Dense Correspondence and Its Applications

CVFX @ NTHU

28 May 2015

Outline

Dense Correspondence

Optical flow

Epipolar Geometry

Stereo Correspondence

Why dense correspondence?

Features are sparse

Some applications of dense correspondence

- Frame-rate conversion
- View morphing, view synthesis
- Stereo

Definition

Compute a vector field $(u(x, y), v(x, y))$ over the pixels of image I_1 so that the pixels $I_1(x, y)$ and $I_2(x + u(x, y), y + v(x, y))$ correspond.

Related topics

Motion vector

Optical flow estimation

Registration

Stereo correspondence, epipolar geometry, rectification

Video matching

Image morphing, view synthesis

Affine transformation

$I_1(x, y)$ and $I_2(x', y')$

$$x' = a_{11}x + a_{12}y + b_1 \qquad y' = a_{21}x + a_{22}y + b_2 \tag{1}$$

Rigid motions (translations and rotations)

Similarity transformations (rigid motions plus a uniform change in scale)

Shape-changing transformations (shears)

Projective transformation

Homography

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}} \qquad y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}} \tag{2}$$

Related to perspective projection

- camera undergoes pure rotation (panning, tilting, and zooming only)
- scene is entirely plannar

Image registration

- Estimating the parameters

Homogeneous coordinates

Representing an image location (x, y) as a triple $(x, y, 1)$

- any triple (x, y, z) can be converted backt to an image coordinate $\left(\frac{x}{z}, \frac{y}{z}\right)$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \sim \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad (3)$$

- The simbol $\sim$ means that the two vectors are equivalent up to a scalar multiple

Normalized direct linear transform (Hartley and Zisserman)

Initial estimation of the parameters of a projective transformation given a set of feature matches

- Two sets of features $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ and $\{(x'_1, y'_1), \ldots, (x'_n, y'_n)\}$
- Normalize each set to have zero mean and average distance from the origin $\sqrt{2}$
- Construct a $2n \times 9$ matrix A, where each feature match generates two rows of A

$$A_i = \begin{bmatrix} 0 & 0 & 0 & x_i & y_i & 1 & -y'_i x_i & -y'_i y_i & -y'_i \\ x_i & y_i & 1 & 0 & 0 & 0 & -x'_i x_i & -x'_i y_i & -x'_i \end{bmatrix} \tag{4}$$

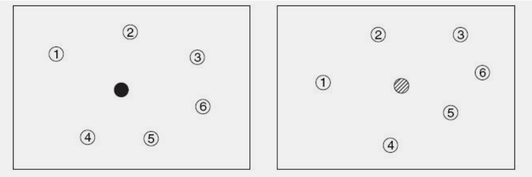
$(A_i [h_{11}\ h_{12}\ h_{13}\ h_{21}\ h_{22}\ h_{23}\ h_{31}\ h_{32}\ h_{33}]^T = 0)$

- Singular value decomposition $A = UDV^T$. Let h be the last column of V.
- Reshape h into a 3×3 matrix $\hat{H}$
- Recover the final projective transformation estimate as $H = T'^{-1} \hat{H} T$

Scattered data interpolation

Find a continuous function $f(x, y)$ defined over the first image plane so that $f(x_i, y_i) = (x'_i, y'_i), i = 1, \ldots, n$.

- given a set of feature matches unevenly distributed in each image
- the goal is to generate a dense correspondence for every point in the first image plane.



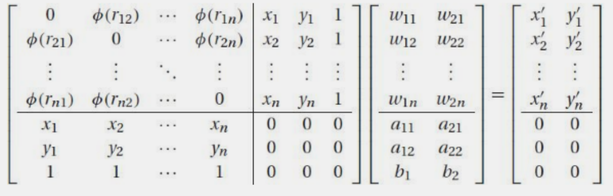
Thin-plate spline interpolation

$$f(x,y) = \left[\begin{array}{c} \sum_{i=1}^{n} w_{1i}\phi(r_i) + a_{11}x + a_{12}y + b_1 \\ \sum_{i=1}^{n} w_{2i}\phi(r_i) + a_{21}x + a_{22}y + b_2 \end{array} \right] \quad (5)$$

where $\phi(r)$ is called a radial bassis function and

$$r_i = \left\| \left[\begin{array}{c} x \\ y \end{array} \right] - \left[\begin{array}{c} x_i \\ y_i \end{array} \right] \right\|_2 \quad (6)$$

Solving a linear system

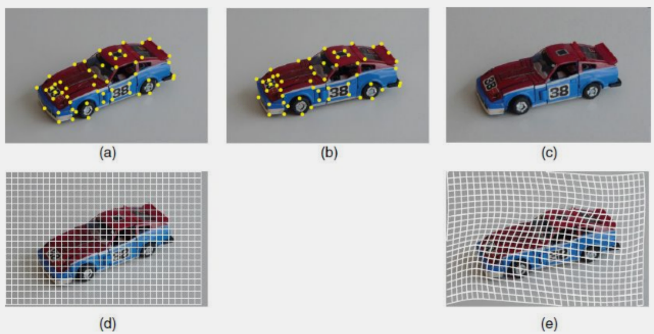


$$\left[\begin{array}{cccc|ccc} 0 & \phi(r_{12}) & \cdots & \phi(r_{1n}) & x_1 & y_1 & 1 \\ \phi(r_{21}) & 0 & \cdots & \phi(r_{2n}) & x_2 & y_2 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \phi(r_{n1}) & \phi(r_{n2}) & \cdots & 0 & x_n & y_n & 1 \\ \hline x_1 & x_2 & \cdots & x_n & 0 & 0 & 0 \\ y_1 & y_2 & \cdots & y_n & 0 & 0 & 0 \\ 1 & 1 & \cdots & 1 & 0 & 0 & 0 \end{array}\right] \left[\begin{array}{cc} w_{11} & w_{21} \\ w_{12} & w_{22} \\ \vdots & \vdots \\ w_{1n} & w_{2n} \\ \hline a_{11} & a_{21} \\ a_{12} & a_{22} \\ b_1 & b_2 \end{array}\right] = \left[\begin{array}{cc} x'_1 & y'_1 \\ x'_2 & y'_2 \\ \vdots & \vdots \\ x'_n & y'_n \\ \hline 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{array}\right]$$

where

$$r_{ij} = \left\| \begin{bmatrix} x_i \\ y_i \end{bmatrix} - \begin{bmatrix} x_j \\ y_j \end{bmatrix} \right\|_2 \tag{7}$$

Thin-plate spline interpolation



(a) (b) (c)

(d) (e)

n correspondences, $2(n+3)$ parameters including 6 global affine motion parameters and $2n$ coefficients for rbfs

Choice of rbf

Smallest bending energy

Minimizingthe integral

$$\int\int\left(\frac{\partial^2 f}{\partial x^2}\right)^2+\left(\frac{\partial^2 f}{\partial x \partial y}\right)^2+\left(\frac{\partial^2 f}{\partial y^2}\right)^2 dx\,dy \tag{8}$$

$$\phi(r)=r^2\log r \tag{9}$$

B-spline interpolation

- Basis spline
- Not at the feature point locations but at control points on a lattice overlaid on the first image plane

$$f(x, y) = \sum_{k=0}^{3} \sum_{l=0}^{3} w_{kl}(x, y)\psi_{kl}(x, y) \tag{10}$$

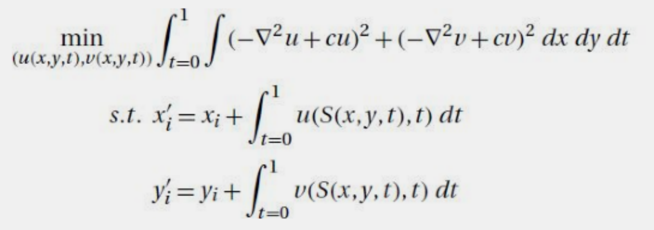
ψ: cubic polynomials

Diffeomorphisms

- ‘flow’ the first image I_1 to the second image I_2 over time interval $t \in [0, 1]$
- the flow is represented by $(u(x, y, t), v(x, y, t))$
- a mapping $S(x, y, t)$ specifying the flowed location of each point (x, y) at time t, $t = 0$ for I_1 and $t = 1$ for I_2

$$\frac{\partial S(x, y, t)}{\partial t} = \left[\begin{array}{c} u(x, y, t) \\ v(x, y, t) \end{array} \right], \quad t \in [0, 1] \tag{11}$$

Optimization



$$
\begin{aligned}
\min_{(u(x,y,t),v(x,y,t))} \int_{t=0}^{1} \int (-\nabla^2 u + cu)^2 + (-\nabla^2 v + cv)^2 \, dx\, dy\, dt \\
s.t.\ x_i' = x_i + \int_{t=0}^{1} u(S(x,y,t),t)\, dt \\
y_i' = y_i + \int_{t=0}^{1} v(S(x,y,t),t)\, dt
\end{aligned}
$$

The result is a diffeomorphism that does not suffer from the grid line self-intersection problem

As-rigid-as-possible deformation

- $f(x,y) = T_{x,y}(x,y)$, where $T_{x,y}$ is a rigid transformation defined at (x,y)
- each $T_{x,y}$ is estimated by minimizing

$$\sum_{i=1}^{n} \frac{\| T_{x,y}(x_i, y_i) - (x_i', y_i')\|_2^2}{\|(x,y) - (x_i, y_i)\|_2^{2\alpha}} \tag{12}$$

Problems with scattered data interpolation techniques

Hard to control (e.g. to keep straight lines straight)

Independent of the underlying image intensities

Might reuqire significant user interaction

Optical flow

Estimating a motion vector (u, v) at every point (x, y) such that $I_1(x, y)$ and $I_2(x + u, y + v)$ correspond:

Horn-Schunck method

Lucas-Kanade method

The Horn-Schunck method

Brightness constancy assumption

$$I(x+u, y+v, t+1) = I(x, y, t) \tag{13}$$

Lambertian assumption

- same brightness regardless of viewing direction

Ideal image formation process

- no vignetting

The Horn-Schunck method

Taylor expansion

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = 0 \tag{14}$$

$$\nabla I \cdot \left[\begin{array}{c} u \\ v \end{array} \right] = -\frac{\partial I}{\partial t} \tag{15}$$

The component of the flow vector in the direction of the gradient is

$$-\frac{\frac{\partial I}{\partial t}}{\|\nabla I\|_2^2}\nabla I \tag{16}$$

The Horn-Schunck method

Considering smoothness

$$E_{\mathrm{HS}}(u,v) = E_{\mathrm{data}}(u,v) + \lambda E_{\mathrm{smoothness}}(u,v) \tag{17}$$

- λ is a regularization parameter

$$E_{\mathrm{data}}(u,v) = \sum_{x,y} \left(\frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v + \frac{\partial I}{\partial t} \right)^2 \tag{18}$$

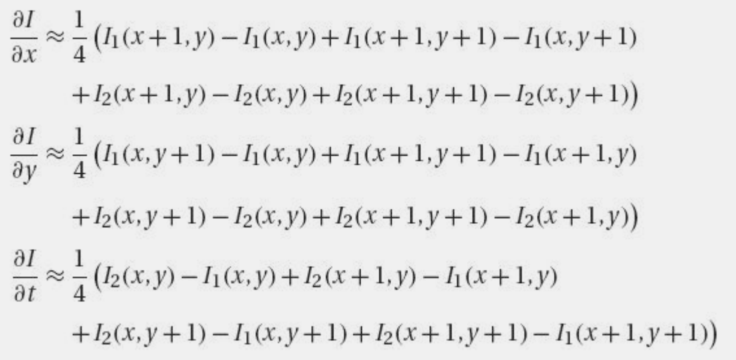
$$E_{\mathrm{smoothness}} = \sum_{x,y} \left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial u}{\partial y} \right)^2 + \left(\frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 \tag{19}$$

Euler-Lagrange

$$\lambda \nabla^2 u = \left(\frac{\partial I}{\partial x} \right)^2 u + \frac{\partial I}{\partial x} \frac{\partial I}{\partial y} v + \frac{\partial I}{\partial x} \frac{\partial I}{\partial t} \tag{20}$$

$$\lambda \nabla^2 v = \left(\frac{\partial I}{\partial y} \right)^2 v + \frac{\partial I}{\partial x} \frac{\partial I}{\partial y} u + \frac{\partial I}{\partial y} \frac{\partial I}{\partial t} \tag{21}$$

Finite differences



$$\begin{aligned}
\frac{\partial I}{\partial x} &\approx \frac{1}{4}\big(I_1(x+1,y) - I_1(x,y) + I_1(x+1,y+1) - I_1(x,y+1) \\
&\quad + I_2(x+1,y) - I_2(x,y) + I_2(x+1,y+1) - I_2(x,y+1)\big) \\
\frac{\partial I}{\partial y} &\approx \frac{1}{4}\big(I_1(x,y+1) - I_1(x,y) + I_1(x+1,y+1) - I_1(x+1,y) \\
&\quad + I_2(x,y+1) - I_2(x,y) + I_2(x+1,y+1) - I_2(x+1,y)\big) \\
\frac{\partial I}{\partial t} &\approx \frac{1}{4}\big(I_2(x,y) - I_1(x,y) + I_2(x+1,y) - I_1(x+1,y) \\
&\quad + I_2(x,y+1) - I_1(x,y+1) + I_2(x+1,y+1) - I_1(x+1,y+1)\big)
\end{aligned}$$

Strength and weakness

Advantage

- difussion equation, creates good estimates of the dense correspondence field even when the local gradient is nearly zero
- smoothly filling in reasonable flow vectors based on nearby locations

Disadvantage

- Taylor series approximation is noly valid when (u, v) is close to zero
- I_1 and I_2 are already very similar

Hierarchical or multiresolution approach

Hierarchical approach

Laplacian pyramid

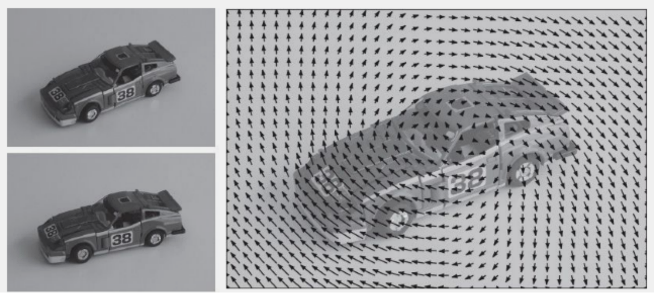
Flow is computed for the coarset level of the pyramid

Warp the second image plane to the first using the estimated flow

- the flow between the first image and the warped second image is expected to be nearly zero

Improvements

Applying a median filter to the incremental flow field during the warping process



D. Sun, S. Roth, and M. Black. Secrets of optical flow estimation and their principles. CVPR 2010.

The Lucas-Kanade method

Optical flow vector at a point (x_0, y_0) is defined as the minimizer (u, v) of

$$E_{\mathrm{LK}}(u, v) = \sum_{(x,y)} w(x, y)(I(x + u, y + v, t + 1) - I(x, y, t))^2 \qquad (22)$$

where $w(x, y)$ is a window function centered at (x_0, y_0).

Solution of the linear system

$$H = \begin{bmatrix} \sum_{(x,y)} w(x,y) \left(\frac{\partial I}{\partial x}(x,y) \right)^2 & \sum_{(x,y)} w(x,y) \left(\frac{\partial I}{\partial x}(x,y) \frac{\partial I}{\partial y}(x,y) \right) \\ \sum_{(x,y)} w(x,y) \left(\frac{\partial I}{\partial x}(x,y) \frac{\partial I}{\partial y}(x,y) \right) & \sum_{(x,y)} w(x,y) \left(\frac{\partial I}{\partial y}(x,y) \right)^2 \end{bmatrix} \tag{23}$$

$$H \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum_{(x,y)} w(x,y) \left(\frac{\partial I}{\partial x}(x,y) \frac{\partial I}{\partial t}(x,y) \right) \\ \sum_{(x,y)} w(x,y) \left(\frac{\partial I}{\partial y}(x,y) \frac{\partial I}{\partial t}(x,y) \right) \end{bmatrix} \tag{24}$$

Comparisons

the Lucas-Kanade algorithm is local

- the flow vector can be computed at each pixel independently
- easier to solve

the Horn-Schunck algorithm is global

- all the flow vectors depend on each other through the differential equation

Both assume that the flow vectors are small

- pyramidal implementation

Refinements and extensions

Changing the data term

Changing the smoothness term

Changing the form of cost functions

Changing the data term

Horn-Schunck brightness constancy assumtion

Uras et al. gradient constancy assumption

$$\nabla I(x+u, y+v, t+1) = \nabla I(x, y, t) \tag{25}$$

- Uras et al. A computational approach to motion perception. Biological Cybernetics, 60(2): 79– 87, Dec. 1988.
- allows some local variation to illumination changes

Changing the data term

Brox et al. assume both brightness and gradient constancy

$$E_{\text{data}}(u,v)=\sum_{x,y}(I_2(x+u,y+v)-I_1(x,y))^2+\gamma\|\nabla I_2(x+u,y+v)-\nabla I_1(x,y)\|_2^2 \tag{26}$$

- Brox et al. High accuracy optical flow estimation based on a theory for warping. ECCV 2004.
- γ weights the contribution of the terms (typically $\gamma \simeq 100$)
- directly expresses the deviation from the constancy assumptions, instead of using the Taylor approximation that is only valid for small u and v
- Xu et al. introduce a binary swithch variable choosing either the brightness constancy or gradient constancy assumption

Changing the data term

LK + HS

$$E_{\text{data}}(u,v) = \sum_{(x,y)} w(x,y)(I_2(x+u,y+v) - I_1(x,y))^2 \tag{27}$$

- A. Bruhn, J. Weickert, and C. Schnöorr. Lucas/ Kanade meets Horn/ Schunck: Combining local and global optic flow methods. IJCV, 61(3): 211– 31, Feb. 2005.
- small window size, local spatial smoothing, which makes the data term robust to noise
- global regularization, which makes the flow field smooth and dense

Probabilistic data term (Sun et al.)

- learned distribution of $I_2(x+u,y+v) - I_1(x,y)$
- mixture of Gaussian

Changing the smoothnes term

The smoothness term should be downweighted perpendicular to image edges

- correspond to depth discontinuities
- Nagel and Enkelmann . An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-8(5): 565– 93, Sept. 1986.

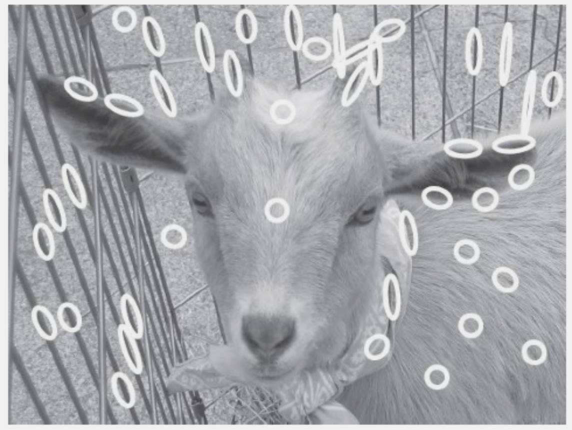
$$E_{\text{smoothness}}(u,v) = \text{trace}\left(\begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial v}{\partial x} \\ \frac{\partial v}{\partial y} & \frac{\partial v}{\partial y} \end{bmatrix}^T D(\nabla I_1(x,y)) \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial v}{\partial x} \\ \frac{\partial v}{\partial y} & \frac{\partial v}{\partial y} \end{bmatrix} \right) \tag{28}$$

$D : \mathbb{R} \to \mathbb{R}^{2\times2}$ is the anisotropic diffusion tensor

$$D(g) = \frac{1}{\|g\|_2^2 + 2\beta^2}\left(g^{\perp} {g^{\perp}}^T + \beta^2 \mathbf{I}_{2\times2} \right) \tag{29}$$

Anisotropic diffusion and eigenvalues

- the major and minor axes of each ellipse are aligned with the tensor's eigenvectors and weighted by the corresponding eigenvalues



Changing the smoothness term

as a prior term on the optical flow field

- reflects the assumptions about ‘good’ flows
- S. Roth and M. Black. On the spatial statistics of optical flow. IJCV, 74(1): 33– 50, Aug. 2007.
- Field-of-Experts model

Changing the cost functions

Horn and Schunck: quadratic functions

Robust estimation

- M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. CVIU, 63(1): 75– 104, Jan. 1996.
- Robust data term

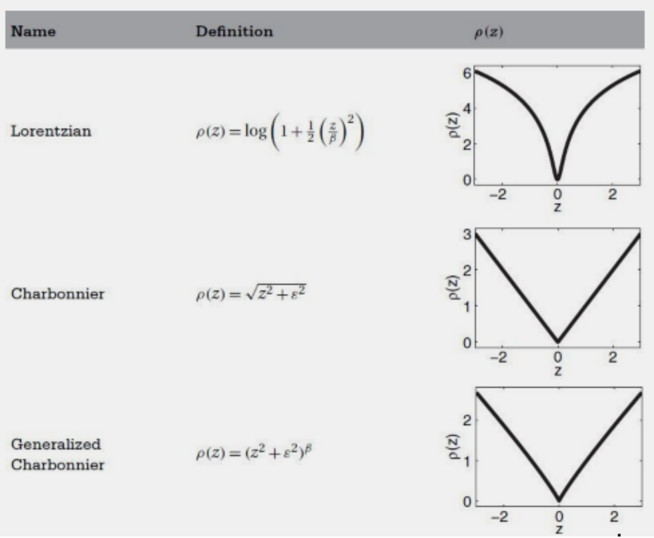
$$E_{\text{data}}(u, v) = \sum_{x,y} \rho\left(\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t}\right) \tag{30}$$

- Robust smoothness term

$$E_{\text{smoothness}}(u, v) = \rho\left(\left\|\left[\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial v}{\partial x}, \frac{\partial v}{\partial y}\right]\right\|_2\right) \tag{31}$$

- $\rho(z) = z^2$: Horn-Schunck
- other choices

Robust penalty functions typically used for optical flow

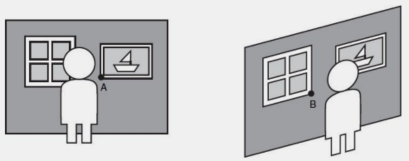


Name	Definition	$\rho(z)$
Lorentzian	$\rho(z) = \log\left(1 + \frac{1}{2}\left(\frac{z}{\beta}\right)^2\right)$	
Charbonnier	$\rho(z) = \sqrt{z^2 + \varepsilon^2}$	
Generalized Charbonnier	$\rho(z) = (z^2 + \varepsilon^2)^\beta$	

ρ(z)
6
4
2
0
−2
0
2
z
ρ(z)
3
2
1
0
−2
0
2
z
ρ(z)
2
1
0
−2
0
2
z

Occlusion

- Implicit assumption: every pixel in I_2 corresponds to some pixel in I_1. This assumption is violated at occlusions.



- Cross checking, or left-right checking

$$(u,v)^{\text{fwd}}(x,y) = -(u,v)^{\text{bwd}}(I_2(x+u^{\text{fwd}}(x,y), y+v^{\text{fwd}}(x,y))) \quad (32)$$

- Robust cost functions can partially alleviate the occlusion problem

Layered Flow

Layered motion for video matting

Post-converting a monocular film into stereo

- M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. CVIU, 63(1): 75– 104, Jan. 1996. (using robust penalty functions to estimate a dominant motion in the scene, classify inlier pixels into a solved layer, and re-apply the process to outlier pixels)
- M. Black and A. Jepson. Estimating optical flow in segmented images using variable-order parametric models with local deformations. PAMI, 18(10): 972– 86, Oct. 1996. (flow field in each region can be represented by a low-dimensinal parametric transformation (e.g. affine), coarse to fine)
- Y. Weiss. Smoothness in layers: Motion segmentation using nonparametric mixture estimation. CVPR 1997. (the non-parametric flow field is smooth, using EM to estimate layer memberships and estimate the flow field in each layer)

Large-displacement optical flow

T. Brox, C. Bregler, and J. Malik. Large displacement optical flow. CVPR 2009.

- small structure, fast motion, e.g. hand waving
- starts with the segmentation of each image into roughly constant-texture patches, each of which is described by a SIFT-like descriptor
- matching descriptors, as additional constraints

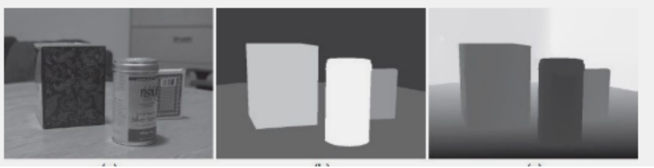
C. Liu, J. Yuen, and A. Torralba. SIFT flow: Dense correspondence across scenes and its applications. PAMI, 33(5): 978– 94, May 2011.

- replaces the brightness constancy assumption with a SIFT-descriptor constancy assumption, $\|S_2(x+u, y+v) - S_1(x, y)\|$
- dense SIFT
- designed to match between different scenes

Human-assisted motion annotation

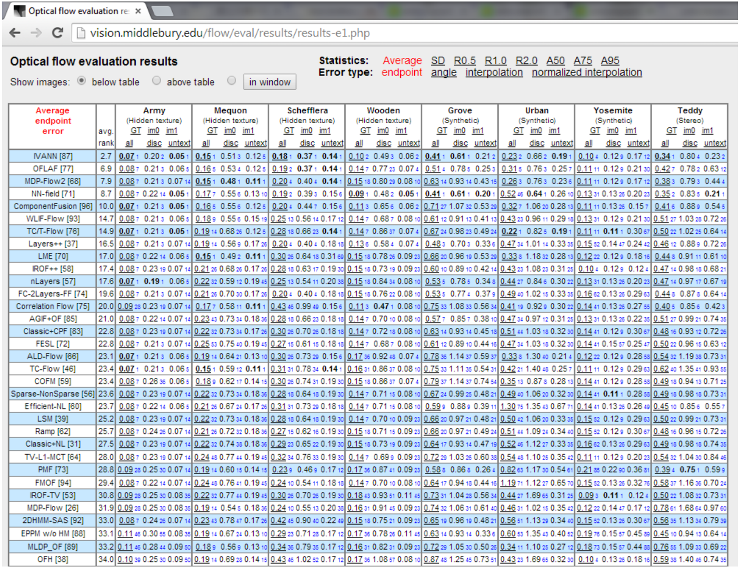
C . Liu, W. Freeman, E. Adelson, and Y. Weiss. Human-assisted motion annotation . CVPR 2008.

- semi-automatically created layers



Evaluation

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



Optical flow evaluation re ×
vision.middlebury.edu/flow/eval/results/results-e1.php

Optical flow evaluation results

Show images: ● below table ○ above table ○ in window

Statistics: Average SD R0.5 R1.0 R2.0 A50 A75 A95

Error type: endpoint angle interpolation normalized interpolation

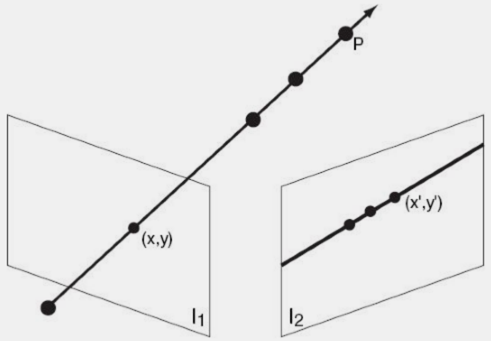
Average endpoint error	avg. rank	Army (Hidden texture) GT im0 im1			Mequon (Hidden texture) GT im0 im1			Schefflera (Hidden texture) GT im0 im1			Wooden (Hidden texture) GT im0 im1		
	rank	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
IVANN [87]	2.7	**0.07** 1	0.20 2	**0.05** 1	**0.15** 1	0.51 3	0.12 5	**0.18** 1	**0.37** 1	**0.14** 1	0.10 2	0.49 3	0.06 2
OFLAF [77]	6.9	0.08 7	0.21 3	0.06 5	0.16 5	0.53 4	0.12 5	0.19 2	**0.37** 1	**0.14** 1	0.14 7	0.77 23	0.07 4
MDP-Flow2 [68]	7.9	0.08 7	0.21 3	0.07 14	**0.15** 1	**0.48** 1	**0.11** 1	0.20 4	0.40 4	**0.14** 1	0.15 18	0.80 29	0.08 10
NN-field [71]	8.7	0.08 7	0.22 14	**0.05** 1	0.17 7	0.55 6	0.13 10	0.19 2	0.39 3	0.15 6	**0.09** 1	0.48 2	**0.05** 1
ComponentFusion [96]	10.0	**0.07** 1	0.21 3	**0.05** 1	0.16 5	0.55 6	0.12 5	0.20 4	0.44 7	0.15 6	0.11 3	0.65 6	0.06 2
WLIF-Flow [93]	14.7	0.08 7	0.21 3	0.06 5	0.18 9	0.55 6	0.15 19	0.25 13	0.56 14	0.17 12	0.14 7	0.68 7	0.08 10
TC/T-Flow [76]	14.9	**0.07** 1	0.21 3	**0.05** 1	0.19 14	0.68 26	0.12 5	0.28 18	0.66 23	**0.14** 1	0.14 7	0.86 37	0.07 4
Layers++ [37]	16.5	0.08 7	0.21 3	0.07 14	0.19 14	0.56 9	0.17 26	0.20 4	0.40 4	0.18 18	0.13 6	0.58 4	0.07 4
LME [70]	17.0	0.08 7	0.22 14	0.06 5	**0.15** 1	0.49 2	**0.11** 1	0.30 26	0.64 18	0.31 69	0.15 18	0.78 26	0.09 23
IROF++ [58]	17.4	0.08 7	0.23 19	0.07 14	0.21 26	0.68 26	0.17 26	0.28 18	0.63 17	0.19 30	0.15 18	0.73 19	0.09 23
nLayers [57]	17.6	**0.07** 1	**0.19** 1	0.06 5	0.22 32	0.59 12	0.19 45	0.25 13	0.54 11	0.20 38	0.15 18	0.84 34	0.08 10
FC-2Layers-FF [74]	19.6	0.08 7	0.21 3	0.07 14	0.21 26	0.70 30	0.17 26	0.20 4	0.40 4	0.18 18	0.15 18	0.76 22	0.08 10
Correlation Flow [75]	20.0	0.09 28	0.23 19	0.07 14	0.17 7	0.58 11	**0.11** 1	0.43 46	0.99 49	0.15 6	0.11 3	**0.47** 1	0.08 10
AGIF+OF [85]	21.0	0.08 7	0.22 14	0.07 14	0.23 43	0.73 34	0.18 36	0.28 18	0.66 23	0.18 18	0.14 7	0.70 10	0.08 10
Classic+CPF [83]	22.8	0.08 7	0.23 19	0.07 14	0.22 32	0.73 34	0.17 26	0.30 26	0.70 26	0.18 18	0.14 7	0.72 18	0.08 10
FESL [72]	22.8	0.08 7	0.21 3	0.07 14	0.25 53	0.75 40	0.19 45	0.27 15	0.61 15	0.18 18	0.14 7	0.68 7	0.08 10
ALD-Flow [66]	23.1	**0.07** 1	0.21 3	0.06 5	0.19 14	0.64 21	0.13 10	0.30 26	0.73 29	0.15 6	0.17 36	0.92 48	0.07 4
TC-Flow [46]	23.4	**0.07** 1	0.21 3	0.06 5	**0.15** 1	0.59 12	**0.11** 1	0.31 31	0.78 34	**0.14** 1	0.16 31	0.86 37	0.08 10
COFM [59]	23.4	0.08 7	0.26 36	0.06 5	0.18 9	0.62 17	0.14 15	0.30 26	0.74 31	0.19 30	0.15 18	0.86 37	0.07 4
Sparse-NonSparse [56]	23.6	0.08 7	0.23 19	0.07 14	0.22 32	0.73 34	0.18 36	0.28 18	0.64 18	0.19 30	0.14 7	0.71 15	0.08 10
Efficient-NL [60]	23.7	0.08 7	0.22 14	0.06 5	0.21 26	0.67 24	0.17 26	0.31 31	0.73 29	0.18 18	0.14 7	0.71 15	0.08 10
LSM [39]	25.2	0.08 7	0.23 19	0.07 14	0.22 32	0.73 34	0.18 36	0.28 18	0.64 18	0.19 30	0.14 7	0.70 10	0.09 23
Ramp [62]	25.7	0.08 7	0.24 26	0.07 14	0.21 26	0.72 32	0.18 36	0.27 15	0.62 16	0.19 30	0.15 18	0.71 15	0.09 23
Classic+NL [31]	27.5	0.08 7	0.23 19	0.07 14	0.22 32	0.74 38	0.18 36	0.29 23	0.65 22	0.19 30	0.15 18	0.73 19	0.09 23
TV-L1-MCT [64]	28.0	0.08 7	0.23 19	0.07 14	0.24 48	0.77 44	0.19 45	0.32 34	0.76 33	0.19 30	0.14 7	0.69 9	0.09 23
PMF [73]	28.8	0.09 28	0.25 30	0.07 14	0.19 14	0.60 15	0.14 15	0.23 9	0.46 9	0.17 12	0.17 36	0.87 41	0.09 23
FMOF [94]	29.4	0.08 7	0.22 14	0.07 14	0.24 48	0.76 41	0.19 45	0.24 10	0.54 11	0.18 18	0.14 7	0.70 10	0.08 10
IROF-TV [53]	30.8	0.09 28	0.25 30	0.08 35	0.22 32	0.77 44	0.19 45	0.30 26	0.70 26	0.19 30	0.18 43	0.93 51	0.11 45
MDP-Flow [26]	31.9	0.09 28	0.25 30	0.08 35	0.19 14	0.54 5	0.18 36	0.24 10	0.55 13	0.20 38	0.16 31	0.91 45	0.09 23
2DHMM-SAS [92]	33.0	0.08 7	0.24 26	0.07 14	0.23 43	0.78 47	0.17 26	0.42 45	0.90 40	0.22 49	0.15 18	0.75 21	0.09 23
EPPM w/o HM [88]	33.1	0.11 46	0.30 55	0.08 35	0.19 14	0.67 24	0.13 10	0.29 23	0.71 28	0.17 12	0.17 36	0.78 26	0.11 45
MLDP_OF [89]	33.2	0.11 46	0.28 44	0.09 50	0.18 9	0.56 9	0.13 10	0.34 36	0.79 35	0.17 12	0.16 31	0.82 31	0.09 23
OFH [38]	34.0	0.10 39	0.25 30	0.09 50	0.19 14	0.69 28	0.14 15	0.43 46	1.02 52	0.17 12	0.17 36	1.08 57	0.08 10

Average endpoint error	Grove (Synthetic) GT im0 im1			Urban (Synthetic) GT im0 im1			Yosemite (Synthetic) GT im0 im1			Teddy (Stereo) GT im0 im1		
	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
IVANN [87]	**0.41** 1	**0.61** 1	0.21 2	0.23 2	0.66 2	**0.19** 1	0.10 4	0.12 9	0.17 12	**0.34** 1	0.80 4	0.23 2
OFLAF [77]	0.51 4	0.78 5	0.25 3	0.31 5	0.76 3	0.25 7	0.11 11	0.12 9	0.21 30	0.42 7	0.78 2	0.63 12
MDP-Flow2 [68]	0.63 14	0.93 14	0.43 15	0.26 3	0.76 3	0.23 6	0.11 11	0.12 9	0.17 12	0.38 3	0.79 3	0.44 4
NN-field [71]	**0.41** 1	**0.61** 1	**0.20** 1	0.52 46	**0.64** 1	0.26 10	0.13 31	0.13 26	0.20 23	0.35 2	0.83 5	**0.21** 1
ComponentFusion [96]	0.71 27	1.07 32	0.53 29	0.32 7	1.06 20	0.28 13	0.11 11	0.13 26	0.15 7	0.41 6	0.88 9	0.54 5
WLIF-Flow [93]	0.61 12	0.91 13	0.41 13	0.43 23	0.96 11	0.29 18	0.13 31	0.12 9	0.21 30	0.51 27	1.03 28	0.72 26
TC/T-Flow [76]	0.67 24	0.98 23	0.49 24	**0.22** 1	0.82 5	**0.19** 1	0.11 11	**0.11** 1	0.30 67	0.50 22	1.02 25	0.64 14
Layers++ [37]	0.48 3	0.70 3	0.33 6	0.47 34	1.01 14	0.33 35	0.15 52	0.14 47	0.24 42	0.46 12	0.88 9	0.72 26
LME [70]	0.66 20	0.96 19	0.53 29	0.33 8	1.18 32	0.28 13	0.12 22	0.12 9	0.18 16	0.44 8	0.91 11	0.61 10
IROF++ [58]	0.60 10	0.89 10	0.42 14	0.43 23	1.08 23	0.31 25	0.10 4	0.12 9	0.12 4	0.47 14	0.98 18	0.68 21
nLayers [57]	0.53 5	0.78 5	0.34 8	0.44 27	0.84 6	0.30 22	0.13 31	0.13 26	0.20 23	0.47 14	0.97 17	0.67 19
FC-2Layers-FF [74]	0.53 5	0.77 4	0.37 9	0.49 40	1.02 15	0.33 35	0.16 62	0.13 26	0.29 63	0.44 8	0.87 8	0.64 14
Correlation Flow [75]	0.75 33	1.08 33	0.56 34	0.41 19	0.92 9	0.30 22	0.14 41	0.13 26	0.27 55	0.40 5	0.85 6	0.42 3
AGIF+OF [85]	0.57 7	0.85 7	0.38 10	0.47 34	0.97 12	0.31 25	0.13 31	0.13 26	0.22 35	0.51 27	0.99 21	0.74 35
Classic+CPF [83]	0.63 14	0.93 14	0.45 18	0.51 44	1.03 18	0.32 30	0.14 41	0.12 9	0.30 67	0.48 16	0.93 12	0.72 26
FESL [72]	0.61 12	0.89 10	0.44 16	0.47 34	1.03 18	0.32 30	0.14 41	0.15 57	0.25 47	0.50 22	0.96 15	0.63 12
ALD-Flow [66]	0.78 36	1.14 37	0.59 37	0.33 8	1.30 40	0.21 4	0.12 22	0.12 9	0.28 58	0.54 32	1.19 38	0.73 31
TC-Flow [46]	0.75 33	1.11 35	0.54 31	0.42 21	1.40 48	0.25 7	0.11 11	0.12 9	0.29 63	0.62 40	1.35 41	0.93 55
COFM [59]	0.79 37	1.14 37	0.74 54	0.35 13	0.87 8	0.28 13	0.14 41	0.12 9	0.28 58	0.49 18	0.94 13	0.71 25
Sparse-NonSparse [56]	0.67 24	0.99 25	0.48 21	0.49 40	1.06 20	0.32 30	0.14 41	**0.11** 1	0.28 58	0.49 18	0.98 18	0.73 31
Efficient-NL [60]	0.59 9	0.88 9	0.39 11	1.30 75	1.35 43	0.67 71	0.14 41	0.13 26	0.26 49	0.45 10	0.85 6	0.55 7
LSM [39]	0.66 20	0.97 21	0.48 21	0.50 42	1.06 20	0.33 35	0.15 52	0.12 9	0.29 63	0.50 22	0.99 21	0.73 31
Ramp [62]	0.66 20	0.97 21	0.49 24	0.51 44	1.09 24	0.34 40	0.15 52	0.12 9	0.30 67	0.48 16	0.96 15	0.72 26
Classic+NL [31]	0.64 17	0.93 14	0.47 19	0.52 46	1.12 27	0.33 35	0.16 62	0.13 26	0.29 63	0.49 18	0.98 18	0.74 35
TV-L1-MCT [64]	0.72 29	1.03 26	0.60 38	0.54 48	1.10 25	0.35 42	0.11 11	0.12 9	0.20 23	0.54 32	1.04 30	0.84 46
PMF [73]	0.58 8	0.86 8	0.26 4	0.82 63	1.17 30	0.54 61	0.21 85	0.22 90	0.36 81	0.39 4	**0.75** 1	0.59 9
FMOF [94]	0.64 17	0.94 18	0.44 16	1.19 71	1.12 27	0.65 70	0.15 52	0.13 26	0.32 76	0.58 37	1.16 36	0.70 24
IROF-TV [53]	0.73 31	1.04 28	0.56 34	0.44 27	1.69 65	0.31 25	0.09 3	**0.11** 1	0.12 4	0.50 22	1.08 32	0.73 31
MDP-Flow [26]	0.74 32	1.06 31	0.61 40	0.46 31	1.02 15	0.35 42	0.12 22	0.14 47	0.17 12	0.78 61	1.68 64	0.97 60
2DHMM-SAS [92]	0.65 19	0.96 19	0.48 21	0.56 51	1.13 29	0.34 40	0.15 52	0.13 26	0.30 67	0.56 35	1.13 34	0.79 39
EPPM w/o HM [88]	0.63 14	0.93 14	0.33 6	0.60 53	1.35 43	0.40 52	0.19 76	0.15 57	0.45 89	0.45 10	0.94 13	0.64 14
MLDP_OF [89]	0.72 29	1.05 30	0.50 26	0.34 11	1.10 25	0.27 12	0.18 73	0.15 57	0.44 88	0.76 55	1.09 33	0.69 22
OFH [38]	0.87 48	1.25 45	0.73 51	0.43 23	1.69 65	0.32 30	0.10 4	0.13 26	0.18 16	0.59 38	1.40 46	0.74 35

Epipolar Geometry

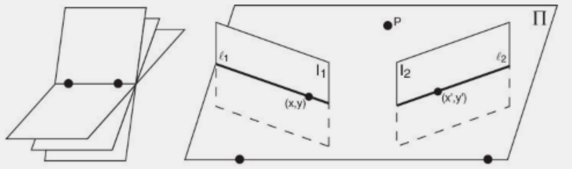
Two images taken at the same time, in different positions, by different cameras

- don't need to search across the entire image to estimate the motion vector (u, v)
- only one degree of freedom for the possible correspondence



Epipolar line and epipole

- all of the epipolar lines in one image intersect at the epipole, the projection of the camera center of the other image



- every point in the scene has to lie on one of those planes

The fundamental matrix

- the epipolar line in one image corresponding to a point in the other image can be computed from the fundamental matrix
- a 3×3 matrix F that concisely expresses the relationship between any two matching points
- any correspondence $\{(x, y) \in I_1, (x', y') \in I_2\}$ must satisfy

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}^T F \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0 \tag{33}$$

- the fundamental matrix F is defined up to scale
- F has rank 2
- what are the coefficients on x', y', and 1?

Geometric argument to see why F has rank 2

- all the epipolar lines in I_1 intersect at the epipole $e = (x_e, y_e)$
- for any $(x', y') \in I_2$, e lies on the corresponding epipolar line

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}^T \left(F \begin{bmatrix} x_e \\ y_e \\ 1 \end{bmatrix} \right) = 0 \tag{34}$$

holds for every (x', y')

- it means that

$$F \begin{bmatrix} x_e \\ y_e \\ 1 \end{bmatrix} = 0 \tag{35}$$

so $[x_e, y_e, 1]^T$ is an eigenvector of F with eigenvalue 0.

- similarly, $[x_e', y_e', 1']^T$ is an eigenvector of F^T with eigenvalue 0.

Representating F as a factorization

- based on the epipole in the second image

$$F = \begin{bmatrix} x'_e \\ y'_e \\ 1 \end{bmatrix}_\times M \tag{36}$$

where M is a full-rank 3×3 matrix. The notation $[e]_\times$:

$$[e]_\times = \begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix} \tag{37}$$

- rank?
- the fundamental matrix is not defined for an image pair that shares the same camera center
- a project transformation that directly specifies each pair of corresponding points; the same type of relationship holds when the scene contains only a single plane
- the fundamental matrix for an image pair taken by a pair of separated cameras of a real-world scene is defined and unique (up to scale)

Estimating the fundamental matrix

- like a projective transformation, the fundamental matrix can be estimated from a set of feature matches

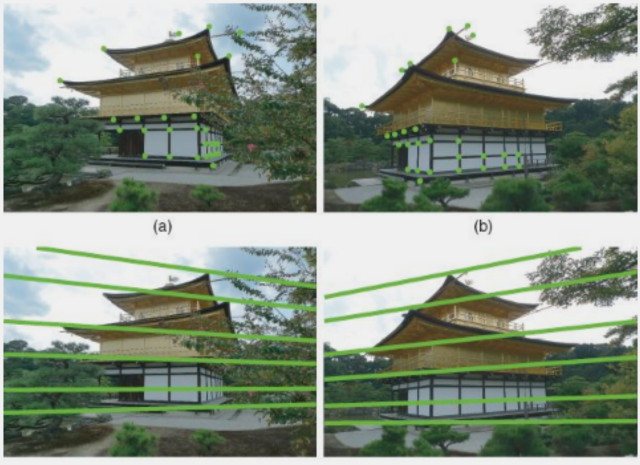
$$[x_i' x_i,\ x_i' y_i,\ x_i',\ y_i' x_i,\ y_i' y_i,\ y_i',\ x_i\ y_i\ 1][f_{11}\ f_{12}\ f_{13}\ f_{21}\ f_{22}\ f_{23}\ f_{31}\ f_{32}\ f_{33}]^T = 0 \tag{38}$$

- collecting the linear equations for each point yields an $n \times 9$ linear system $Af = 0$
- basic algorithm: normalized eight-point algorithm

Normalized eight-point algorithm

1. The input is two sets of features $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ and $\{(x'_1, y'_1), \ldots, (x'_n, y'_n)\}$. Normalize each set of feature matches to have zero mean and average distance from the origin $\sqrt{2}$. Done by a pair of similarity transformations S and S'.
2. Construct the $n \times 9$ matrix A, where each feature match generates a row.
3. Compute the singular value decomposition of A, $A = UDV^T$. Let f be the last column of V (corresponding to the smallest sigular value).
4. Reshape f into a 3×3 matrix $\hat{F}$, filing in the elements from left to right in each row.
5. Compute the singular value decomposition of $\hat{F}$, $\hat{F} = UDV^T$. Set the lower right $(3, 3)$ entry of D equal to zero to create $\hat{D}$ and replace $\hat{F}$ with $U\hat{D}V^T$.
6. Recover the final fundamental matrix estimate as $F = S'^T \hat{F} S$.

Example



(a)

(b)

(c)

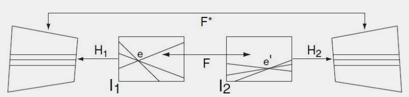
(d)

Extensions

- Maximum likelihood estimate under the assumptions that the measurement errors in each feature location are Gaussian
- Non-linear optimization and RANSAC
- When each camera's location, orientation, and internal configuration are known, the funcamental matrix can be computed directly

Image rectification

- epipolar lines are typically slanted, which can make estimating correspondences along conjugate epipolar lines complicated due to repeated image resampling operations
- rectification: making conjugate epipolar lines coincide with scanlines
- epipolar lines are parallel and the epipoles are 'at infinity', represented by the homogeneous coordinate $[1, 0, 0]$



- the fundamental matrix for a rectified image pair is given by

$$F^* = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \tag{39}$$

- for a correspondence in the rectified pair of images, $y' = y$, corresponding to the definition of rectification.

Rectification method proposed by Hartley

- the idea is to estimate a projective transformation H_2 for the second image that moves the epipole to the homogeneous coordinate $[1, 0, 0]$ while resembling a rigid transformation as much as possible
- Estimate the fundamental matrix F from the feature matches using the eight-point algorithm.
- Factor F in the form of $F = [e']_\times M$, where $e' = [x'_e, y'_e, 1]^T$ is the homogeneous coordinate of the epipole in the second image.
- Choose a location (x'_0, y'_0) in the second image, e.g., the center of the image, and determine a 3×3 homogeneous translation matrix T that moves (x'_0, y'_0) to the origin.
- Determine a rotation matrix R that movels the epipole onto the x-axis; let its new location be $(x^*, 0)$.
- Compute H_2 as

$$H_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1/x^* & 0 & 1 \end{bmatrix} RT \tag{40}$$

Rectification method proposed by Hartley (cont.)

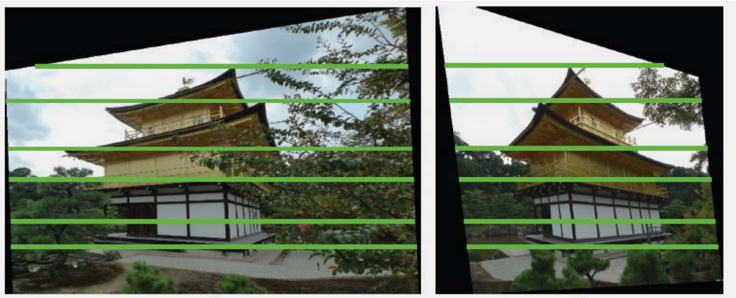
5.(cont.) The first matrix moves the epipole to infinity along the x-axis.
6.Apply the projective transformation $H_2 M$ to the features in I_1 and the projective transformation H_2 to the features in I_2 to get a transformed set of feature matches.
7.At this point, the two images are rectified, but applying $H_2 M$ to I_1 may result in an unacceptably distorted image. The next step is to find a horizontal shear and translation that bring the feature matches as close together as possible.

$$\sum_{i=1}^{n} (a\hat{x}_i + b\hat{y}_i + c - \hat{x}'_i)^2 \tag{41}$$

8.Compute H_1 as

$$H_1 = \begin{bmatrix} a & b & c \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} H_2 M \tag{42}$$

Example



Stereo correspondence

Estimating the disparity map

Quantized disparity values

Occlusions are explicitly modeled

Monotonicity assumption

Calibration is assumed known

Images are acquired simultaneously