Pixel Labeling: Stereo Vision and Optic Flow[1]

(80 min lecture)

See Material in
Reinhard Klette: Concise Computer Vision
Springer-Verlag, London, 2014

ccv.wordpress.fos.auckland.ac.nz

[1] See last slide for copyright information.

Agenda

Generic Model for Matching

Given: Left image L and right image R

One is the *base image* B, the other one the *match image* M

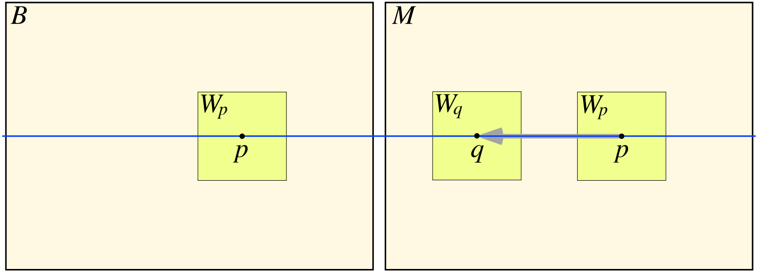
Matching Task

For $(x, y, B(x, y))$ search *corresponding pixel* $(x + d, y, M(x + d, y))$

Epipolar line identified by row y, and d is the disparity

Two pixels are corresponding *iff*
they are projections of the same point $P = (X, Y, Z)$ in the shown scene

$B = L$ and $M = R$



Basic Idea

Start at pixel p in B, consider its neighborhood defined by a square window

Compare with neighborhoods around pixels q on the epipolar line in M

Search for best match of pixel neighborhoods

Search Interval for $B = L$ and $M = R$

Initiate search by selecting $p = (x, y)$ in B

Search interval: $\max\{x - d_{\max}, 1\} \leq x + d \leq x$ for $q = (x + d, y)$ in M

In other words:

$$0 \leq -d \leq \min\{d_{\max}, x - 1\}$$

Example

Start at $p = (1, y)$ in B

Then we can only consider $d = 0$ (i.e. a point P "at infinity")

If no "reasonable" similarity of neighborhoods of $p = (1, y)$ in B and $q = (1, y)$ in M then do not assign disparity 0 to p

If Also Considering Smoothness Cost ...

Stereo matcher assigns disparity f_p to pixel location $p \in \Omega$

$E_{data}(p, f_p) =$ dissimilarity cost (error) between
local neighbourhood around p in B and
local neighbourhood around pixel in M defined by disparity f_p

$E_{smooth}(f_p, f_q) =$ dissimilarity cost (error) between disparity f_p at p and disparity f_q at an adjacent location q

Goal for a stereo matcher: Minimise the total error

$$E(f) = \sum_{p \in \Omega} \left[E_{data}(p, f_p) + \sum_{q \in A(p)} E_{smooth}(f_p, f_q) \right]$$

Will be discussed in detail in the next (i.e. the MRF) lecture

Markov, Bayes, Gibbs, and Pixel-interaction

The Russian mathematician A. A. Markov (1856 – 1922) studied stochastic processes where the interaction of multiple random variables can be modeled by an undirected graph. These models are today known as *Markov random fields* (MRFs).

If the underlying graph is directed and acyclic, then we have a *Bayesian network*, named after the English mathematician T. Bayes (1701 – 1761).

If we only consider strictly positive random variables then an MRF is called a *Gibbs random field*, named after the US-American scientist J. W. Gibbs (1839 – 1903).

Here: Error- (or energy-) minimisation by pixel-interaction on undirected pixel-adjacency graphs; labels assigned to pixels play the role of random variables; assigned labels and pixel-interaction specify an MRF model

Agenda

Neighborhoods for Correspondence Search

Consider $(2l+1) \times (2k+1)$ windows

$W_p^{l,k}(B)$ around reference point p in image B and

$W_q^{l,k}(M)$ around reference point q in image M

Consider image row y (the current epipolar line) and

compare values in those local neighborhoods of p and q

Examples of Simple Data Cost Terms

$p = (x, y)$ and $q = (x + d, y)$

SSD data cost measure

$$E_{SSD}(p, d) = \sum_{i=-l}^{l} \sum_{j=-k}^{k} [B(x + i, y + j) - M(x + d + i, y + j)]^2$$

SSD for *sum of squared differences*

SAD for *sum of absolute differences*

SAD data cost measure

$$E_{SAD}(p, d) = \sum_{i=-l}^{l} \sum_{j=-k}^{k} |B(x + i, y + j) - M(x + d + i, y + j)|$$

Five Reasons Why Just SSD or SAD Will Not Work

1. *Invalidity of Intensity Constancy Assumption* (ICA). Intensity values at corresponding pixels, and in their neighborhoods, typically impacted by lighting variations, or just by image noise
2. *Local reflectance differences*. Due to different viewing angles, P and its neighborhood reflect light differently to cameras recording B and M
3. *Differences in cameras*. Different gains or offsets in cameras used result in high SAD or SSD errors
4. *Perspective distortion*. 3D point $P = (X, Y, Z)$ is on a sloped surface; local neighborhood around P on this surface is differently projected into images B and M
5. *No unique minimum*. There might be several pixel locations q defining the same minimum

Zero-Mean Version

Calculate mean $\overline{B}_x$ of a used window $W_x^{l,k}(B)$, and mean $\overline{M}_{x+d}$ of window $W_{x+d}^{l,k}(M)$, subtract $\overline{B}_x$ from all values in $W_x^{l,k}(B)$, and $\overline{M}_{x+d}$ from all values in $W_{x+d}^{l,k}(M)$, calculate this way the data-cost function in its *zero-mean version*

Option for reducing impact of lighting artifacts (i.e. not depending on ICA)

Indicated by starting subscript of data-cost function with a Z

Example: E_{ZSSD} or E_{ZSAD} are zero-mean SSD or zero-mean SAD data-cost functions

$$E_{ZSSD}(x,d) = \sum_{i=-l}^{l} \sum_{j=-k}^{k} \left[(B_{x+i,y+j} - \overline{B}_x) - (M_{x+i+d,y+j} - \overline{M}_{x+d}) \right]^2$$

$$E_{ZSAD}(x,d) = \sum_{i=-l}^{l} \sum_{j=-k}^{k} \left| [B_{x+i,y+j} - \overline{B}_x] - [M_{x+d+i,y+j} - \overline{M}_{i+d}] \right|$$

NCC Data Cost

Normalized cross correlation (NCC) already used for comparing two images

Already defined by zero-mean normalization, but we add Z to the index for uniformity in notation; let $E_{ZNCC}(x, d) =$

$$1 - \frac{\sum_{i=-l}^{l} \sum_{j=-k}^{k} \left[B_{x+i,y+j} - \overline{B}_x\right] \left[M_{x+d+i,y+j} - \overline{M}_{x+d}\right]}{\sqrt{\sigma^2_{B,x} \cdot \sigma^2_{M,x+d}}}$$

where

$$\sigma^2_{B,x} = \sum_{i=-l}^{l} \sum_{j=-k}^{k} \left[B_{x+i,y+j} - \overline{B}_x\right]^2$$

$$\sigma^2_{M,x+d} = \sum_{i=-l}^{l} \sum_{j=-k}^{k} \left[M_{x+d+i,y+j} - \overline{M}_{x+d}\right]^2$$

Census Data-Cost Function

The *zero-mean normalized census cost function*

$$E_{ZCEN}(x, d) = \sum_{i=-l}^{l} \sum_{j=-k}^{k} \rho(x+i, y+j, d)$$

with

$$\rho(u, v, d) = \begin{cases} 0 & B_{uv} \perp \overline{B}_x \text{ and } M_{u+d,v} \perp \overline{M}_{x+d} \\ 1 & \text{otherwise} \end{cases}$$

where $\perp$ either $<$ or $>$

By using B_x instead of $\overline{B}_x$, and M_{x+d} instead of $\overline{M}_{x+d}$, we have the census data-cost function E_{CEN}

Example for Census Data Cost

Windows $W_x(B)$ and $W_{x+d}(M)$

2	1	6
1	2	4
2	1	3

5	5	9
7	6	7
5	4	6

Have $\overline{B}_x \approx 2.44$ and $\overline{M}_{x+d} \approx 6.11$

$i = j = -1$ results in $u = x - 1$ and $v = y - 1$

$B_{x-1,y-1} = 2 < 2.44$ and $M_{x-1+d,y-1} = 5 < 6.11$

Thus $\rho(x-1, y-1, d) = 0$

$i = j = +1$ results in $u = x + 1$ and $v = y + 1$

$B_{x+1,y+1} = 3 > 2.44$ but $M_{x+1+d,y+1} = 6 < 6.11$

Thus $\rho(x+1, y+1, d) = 1$

$i = j = -1$: values in the same relation with respect to the mean

$i = j = +1$: opposite relationships

Result for Example

For the given example: $E_{ZCEN} = 2$

Spatial distribution of ρ-values

0	0	0
1	0	0
0	0	1

Vector $\mathbf{c}_{x,d}$ lists these ρ-values in a left-to-right, top-to-bottom order:

$$[0, 0, 0, 1, 0, 0, 0, 0, 1]^\top$$

Hamming Distance

Let $\mathbf{b}_x$ be the vector listing results $\mathrm{sgn}(B_{x+i,y+j} - \overline{B}_x)$ in a left-to-right, top-to-bottom order, where sgn is the signum function

Similarly, $\mathbf{m}_{x+d}$ lists values $\mathrm{sgn}(M_{x+i+d,y+j} - \overline{M}_{x+d})$

For the values in previous example

$$\begin{aligned}
\mathbf{b}_x &= [-1,-1,+1,-1,-1,+1,-1,-1,+1]^\top \\
\mathbf{m}_{x+d} &= [-1,-1,+1,+1,-1,+1,-1,-1,-1] \\
\mathbf{c}_{x,d} &= [\ 0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0,\ 0,\ 1\ ]^\top
\end{aligned}$$

Vector $\mathbf{c}_{x,d}$ shows positions where $\mathbf{b}_x$ and $\mathbf{m}_{x+d}$ differ; the number of positions where two vectors differ is known as *Hamming distance*

Efficient Calculation

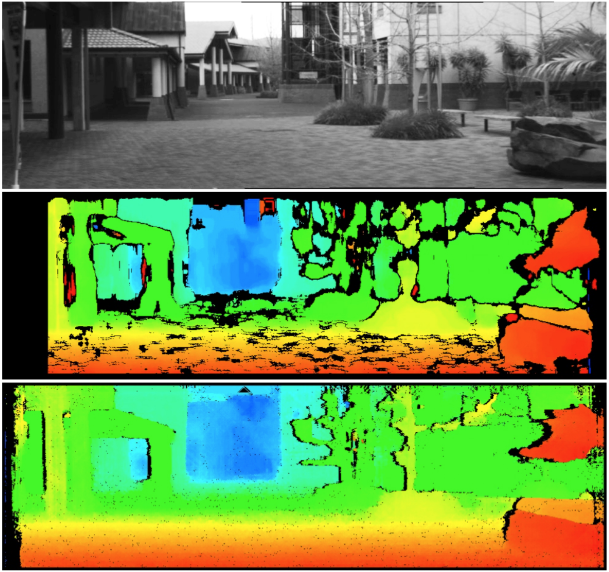
Observation The zero-mean normalized census data cost $E_{ZCEN}(x, d)$ equals the Hamming distance between vectors $\mathbf{b}_x$ and $\mathbf{m}_{x+d}$

By replacing values "-1" by "0" in vectors $\mathbf{b}_x$ and $\mathbf{m}_{x+d}$, Hamming distance for resulting binary vectors can be calculated very time-efficiently

Steps Towards Stereo Vision

1. Choose 2 (or more) cameras appropriate for application
2. Aim at “CSG installation” of cameras
3. Calibrate cameras
4. Rectify recorded images using calibration results
5. Choose a stereo matcher for finding corresponding points
6. Possibly use $B = L$ and $M = R$, followed by $B = R$ and $M = L$
7. Evaluate calculated disparities (apply a confidence measure)
8. Calculate depth from disparities
9. Possibly approximate a surface model based on depth values

Varying Qualities of Stereo Matchers



Caption to Figure on Page Before

Top: Input image of a stereo sequence
recorded at Tamaki campus, The University of Auckland

Middle: Disparity map using a local matcher (block matching, as available in `OpenCV` beginning of 2013

Bottom: Disparity map using `iSGM` as stereo matcher which applies a 3×9 zero-mean normalized census data cost term

Comparative Evaluations of Stereo Matchers

For examples of test data and performance of stereo matchers, see

1. KITTI: `www.cvlibs.net/datasets/kitti/index.php`
2. HCI: `ci.iwr.uni-heidelberg.de/Static/challenge2012`
3. EISATS: `ccv.wordpress.fos.auckland.ac.nz/eisats`
4. Middlebury Stereo Vision: `vision.middlebury.edu/stereo/`

It is also an important task to evaluate the provided test data (what kind of challenges are given by a set of data); the performance of stereo matchers depends on input data (lighting, complexity of scene, trajectories of moving objects, etc.)

For a clip showing iSGM results on HCI test data, see `www.mi.auckland.ac.nz/DATA/CCV/VideoStereoGrey`

Agenda

Frames in a Video Sequence and Optic Flow

We consider a sequence of scalar images, also called *frames*

Time difference δt between two subsequent time slots

$I(.,.,t)$ is the frame at time slot t with values $I(x, y, t)$

Example: $\delta t = 1/30$ s means 30 Hz (read: "hertz") or

30 fps (read: "frames per second") or 30 pps (read: "pictures per second")

The *optic flow* $\mathbf{u}(x, y) = (u(x, y), v(x, y))$

is the visible motion of a pixel at (x, y) into a pixel at $(x + u(x, y), y + v(x, y))$ between two subsequent frames

The Horn-Schunck Algorithm

Taylor expansion for frame sequence:

$$\begin{aligned} I(x+\delta x, y+\delta y, t+\delta t) \\ &= I\left(x,y,t\right) + \delta x \cdot \frac{\partial I}{\partial x}\left(x,y,t\right) + \delta y \cdot \frac{\partial I}{\partial y}\left(x,y,t\right) \\ &+ \delta t \cdot \frac{\partial I}{\partial t}\left(x,y,t\right) + e \end{aligned}$$

Assumption 1.
Let $e = 0$, i.e. $I(.,.,.)$ linear for *small* values of δx, δy, and δt

Assumption 2.
δx and δy model the motion u and v of one pixel between t and $t+1$

Assumption 3.
Intensity constancy assumption $\quad I\left(x+\delta x, y+\delta y, t+\delta t\right) = I\left(x,y,t\right)$

Horn-Schunck Constraint or Optic Flow Equation

$$0 = \frac{\delta x}{\delta t} \cdot \frac{\partial I}{\partial x}(x, y, t) + \frac{\delta y}{\delta t} \cdot \frac{\partial I}{\partial y}(x, y, t) + \frac{\partial I}{\partial t}(x, y, t)$$

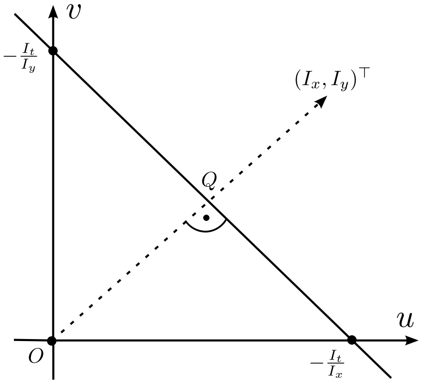
Changes in x- and y-coordinate during δt as optic flow

$$0 = u(x, y, t) \cdot \frac{\partial I}{\partial x}(x, y, t) + v(x, y, t) \cdot \frac{\partial I}{\partial y}(x, y, t) + \frac{\partial I}{\partial t}(x, y, t)$$

Short form:

$$0 = uI_x + vI_y + I_t$$

The *uv* Velocity Space



Straight line

$$-I_t = u \cdot I_x + v \cdot I_y = \mathbf{u} \cdot \nabla_{x,y} I$$

in *uv* velocity space, with optic flow vector $\mathbf{u} = [u, v]^\top$

Agenda

Labeling Model, Constraints, and an MRF

Labeling function f assigns *label* (u, v) to $p \in \Omega$ in $I(., ., t)$

Possible set of vectors $(u, v) \in \mathbb{R}^2$ defines the set of labels

Data error or *data energy*

$$E_{data}(f) = \sum_{\Omega} [\, u \cdot I_x + v \cdot I_y + I_t \,]^2$$

Smoothness error or *smoothness energy*

$$E_{smooth}(f) = \sum_{\Omega} u_x^2 + u_y^2 + v_x^2 + v_y^2$$

where u_x is the 1^{st} order derivative of u with respect to x, and so forth

Derivatives define dependencies between adjacent pixels: our first MRF

The Optimization Problem

Task: Calculate labelling function f which minimizes

$$E_{total}(f) = E_{data}(f) + \lambda \cdot E_{smooth}(f)$$

where $\lambda > 0$ is a weight, e.g. $\lambda = 0.1$

Characterization: *Total variation* (TV)

Search for an optimum f in the set of all possible labelings
We apply L_2-penalties for error terms, thus TVL_2 optimization

Applied solution strategy: *least-square error* (LSE) *optimization*

1. Define an error or energy function. – DONE
2. Calculate derivatives of this function with respect to all the unknown parameters. – NEXT ON OUR LIST
3. Set derivatives equal to zero and solve equational system with respect to the unknowns. Result defines minimum of the error function.

Copyright Information

This slide show was prepared by Reinhard Klette
with kind permission from Springer Science+Business Media B.V.

The slide show can be used freely for presentations.
However, *all the material* is copyrighted.

In case of citation: just cite the book, that's fine.