Intelligent Control Systems

Visual Tracking (1)

— Pixel-intensity-based methods —

Shingo Kagami

Graduate School of Information Sciences,

Tohoku University

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

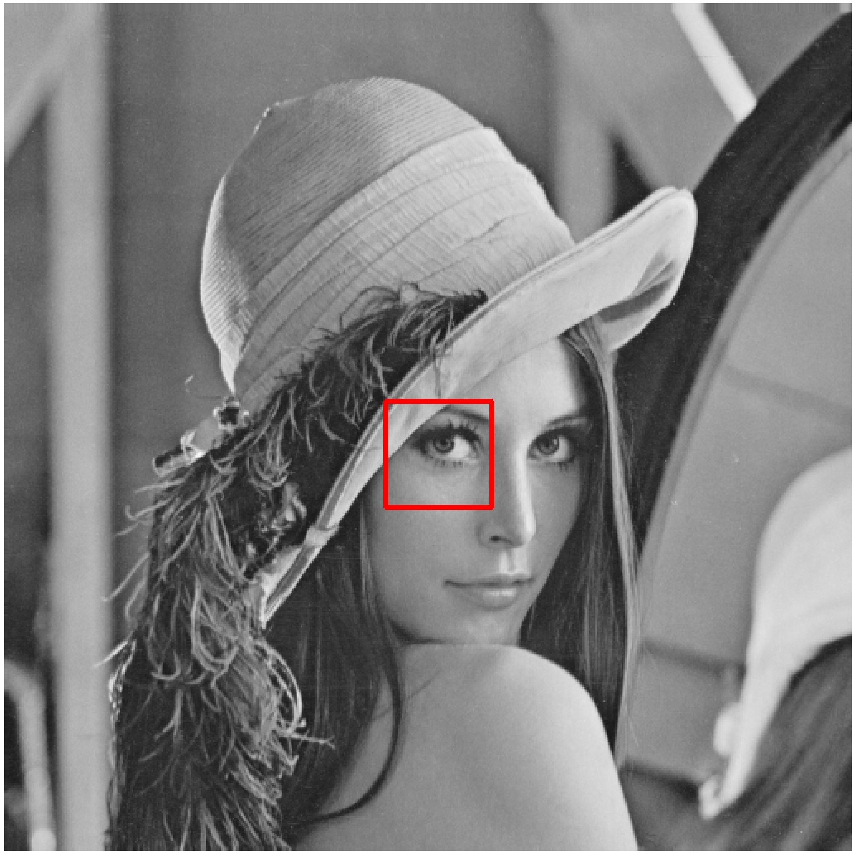
http://www.ic.is.tohoku.ac.jp/ja/swk/

Agenda

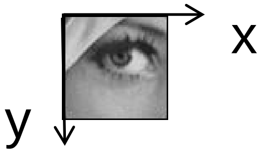
- Template Matching (Brute force)
- Gradient-based Template Matching
- Optical Flow Computation
- Lucas-Kanade method for General Warps

Visual Tracking

input image



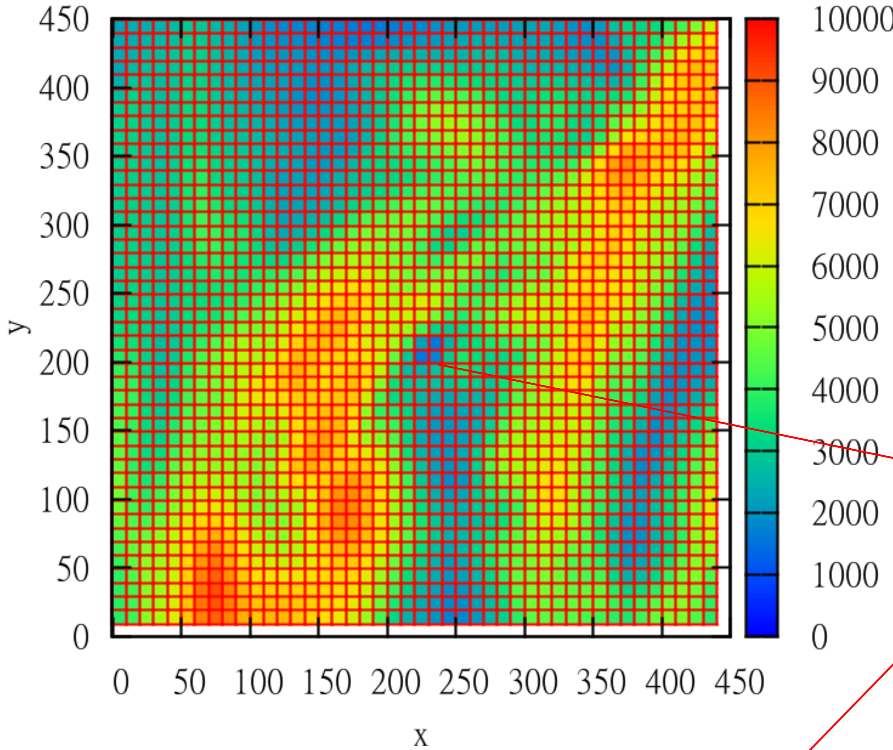
template image $T_{x,y}$

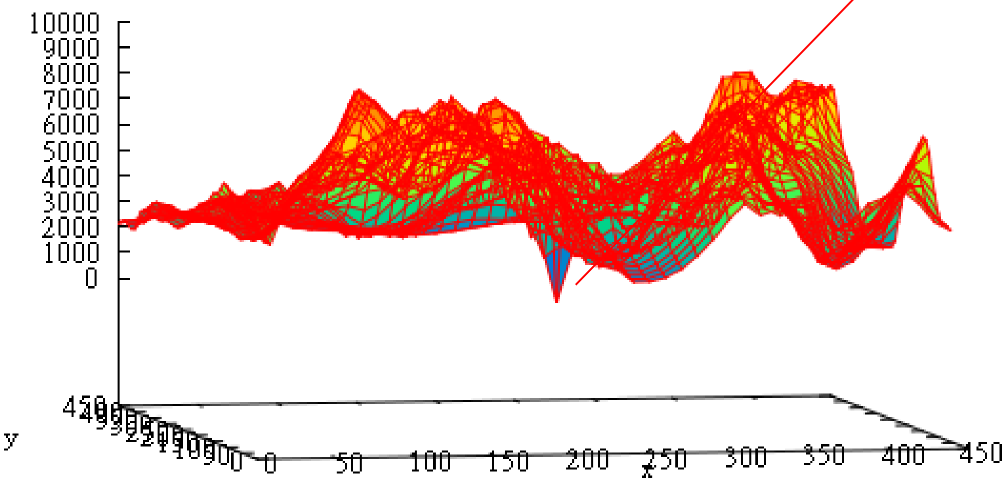


Matching (Detection) Problem:
Find the area with the best similarity to the template

Matching is often called "tracking" when it is sequentially done with time

- Evaluate a dissimilarity measure for every possible position





(3)

Minimum point of dissimilarity measure (here, sum of squared difference of pixel intensities)

Examples of Evaluation Functions

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

: sum of squared differences
! 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
! min

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

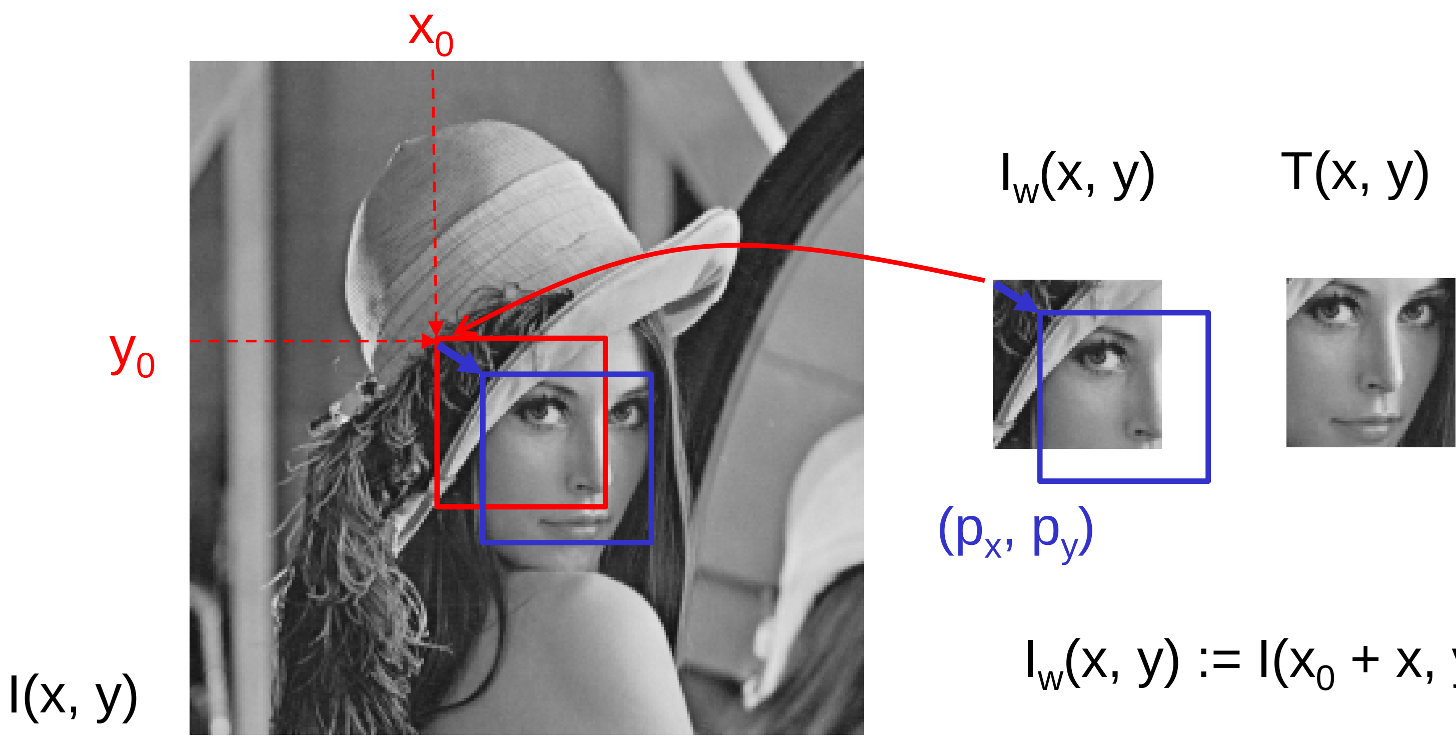
: cross correlation
! max

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
! max

Gradient-based Optimization



$$I_w(x, y) := I(x_0 + x, y_0 + y)$$

Instead of brute force search for the minimum, let us consider application of Gauss-Newton optimization method to

$$E(p_x, p_y) = \sum_{m,n} \{I_W(p_x + m, p_y + n) - T(m, n)\}^2$$

Lucas-Kanade Method (forward algorithm)

1st order approximation is applied and partial derivatives are equated to 0:

$$E(p_x, p_y) \simeq \sum_{m,n} \left\{ I_{\mathrm{w}}(m,n) + \frac{\partial I_{\mathrm{w}}}{\partial x}(m,n)p_x + \frac{\partial I_{\mathrm{w}}}{\partial y}(m,n)p_y - T(m,n) \right\}^2$$

$$= \sum_{m,n} \left\{ e(m,n) + \frac{\partial I_{\mathrm{w}}}{\partial x}(m,n)p_x + \frac{\partial I_{\mathrm{w}}}{\partial y}(m,n)p_y \right\}^2 \to \min_{p_x, p_y}$$

$$\frac{\partial E}{\partial p_x} = 2\sum_{m,n} \left\{ e(m,n) + \frac{\partial I_{\mathrm{w}}}{\partial x}(m,n)p_x + \frac{\partial I_{\mathrm{w}}}{\partial y}(m,n)p_y \right\} \frac{\partial I_{\mathrm{w}}}{\partial x}(m,n) = 0$$

$$\frac{\partial E}{\partial p_y} = 2\sum_{m,n} \left\{ e(m,n) + \frac{\partial I_{\mathrm{w}}}{\partial x}(m,n)p_x + \frac{\partial I_{\mathrm{w}}}{\partial y}(m,n)p_y \right\} \frac{\partial I_{\mathrm{w}}}{\partial y}(m,n) = 0$$

By solving the following equation, motion vector (p_x, p_y) is obtained

$$\begin{pmatrix} \sum(\frac{\partial I_{\mathrm{w}}}{\partial x})^2 & \sum \frac{\partial I_{\mathrm{w}}}{\partial x}\frac{\partial I_{\mathrm{w}}}{\partial y} \\ \sum \frac{\partial I_{\mathrm{w}}}{\partial x}\frac{\partial I_{\mathrm{w}}}{\partial y} & \sum(\frac{\partial I_{\mathrm{w}}}{\partial y})^2 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \end{pmatrix} = - \begin{pmatrix} \sum \frac{\partial I_{\mathrm{w}}}{\partial x} e \\ \sum \frac{\partial I_{\mathrm{w}}}{\partial y} e \end{pmatrix}$$

Terminologies

$$\begin{pmatrix} \sum(\frac{\partial I_\mathrm{w}}{\partial x})^2 & \sum \frac{\partial I_\mathrm{w}}{\partial x}\frac{\partial I_\mathrm{w}}{\partial y} \\ \sum \frac{\partial I_\mathrm{w}}{\partial x}\frac{\partial I_\mathrm{w}}{\partial y} & \sum(\frac{\partial I_\mathrm{w}}{\partial y})^2 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \end{pmatrix} = -\begin{pmatrix} \sum \frac{\partial I_\mathrm{w}}{\partial x} e \\ \sum \frac{\partial I_\mathrm{w}}{\partial y} e \end{pmatrix}$$

$$J^T J \begin{pmatrix} p_x \\ p_y \end{pmatrix} = -J^T \boldsymbol{e}$$

1st pixel
2nd pixel
…

$$J = \begin{pmatrix} \frac{\partial I_{w,1}}{\partial x} & \frac{\partial I_{w,1}}{\partial y} \\ \frac{\partial I_{w,2}}{\partial x} & \frac{\partial I_{w,2}}{\partial y} \\ \vdots & \vdots \end{pmatrix} \quad \boldsymbol{e} = \begin{pmatrix} e_1 \\ e_2 \\ \vdots \end{pmatrix} \quad J^T J$$

Jacobian matrix — error vector — (Gauss-Newton approximation of) Hessian matrix

- (p_x, p_y) is only approximately obtained because of the 1st order Taylor approximation. We usually need to iteratively run the above process by updating

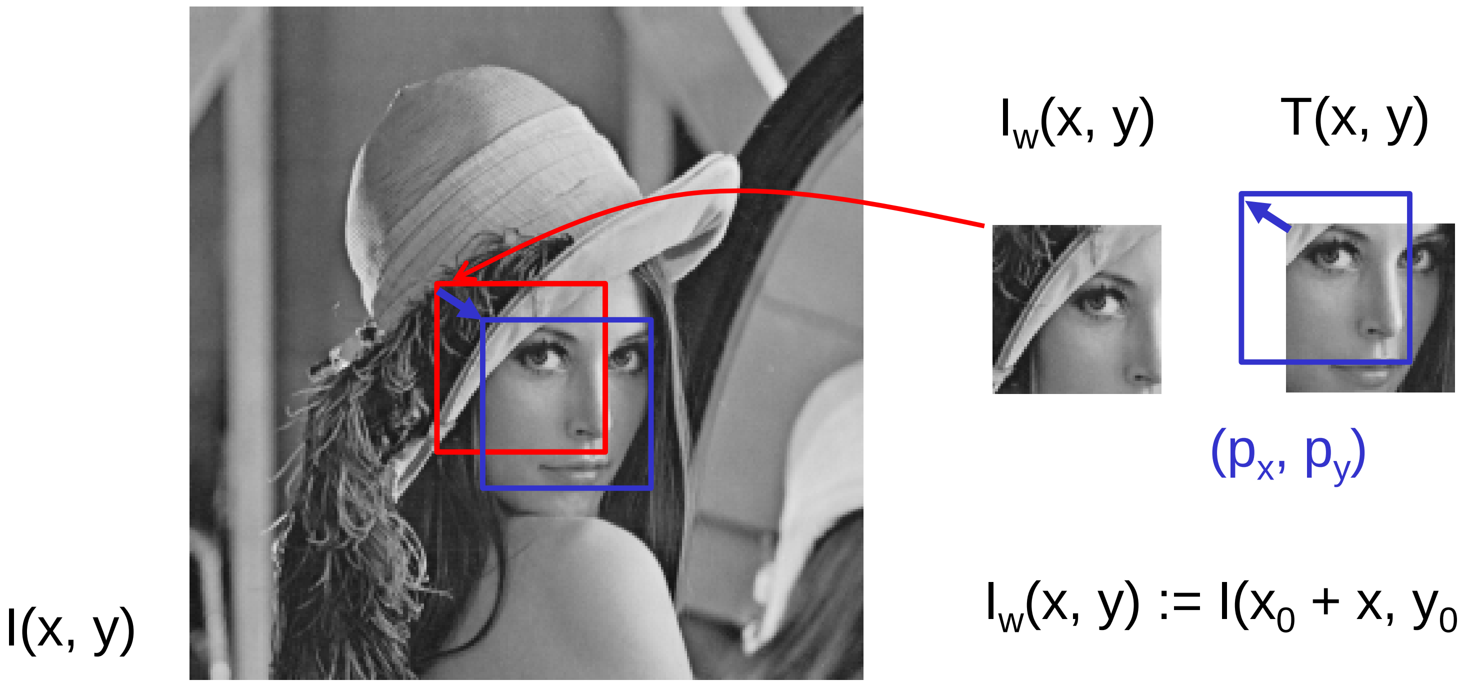
  $x_0 := x_0 + p_x$

  $y_0 := y_0 + p_y$

  and obtaining $I_w(x, y) := I(x_0 + x, y_0 + y)$ with new (x_0, y_0)

- Because I_w changes, the derivatives and their products must be recomputed for each iteration

Inverse Algorithm



$$I_w(x, y) := I(x_0 + x, y_0 + y)$$

The recomputation of derivatives and their products can be avoided by exchanging the role of T and I_w

$$\tilde{E}(p_x, p_y) = \sum_{m,n} \{T(p_x + m, p_y + n) - I_W(m, n)\}^2$$

$$\tilde{E}(p_x, p_y) \simeq \sum_{m,n} \left\{ T(m,n) + \frac{\partial T}{\partial x}(m,n)p_x + \frac{\partial T}{\partial y}(m,n)p_y - I_{\mathrm{w}}(m,n) \right\}^2$$

$$= \sum_{m,n} \left\{ -e(m,n) + \frac{\partial T}{\partial x}(m,n)p_x + \frac{\partial T}{\partial y}(m,n)p_y \right\}^2 \rightarrow \min_{p_x, p_y}$$

$$\begin{pmatrix} \sum(\frac{\partial T}{\partial x})^2 & \sum \frac{\partial T}{\partial x}\frac{\partial T}{\partial y} \\ \sum \frac{\partial T}{\partial x}\frac{\partial T}{\partial y} & \sum(\frac{\partial T}{\partial y})^2 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \end{pmatrix} = \begin{pmatrix} \sum \frac{\partial T}{\partial x} e \\ \sum \frac{\partial T}{\partial y} e \end{pmatrix}$$

After solving (p_x, p_y), Redefine $I_w()$ by updating

$x_0 := x_0 - p_x$

$y_0 := y_0 - p_y$

and resample $I_w(x, y)$ with the new (x_0, y_0)

Application to Optical Flow Computation

Distribution of the motion vectors over the image is called optical flow

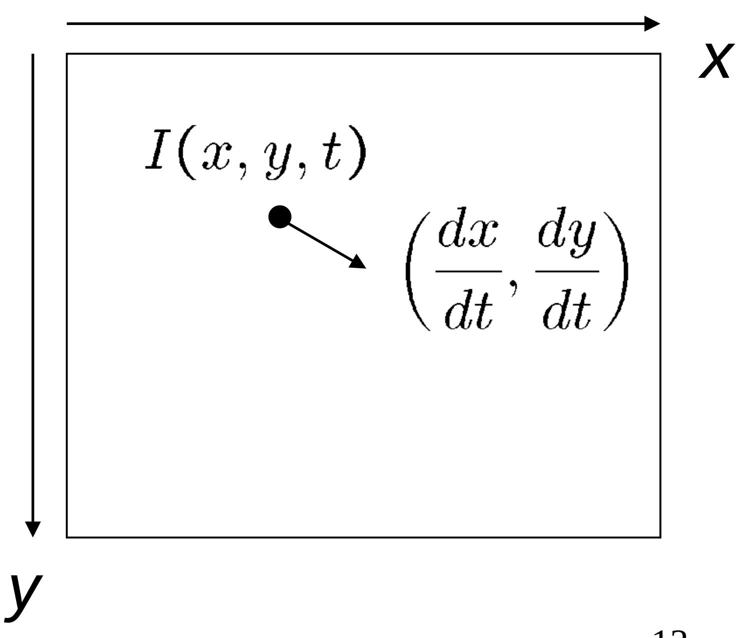
- Sometimes the terms "motion vector" and "optical flow" are used interchangeably (depending on the context)

Optical Flow Constraint

Assuming that the intensity of the tracked point is constant and ignoring 2nd order or higher terms,

$$I(x,y,t) = I(x+dx, y+dy, t+dt)$$
$$= I(x,y,t) + \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt$$

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



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

[Horn and Schunck 1981]

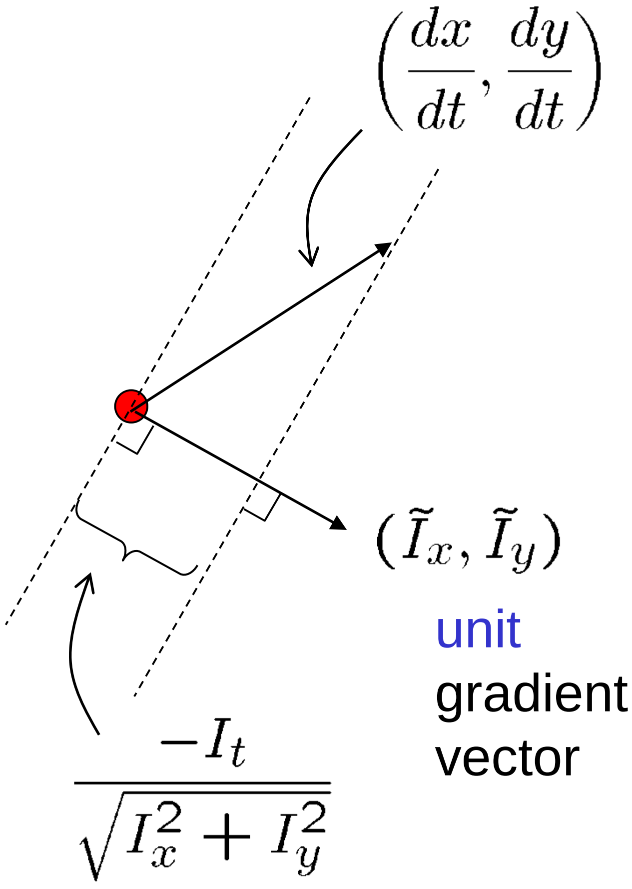
Interpretation of the Constraint

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)



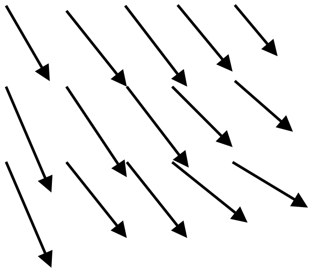
Only the component in the direction of the gradient vector is determined (aperture problem)

Additional Assumptions

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

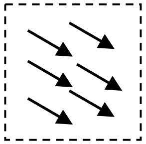
ex1) Optical flow changes smoothly in space

- [Horn and Schunck 1981]

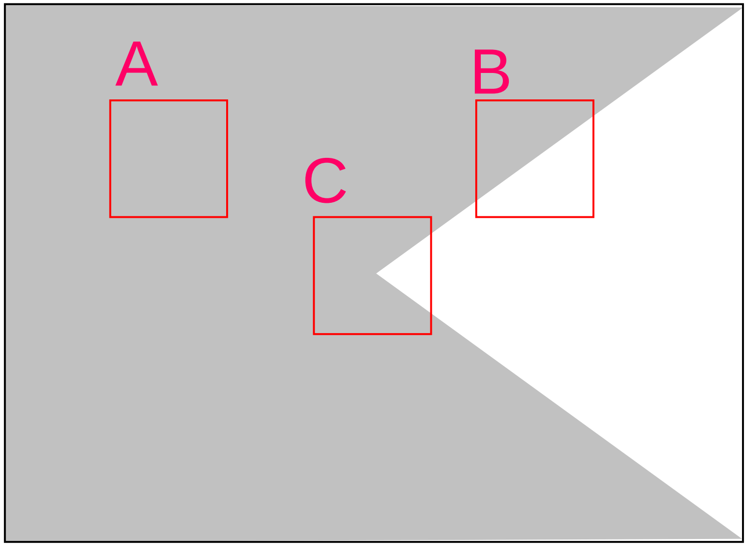


ex2) Optical flow is constant within a small neighborhood of a point

- Hence the problem comes down to tracking of small blocks



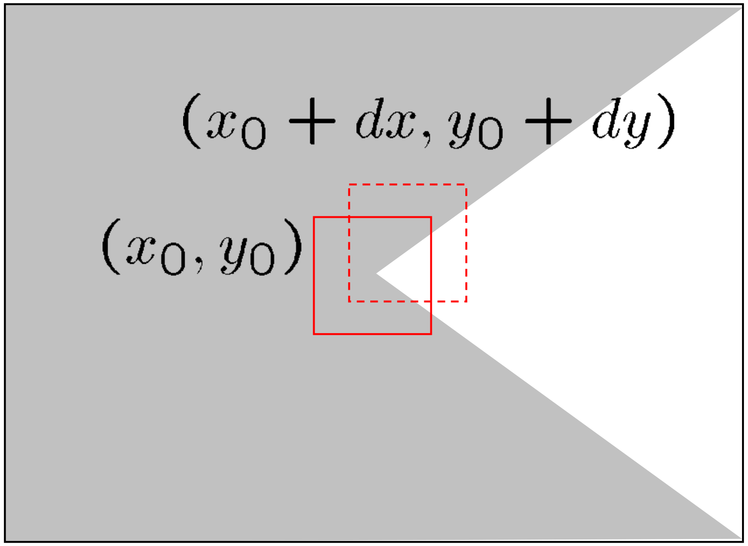
What is Good Point to Track



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 $(x_0 + dx, y_0 + dy)$

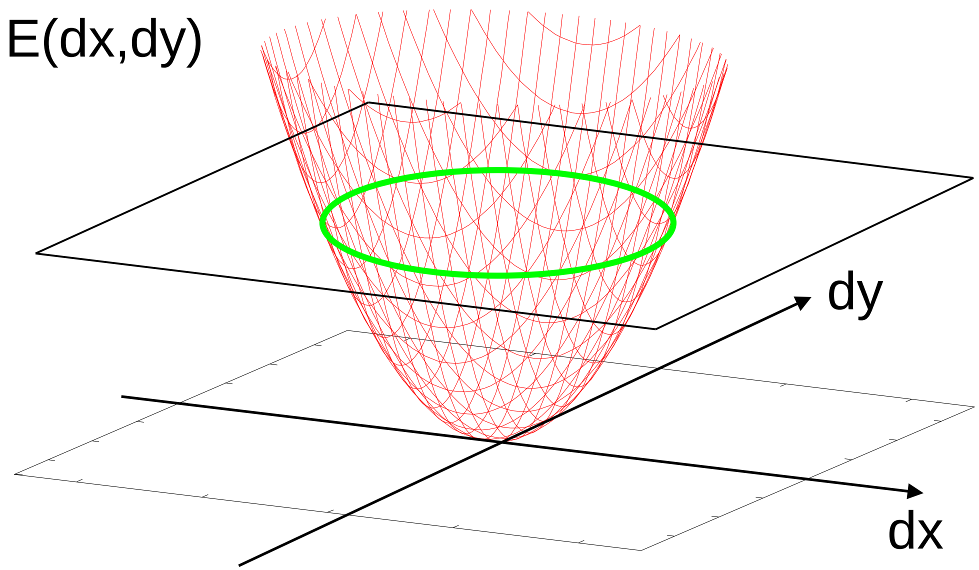
These two blocks should not resemble each other for any choice of (dx, dy)

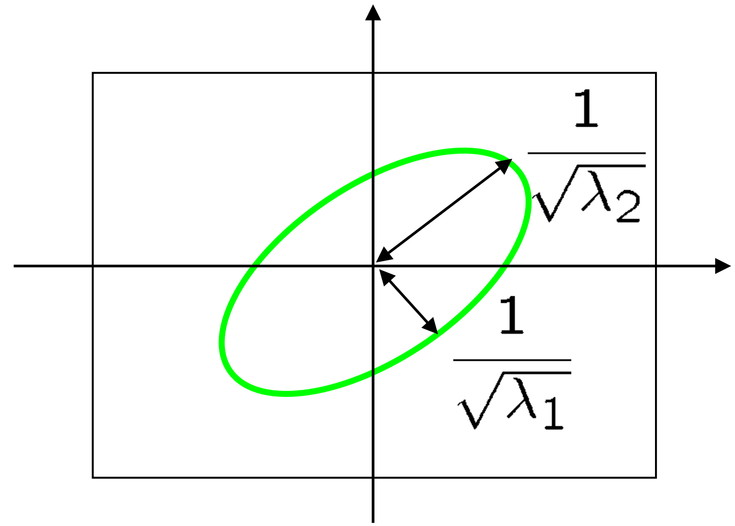
Let's measure how they do not resemble by SSD

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

With 1st order Taylor expansion,

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





E(dx,dy) = 1

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

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 H should be large enough and close to each other.

Compatible with numerical stability in solving $H \begin{pmatrix} dx \\ dy \end{pmatrix} = -J^t e$

(Just in case you forget linear algebra)

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

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

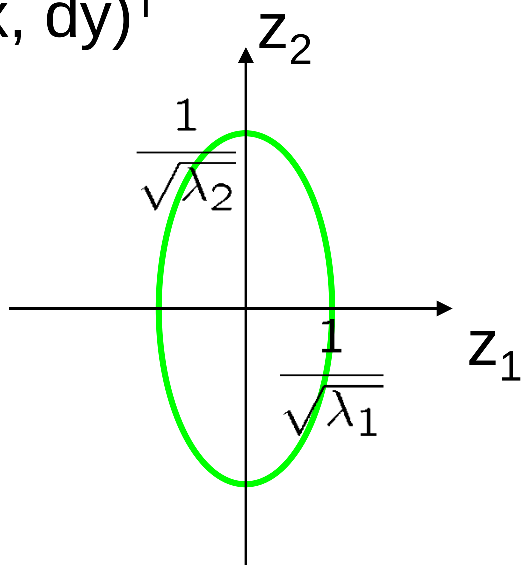
$$(dx, dy)P \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} P^T \begin{pmatrix} dx \\ dy \end{pmatrix} = 1$$

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]

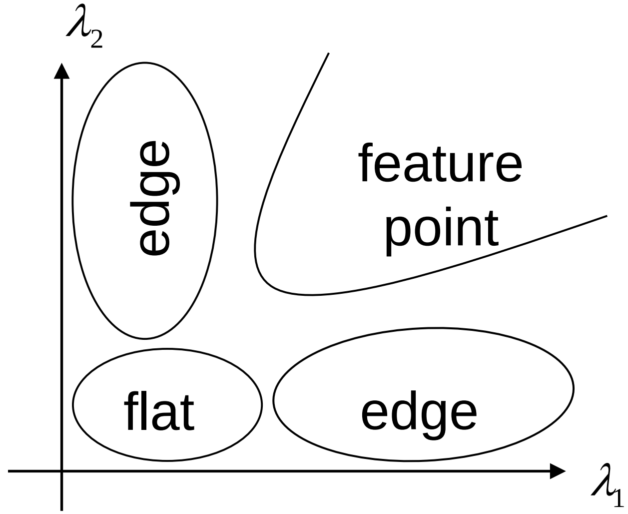
$$\det H - k(\mathrm{tr}\, H)^2$$

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

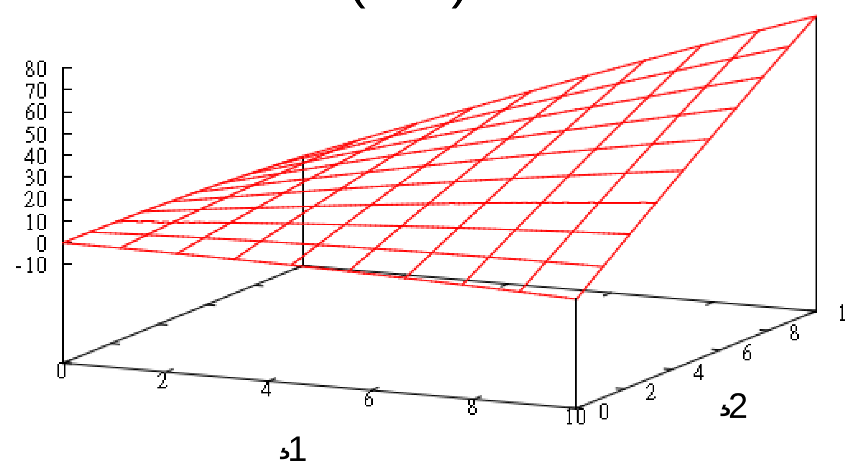
Good Features to Track

[Tomasi and Kanade 1991]

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



det H – k(tr H)^2



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

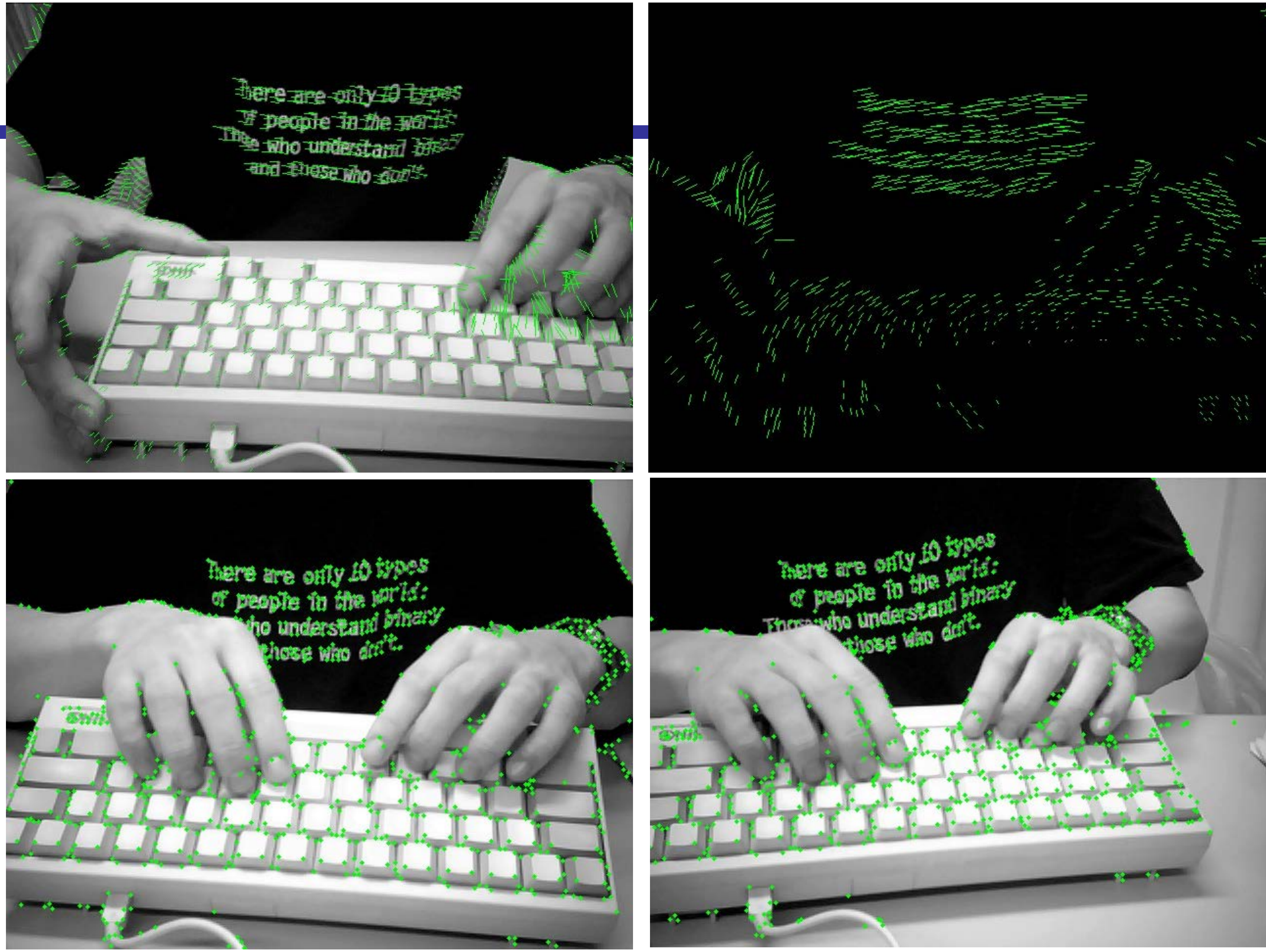
Other Feature Point Detectors

SIFT detector [Lowe 2004]

- Build a Gaussian scale space and apply (an approximate) Laplacian operator in each scale
- Detect extrema of the results (i.e. strongest responses among their neighbor in space as well as in scale)
- Eliminate edge responses
- (Often followed by encoding of edge orientation histogram in the neighborhood into a fixed-size vector, called a feature point descriptor, which can be compared with each other by Euclidean distance)

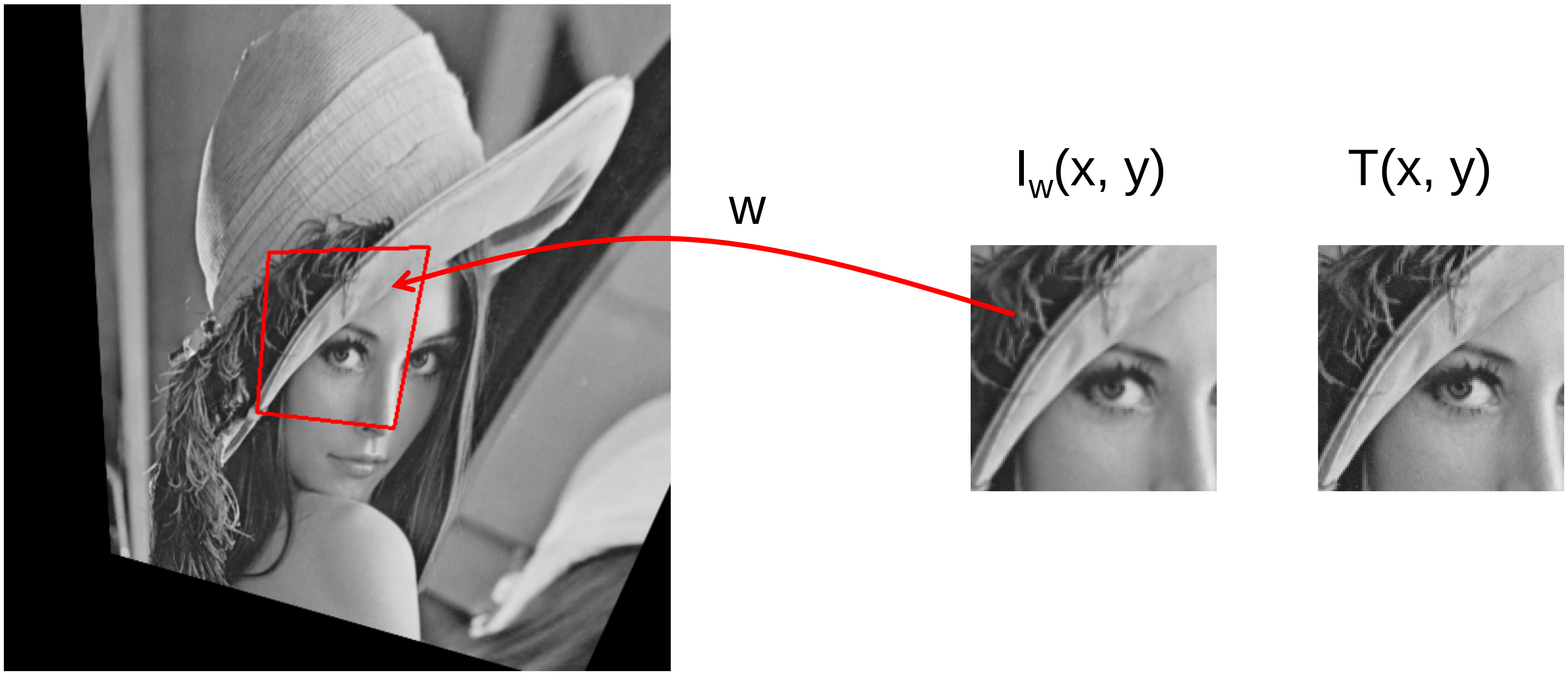
FAST detector [Rosten et al. 2010]

- Heuristics based on pixel values along a surrounding circle
- Optimized for speed and quality by machine learning approach



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

Generalization to Different Warps



What kind of modifications are needed?

$$\begin{pmatrix} \sum (\frac{\partial T}{\partial x})^2 & \sum \frac{\partial T}{\partial x}\frac{\partial T}{\partial y} \\ \sum \frac{\partial T}{\partial x}\frac{\partial T}{\partial y} & \sum (\frac{\partial T}{\partial y})^2 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \end{pmatrix} = \begin{pmatrix} \sum \frac{\partial T}{\partial x} e \\ \sum \frac{\partial T}{\partial y} e \end{pmatrix}$$

Warps and Their Parametrizations

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

p = [t_x, t_y]

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

p = [t_x, t_y, θ]

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1+p_1 & p_2 & p_3 \\ p_4 & 1+p_5 & p_6 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} \propto \begin{pmatrix} 1+p_1 & p_2 & p_3 \\ p_4 & 1+p_5 & p_6 \\ p_7 & p_8 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

For convenience, we usually define the parameters so that **p** = **0** corresponds to identity warp

Warp Functions and Their Derivatives

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \boldsymbol{w_p}(\begin{pmatrix} x \\ y \end{pmatrix}) = \begin{pmatrix} x\cos\theta - y\sin\theta + t_x \\ x\sin\theta + y\cos\theta + t_y \end{pmatrix}$$

$$\frac{\partial}{\partial \boldsymbol{p}} \boldsymbol{w_p}(\begin{pmatrix} x \\ y \end{pmatrix}) \Big|_{\boldsymbol{p}=\boldsymbol{0}} = \begin{pmatrix} 1 & 0 & -y \\ 0 & 1 & x \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} \propto \begin{pmatrix} 1+p_1 & p_2 & p_3 \\ p_4 & 1+p_5 & p_6 \\ p_7 & p_8 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \boldsymbol{w_p}(\begin{pmatrix} x \\ y \end{pmatrix}) \Big|_{\boldsymbol{p}=\boldsymbol{0}} = \begin{pmatrix} \dfrac{(1+p_1)x + p_2 y + p_3}{p_7 x + p_8 y + 1} \\ \dfrac{p_4 x + (1+p_5) y + p_6}{p_7 x + p_8 y + 1} \end{pmatrix}$$

$$\frac{\partial}{\partial \boldsymbol{p}} \boldsymbol{w_p}(\begin{pmatrix} x \\ y \end{pmatrix}) = \begin{pmatrix} x & y & 1 & 0 & 0 & 0 & -x^2 & -xy \\ 0 & 0 & 0 & x & y & 1 & -xy & -y^2 \end{pmatrix}$$

Inverse "Additive" Algorithm

Warp the input image with parameter p_0 to obtain I_w

$$\tilde{E}(\boldsymbol{p}) \simeq \sum_{m,n} \left\{ -e(m,n) + \sum_k \frac{\partial T}{\partial p_k}(m,n) \cdot p_k \right\}^2 \rightarrow \min_{\boldsymbol{p}}$$

$$J^T J \boldsymbol{p} = J^T \boldsymbol{e}$$

1^{st} pixel
2^{nd} pixel
…

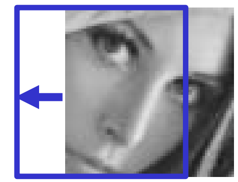
$$J = \begin{pmatrix} \frac{\partial T_1}{\partial p_1} & \frac{\partial T_1}{\partial p_2} & \cdots \\ \frac{\partial T_2}{\partial p_1} & \frac{\partial T_2}{\partial p_2} & \cdots \\ \vdots & \vdots & \end{pmatrix} = \begin{pmatrix} \left(\frac{\partial T_1}{\partial x}, \frac{\partial T_1}{\partial y}\right) \frac{\partial \boldsymbol{w}_{\boldsymbol{p},1}}{\partial p_1} & \left(\frac{\partial T_1}{\partial x}, \frac{\partial T_1}{\partial y}\right) \frac{\partial \boldsymbol{w}_{\boldsymbol{p},1}}{\partial p_2} & \cdots \\ \left(\frac{\partial T_2}{\partial x}, \frac{\partial T_2}{\partial y}\right) \frac{\partial \boldsymbol{w}_{\boldsymbol{p},2}}{\partial p_1} & \left(\frac{\partial T_2}{\partial x}, \frac{\partial T_2}{\partial y}\right) \frac{\partial \boldsymbol{w}_{\boldsymbol{p},2}}{\partial p_2} & \cdots \\ & \vdots & \end{pmatrix}$$

Then, resample I_w by updating the parameters as $p_0 := p_0 - p$ … Is this OK?

This works for 2D translation, but not always for general warps

$I_w(x, y)$

$T(x, y)$



$(p_x, p_y, p_\theta) = (-10, 0, 0)$

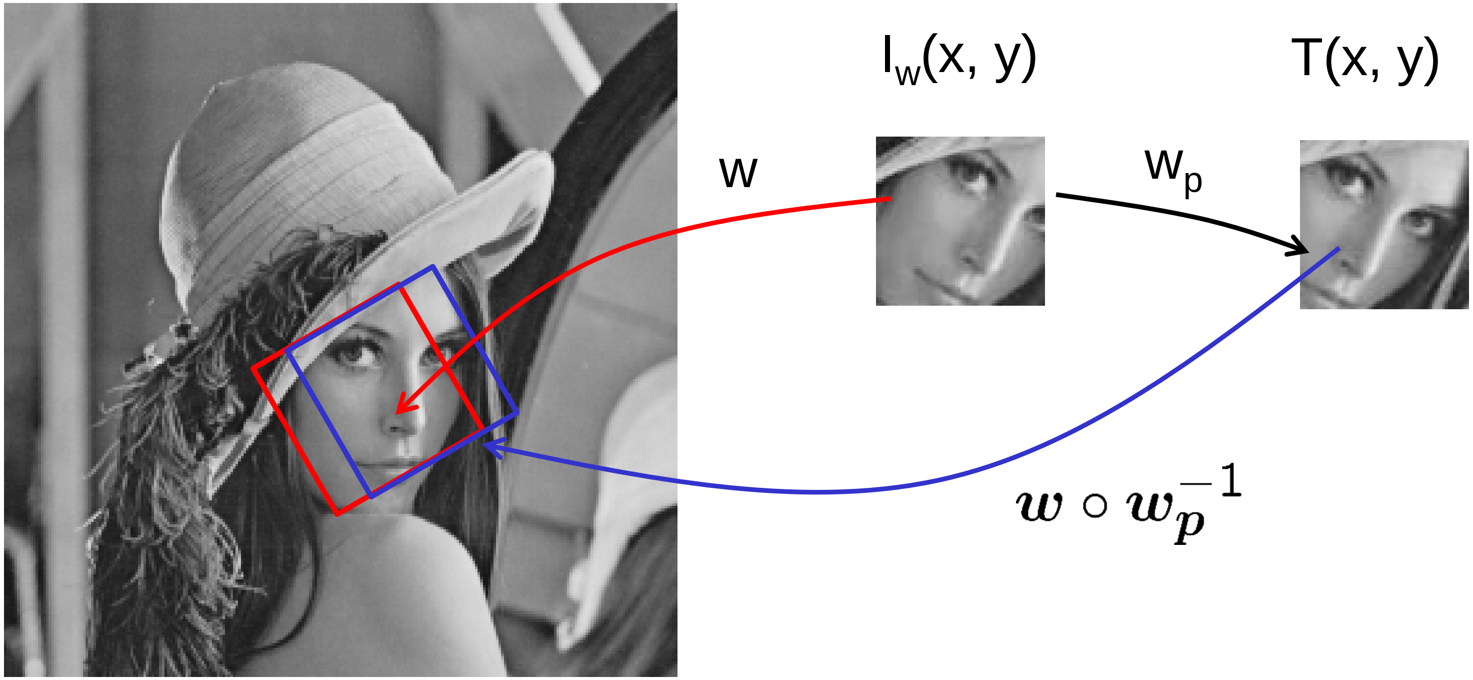
$I(x, y)$

Is it OK if we resample I_w at new position and orientation (+10, 0, 0)?

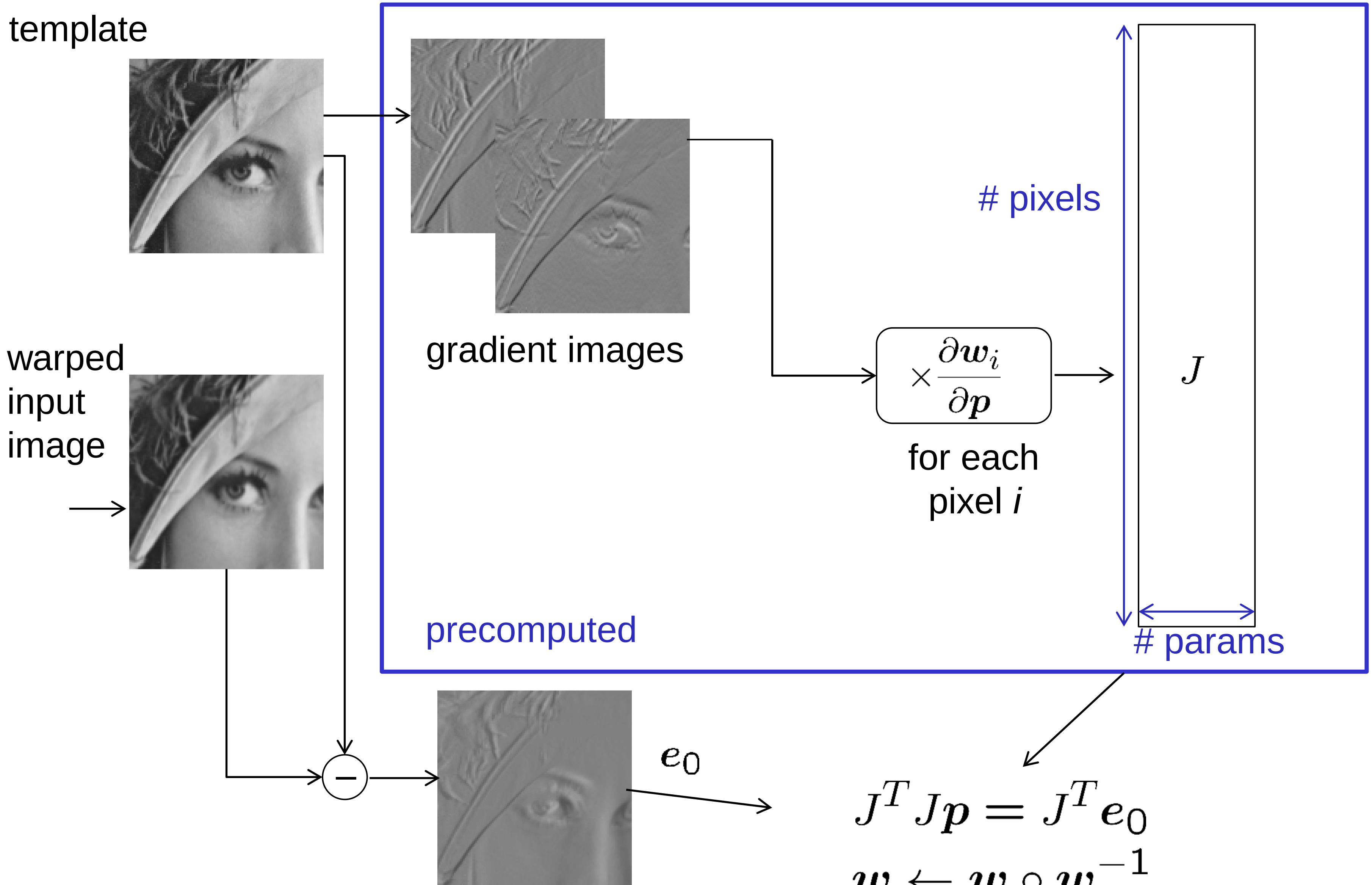
Obviously no.

Inverse Compositional Algorithm

Instead of updating the warp parameters additively, the warp is compositionally updated [Baker & Matthews 2004]: $\boldsymbol{w} \leftarrow \boldsymbol{w} \circ \boldsymbol{w}_{\boldsymbol{p}}^{-1}$



$$\tilde{E}(\boldsymbol{p}) = \sum_{\boldsymbol{x}} \{T(\boldsymbol{w}_{\boldsymbol{p}}(x)) - I_w(x)\}^2$$
$$= \sum_{\boldsymbol{x}} \{T(\boldsymbol{w}_{\boldsymbol{p}}(x)) - I(\boldsymbol{w}(x))\}^2 \rightarrow \min_{\boldsymbol{p}}$$



$$\times \frac{\partial \boldsymbol{w}_i}{\partial \boldsymbol{p}}$$

$$J^T J \boldsymbol{p} = J^T \boldsymbol{e}_0$$

$$\boldsymbol{w} \leftarrow \boldsymbol{w} \circ \boldsymbol{w}_{\boldsymbol{p}}^{-1}$$

Other Choices of Optimization Methods

Other optimization methods

- Levenberg-Marquardt method

$$(J^T J + \lambda I)\boldsymbol{p} = J^T \boldsymbol{e}_0$$

I : identity matrix
λ : scalar coefficient
(small λ : Gauss-Newton, large λ : steepest descent)

- Efficient Second-order Minimization method [Banhimane and Malis 2007]

$$(J^T J)\boldsymbol{p} = J^T \boldsymbol{e}_0, \; J = (J_1 + J_2)/2$$

J_1: derivative of template image w.r.t. param.
J_2: derivative of current warped image w.r.t. param.
(Possible when parametrized with special care)

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.
- S. Baker and I. Matthews: Lucas-Kanade 20 Years On: A Unifying Framework, International Journal of Computer Vision, vol. 56, no. 3, 2004.
- S. Benhimane and E. Malis: Homography-based 2D Visual Tracking and Servoing, International Journal of Robotics Research, vol. 26, no. 7, pp.661-676, 2007.
- D. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, International Journal of Computer Vision, vol. 60, no. 2, pp. 91-110, 2004.
- E. Rosten, R. Porter and T. Drummond: Faster and Better: A Machine Learning Approach to Corner Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 1, pp. 105-119, 2010.
- 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, Carnegie Mellon University, 1991.