Intelligent Control Systems

Visual Tracking (1)

— Feature Point Tracking and Block Matching —

Shingo Kagami

Graduate School of Information Sciences,

Tohoku University

swk(at)ic.is.tohoku.ac.jp

http://www.ic.is.tohoku.ac.jp/ja/swk/

Outline

- Tracking of a Point
- Block Matching
- Gradient Methods
- Feature Point Detector

Tracking of “a Point”

To track a point
= To determine the motion vector of a point from a frame to its next frame (discrete time)
$\simeq$ To determine the velocity vector of a point (continuous time)

- Distribution of the motion vectors over the image is called optical flow
  - dense optical flow
  - sparse optical flow

- Sometimes the terms “motion vector” and “optical flow” are used interchangeably (depending on the context)

Optical Flow Constraint

Assuming that the intensity of the tracked point is constant,

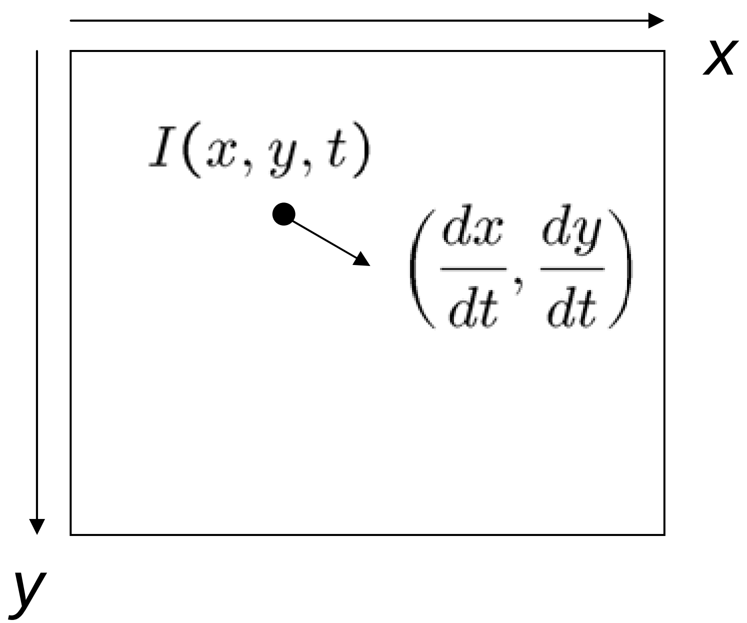
$$\begin{aligned} I(x,y,t) &= I(x+dx, y+dy, t+dt) \\ &= I(x,y,t) + \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt + \epsilon \end{aligned}$$

Ignoring the 2nd order or higher terms ϵ yields

$$\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = 0$$

This single equation is not enough to determine the two components $\left(\frac{dx}{dt}, \frac{dy}{dt}\right)$

[Horn and Schunck 1981]



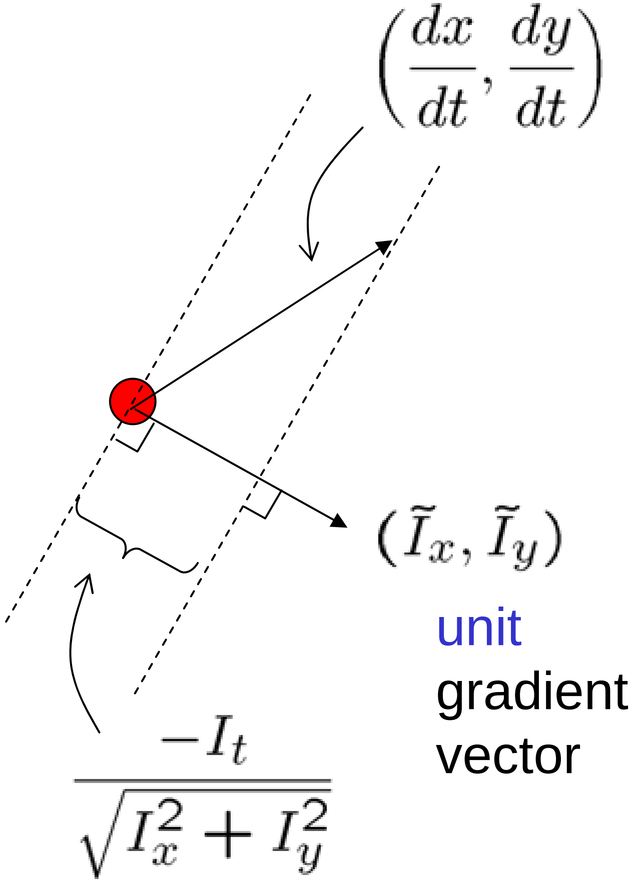
Interpretation of the Constraint

With $\frac{\partial I}{\partial x} = I_x, \frac{\partial I}{\partial y} = I_y, \frac{\partial I}{\partial t} = I_t$

$$(I_x, I_y)\left(\frac{dx}{dt}, \frac{dy}{dt}\right)^T = -I_t$$

$$(\tilde{I}_x, \tilde{I}_y)\left(\frac{dx}{dt}, \frac{dy}{dt}\right)^T = \frac{-I_t}{\sqrt{I_x^2 + I_y^2}}$$

where $(\tilde{I}_x, \tilde{I}_y)$ is a unit vector parallel to (I_x, I_y)



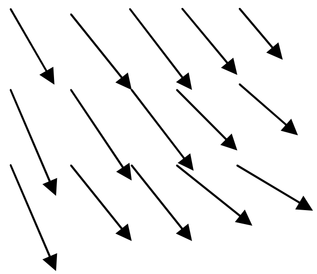
Only the component in the direction of the gradient vector is determined (aperture problem)

Additional Assumptions

Thus we cannot determine the optical flow from I_x, I_y and I_t. Additional assumptions are needed.

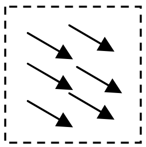
ex1) Optical flow changes smoothly in space

- [Horn and Schunck 1981]



ex2) Optical flow is constant within a small neighborhood of a point

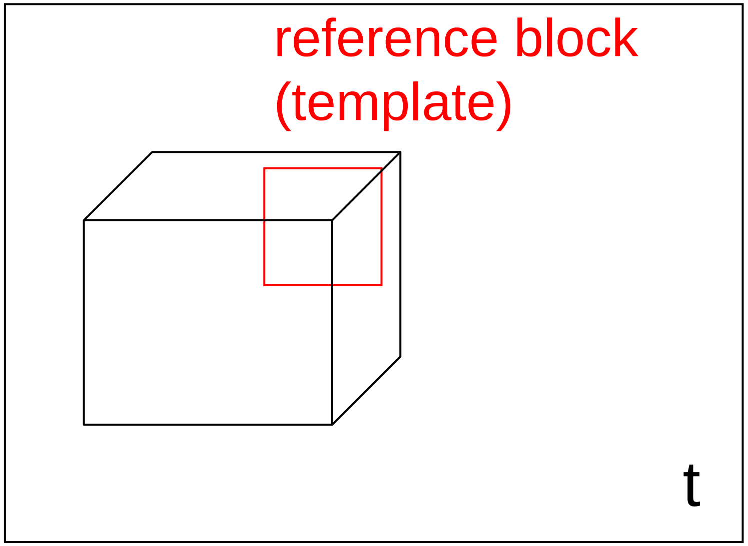
- We will investigate this in the followings
- So, what we call “point tracking” is actually “patch (block) tracking”

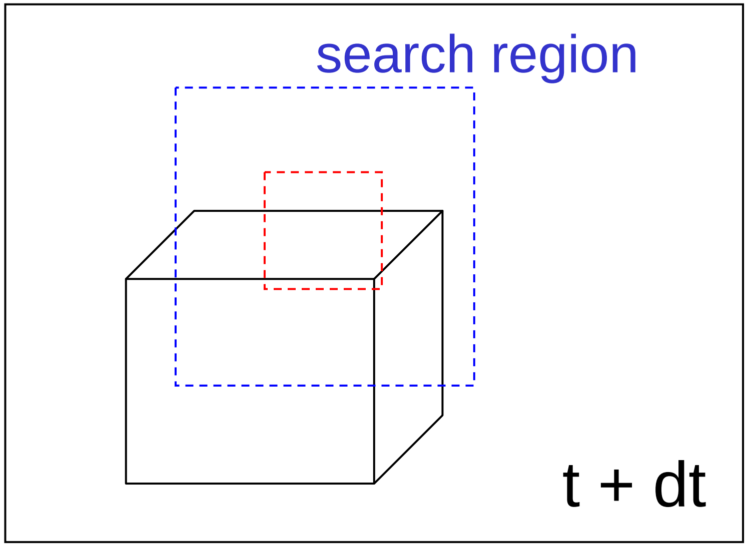


Outline

- Tracking of a Point
- Block Matching
- Gradient Methods
- Feature Point Detector

Block Matching





Slides and compares the reference block through the search region

- How to compare?: by computing evaluation functions
- So, this is full-search optimization

Evaluation Functions

Letting $T_{i,j}$ denote the pixel value at (i, j) in the reference block:

$$d_{\mathbf{SSD}}(x,y) = \sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(T_{i,j} - I_{x+i,y+j})^2$$

: sum of squared differences
$\rightarrow$ min

$$d_{\mathbf{SAD}}(x,y) = \sum_{i=0}^{m-1}\sum_{j=0}^{n-1}|T_{i,j} - I_{x+i,y+j}|$$

: sum of absolute differences
$\rightarrow$ min

$$C(x,y) = \sum_{i=0}^{m-1}\sum_{j=0}^{n-1}T_{i,j}I_{x+i,y+j}$$

: cross correlation
$\rightarrow$ max

average

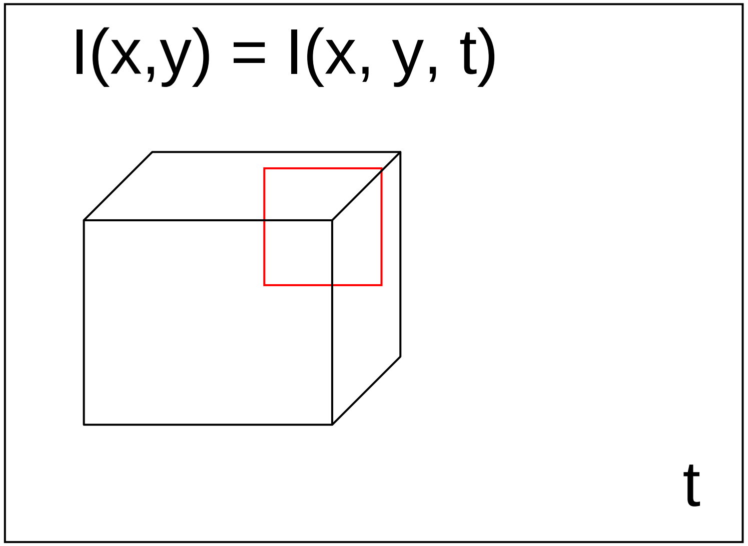
$$C_{\mathbf{n}}(x,y) = \frac{\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(T_{i,j} - \bar{T})(I_{x+i,y+j} - \bar{I}_{x,y})}{\sqrt{\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(T_{i,j} - \bar{T})^2 \sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(I_{x+i,y+j} - \bar{I}_{x,y})^2}}$$

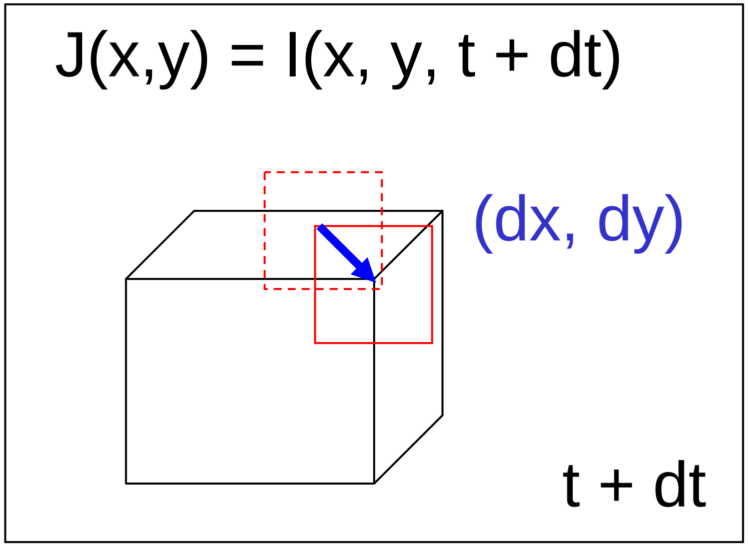
: normalized cross correlation
$\rightarrow$ max

Outline

- Tracking of a Point
- Block Matching
- Gradient Methods
- Feature Point Detector

Utilize Gradients to Explore the Solution





$$I(x, y, t) = I(x + dx, y + dy, t + dt)$$

$$I(x - dx, y - dy, t) = I(x, y, t + dt)$$

Writing $I(x, y, t) = I(x, y),\ I(x, y, t + dt) = J(x, y)$

$$I(x - dx, y - dy) = J(x, y)$$

Using 1st order Taylor expansion of I(x, y), SSD is written as

$$
\begin{aligned}
E(dx, dy) &= \sum_{u,v} \{I(x - dx + u, y - dy + v) - J(x + u, y + v)\}^2 \\
&= \sum_{u,v} \{I(x + u, y + v) - I_x(x + u, y + v)dx \\
&\qquad - I_y(x + u, y + v)dy - J(x + u, y + v)\}^2 \\
&= \sum \{h - I_x dx - I_y dy\}^2 \quad (h \equiv I - J) \\
&= \sum \left\{h - (I_x, I_y) \begin{pmatrix} dx \\ dy \end{pmatrix} \right\}^2
\end{aligned}
$$

To minimize E, derivative of E w.r.t. (dx, dy) is equated to 0

$$
\sum \left\{h - (I_x, I_y) \begin{pmatrix} dx \\ dy \end{pmatrix} \right\} \begin{pmatrix} I_x \\ I_y \end{pmatrix} = 0
$$

$$\sum \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \sum h \begin{pmatrix} I_x \\ I_y \end{pmatrix}$$

$$\begin{pmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} \sum (I - J) I_x \\ \sum (I - J) I_y \end{pmatrix}$$

$$G \begin{pmatrix} dx \\ dy \end{pmatrix} = e$$

Then, (dx, dy) is obtained by solving this linear equation (Lucas-Kanade method [Lucas and Kanade 1981])

- (dx, dy) is only approximately obtained because of the 1st order Taylor approximation. We often need to iteratively run the above process by setting J(x, y) := J(x + dx, y + dy) to obtain a good result

Generalization

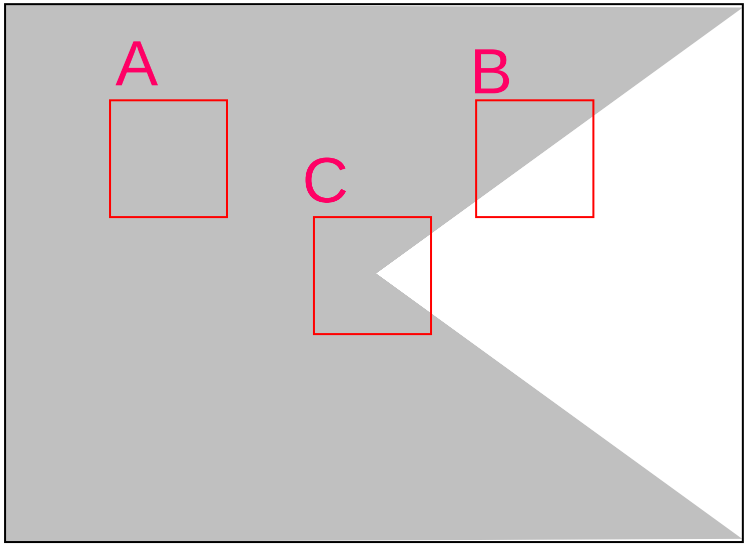
Lucas-Kanade method is viewed as an application of Gauss-Newton method (an iterative non-linear optimization method for least square problems).

- Formulations for more general image transformations are possible: e.g.
  - translation + rotation
  - translation + rotation + magnification
  - affine transformation
  - perspective transformation
- Other optimization methods could be employed: e.g.
  - Levenberg-Marquardt method
  - Efficient Second-order Minimization method

Outline

- Tracking of a Point
- Block Matching
- Gradient Methods
- Feature Point Detector

What is Good Point to Track

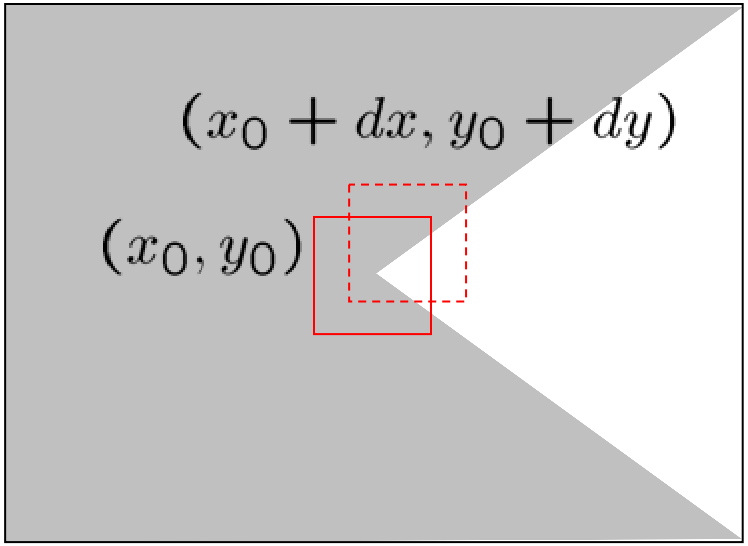


Recall that we aggregate many flows within a small block to obtain enough constraints

A: Block with constant intensity is not suitable (0 constraint)

B: Block including only edges with the same direction is also not suitable (essentially 1 constraint)

How to find a block like C?



Consider two blocks

- around a point of interest (x_0, y_0)
- around the point $(x_0 + dx, y_0 + dy)$

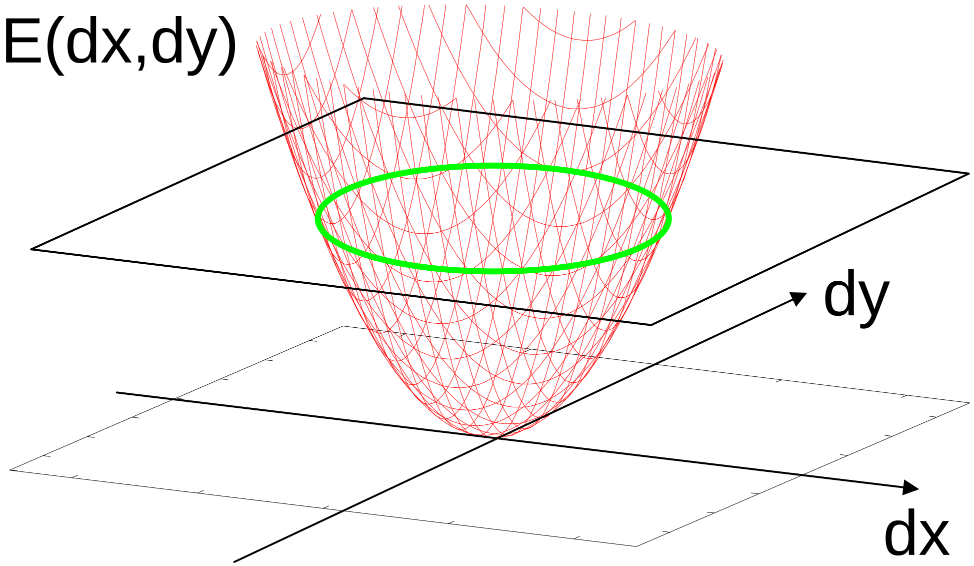
These two blocks should not resemble each other for any choice of (dx, dy)

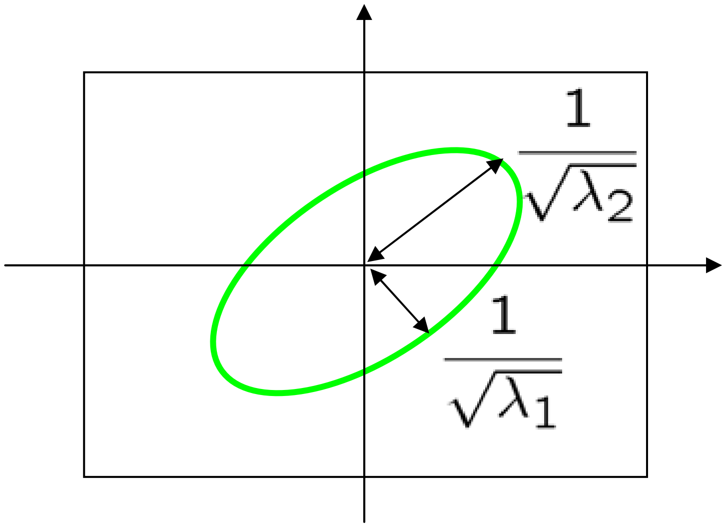
Let’s measure how they do not resemble by SSD

$$E(dx, dy) \equiv \sum_{u,v} \{I(x_0 + dx + u, y_0 + dy + v) - I(x_0 + u, y_0 + v)\}^2$$

With 1st order Taylor expansion,

$$\begin{aligned} E(dx, dy) &= \sum_{u,v} \{I_x(x_0 + u, y_0 + v)dx + I_y(x_0 + u, y_0 + v)dy\}^2 \\ &= \sum I_x^2 dx^2 + 2\sum I_x I_y dxdy + \sum I_y^2 dy^2 \\ &= (dx, dy) \begin{pmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} \\ &= (dx, dy) G \begin{pmatrix} dx \\ dy \end{pmatrix} \end{aligned}$$





E(dx,dy) = 1

$$E(dx, dy) = (dx, dy)G\begin{pmatrix} dx \\ dy \end{pmatrix} = 1$$

is an ellipse in (dx, dy) plane. This ellipse should be as small as possible and should be close to true circle.

i.e.: Eigenvalues λ_1, λ_2 of G should be large enough and close to each other.

This is compatible with numerical stability in solving $G\begin{pmatrix} dx \\ dy \end{pmatrix} = e$

(Just in case you forget linear algebra)

$$(dx, dy)G \begin{pmatrix} dx \\ dy \end{pmatrix} = 1$$

Noting that G is symmetric, G can be diagonalized by an orthonormal matrix P (i.e. $P^{-1} = P^T$) so that $P^T G P = \mathrm{diag}(\lambda_1, \lambda_2)$

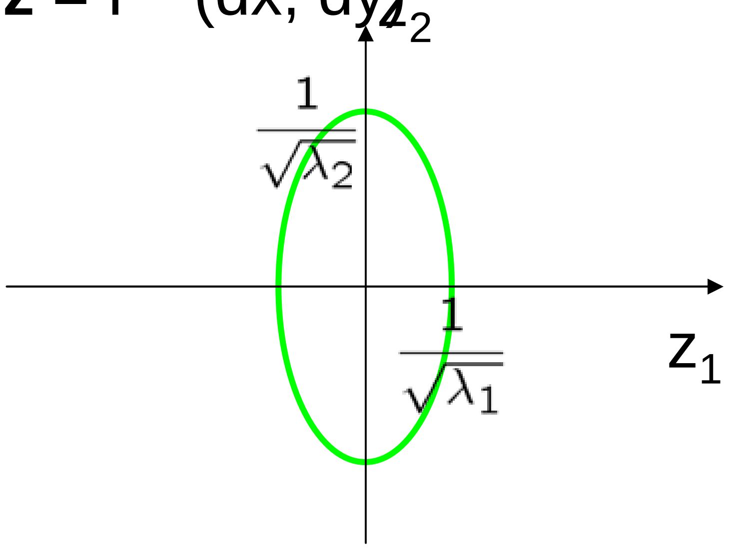
$$(dx, dy)P \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} P^T \begin{pmatrix} dx \\ dy \end{pmatrix} = 1$$

Viewed in a new coordinate system $\mathbf{z} = P^T (dx, dy)^T$

$$z^T \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} z = 1$$

Or, equivalently

$$\lambda_1 z_1^2 + \lambda_2 z_2^2 = 1$$



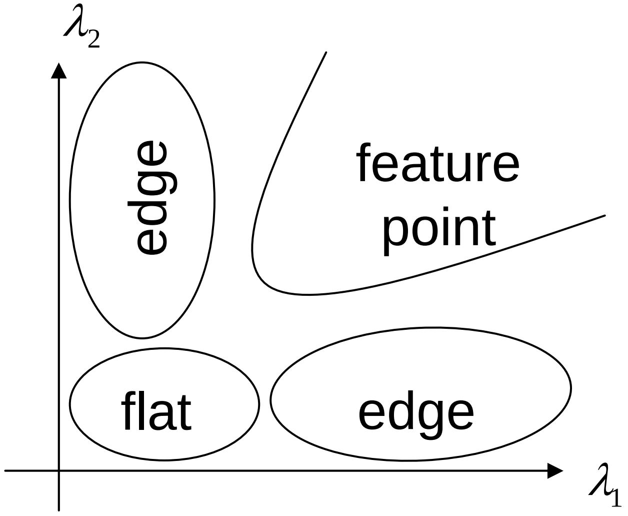
Feature Point Detector

Harris operator

[Harris and Stephens 1988]

$$\det G - k(\operatorname{tr} G)^2$$

$$= \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2$$

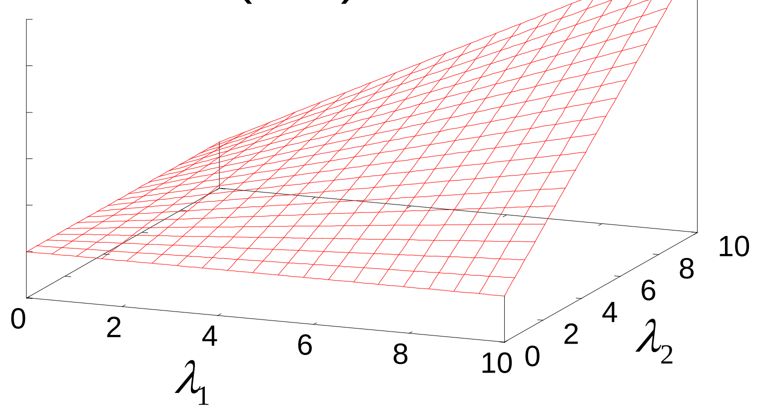


Good Features to Track

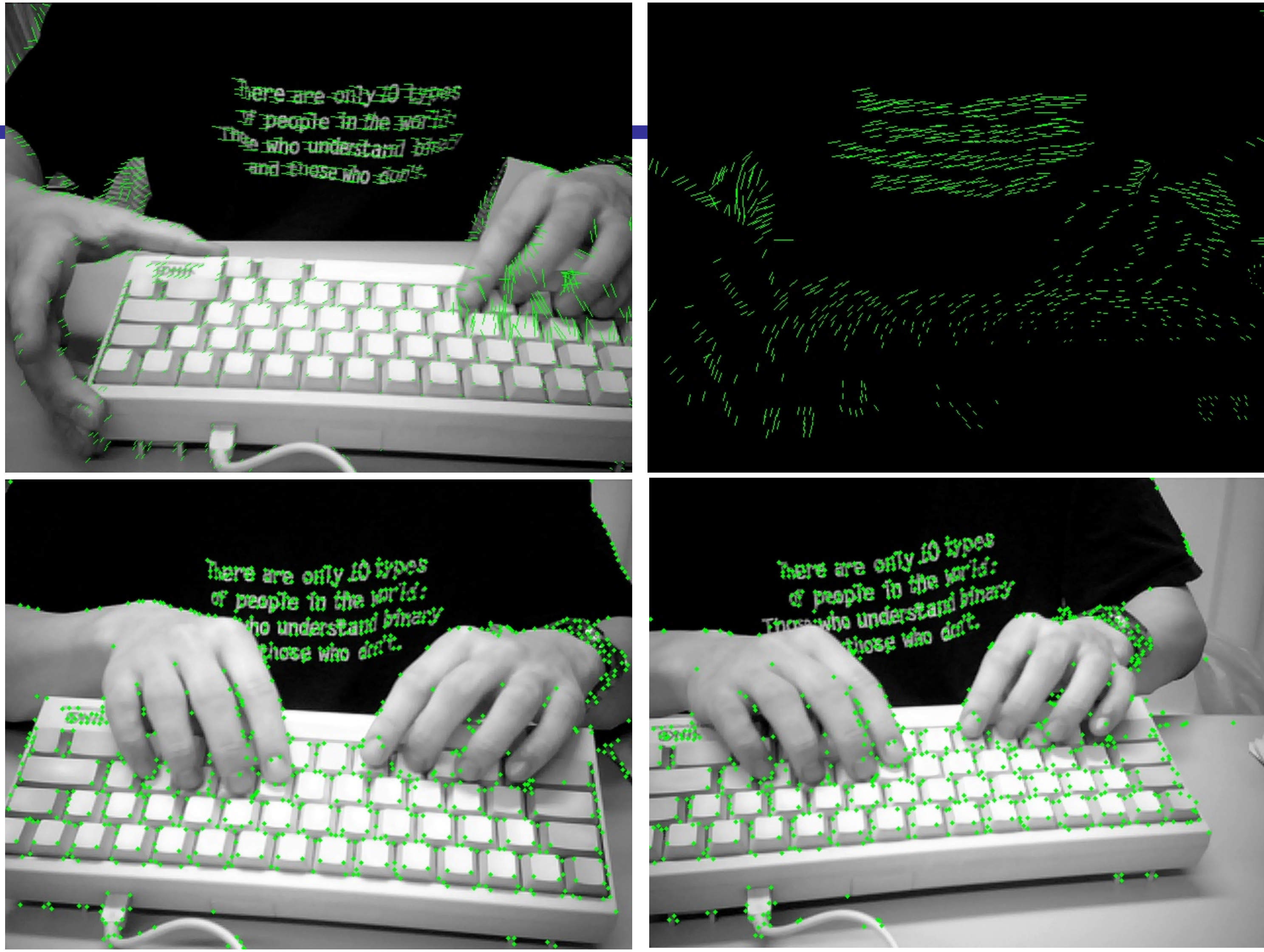
[Tomasi and Kanade 1991]

$$\min(\lambda_1, \lambda_2)$$

det G – k(tr G)^2



These “good” points for tracking and/or matching are called feature point, interest point, keypoint and so on.



Lucas-Kanade method applied to "Good-features-to-track" points is often called KLT (Kanade-Lucas-Tomasi) tracker

Summary

- Tracking of a Point
  - is an ill-posed problem
  - a block is often considered instead of a point

- Block Matching
  - full-search optimization of an evaluation function
    - SSD, SAD, (normalized) cross correlation

- Lucas-Kanade method
  - a Gradient method for optimization of SSD

- Feature Point Detector
  - Harris operator, Good feature to track
  - KLT tracker

References

- B. K. P. Horn and B. G. Schunck: Determining Optical Flow, Artificial Intelligence, vol.17, pp.185-203, 1981.
- C. Harris and M. Stephens: A Combined Corner and Edge Detector, Proc. 14th Alvey Vision Conference, pp.147-151, 1988.
- B. D. Lucas and T. Kanade: An Iterative Image Registration Technique with an Application to Stereo Vision, Proc. 7th International Conference on Artificial Intelligence, pp.674-679, 1981.
- C. Tomasi and T. Kanade: Detection and Tracking of Point Features, Shape and Motion from Image Streams: a Factorization Method –Part 3, Technical Report CMU-CS-91-132, School of Computer Science, Carnegie Mellon University, 1991.

Sample codes are in sample20140701.zip available at
http://www.ic.is.tohoku.ac.jp/~swk/lecture/