图像对齐算法

Image Alignment Algorithm

主　讲　人：龚益群

公　众　号：3D视觉工坊

Overview

- Forward Additive Image Alignment（前向加法）
- Forward Compositional Image Alignment（前向组合）
- Inverse Compositional Image Alignment（逆向组合）
- Lucas-Kanade Optical Flow
- Kanade-Lucas-Tomasi(KLT) Tracker
- Pyramidal Iterative KLT
- Others Improvements

Image Alignment

Goal $\min \sum_{\mathbf{x}} [I(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{x})]^2$

采用 **S**um of **S**quared **D**ifferences (SSD)

Template

如何在右图中找到模板?

Warped Image

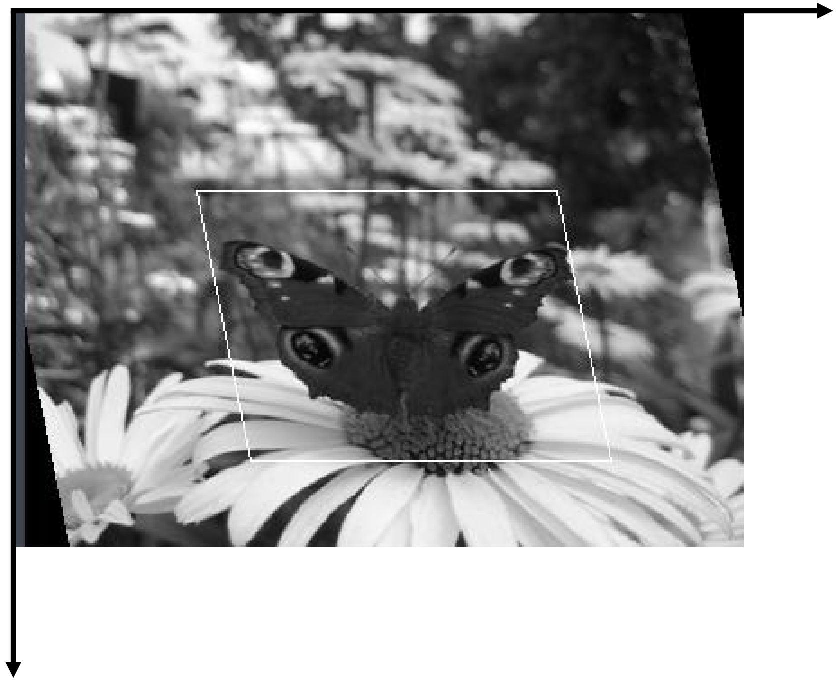


Image Alignment

$$\min_{\boldsymbol{p}} \sum_{\boldsymbol{x}} \left[I(\mathbf{W}(\boldsymbol{x};\boldsymbol{p})) - T(\boldsymbol{x}) \right]^2$$

- 2D image transformation

$$\mathbf{W}(\boldsymbol{x};\boldsymbol{p})$$

- 2D image coordinate

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix}$$

- Parameters of the transformation

$$\boldsymbol{p} = \{p_1, \ldots, p_N\}$$

- Warped image Pixel value at a coordinate

$$I(\boldsymbol{x}') = I(\mathbf{W}(\boldsymbol{x};\boldsymbol{p}))$$

Affine **Translation**

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$\mathbf{W}(\boldsymbol{x};\boldsymbol{p}) = \begin{bmatrix} p_1 x + p_2 y + p_3 \\ p_4 x + p_5 y + p_6 \end{bmatrix}$$

$$= \begin{bmatrix} p_1 & p_2 & p_3 \\ p_4 & p_5 & p_6 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\mathbf{p} = (p_1, p_2, p_3, p_4, p_5, p_6)^T$$

Forward Additive

$$\sum_{\boldsymbol{x}} \left[\underline{I(\mathbf{W}(\boldsymbol{x};\boldsymbol{p}+\Delta\boldsymbol{p}))} - T(\boldsymbol{x}) \right]^2$$

- 非线性问题，做一阶泰勒展开

$$\begin{aligned} I(\mathbf{W}(\boldsymbol{x};\boldsymbol{p}+\Delta\boldsymbol{p})) &\approx I(\mathbf{W}(\boldsymbol{x};\boldsymbol{p})) + \frac{\partial I(\mathbf{W}(\boldsymbol{x};\boldsymbol{p}))}{\partial \boldsymbol{p}} \Delta\boldsymbol{p} \\ &= I(\mathbf{W}(\boldsymbol{x};\boldsymbol{p})) + \frac{\partial I(\mathbf{W}(\boldsymbol{x};\boldsymbol{p}))}{\partial \boldsymbol{x}'} \frac{\partial \mathbf{W}(\boldsymbol{x};\boldsymbol{p})}{\partial \boldsymbol{p}} \Delta\boldsymbol{p} \\ &= I(\mathbf{W}(\boldsymbol{x};\boldsymbol{p})) + \nabla I \frac{\partial \mathbf{W}}{\partial \boldsymbol{p}} \Delta\boldsymbol{p} \end{aligned}$$

- 得到线性近似函数

$$\sum_{\mathbf{x}} \left[I(\mathbf{W}(\mathbf{x};\mathbf{p})) + \nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} - T(\mathbf{x}) \right]^2 .$$

带有未知数的线性方程

Forward Additive

$\mathbf{W}(\boldsymbol{x};\boldsymbol{p})$的 Jacobian

- 将变换按照坐标展开

$$\mathbf{W}(\mathbf{x};\mathbf{p}) = (W_x(\mathbf{x};\mathbf{p}),\, W_y(\mathbf{x};\mathbf{p}))^{\mathrm{T}}$$

- 每个坐标对于每个参数求导

$$\frac{\partial \mathbf{W}}{\partial \mathbf{p}} = \begin{pmatrix} \frac{\partial W_x}{\partial p_1} & \frac{\partial W_x}{\partial p_2} & \cdots & \frac{\partial W_x}{\partial p_n} \\ \frac{\partial W_y}{\partial p_1} & \frac{\partial W_y}{\partial p_2} & \cdots & \frac{\partial W_y}{\partial p_n} \end{pmatrix}.$$

$$\mathbf{W}(\boldsymbol{x};\boldsymbol{p}) = \begin{bmatrix} p_1 x + p_3 y + p_5 \\ p_2 x + p_4 y + p_6 \end{bmatrix}$$

$$\frac{\partial W_x}{\partial p_1} = x \qquad \frac{\partial W_x}{\partial p_2} = 0 \qquad \cdots$$

$$\frac{\partial W_y}{\partial p_1} = 0 \qquad \cdots$$

$$\frac{\partial \mathbf{W}}{\partial \boldsymbol{p}} = \begin{bmatrix} x & 0 & y & 0 & 1 & 0 \\ 0 & x & 0 & y & 0 & 1 \end{bmatrix}$$

Forward Additive

$$\hat{x} = \arg\min_{x} ||Ax - b||^2 \qquad x = (A^\top A)^{-1} A^\top b$$

- 各个变量的维度

$$\sum_{\mathbf{x}} \left[I(\mathbf{W}(\mathbf{x};\mathbf{p})) + \underbrace{\nabla I}_{1\times 2} \underbrace{\frac{\partial \mathbf{W}}{\partial \mathbf{p}}}_{2\times 6} \underbrace{\Delta \mathbf{p}}_{6\times 1} - T(\mathbf{x}) \right]^2$$

- 求解方法

$$\min_{\Delta \boldsymbol{p}} \sum_{\boldsymbol{x}} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \boldsymbol{p}} \Delta \boldsymbol{p} - \{T(\boldsymbol{x}) - I(\mathbf{W}(\boldsymbol{x};\boldsymbol{p}))\} \right]^2$$

$$\mathbf{Ax} \quad - \quad \mathbf{b}$$

- 对变量求导，然后令其等于0

$$\Rightarrow 2\sum_{\mathbf{x}} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^T \left[I(\mathbf{W}(\mathbf{x};\mathbf{p})) + \nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta \mathbf{p} - T(\mathbf{x}) \right]$$

$$\Delta \mathbf{p} = H^{-1} \sum_{\mathbf{x}} \left[\boldsymbol{\nabla} I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^{\mathrm{T}} [T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p}))]$$

其中 $H = \sum_{\mathbf{x}} \left[\boldsymbol{\nabla} I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^{\mathrm{T}} \left[\boldsymbol{\nabla} I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right].$

- 使用加法更新，迭代求解

$$\mathbf{p} \leftarrow \mathbf{p} + \Delta \mathbf{p} \quad \textbf{Until} \quad \|\Delta \mathbf{p}\| \le \xi$$

Forward Additive

Iterate:

1. Warp I with $\mathbf{W}(\mathbf{x};\mathbf{p})$ to compute $I(\mathbf{W}(\mathbf{x};\mathbf{p}))$
2. Compute the error image $T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p}))$
3. Warp the gradient of image I to compute ∇I
4. Evaluate the Jacobian $\frac{\partial \mathrm{W}}{\partial \mathrm{p}}$ at $\mathbf{W}(\mathbf{x};\mathbf{p})$
5. Compute the steepest descent images $\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}}$
6. Compute the Hessian matrix $H = \sum_{\mathbf{x}} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^T \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]$
7. Compute $\sum_{\mathbf{x}} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^T [T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p}))]$
8. Compute $\Delta\mathbf{p} = H^{-1} \sum_{\mathbf{x}} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^T [T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p}))]$
9. Update the parameters $\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}$

until $\|\Delta \mathrm{p}\| \leq \xi$

Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8	Step 9	Total
$O(nN)$	$O(N)$	$O(nN)$	$O(nN)$	$O(nN)$	$O(n^2N)$	$O(nN)$	$O(n^3)$	$O(n)$	$O(n^2N + n^3)$

Forward Compositional

$$\min_{\boldsymbol{p}} \sum_{\boldsymbol{x}} [I(\mathbf{W}(\boldsymbol{x};\boldsymbol{p})) - T(\boldsymbol{x})]^2$$

是否可以以其它的形式更新参数?

Additive Alignment

$$\sum_{\boldsymbol{x}} [I(\mathbf{W}(\boldsymbol{x};\boldsymbol{p}+\Delta\boldsymbol{p})) - T(\boldsymbol{x})]^2$$

对参数进行增量扰动

Compositional Alignment

$$\sum_{\boldsymbol{x}} [I(\mathbf{W}(\ \mathbf{W}(\boldsymbol{x};\Delta\boldsymbol{p});\boldsymbol{p}\ ) - T(\boldsymbol{x})]^2$$

对warp进行增量扰动

Forward Compositional

加法策略

Forward Compositional

组合策略

Forward Compositional

$$\sum_{\boldsymbol{x}} [I(\mathbf{W}(\underline{\ \mathbf{W}(\boldsymbol{x};\Delta\boldsymbol{p});\boldsymbol{p}\ }) - T(\boldsymbol{x})]^2$$

- 进行线性化

$$\sum_{\mathbf{x}} \left[I(\mathbf{W}(\mathbf{W}(\mathbf{x};\mathbf{0});\mathbf{p})) + \nabla I(\mathbf{W}) \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} - T(\mathbf{x}) \right]^2 .$$

其中 $\mathbf{W}(\boldsymbol{x};\mathbf{0})$表示单位变换

- 新的更新方式

$$\mathbf{W}(\mathbf{x};\mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p}) \quad \textbf{Until} \quad \|\Delta\mathbf{p}\| \leq \xi$$

其中 $\mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p}) \equiv \mathbf{W}(\mathbf{W}(\mathbf{x};\Delta\mathbf{p});\mathbf{p})$

$$\sum_{\boldsymbol{x}} \left[I(\mathbf{W}(\boldsymbol{x};\boldsymbol{p})) + \nabla I(\boldsymbol{x}') \boxed{\frac{\partial \mathbf{W}(\boldsymbol{x};\mathbf{0})}{\partial \boldsymbol{p}}} \Delta\boldsymbol{p} - T(\boldsymbol{x}) \right]^2$$

Warp的Jacobian是在0处展开，不
会改变随着迭代，可以提前计算

Forward Compositional

Per-compute:

4. Evaluate the Jacobian $\frac{\partial W}{\partial p}$ at $(\mathbf{x}; 0)$

Iterate:

1. Warp I with $\mathbf{W}(\mathbf{x};\mathbf{p})$ to compute $I(\mathbf{W}(\mathbf{x};\mathbf{p}))$
2. Compute the error image $T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p}))$
3. Compute the gradient $\nabla I(\mathbf{W})$ of image $I(\mathbf{W}(\mathbf{x};\mathbf{p}))$
5. Compute the steepest descent images $\nabla I(\mathrm{W}) \frac{\partial \mathbf{W}}{\partial \mathbf{p}}$
6. Compute the Hessian matrix $H = \sum_{\mathbf{x}} \left[\nabla I(\mathrm{W}) \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^T \left[\nabla I(\mathrm{W}) \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]$
7. Compute $\sum_{\mathbf{x}} \left[\nabla I(\mathrm{W}) \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^T [T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p}))]$
8. Compute $\Delta\mathbf{p} = H^{-1} \sum_{\mathbf{x}} \left[\nabla I(\mathrm{W}) \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^T [T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p}))]$
9. Update the parameters $\mathbf{W}(\mathbf{x};\mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p})$

until $\|\Delta \mathrm{p}\| \leq \xi$

Pre-computation

Step 4	Total
$O(nN)$	$O(nN)$

Per-iteration

Step 1	Step 2	Step 3	Step 5	Step 6	Step 7	Step 8	Step 9	Total
$O(nN)$	$O(N)$	$O(N)$	$O(nN)$	$O(n^2N)$	$O(nN)$	$O(n^3)$	$O(n^2)$	$O(n^2N + n^3)$

Inverse Compositional

3D视觉工坊

$$\min_{\boldsymbol{p}} \sum_{\boldsymbol{x}} \left[I(\mathbf{W}(\boldsymbol{x};\boldsymbol{p})) - T(\boldsymbol{x}) \right]^2$$

是否可以对模板进行warp变换?

Additive Alignment

$$\sum_{\boldsymbol{x}} \left[I(\mathbf{W}(\boldsymbol{x};\boldsymbol{p}+\Delta\boldsymbol{p})) - T(\boldsymbol{x}) \right]^2$$

对参数进行增量扰动

Compositional Alignment

$$\sum_{\boldsymbol{x}} \left[I(\mathbf{W}(\ \mathbf{W}(\boldsymbol{x};\Delta\boldsymbol{p});\boldsymbol{p}\ )) - T(\boldsymbol{x}) \right]^2$$

对warp进行增量扰动

Inverse Compositional Alignment

$$\sum_{\boldsymbol{x}} \left[T(\mathbf{W}(\boldsymbol{x};\Delta\boldsymbol{p})) - I(\mathbf{W}(\boldsymbol{x};\boldsymbol{p})) \right]^2$$

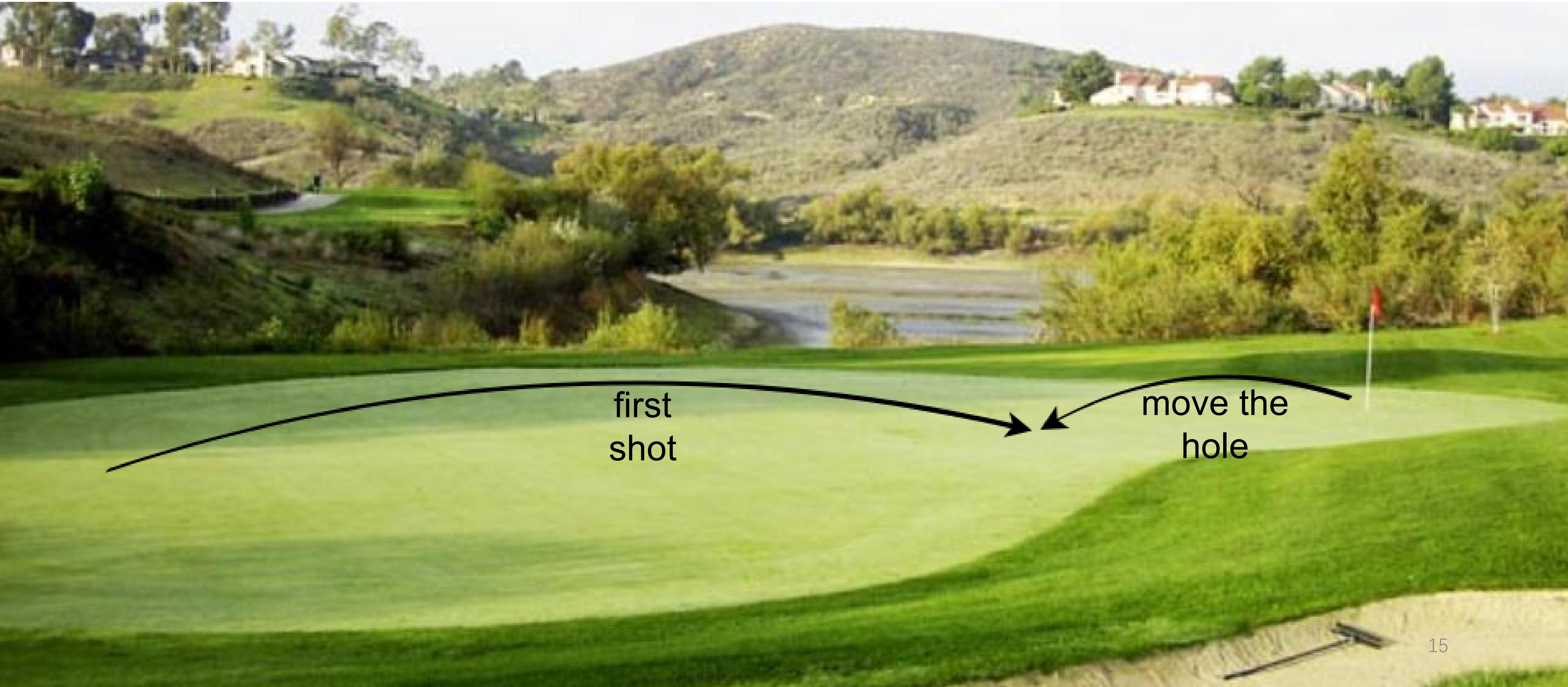
图像T 和图像I 的角色调换

Inverse Compositional

组合策略

Inverse Compositional

逆向组合策略



Inverse Compositional

3D视觉工坊

$$\sum_{\boldsymbol{x}} [T(\mathbf{W}(\boldsymbol{x};\Delta\boldsymbol{p})) - I(\mathbf{W}(\boldsymbol{x};\boldsymbol{p}))]^2$$

- 进行线性化

$$\sum_{\mathbf{x}} \left[T(\mathbf{W}(\mathbf{x};\mathbf{0})) + \nabla T \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta \mathbf{p} - I(\mathbf{W}(\mathbf{x};\mathbf{p})) \right]^2.$$

$$\sum_{\mathbf{x}} \left[T(\mathbf{x}) + \nabla T \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta \mathbf{p} - I(\mathbf{W}(\mathbf{x};\mathbf{p})) \right]^2$$

不会改变

- 求解

$$\boxed{H = \sum_{\mathbf{x}} \left[\nabla T \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^T \left[\nabla T \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]}$$

$$\Delta \mathbf{p} = \boxed{H^{-1} \sum_{\mathbf{x}} \left[\nabla T \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^T} \left[I(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{x}) \right]$$

可以提前计算

- 新的更新方式

$$\mathbf{W}(\mathbf{x};\mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p})^{-1} \quad \textbf{Until} \quad \|\Delta \mathbf{p}\| \le \xi$$

要求可逆

Inverse Compositional

Per-compute:

3. Evaluate the gradient ∇T of image $T(\mathbf{x})$
4. Evaluate the Jacobian $\frac{\partial \mathrm{W}}{\partial \mathrm{p}}$ at $(\mathbf{x}; 0)$
5. Compute the steepest descent images $\nabla T \frac{\partial \mathbf{W}}{\partial \mathbf{p}}$
6. Compute the Hessian matrix $H = \sum_{\mathbf{x}} \left[\nabla T \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^T \left[\nabla T \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]$

Iterate:

1. Warp I with $\mathbf{W}(\mathbf{x};\mathbf{p})$ to compute $I(\mathbf{W}(\mathbf{x};\mathbf{p}))$
2. Compute the error image $I(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{x})$
7. Compute $\sum_{\mathbf{x}} \left[\nabla T \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^T [I(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{x})]$
8. Compute $\Delta \mathbf{p} = H^{-1} \sum_{\mathbf{x}} \left[\nabla T \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^T [I(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{x})]$
9. Update the parameters $\mathbf{W}(\mathbf{x};\mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p})^{-1}$

until $\|\Delta \mathrm{p}\| \le \xi$

Pre-computation					
Step 3	Step 4	Step 5	Step 6	Total	
$O(N)$	$O(n\,N)$	$O(n\,N)$	$O(n^2\,N)$	$O(n^2\,N)$	
Per iteration					
Step 1	Step 2	Step 7	Step 8	Step 9	Total
$O(n\,N)$	$O(N)$	$O(n\,N)$	$O(n^3)$	$O(n^2)$	$O(n\,N + n^3)$

Summary

3D视觉工坊

Algorithm	For example	Complexity	can be applied to
Forwards Additive	Lucas-Kanade (1981)	$O(n^2N + n^3)$	Any warp
Forwards Compositional	Shum-Szeliski (2000)	$O(n^2N + n^3)$	Any semi-group
Inverse Additive	Hager-Belhumeur (1998)	$O(nN + n^3)$	Simple linear 2D +
Inverse Compositional	Baker-Matthews (2001)	$O(nN + kN + k^3)$	Any group

[1] Baker S, Matthews I. Lucas-Kanade 20 Years On: A Unifying Framework[J]. International Journal of Computer Vision, 2004, 56(3):221-255.

Optical Flow

An Iterative Image Registration Technique with an Application to Stereo Vision.

1981

Detection and Tracking of Feature Points.

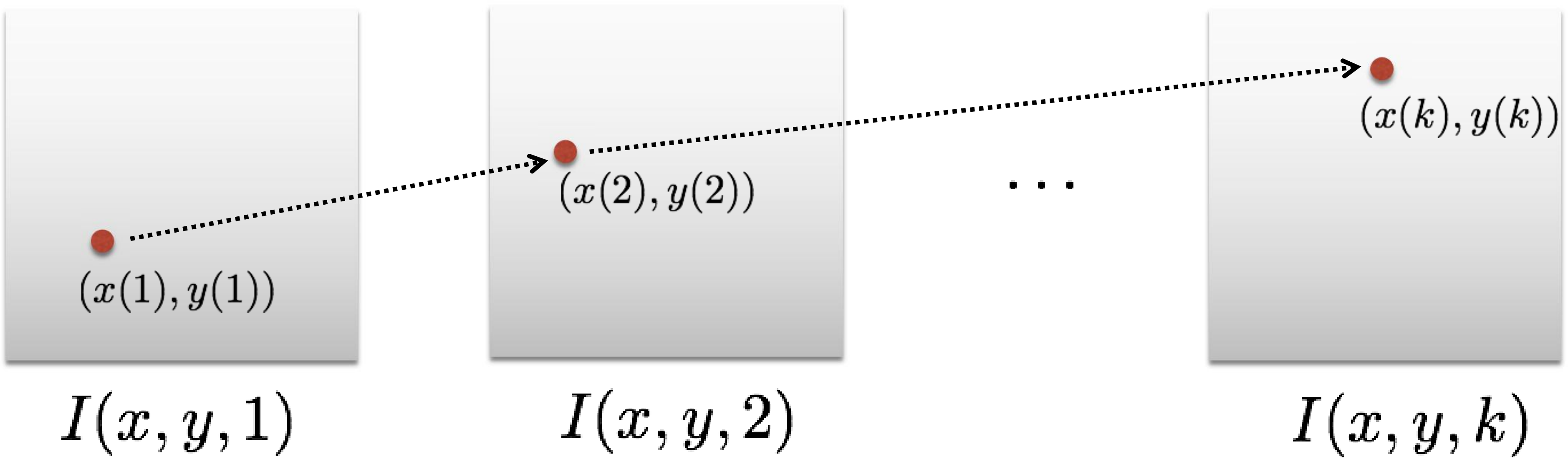
1991

Good Features to Track.

1994

Forward Additive

LK Optical Flow



- 目标：估计出帧间像素的运动
- 假设1：像素光度恒定（像素对比需要）
- 假设2：小运动（线性化需要）
- 假设3：空间一致性，邻近点运动相似

$$I(x(t), y(t), t) = C$$

$$(\delta x, \delta y) = (u\delta t, v\delta t)$$

LK Optical Flow

3D视觉工坊

$$I(x+u\delta t, y+v\delta t, t+\delta t) = I(x,y,t)$$

- 小运动，因此进行线性化

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

$$\frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t = 0$$

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

flow velocities

$$I_x u + I_y v + I_t = 0$$

Image gradients (at a point p)

temporal gradient

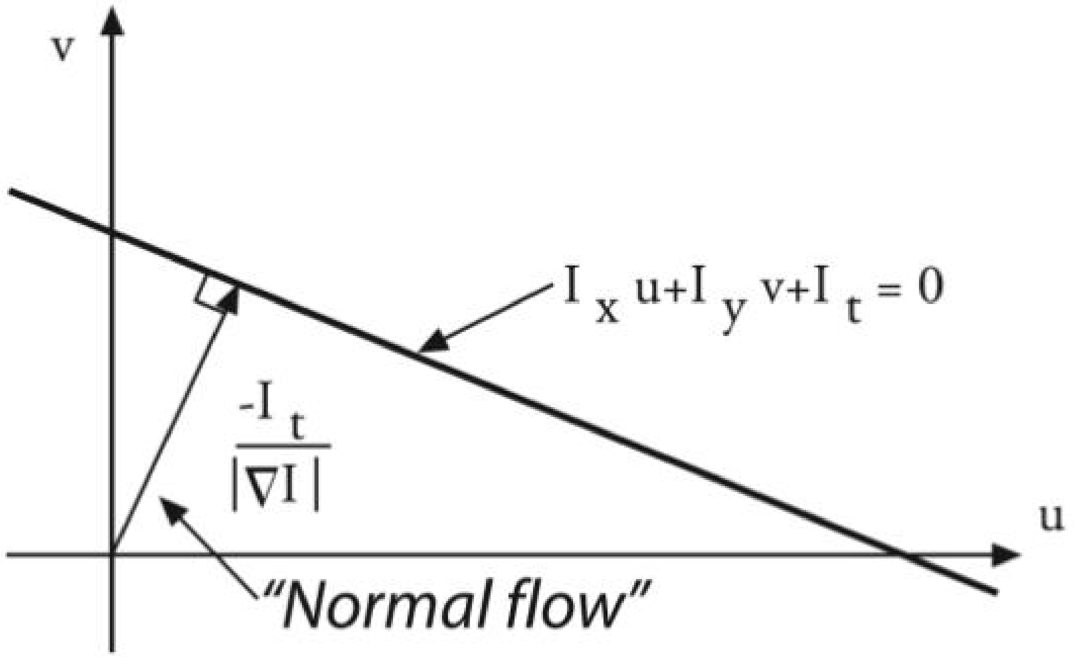
$$\boxed{\nabla I^\top} \boldsymbol{v} + \boxed{I_t} = 0$$

$$\min_{\Delta \boldsymbol{p}} \sum_{\boldsymbol{x}} \left[\boxed{\nabla I \frac{\partial \mathbf{W}}{\partial \boldsymbol{p}}} \Delta \boldsymbol{p} - \boxed{\{T(\boldsymbol{x}) - I(\mathbf{W}(\boldsymbol{x};\boldsymbol{p}))\}} \right]^2$$

LK Optical Flow

3D视觉工坊

$$I_x u + I_y v + I_t = 0$$

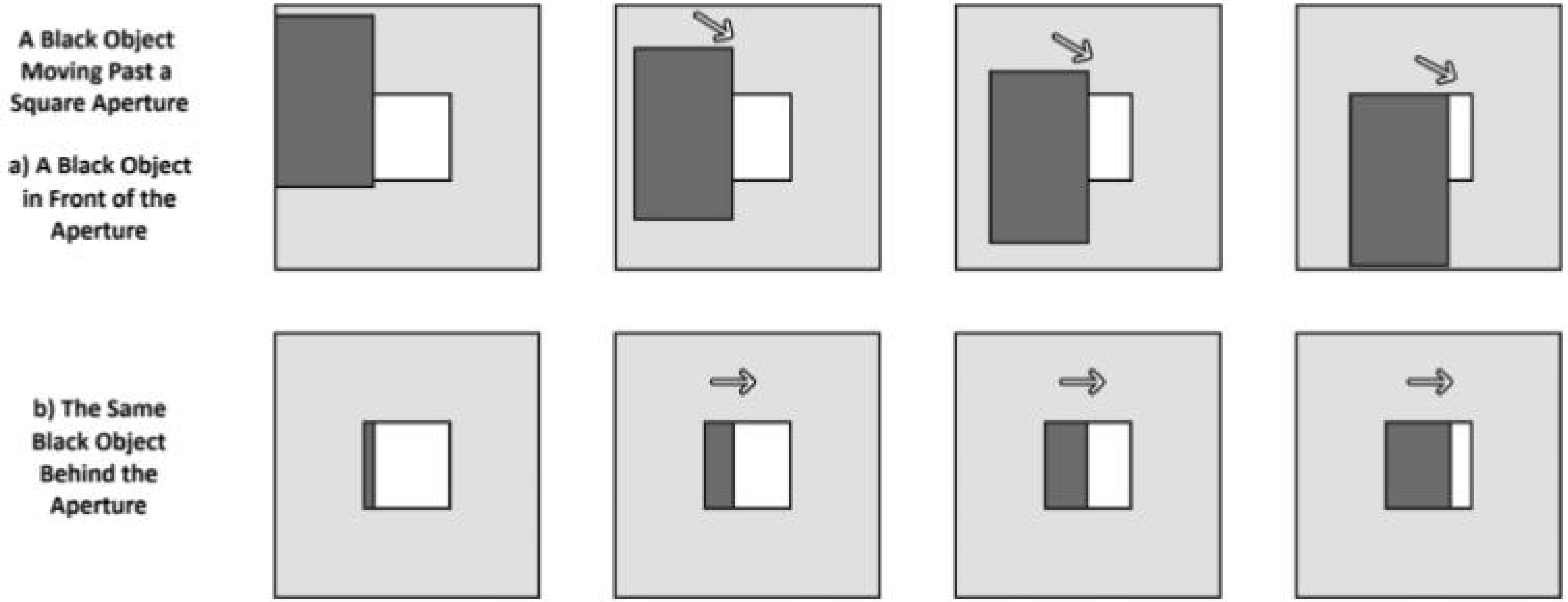


- 根据假设3，可以选一个5×5的patch

$$\begin{aligned} I_x(\boldsymbol{p}_1)u + I_y(\boldsymbol{p}_1)v &= -I_t(\boldsymbol{p}_1) \\ I_x(\boldsymbol{p}_2)u + I_y(\boldsymbol{p}_2)v &= -I_t(\boldsymbol{p}_2) \\ &\vdots \\ I_x(\boldsymbol{p}_{25})u + I_y(\boldsymbol{p}_{25})v &= -I_t(\boldsymbol{p}_{25}) \end{aligned}$$

$$\begin{bmatrix} \sum\limits_{p\in P} I_x I_x & \sum\limits_{p\in P} I_x I_y \\ \sum\limits_{p\in P} I_y I_x & \sum\limits_{p\in P} I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum\limits_{p\in P} I_x I_t \\ \sum\limits_{p\in P} I_y I_t \end{bmatrix}$$

Aperture Problem



KLT

An Iterative Image Registration Technique with an Application to Stereo Vision.

1981

Detection and Tracking of Feature Points.

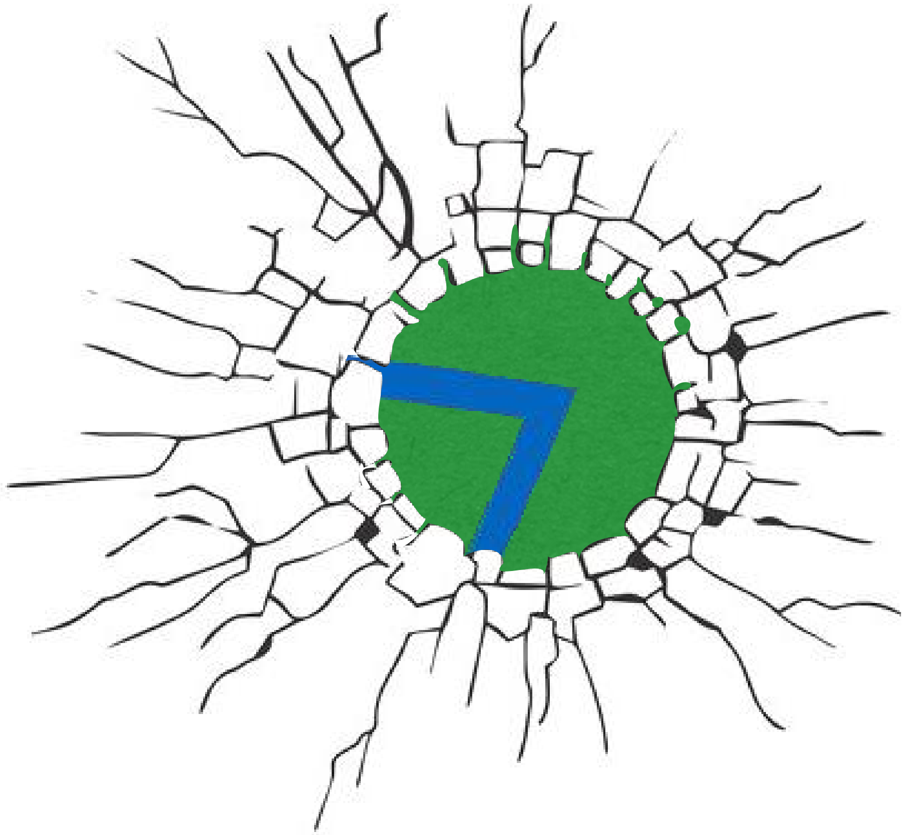
1991

Good Features to Track.

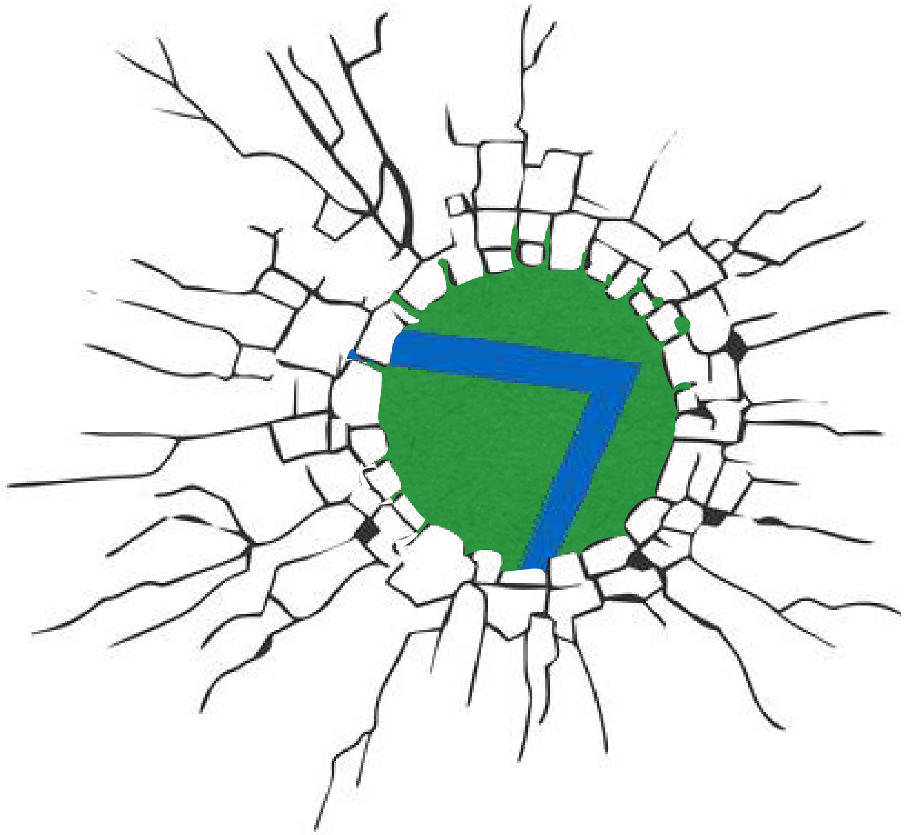
1994

Kanade-Lucas-Tomasi (KLT) Tracker

Aperture Problem



Aperture Problem



$$\Delta \mathbf{p} = H^{-1} \sum_{\mathbf{x}} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^{\mathrm{T}} [T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p}))]$$

怎样选择合适的Patch?

$$H = \sum_{\boldsymbol{x}} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \boldsymbol{p}} \right]^{\top} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \boldsymbol{p}} \right]$$

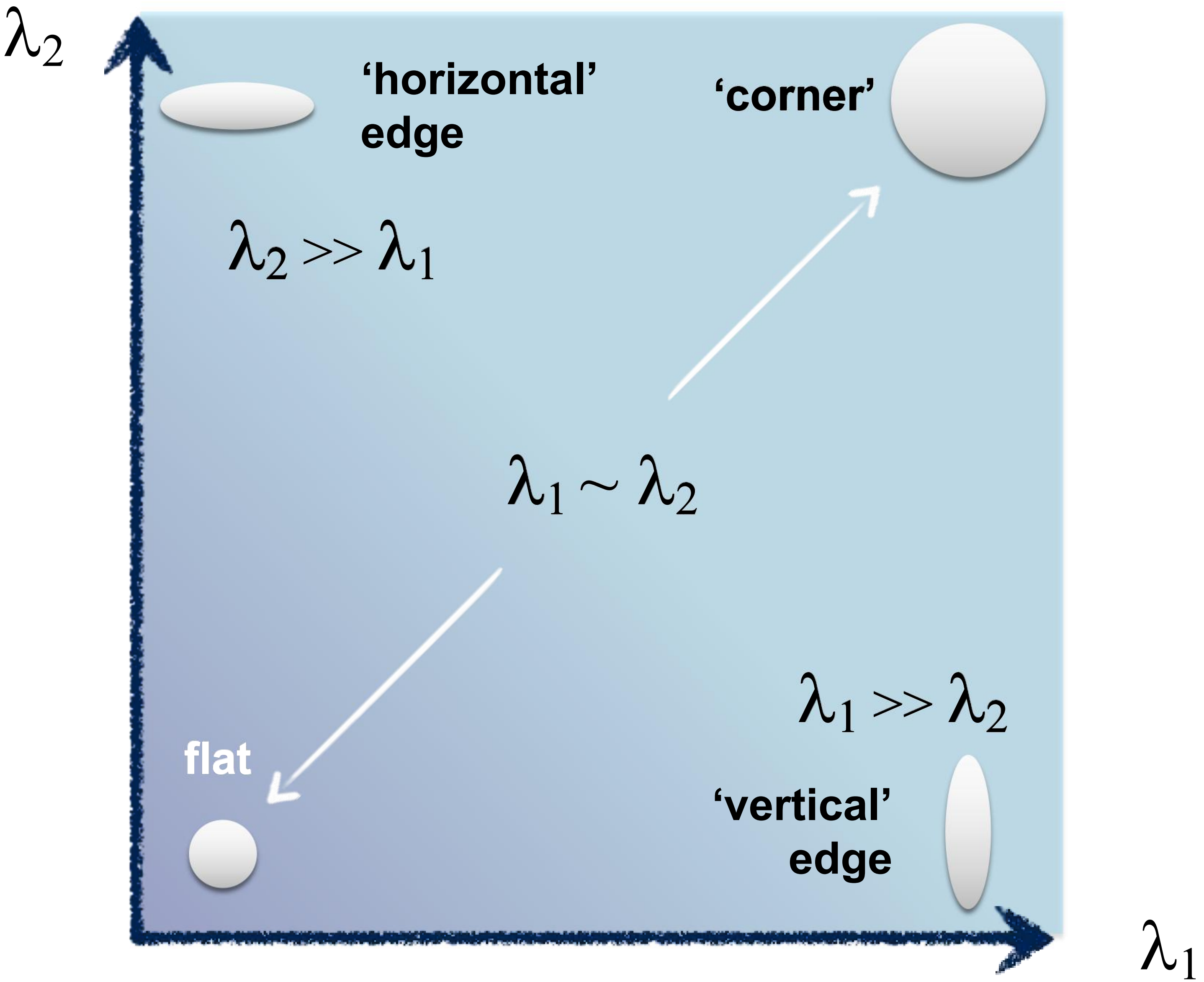
要求可逆

- 做如下假设，即变成光流

$$\mathbf{W}(\boldsymbol{x};\boldsymbol{p}) = \begin{bmatrix} x + p_1 \\ y + p_2 \end{bmatrix}$$

$$\begin{aligned} H &= \sum_{\boldsymbol{x}} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \boldsymbol{p}} \right]^{\top} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \boldsymbol{p}} \right] \\ &= \sum_{\boldsymbol{x}} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I_x \\ I_y \end{bmatrix} \begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \sum_{\boldsymbol{x}} I_x I_x & \sum_{\boldsymbol{x}} I_y I_x \\ \sum_{\boldsymbol{x}} I_x I_y & \sum_{\boldsymbol{x}} I_y I_y \end{bmatrix} \end{aligned}$$

用G表示


λ_2
'horizontal' edge
'corner'
$\lambda_2 >> \lambda_1$
$\lambda_1 \sim \lambda_2$
$\lambda_1 >> \lambda_2$
flat
'vertical' edge
λ_1

像素选取策略

1. 计算每个像素的G矩阵的最小特征值λ_m
2. 取整个图像中λ_m的最大值$\lambda_{\max}$
3. 保留λ_m 大于$\lambda_{\max}$的百分比（10% or 5%）值的像素
4. 对于留下的像素，取局部最大值，即在3×3的邻域内，这个像素的λ_m大于周围的像素
5. 保留和任意像素之间的距离都大于阈值（10 or 5）的像素

Pyramidal Iterative KLT

Linearisation Problem

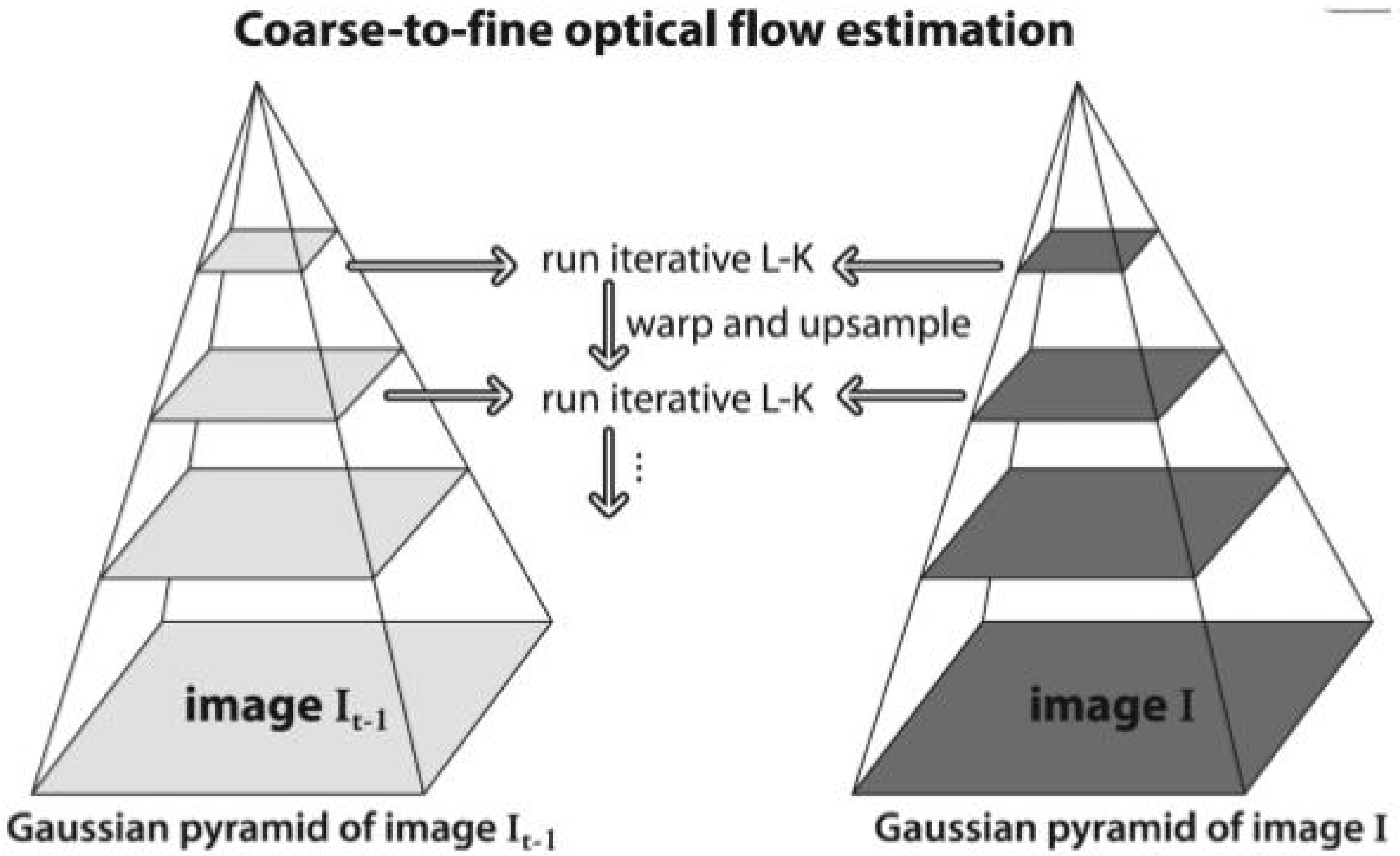
- 当面对大运动，线性化会带来误差，需要大的积分窗口。

平滑掉细节　　违反运动一致性　　精度低　　计算量大

- 小窗口可能会有孔径问题，不够鲁棒，收敛慢。

怎样处理大运动的场景呢?

Pyramidal Iterative KLT



Pyramidal Iterative KLT

Goal: Let $\mathbf{u}$ be a point on image I. Find its corresponding location $\mathbf{v}$ on image J

Build pyramid representations of I and J: $\{I^L\}_{L=0,\dots,L_m}$ and $\{J^L\}_{L=0,\dots,L_m}$

Initialization of pyramidal guess: $\boxed{\mathbf{g}^{L_m} = [g_x^{L_m} \;\; g_x^{L_m}]^T = [0 \;\; 0]^T}$

for $L = L_m$ **down to** 0 **with step of** -1

Location of point $\mathbf{u}$ on image I^L: $\mathbf{u}^L = [p_x \;\; p_y]^T = \mathbf{u}/2^L$

Derivative of I^L with respect to x:
$$I_x(x,y) = \frac{I^L(x+1,y) - I^L(x-1,y)}{2}$$

Derivative of I^L with respect to y:
$$I_y(x,y) = \frac{I^L(x,y+1) - I^L(x,y-1)}{2}$$

Spatial gradient matrix:
$$G = \sum_{x=p_x-\omega_x}^{p_x+\omega_x} \sum_{y=p_y-\omega_y}^{p_y+\omega_y} \begin{bmatrix} I_x^2(x,y) & I_x(x,y)\, I_y(x,y) \\ I_x(x,y)\, I_y(x,y) & I_y^2(x,y) \end{bmatrix}$$

Initialization of iterative L-K: $\overline{\nu}^0 = [0 \;\; 0]^T$

Pyramidal Iterative KLT

for $k = 1$ **to** K **with step of** 1 (or until $\|\overline{\eta}^k\| <$ accuracy threshold)

Image difference: $\delta I_k(x, y) = I^L(x, y) - J^L(x + g_x^L + \nu_x^{k-1}, y + g_y^L + \nu_y^{k-1})$

Image mismatch vector: $\bar{b}_k = \sum_{x=p_x-\omega_x}^{p_x+\omega_x} \sum_{y=p_y-\omega_y}^{p_y+\omega_y} \begin{bmatrix} \delta I_k(x, y)\, I_x(x, y) \\ \delta I_k(x, y)\, I_y(x, y) \end{bmatrix}$

Optical flow (Lucas-Kanade): $\overline{\eta}^k = G^{-1}\, \bar{b}_k$

Guess for next iteration: $\overline{\nu}^k = \overline{\nu}^{k-1} + \overline{\eta}^k$

end of for-loop on k

Final optical flow at level L: $\mathbf{d}^L = \overline{\nu}^K$

Guess for next level $L - 1$: $\boxed{\mathbf{g}^{L-1} = [g_x^{L-1} \;\; g_y^{L-1}]^T = 2\,(\mathbf{g^L} + \mathbf{d}^L)}$

end of for-loop on L

Final optical flow vector: $\mathbf{d} = \mathbf{g}^0 + \mathbf{d}^0$

Location of point on J: $\mathbf{v} = \mathbf{u} + \mathbf{d}$

Solution: The corresponding point is at location $\mathbf{v}$ on image J

Other Improvements

3D视觉工坊

针对光度恒定假设改进

- 使用Locally-scaled sum of squared differences(LSSD)

$$r_i(\boldsymbol{\xi}) = \frac{I_{t+1}(\mathbf{T}\mathbf{x}_i)}{\overline{I_{t+1}}} - \frac{I_t(\mathbf{x}_i)}{\overline{I_t}} \quad \forall \mathbf{x}_i \in \Omega.$$

- 使用全局光度变换模型

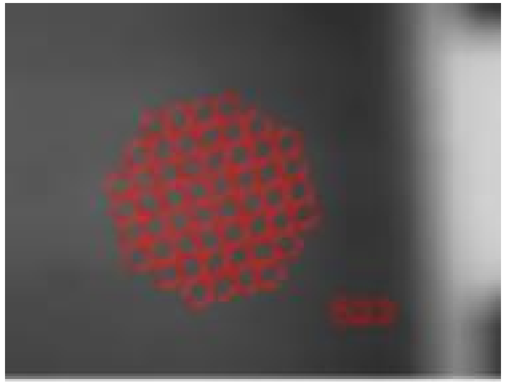
$$\sum_{\mathbf{x}} \left[\mathbf{e}^a I\big(\mathbf{W}(\mathbf{x};\mathbf{p})\big) + b - T(\mathbf{x}) \right]^2$$

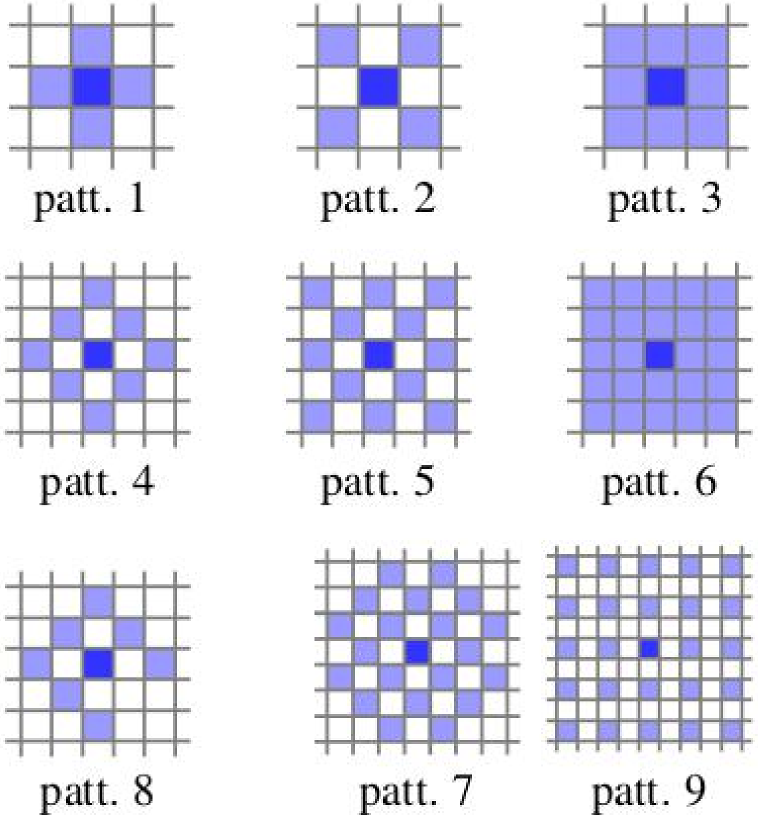
- 假设梯度不变进行跟踪

Other Improvements

针对鲁棒性改进

- 改变Patch的形状，应对旋转等特殊运动





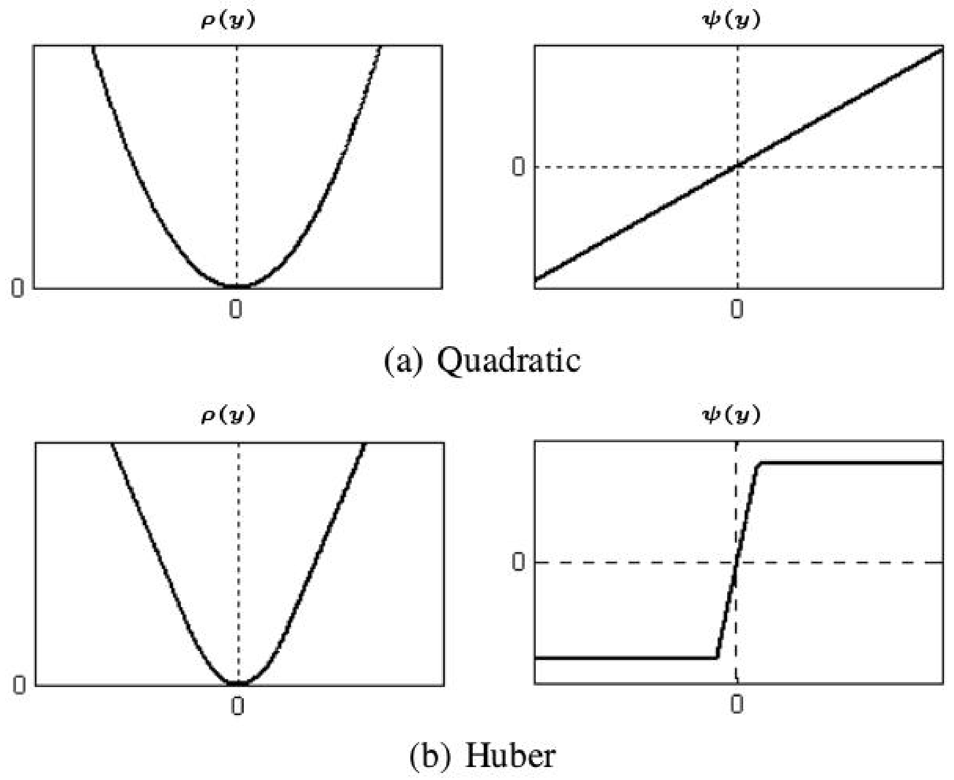
针对鲁棒性改进

- 使用Robust Kernel Function，加权处理外点

$$\sum_{\mathbf{x}} \boldsymbol{\rho}\left(I(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{x})\right)^2$$

- 使用仿射变换模型

$$J(Ax + d) = I(x),\ \text{其中}\ A = 1 + D = 1 + \begin{bmatrix} d_{xx} & d_{xy} \\ d_{yx} & d_{yy} \end{bmatrix}$$



Reference

- yves Bouguet J. Pyramidal implementation of the lucas kanade feature tracker[J]. Intel Corporation, Microprocessor Research Labs, 2000, 1.
- Baker S, Matthews I. Lucas-kanade 20 years on: A unifying framework[J]. International journal of computer vision, 2004, 56(3): 221-255.
- http://www.cs.cmu.edu/~16385/

3D视觉从入门到精通知识星球：针对3D视觉领域的知识点汇总、入门进阶学习路线、最新paper分享、疑问解答四个方面进行深耕，更有各类大厂的算法工程人员进行技术指导，近1000+的星球成员为创造更好的AI世界共同进步，知识星球入口：

欢迎关注3D视觉工坊

公众号

我们这里有3D视觉算法、VSLAM算法、计算机视觉、深度学习、自动驾驶、图像处理等干货分享！

交流群请添加客服

如果你也想成为主讲人，欢迎加入我们。

➢ 报名方式：请发送邮件至vision3d@yeah.net

Thanks