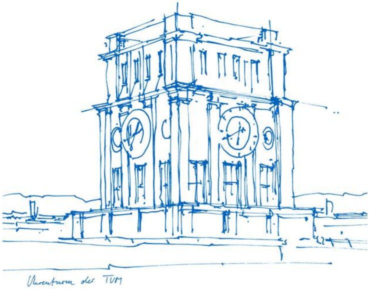
# Computer Vision II: Multiple View Geometry (IN2228)

## Chapter 05 Correspondence Estimation
## (Part 1 Small Motion)

Dr. Haoang Li

11 May 2023  11:00 to 11:45



# Announcement before Class

## ➢ Course Content, Exercise Session, and Exam

✓ Course content and exercise session

- This year, we have new lecturers and new teaching assistants. Slides for lectures are totally new, but exercise questions are partly based on the materials from the previous years.
- **I will try to introduce more detailed knowledge required by the exercise session in the future**.

✓ Couse content and exam

- **Exam questions will be most aligned to the course content**, so they will be partly different from questions from previous years. (Note: there still will be some overlaps.)
- All the involved knowledge in the exam will be clearly introduced in our class. Therefore, **as long as you understand the knowledge introduced in our class, you should obtain a good grade**.
- I will prepare **a class to review knowledge important for the exam (tentatively on 13 July).** P.S.: Exam is on 04 August.

# Announcement before Class

## ➢ Programming Assignments and Bonus

✓ Programming assignment

- We received some feedback and we have discussed potential solutions.
- For example, our teaching assistants will add additional feedback on the sample tests to help students pass the tests more easily.
- Please note that there are also hidden test cases that your solution is being evaluated against.

✓ Bonus

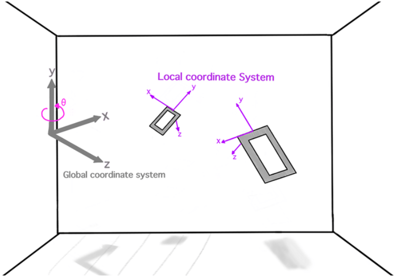
- From my perspective, this bonus is not very easy to obtain.
- If we think you have to spend much more time than you expected, i.e., it is not very “economical”, please mainly focus on the content of our lecture.
- You can still obtain a satisfactory grade even without bonus.

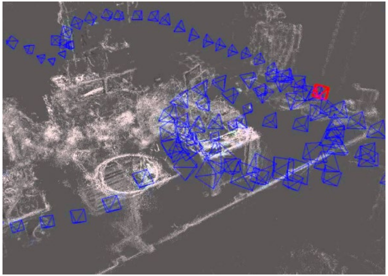
# Clarification before Class

➢ Single Camera and Multiple Camera Frames

✓ In VO/SLAM/SFM, we use a **single camera** to obtain **multiple images** from different view points. **Multiple view points** correspond to **multiple camera frames**.

✓ However, in practice, we may do not differentiate between “cameras” and “camera frames”.



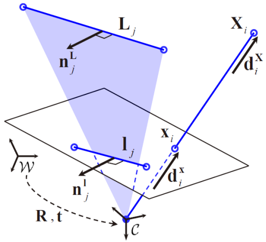
# Clarification before Class

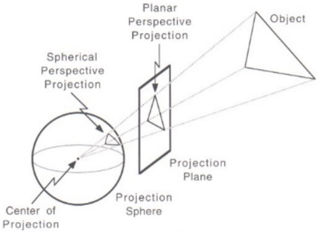
➢ Projection Plane and Image Plane

✓ Strictly, projection plane refers to the plane defined by the origin of a coordinate system and a 3D line/2D line.

✓ However, in practice, projection plane may also correspond to the image plane.

✓ Projection ray refers to the 3D direction defined by the origin of a coordinate system and a 3D point/2D point.





# Explanation before Class

➢ Homogeneous Coordinates of 2D line

Two representative methods introduced in the middle school

✓ $y=kx+b$

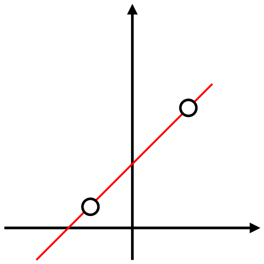
✓ $ax+by+c=0$

- $(a, b, c)$ is the homogenous coordinates of 2D line
- $(1, 2, 3)$ is equivalent to $(2, 4, 6)$

- Two points $(x_0, y_0)$ and $(x_1, y_1)$ determine a 2D line

$$\begin{cases} ax_0+by_0+c=0 \\ ax_1+by_1+c=0 \end{cases}$$

To solve this linear system, we can choose an arbitrary value of c



- We can directly obtain $(a, b, c)$ by the cross product between $(x_0, y_0, 1)$ and $(x_1, y_1, 1)$

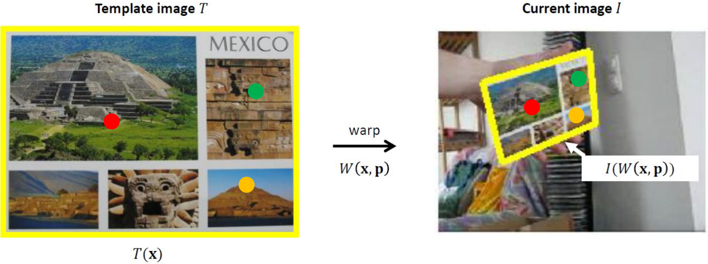
# Today's Outline

- ➢ Overview of Matching/Tracking Problem
- ➢ KLT Tracker for Small Motion
- • Simplified Case: Pure Translation
- • General Case

# Overview of Matching/Tracking Problem

➢ Problem Formulation

✓ A practical task: estimate the transformation $W$ (warping) between a template image $T$ and the current image $I$.




Template image

Current image

✓ Clue: All the (inlier) 2D-2D point correspondences should satisfy the same warping model.

# Overview of Matching/Tracking Problem

➢ Problem Formulation

✓ The warping estimation problem can be reformulated as the correspondence finding problem.

✓ Example of Euclidian transformation

$$x' = x\cos(a_3) - y\sin(a_3) + a_1$$
$$y' = x\sin(a_3) + y\cos(a_3) + a_2$$

$(x, y)$ and $(x', y')$ constitute a pair of unknown-but-sought correspondence

$\mathbf{p} = (a_1, a_2, \dots, a_n)$ are warping parameters to estimate

A "chicken-and-egg" problem

✓ Two types of solutions to find correspondences exit: indirect and direct methods.

# Overview of Matching/Tracking Problem

➢ Problem Formulation

✓ Indirect methods (next week)

- They work by detecting and matching features (points or lines)
- Pros: They can cope with large frame-to-frame motions and strong illumination changes
- Cons: They are slow due to costly feature extraction, matching, and outlier removal (e.g., RANSAC)


Matched points

# Overview of Matching/Tracking Problem

➢ Problem Formulation

✓ Indirect methods (next week)

1. Detect and match features that are invariant to scale, rotation, view point changes (e.g., SIFT)

2. Geometric verification (RANSAC) (e.g., 4 point RANSAC for planar objects, or 5 or 8 point RANSAC for 3D objects)

3. Refine estimate by minimizing the sum of squared reprojection errors between the observed feature $\boldsymbol{f}^i$ in the current image and the warped corresponding feature $W(\mathbf{x}^i, \mathbf{p})$ from the template

$$\mathbf{p} = argmin_{\mathbf{p}} \sum_{i=1}^{N} \left\| W(\mathbf{x}^i, \mathbf{p}) - \boldsymbol{f}^i \right\|^2$$

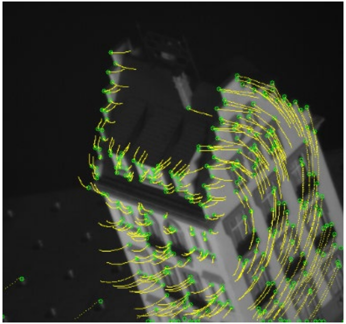
Feature distance

# Overview of Matching/Tracking Problem

➢ Problem Formulation

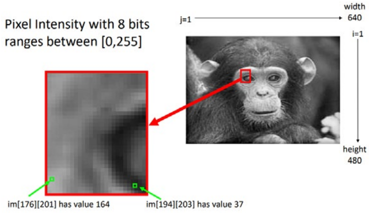
✓ Direct methods (today)

- Pros: All information in the image can be exploited (higher accuracy, higher robustness to motion blur and weak texture (i.e., weak gradients))
- Pros: Increasing the camera frame rate reduces computational cost per frame (no RANSAC needed)
- Cons: Very sensitive to initial value limited frame to frame motion



Tracked points

# Overview of Matching/Tracking Problem



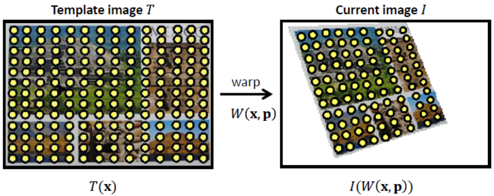

➢ Problem Formulation

✓ Direct methods (today)

- They work by directly processing pixel intensities.
- Technically, they estimate the parameters p of the transformation $W(\mathbf{x},\mathbf{p})$ that minimize the Sum of Squared Differences:

$$\mathbf{p} = argmin_{\mathbf{p}} \sum_{\mathbf{x} \in \mathbf{T}} \left[ I(W(\mathbf{x},\mathbf{p})) - T(\mathbf{x}) \right]^2$$

Intensity/brightness distance




Every yellow dot in this image denotes a pixel

# Overview of Matching/Tracking Problem

## ➢ Assumptions of Direct Methods

✓ Brightness constancy

- The intensity of the pixels to track does not change much over consecutive frames
- It does not cope with strong illumination changes

# Overview of Matching/Tracking Problem

➢ Assumptions of Direct Methods

✓ Temporal consistency

- Small frame-to-frame motion (1-2 pixels).
- It does not cope with large frame to frame motion. However, this can be addressed using coarse to fine multi scale implementations (see later)

✔

✘

# Overview of Matching/Tracking Problem

## ➢ Assumptions of Direct Methods

✓ Spatial coherence

- All pixels in the template undergo the same transformation (i.e., they roughly lie on the same 3D surface)

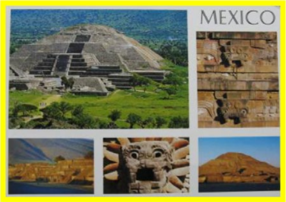

Point roughly lying on the same surface (face)

# Overview of Matching/Tracking Problem

➢ Assumptions of Direct Methods

✓ Spatial coherency

- No errors in the template image boundaries: only the object to track appears in the template image
- No occlusion: the entire template is visible in the input image




✔


Foreground and background have different motions

✘


Occlusion

✘

# KLT Tracker for Small Motion

➢ Overview

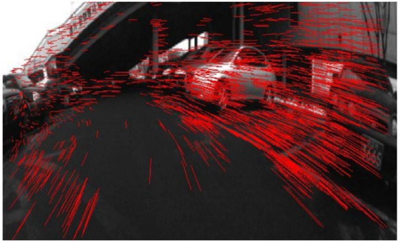
The Kanade-Lucas-Tomasi (KLT) tracker tackles two problems:

✓ How should we select features?

- Tomasi-Kanade: Method for choosing the best feature (image patch) for tracking

✓ How should we track them from frame to frame?

- Lucas-Kanade: Method for aligning (tracking) an image patch



----------------------------------------------------

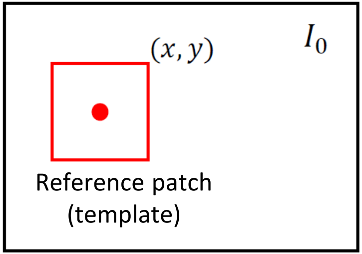
✓ Structure of our introduction

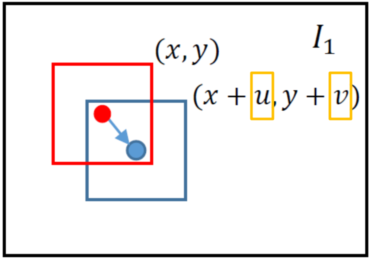
- Simplified case: pure translation
- General case

# KLT Tracker for Small Motion

➢ Simplified Case: Pure Translation

- ✓ Consider the reference patch centered at $(x,y)$ in image $I_0$ and the shifted patch centered at $(x+u,y+v)$ in image $I_1$. The patch has size $\Omega$.
- ✓ We want to find the motion vector $(u,v)$ that minimizes the Sum of Squared Differences (SSD) w.r.t. the intensity (based on the **intensity invariance assumption**):



# KLT Tracker for Small Motion

## ➢ Simplified Case: Pure Translation

✓ Recap on mathematical knowledge

- Derivative and gradient

Function: $f(x)$ — Single **univariate** function

Derivative: $f'(x) = \frac{df}{dx}$, where $x$ is a scalar

---

Function: $f(x_1, x_2, \ldots, x_n)$ — Single **multivariate** function

Gradient: $\nabla f(x_1, x_2, \ldots, x_n) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n}\right)$

⇨

Multiple **multivariate** function

$$F(x_1, x_2, \ldots, x_n) = \begin{bmatrix} f_1(x_1, x_2, \ldots, x_n) \\ \vdots \\ f_m(x_1, x_2, \ldots, x_n) \end{bmatrix}$$ is a vector-valued function

The derivative in this case is called Jacobian $\frac{\partial F}{\partial \mathbf{x}}$:

$$\frac{\partial F}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}, \ldots, \frac{\partial f_1}{\partial x_n} \\ \vdots \\ \frac{\partial f_m}{\partial x_1}, \ldots, \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

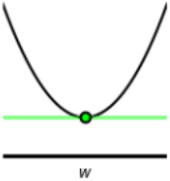
# KLT Tracker for Small Motion

➢ Simplified Case: Pure Translation

✓ Recap on mathematical knowledge

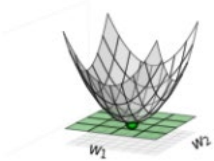
- First-order optimality condition

Zero-valued derivative(s) with respect to the unknown parameter(s) correspond to the minimum.




$$\frac{\mathrm{d}}{\mathrm{d}w} g\left(w\right) = 0$$

1D function




$$\frac{\partial}{\partial w_1} g(\mathbf{v}) = 0$$
$$\frac{\partial}{\partial w_2} g(\mathbf{v}) = 0$$
$$\vdots$$
$$\frac{\partial}{\partial w_N} g(\mathbf{v}) = 0$$

2D function

$$\Rightarrow \nabla g\left(\mathbf{v}\right) = \mathbf{0}_{N\times 1}$$

Gradient notation

# KLT Tracker for Small Motion

➢ Simplified Case: Pure Translation

✓ Cost function (quadratic function) w.r.t. two variables $(u,v)$

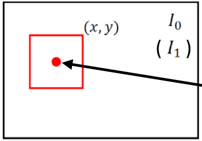
$$SSD(u,v) = \sum_{x,y\in\Omega} \left(I_0(x,y) - I_1(x+u, y+v)\right)^2$$

$$I_1(x+u, y+v) \cong I_1(x,y) + I_x u + I_y v$$

Directional derivative

$$\Rightarrow SSD(u,v) \cong \sum_{x,y\in\Omega} \left(I_0(x,y) - I_1(x,y) - I_x u - I_y v\right)^2$$

$$\Rightarrow SSD(u,v) \cong \sum_{x,y\in\Omega} \left(\Delta I - I_x u - I_y v\right)^2$$




Intensity difference at (x, y)

$$\sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n$$

$$f(a) + \frac{f'(a)}{1!}(x-a)$$

The first-order Taylor polynomial

# KLT Tracker for Small Motion

- Simplified Case: Pure Translation

$$\Rightarrow SSD(u,v) \cong \sum_{x,y\in\Omega} (\Delta I - I_x u - I_y v)^2$$

✓ To minimize it, we differentiate it with respect to $(u,v)$ and equate it to zero:

$$\frac{\partial SSD}{\partial u} = 0 \;,\; \frac{\partial SSD}{\partial v} = 0$$

$$\frac{\partial SSD}{\partial u} = 0 \;\Rightarrow\; -2\sum I_x(\Delta I - I_x u - I_y v) = 0$$

$$\frac{\partial SSD}{\partial v} = 0 \;\Rightarrow\; -2\sum I_y(\Delta I - I_x u - I_y v) = 0$$

# KLT Tracker for Small Motion

- Simplified Case: Pure Translation

$$\sum I_x(\Delta I - I_x u - I_y v) = 0$$

$$\sum I_y(\Delta I - I_x u - I_y v) = 0$$

✓ Linear system of two equations w.r.t. two unknown parameters $(u,v)$
✓ We can write them in matrix form:

$$\underbrace{\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}}_{\text{M}} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum I_x \Delta I \\ \sum I_y \Delta I \end{bmatrix} \quad \Rightarrow \quad \begin{bmatrix} u \\ v \end{bmatrix} = \underbrace{\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}^{-1}}_{\text{Inverse of M}} \begin{bmatrix} \sum I_x \Delta I \\ \sum I_y \Delta I \end{bmatrix}$$

# KLT Tracker for Small Motion

➢ Simplified Case: Pure Translation

Eigenvector

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$$

Eigenvalue

$$\det(A) = \lambda_1 \lambda_2 \cdots \lambda_n$$

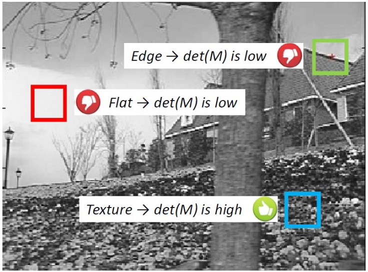
- ✓ For $M$ to be invertible, $\det(M)$ should be non zero, which means that its eigenvalues should be large (i.e., not a flat region, not an edge)
- ✓ In practice, it should be a corner or more generally contain texture

Eigenvalues

$$M = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$

Eigenvectors

$I_x$ and $I_y$: Directional derivatives



# KLT Tracker for Small Motion

## ➢ Simplified Case: Pure Translation

✓ Answer to our two main tasks

- How should we select features?

Patch whose associated M matrix has large eigen values.

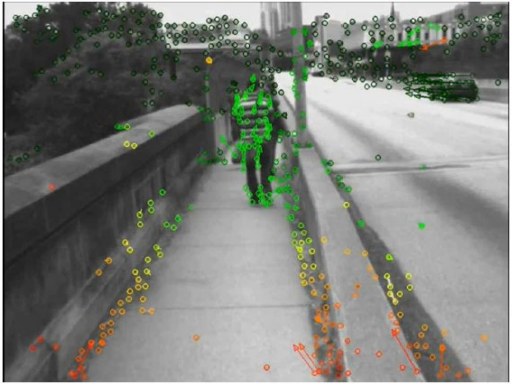
$$M = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}$$

- How should we track them from frame to frame?

(u, v) is the displacement vector.

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}^{-1} \begin{bmatrix} \sum I_x \Delta I \\ \sum I_y \Delta I \end{bmatrix}$$

Intensity difference at (x, y)



Color encodes motion direction

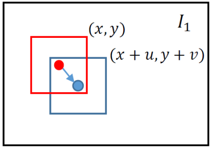
# KLT Tracker for Small Motion

## ➢ General Case

✓ Relationship between pure translation and general motion

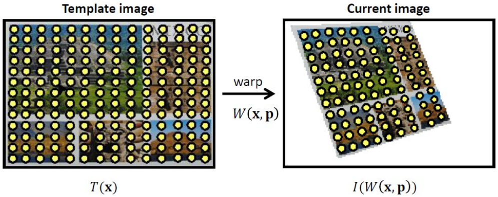
- Definition of cost function

$$SSD(u,v) = \sum_{x,y\in\Omega} \left(I_0(x,y) - \underbrace{I_1(x+u, y+v)}_{\text{Pure translation}}\right)^2$$




$$SSD = \sum_{\mathbf{x}\in\mathbf{T}} \left[I(W(\mathbf{x},\mathbf{p})) - T(\mathbf{x})\right]^2$$

General transformation (warping)
a vector-valued function
**p** represents the warping parameters



# KLT Tracker for Small Motion

➢ General Case

$$I_1(x+u, y+v))^2$$

$$I_1(x+u, y+v) \cong I_1(x,y) + I_x u + I_y v$$

Recap on first-order approximation in pure translation case

✓ Relationship between pure translation and general motion

- Minimization of cost function

**Similarity:**
In both case, we apply a first order approximation of the warping.

$$\frac{\partial SSD}{\partial u} = 0 \,, \frac{\partial SSD}{\partial v} = 0$$

Recap on zero-valued partial derivatives in pure translation case

**Difference:**

- In pure translation case, we equate partial derivatives to zero and directly obtain solutions (u, v).
- In the general case, we leverage Gauss-Newton method to minimize the SSD iteratively. (We can still use first-order optimality condition to generate equations w.r.t. warping parameters, but they may be difficult to solve.)

# KLT Tracker for Small Motion

$$SSD = \sum_{\mathbf{x} \in \mathbf{T}} \left[ I(W(\mathbf{x}, \mathbf{p})) - T(\mathbf{x}) \right]^2$$

## ➢ General Case

✓ Overview

We incrementally update the warping parameters **p** to continuously reduce the value of cost function.

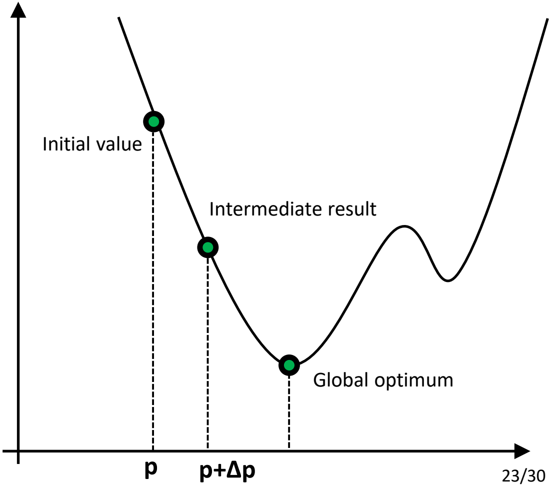
✓ One iteration

Assume that an initial estimate of **p** is known. Then, we want to find the increment $\Delta\mathbf{p}$ that minimizes

Unknown

$$SSD = \sum_{\mathbf{x} \in \mathbf{T}} \left[ I(W(\mathbf{x}, \mathbf{p} + \Delta\mathbf{p})) - T(\mathbf{x}) \right]^2$$

Known (initial guess or result of the previous iteration)



# KLT Tracker for Small Motion

➤ General Case

$$SSD = \sum_{\mathbf{x} \in \mathbf{T}} \left[ I(W(\mathbf{x}, \mathbf{p} + \Delta\mathbf{p})) - T(\mathbf{x}) \right]^2$$

✓ One iteration

First-order Taylor approximation of $I(W(\mathbf{x}, \mathbf{p} + \Delta\mathbf{p}))$ yields to:

$$I(W(\mathbf{x}, \mathbf{p} + \Delta\mathbf{p})) \cong I(W(\mathbf{x}, \mathbf{p})) + \nabla I \frac{\partial W}{\partial \mathbf{p}} \Delta\mathbf{p}$$

$\nabla I = [I_x, I_y]$ = Image gradient evaluated at $W(\mathbf{x}, \mathbf{p})$

Jacobian of the warp $W(\mathbf{x}, \mathbf{p})$

---

$\nabla f(x_1, x_2, \ldots, x_n) = \left( \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n} \right)$ is the image gradient

$F(x_1, x_2, \ldots, x_n) = \begin{bmatrix} f_1(x_1, x_2, \ldots, x_n) \\ \vdots \\ f_m(x_1, x_2, \ldots, x_n) \end{bmatrix}$ is a vector-valued function

Jacobian $\frac{\partial F}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}, \ldots, \frac{\partial f_1}{\partial x_n} \\ \vdots \\ \frac{\partial f_m}{\partial x_1}, \ldots, \frac{\partial f_m}{\partial x_n} \end{bmatrix}$

# KLT Tracker for Small Motion

$$SSD = \sum_{\mathbf{x} \in \mathbf{T}} \left[ I(W(\mathbf{x}, \mathbf{p} + \Delta\mathbf{p})) - T(\mathbf{x}) \right]^2$$

➢ General Case

✓ One iteration

Substitute Taylor approximation of $I(W(\mathbf{x}, \mathbf{p} + \Delta\mathbf{p}))$ into SSD cost, we have new cost function:

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

✓ How do we minimize new cost?

Briefly, we differentiate SSD with respect to $\Delta\mathbf{p}$ and we equate it to zero, i.e., $\frac{\partial SSD}{\partial \Delta\mathbf{p}} = 0$

# KLT Tracker for Small Motion

➢ General Case

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

✓ How do we minimize new cost?

By differentiating SSD with respect to $\Delta\mathbf{p}$ and setting the result as 0, we have

Derivative:

distributive laws

$$2\sum_{\mathbf{x}\in\mathbf{T}} \left[\nabla I \frac{\partial W}{\partial \mathbf{p}}\right]^{\mathrm{T}} \left[ I(W(\mathbf{x},\mathbf{p})) + \nabla I \frac{\partial W}{\partial \mathbf{p}} \Delta\mathbf{p} - T(\mathbf{x}) \right] = 0 \Rightarrow \Delta\mathbf{p} = H^{-1} \sum_{\mathbf{x}\in\mathbf{T}} \left[\nabla I \frac{\partial W}{\partial \mathbf{p}}\right]^{\mathrm{T}} \left[ T(\mathbf{x}) - I(W(\mathbf{x},\mathbf{p})) \right] =$$

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

# KLT Tracker for Small Motion

➢ Another Derivation in General Case (for Exercise Session)

✓ Brightness Constancy

- Video (sequential images) is w.r.t. the time t and a tracked point's position is also w.r.t. the time t.

$$I(x(t), t) = \text{const.} \quad \forall t,$$

- Based on the brininess consistency assumption, we set derivative as 0.

$$\frac{d}{dt} I(x(t), t) = \nabla I^\top \left( \frac{dx}{dt} \right) + \frac{\partial I}{\partial t} = 0.$$ Called "optical flow constraint"

$$V = \frac{dx}{dt}$$ Velocity

# KLT Tracker for Small Motion

➢ Another Derivation in General Case (for Exercise Session)

✓ Constant motion in a neighborhood:

- We assume that the velocity v is constant over a neighborhood W(x) of the point x

$$\nabla I^\top \left( \frac{dx}{dt} \right) + \frac{\partial I}{\partial t} = 0.$$

$$\Downarrow$$

$$\nabla I(x', t)^\top v + \frac{\partial I}{\partial t}(x', t) = 0 \quad \forall x' \in W(x).$$

Image patches

# KLT Tracker for Small Motion

- Another Derivation in General Case (for Exercise Session)

✓ Compute the best velocity vector v for the point x by minimizing the least squares error

$$\nabla I(x', t)^\top v + \frac{\partial I}{\partial t}(x', t) = 0 \quad \forall x' \in W(x). \quad \Rightarrow \quad E(v) = \int_{W(x)} \left| \nabla I(x', t)^\top v + I_t(x', t) \right|^2 dx'.$$

This will be used in exercise session

✓ Expanding the terms and setting the derivative to zero:

$$\frac{dE}{dv} = 2\underline{M}v + 2\underline{q} = 0 \quad \text{where} \quad \underline{M} = \int_{W(x)} \nabla I \, \nabla I^\top dx', \quad \text{and} \quad \underline{q} = \int_{W(x)} I_t \, \nabla I \, dx'.$$

$$\Rightarrow \quad v = -M^{-1} q.$$

# Summary

- ➢ Overview of Tracking Problem
- ➢ KLT Tracker for Small Motion
- • Simplified case: pure translation
- • General case

Thank you for your listening!

If you have any questions, please come to me :-)