Visual Computing: Optical Flow

Prof. Marc Pollefeys
Last lecture

- DCT (JPEG) and DWT (JPEG2000) compression
- JPEG2000 encoding pipeline

Figure 2 Operation flow of the JPEG 2000 standard.
Visual Computing: Optical Flow

Prof. Marc Pollefeys
Optical Flow

- Brightness Constancy
- The Aperture problem
- Regularization
- Lucas-Kanade
- Coarse-to-fine
- Parametric motion models
- Direct depth
- SSD tracking
- Robust flow
- Bayesian flow
Optical Flow: Where do pixels move to?
Motion is a basic cue

Motion can be the only cue for segmentation
Motion is a basic cue

Even impoverished motion data can elicit a strong percept
Applications

- tracking
- structure from motion
- motion segmentation
- stabilization
- compression
- Mosaicing
- ...

ETH
Optical Flow

• Brightness Constancy
• The Aperture problem
• Regularization
• Lucas-Kanade
• Coarse-to-fine
• Parametric motion models
• Direct depth
• SSD tracking
• Robust flow
• Bayesian flow
Definition of Optical Flow

OPTICAL FLOW = apparent motion of brightness patterns

Ideally, the optical flow is the projection of the three-dimensional velocity vectors on the image
Caution required

Two examples:

1. Uniform, rotating sphere
   \[ \Downarrow \]
   \[ \text{O.F.} = 0 \]

2. No motion, but changing lighting
   \[ \Downarrow \]
   \[ \text{O.F.} \neq 0 \]
Caution required
Mathematical formulation

\[ I(x, y, t) = \text{brightness at } (x, y) \text{ at time } t \]

**Brightness constancy assumption:**

\[ I(x + \frac{dx}{dt} \delta t, y + \frac{dy}{dt} \delta t, t + \delta t) = I(x, y, t) \]

**Optical flow constraint equation:**

\[ \frac{dI}{dt} = \frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} + \frac{\partial I}{\partial t} = 0 \]
Optical Flow

- Brightness Constancy
- The Aperture problem
- Regularization
- Lucas-Kanade
- Coarse-to-fine
- Parametric motion models
- Direct depth
- SSD tracking
- Robust flow
- Bayesian flow
The aperture problem

\[ u = \frac{dx}{dt}, \quad v = \frac{dy}{dt} \]

\[ I_x = \frac{\partial I}{\partial y}, \quad I_y = \frac{\partial I}{\partial y}, \quad I_t = \frac{\partial I}{\partial t} \]

\[ I_x u + I_y v + I_t = 0 \]

1 equation in 2 unknowns
The aperture problem

\[ ul_x + vl_y + I_t = 0 \]

one equation, two unknowns

**Figure 12-4.** Local information on the brightness gradient and the rate of change of brightness with time provides only one constraint on the components of the optical flow vector. The flow velocity has to lie along a straight line perpendicular to the direction of the brightness gradient. We can only determine the component in the direction of the brightness gradient. Nothing is known about the flow component in the direction at right angles.
Aperture problem and Normal Flow

The gradient constraint:

\[ I_x u + I_y v + I_t = 0 \]
\[ \nabla I \cdot \vec{U} = 0 \]

Defines a line in the \((u,v)\) space

Normal Flow:

\[ u_\perp = -\frac{I_t}{|\nabla I|} \frac{\nabla I}{|\nabla I|} \]
The aperture problem
Remarks
Apparently an aperture problem
What is Optic Flow, anyway?

• Estimate of observed projected motion field
• Not always well defined!
• Compare:
  – Motion Field (or Scene Flow)
    projection of 3-D motion field
  – Normal Flow
    observed tangent motion
  – Optic Flow
    apparent motion of the brightness pattern
    (hopefully equal to motion field)
• Consider Barber pole illusion
Planar motion examples

- Ideal motion of a plane

What is the motion here?  
translation in X  
translation in Z  
rotation around Z  
rotation around Y

Scene Flow:  
Normal Flow: undef 
Optic Flow: ?, probably 0
Optical Flow

- Brightness Constancy
- The Aperture problem
- Regularization
- Lucas-Kanade
- Coarse-to-fine
- Parametric motion models
- Direct depth
- SSD tracking
- Robust flow
- Bayesian flow
Horn & Schunck algorithm

Additional smoothness constraint:

\[ e_s = \iint ((u_x^2 + u_y^2) + (v_x^2 + v_y^2)) \, dx \, dy, \]

besides OF constraint equation term

\[ e_c = \iint (I_x u + I_y v + I_t)^2 \, dx \, dy, \]

minimize \( e_s + \lambda e_c \)
The Euler-Lagrange equations:

\[ F_u - \frac{\partial}{\partial x} F_{ux} - \frac{\partial}{\partial y} F_{uy} = 0 \]
\[ F_v - \frac{\partial}{\partial x} F_{vx} - \frac{\partial}{\partial y} F_{vy} = 0 \]

In our case, 

\[ F = (u_x^2 + u_y^2) + (v_x^2 + v_y^2) + \lambda(I_x u + I_y v + I_t)^2, \]

so the Euler-Lagrange equations are

\[ \Delta u = \lambda(I_x u + I_y v + I_t)I_x, \]
\[ \Delta v = \lambda(I_x u + I_y v + I_t)I_y, \]

\[ \Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \]

is the Laplacian operator.
Remarks:

1. Coupled PDEs solved using iterative methods and finite differences

\[
\frac{\partial u}{\partial t} = \Delta u - \lambda (I_x u + I_y v + I_t) I_x, \\
\frac{\partial v}{\partial t} = \Delta v - \lambda (I_x u + I_y v + I_t) I_y,
\]

2. More than two frames allow a better estimation of $I_t$

3. Information spreads from corner-type patterns
Horn & Schunk, remarks

1. Errors at boundaries

2. Example of regularisation
   (selection principle for the solution of illposed problems)
Results of an enhanced system
Structure from motion with OF
Optical Flow

- Brightness Constancy
- The Aperture problem
- Regularization
- Lucas-Kanade
- Coarse-to-fine
- Parametric motion models
- SSD tracking
- Bayesian flow
Lucas-Kanade: Integrate over a Patch

Assume a single velocity for all pixels within an image patch

\[
E(u, v) = \sum_{x, y \in \Omega} \left( I_x(x, y)u + I_y(x, y)v + I_t \right)^2
\]

\[
\frac{dE(u, v)}{du} = \sum 2I_x(I_xu + I_yv + I_t) = 0
\]

\[
\frac{dE(u, v)}{dv} = \sum 2I_y(I_xu + I_yv + I_t) = 0
\]

Solve with:

\[
\begin{bmatrix}
\sum I_x^2 & \sum I_xI_y \\
\sum I_xI_y & \sum I_y^2
\end{bmatrix}
\begin{bmatrix}
u \\
v
\end{bmatrix}
= -\begin{bmatrix}
\sum I_xI_t \\
\sum I_yI_t
\end{bmatrix}
\]

On the LHS: sum of the 2x2 outer product tensor of the gradient vector

\[
\left( \sum \nabla I \nabla I^T \right) \vec{U} = -\sum \nabla II_t
\]
Lucas-Kanade: Singularities and the Aperture Problem

Let \( M = \sum (\nabla I)(\nabla I)^T \) and \( b = \begin{bmatrix} - \sum I_x I_t \\ - \sum I_y I_t \end{bmatrix} \)

- Algorithm: At each pixel compute \( U \) by solving \( MU = b \)

- \( M \) is singular if all gradient vectors point in the same direction
  -- e.g., along an edge
  -- of course, trivially singular if the summation is over a single pixel
  -- i.e., only normal flow is available (aperture problem)

- Corners and textured areas are OK

KLT feature tracker:
see “Good Features to Track”, *Shi and Tomasi, CVPR’94*, 1994, pp. 593 - 600.
Iterative Refinement

- Estimate velocity at each pixel using one iteration of Lucas and Kanade estimation
- Warp one image toward the other using the estimated flow field
  \textit{(easier said than done)}
- Refine estimate by repeating the process
Motion and Gradients

Consider 1-d signal; assume linear function of $x$

\[
\frac{dI}{dx} = - \frac{dI}{dt} \frac{dt}{u}
\]

\[
0 = I_x u + I_t
\]

\[
u = - \frac{I_t}{I_x}
\]
Iterative refinement

BUT!!
Optical Flow

- Brightness Constancy
- The Aperture problem
- Regularization
- Lucas-Kanade
- Coarse-to-fine
- Parametric motion models
- SSD tracking
- Bayesian flow
Limits of the (local) gradient method

1. Fails when intensity structure within window is poor
2. Fails when the displacement is large (typical operating range is motion of 1 pixel per iteration!)
   - Linearization of brightness is suitable only for small displacements

Also, brightness is not strictly constant in images
   - actually less problematic than it appears, since we can pre-filter images to make them look similar
Pyramid / “Coarse-to-fine”
Coarse-to-Fine Estimation

\[ I_x \cdot u + I_y \cdot v + I_t \approx 0 \]

\[ \implies \text{small } u \text{ and } v \ldots \]

Pyramid of image J

Pyramid of image I

u = 1.25 pixels

u = 2.5 pixels

u = 5 pixels

u = 10 pixels
OF application: Image stabilization

DeShaker
OF application: MatchMoving
OF application: Slow motion

- Slow motion (generate intermediate frames)
- Technology is also key to 100Hz television
Welcome

slowmoVideo is an OpenSource program that creates slow-motion videos from your footage. But it does not simply make your videos play at 0.1x speed. You can smoothly slow down and speed up your footage, optionally with motion blur.

How does slow motion work? slowmoVideo tries to find out where pixels move in the video (this information is called Optical Flow), and then uses this information to calculate the additional frames.

Features

- Videos in any format supported by ffmpeg can be loaded. Image sequences can also be loaded, so, if you did a timespace with too few frames, slowmoVideo may help as well.
- slowmoVideo does not work with a constant slowdown factor but with curves that allow arbitrary time acceleration/deceleration/reversal.
- Motion blur can be added, as much as you want.

The most recent changes to slowmoVideo can be read in the changelog.

Technologies

These parts are used by slowmoVideo:

- Qt as C++ programming framework
- GPU-OF for calculating the optical flow
- ffmpeg for reading and writing video files

Thesis

I wrote slowmoVideo as my bachelor thesis at ETH Zurich. The thesis can be read [here](http://slowmovideo.granjow.net/).
Optical Flow

- Brightness Constancy
- The Aperture problem
- Regularization
- Lucas-Kanade
- Coarse-to-fine
- Parametric motion models
- SSD tracking
- Bayesian flow
Parametric (Global) Motion Models

Global motion models offer

- more constrained solutions than smoothness (Horn-Schunck)
- integration over a larger area than a translation-only model can accommodate (Lucas-Kanade)
Parametric (Global) Motion Models

2D Models:
(Translation)
Affine
Quadratic
Planar projective transform (Homography)

3D Models:
Instantaneous camera motion models
Homography+epipole
Plane+Parallax
\[ E(\ h\ ) = \sum_{x \in \mathbb{R}} \left[ I(\ x + h\ ) - I_0(x) \right]^2 \]
• Transformations/warping of image

\[ E(\mathbf{h}) = \sum_{x \in \mathbb{R}} \left[ I(x + \mathbf{h}) - I_0(x) \right]^2 \]

Translations:

\[ \mathbf{h} = \begin{bmatrix} \delta x \\ \delta y \end{bmatrix} \]
What about other types of motion?
Generalization

- Transformations/warping of image

\[ E(A, \ h) = \sum_x \sum_{\mathbb{R}} \left[ I(Ax + h) - I_0(x) \right]^2 \]

Affine: \[ A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad h = \begin{bmatrix} \delta x \\ \delta y \end{bmatrix} \]
Generalization

Affine: $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$, $h = \begin{bmatrix} \delta x \\ \delta y \end{bmatrix}$
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 B.C. Equation:

\[ I_x \cdot u + I_y \cdot v + I_t \approx 0 \]

\[ I_x (a_1 + a_2 x + a_3 y) + I_y (a_4 + a_5 x + a_6 y) + I_t \approx 0 \]

Each pixel provides 1 linear constraint in 6 global unknowns

(minimum 6 pixels necessary)

Least Square Minimization (over all pixels):

\[ Err(\vec{a}) = \sum \left[ I_x (a_1 + a_2 x + a_3 y) + I_y (a_4 + a_5 x + a_6 y) + I_t \right]^2 \]
\[ u(x, y) = a_0 + a_1 x + a_2 y \]
\[ v(x, y) = a_3 + a_4 x + a_5 y \]

\[ u(x; a) = (u(x, y), v(x, y)) \]

\[ I(x, t-1) \]

\[ x + u(x; a) \]

\[ I(x + u(x; a), t-1) = I(x, t) \]

(Brightness Constancy Assumption)
KLT: Good features to keep tracking

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.

Simple displacement is sufficient between consecutive frames, but not to compare to reference template.
Generalization

- Transformations/warping of image

\[ E(A) = \sum_{x \in \mathcal{M}} \left[ I(Ax) - I_0(x) \right]^2 \]

Planar perspective: \( A = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & 1 \end{bmatrix} \)
Generalization

Affine +

Planar perspective: $A = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & 1 \end{bmatrix}$
Generalization

- Transformations/warping of image

$$E(h) = \sum_{x \in \mathcal{X}} \left[ I(\mathbf{f}(x, h)) - I_0(x) \right]^2$$

Other parametrized transformations
Generalization

Other parametrized transformations
2D Motion Models summary

**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
\[ u = x' - x, \quad v = y' - y \]
Advanced parametric model

- Optical flow constrained by non-rigid face model

Flexible flow for 3D nonrigid tracking and shape recovery, Brand and Bhotika, CVPR2001.
3D Motion Models summary

**Instantaneous camera motion:**
Global parameters: $\Omega_x, \Omega_y, \Omega_z, T_x, T_y, T_z$
Local Parameter: $Z(x,y)$

$$u = -xy\Omega_x + (1+x^2)\Omega_y - y\Omega_z + (T_x - T_zx)/Z$$
$$v = -(1+y^2)\Omega_x + xy\Omega_y - x\Omega_z + (T_y - T_zx)/Z$$

**Homography + Epipole**
Global parameters: $h_1, \ldots, h_9, t_1, t_2, t_3$
Local Parameter: $\gamma(x,y)$

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

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

**Residual Planar Parallax Motion**
Global parameters: $t_1, t_2, t_3$
Local Parameter: $\gamma(x,y)$

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

Original sequence    Plane-aligned sequence    Recovered shape

Block sequence from  [Kumar-Anandan-Hanna’94]

“Given two views where motion of points on a parametric surface has been compensated, the residual parallax is an epipolar field”
Residual Planar Parallax Motion

\[ I_X u + I_Y v + I_T = 0 \]

The intersection of the two line constraints uniquely defines the displacement.
Figure 1: Model-based tracking is robust to degraded images and transient occlusions. Dots show flexed model in 3/4, frontal, and profile view. Dots on face show where the image is sampled. Dots on neck encode 3D motion parameters.
Optical Flow

- Brightness Constancy
- The Aperture problem
- Regularization
- Lucas-Kanade
- Coarse-to-fine
- Parametric motion models
- SSD tracking
- Bayesian flow
Correlation and SSD

- For large displacements, do template matching as was used in stereo disparity search.
  - 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 interpolation also possible
SSD Surface – Textured area
SSD Surface -- Edge
SSD Surface – homogeneous area
Discrete Search vs. Gradient Based Estimation

Consider image I translated by $u_0, v_0$

\[
I_0(x, y) = I(x, y)
\]
\[
I_1(x + u_0, y + v_0) = I(x, y) + \eta_1(x, y)
\]

\[
E(u, v) = \sum_{x, y} (I(x, y) - I_1(x + u, y + v))^2
\]
\[
= \sum_{x, y} (I(x, y) - I(x - u_0 + u, y - v_0 + v) - \eta_1(x, y))^2
\]

Discrete search simply searches for the best estimate. Gradient method linearizes the intensity function and solves for the estimate.
Optical Flow

- Brightness Constancy
- The Aperture problem
- Regularization
- Lucas-Kanade
- Coarse-to-fine
- Parametric motion models
- SSD tracking
- Bayesian flow
Bayesian Optic Flow

- Some low-level human motion illusions can be explained by adding an uncertainty model to Lucas-Kanade tracking
- Theories from Psychology about normal flow fusion:
  - (VA) vector average (of normal motions)
  - (IOC) intersection of constraints (e.g., Lucas-Kanade):
Rhombus Displays

http://www.cs.huji.ac.il/~yweiss/Rhombus/
Brightness constancy with noise:

\[ I(x,y,t) = I(x + v_x \Delta t, y + v_y \Delta t, t + \Delta t) + \eta \]

Assume Gaussian noise, smooth surfaces, locally constant; take first order linear approximation:

\[
P(I(x_i,y_i,t)|v_i) \propto \exp \left( -\frac{1}{2\sigma^2} \int_{x,y} w(x,y) (I_x(x,y) v_x + I_y(x,y) v_y + I_t(x,y))^2 \, dx \, dy \right)
\]

Prior favoring slow speeds:

\[
P(v) \propto \exp(-\|v\|^2/2\sigma_p^2).
\]

Assume noise is independent across location; apply Bayes:

\[
P(v|I) \propto P(v) \prod_{i} P(I(x_i,y_i,t) | v),
\]

With constant window \( w=1 \),

\[
P(v|I) \propto \exp \left( -\|v\|^2/2\sigma_p^2 - \frac{1}{2\sigma^2} \int_{x,y} (I(x,y) v_x + I_y(x,y) v_y + I_t)^2 \, dx \, dy \right)
\]

Form ‘normal equations’ to arrive at....
Lucas-Kanade with uncertainty:

\[ v^* = - \left( \begin{array}{cc} \Sigma I_x^2 + \frac{\sigma^2}{\sigma_p^2} & \Sigma I_x I_y \\ \Sigma I_x I_y & \Sigma I_y^2 + \frac{\sigma^2}{\sigma_p^2} \end{array} \right)^{-1} \left( \begin{array}{c} \Sigma I_x I_t \\ \Sigma I_y I_t \end{array} \right) \]

One parameter: ratio of observation and prior gaussian spread.

http://www.cs.huji.ac.il/~yweiss/Rhombus

[Weiss, Simoncelli, Adelson Nature Neuroscience 2002]
Figure 4: The response of the Bayesian estimator to a fat rhombus. (replotted from Weiss and Adelson 98)
Figure 3: The response of the Bayesian estimator to a narrow rhombus. (replotted from Weiss and Adelson 98)
Effect of contrast
Thursday: video compression