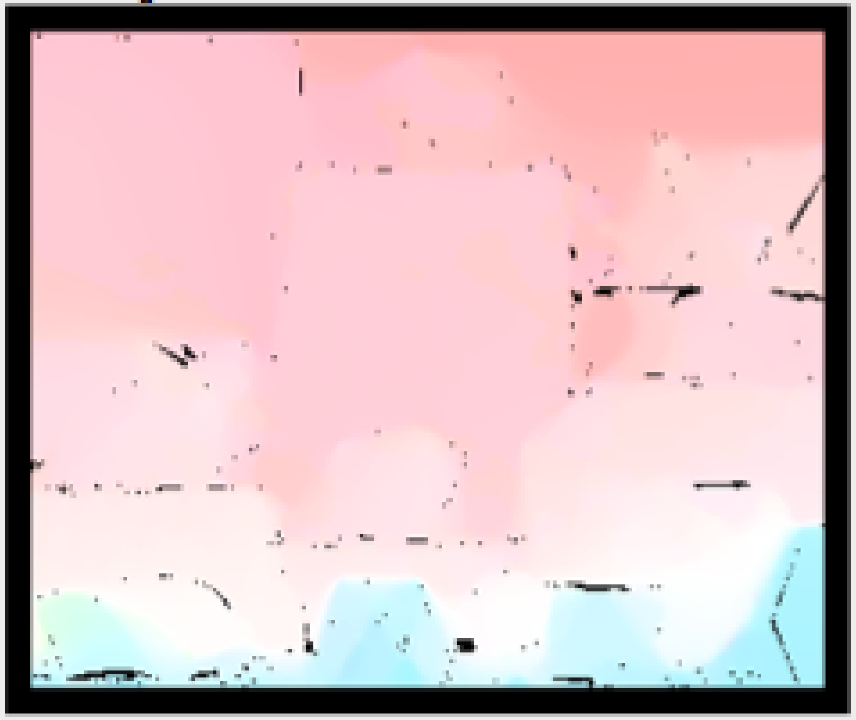
Average angle error	avg. rank	Dimetrodon (Hidden texture) GT im0 im1			Seashell (Hidden texture) GT im0 im1			Rock (Synthetic) GT im0 im1		
		all	disc	untext	all	disc	untext	all	disc	untext
Bruhn et al.	1.6	10.99 3	**9.41** 1	14.22 3	11.09 2	19.48 2	16.21 2	**6.14** 1	**17.41** 1	12.86 2
Black and Anandan	2.1	**9.26** 1	10.11 3	**12.08** 1	11.20 3	19.03 3	17.01 3	7.67 3	18.44 3	16.80 4
Pyramid LK	2.8	10.27 2	9.71 2	13.63 2	**9.46** 1	**18.62** 1	**12.07** 1	6.53 2	18.43 2	**10.95** 1
MediaPlayer™	4.1	15.82 4	26.42 4	16.96 4	23.18 4	27.71 5	21.78 4	9.44 4	22.25 4	15.03 3
Zitnick et al.	4.2	30.10 5	34.27 5	31.58 5	29.07 5	27.55 4	21.78 4	12.38 5	23.93 5	17.59 5

Average angle error	Grove (Synthetic) GT im0 im1			Yosemite (Synthetic) GT im0 im1			Venus (Stereo) GT im0 im1			Moebius (Stereo) GT im0 im1		
	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
Bruhn et al.	**6.32** 1	**12.41** 1	**10.98** 1	**1.69** 1	**2.86** 1	**1.05** 1	8.73 2	31.46 2	8.15 2	**5.85** 1	10.12 2	8.80 2
Black and Anandan	7.89 2	13.55 2	13.96 4	2.65 2	4.18 2	1.88 2	**7.64** 1	**30.13** 1	**7.31** 1	7.05 2	**10.02** 1	**8.41** 1
Pyramid LK	8.14 3	15.08 3	12.78 2	5.22 3	6.64 3	4.29 3	14.61 4	36.18 4	24.67 5	12.98 5	13.85 4	20.61 5
MediaPlayer™	10.99 4	18.15 5	13.64 3	11.09 4	17.16 4	10.66 5	15.48 5	43.56 5	15.09 4	9.98 4	15.04 5	9.47 3
Zitnick et al.	12.55 5	15.56 4	17.35 5	18.50 5	28.00 5	9.41 4	11.42 3	31.46 2	11.12 3	9.88 3	12.83 3	11.28 4

Color encoding of flow vectors



Flow image



Error image

Conclusions

- Difficulty: Data substantially more challenging than Yosemite
- Diversity: Substantial variation in difficulty across the various datasets
- Motion GT vs Interpolation: Best algorithms for one are not the best for the other
- Comparison with Stereo: Performance of existing flow algorithms appears weak

Today’s Objectives

- Parametric Estimation of Motion (§8.2)
- Robust Estimation
- Optical Flow (§8.4)
  - Image Warping