Компьютерное Зрение
Лекция № 11, осень 2021

Задача сопровождения

ЛИТПМ

МФТИ

Кафедра технологий проектирования сложных технических систем

Что будем изучать сегодня

- Feature Tracking
- Simple KLT tracker
- 2D transformations
- Iterative KLT tracker
- Multi-target tracking

Что будем изучать сегодня

- Feature Tracking
- Simple KLT tracker
- 2D transformations
- Iterative KLT tracker
- Multi-target tracking

Постановка задачи

Image sequence

Slide credit: Yonsei Univ.

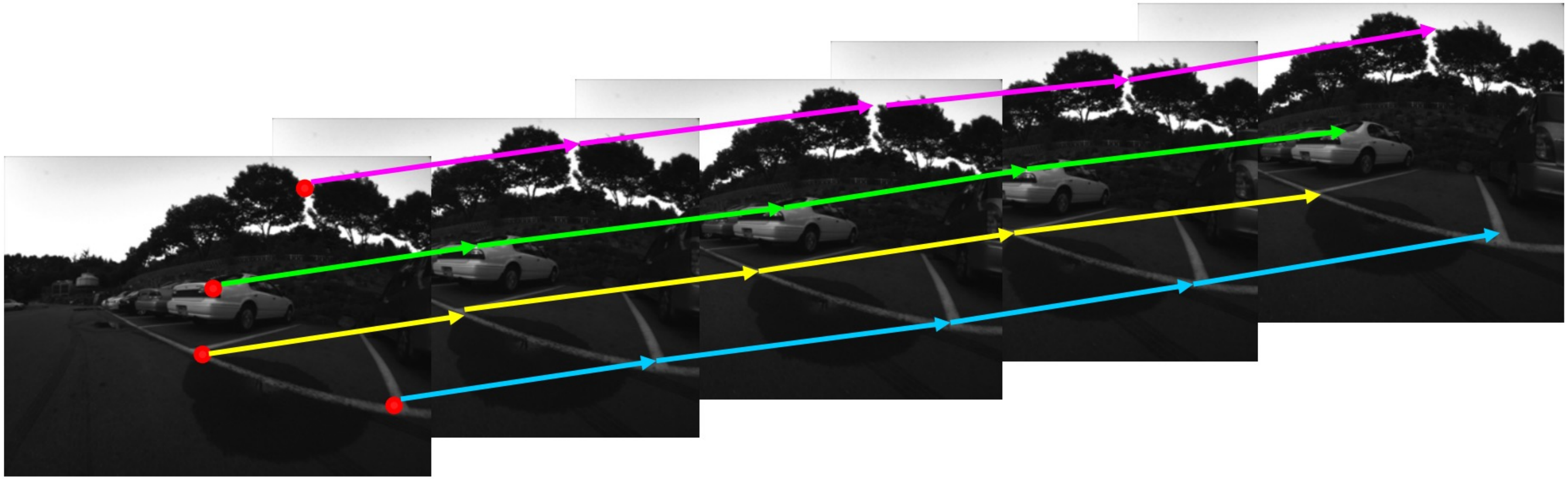
Постановка задачи

Feature point detection

Slide credit: Yonsei Univ.

Problem statement

Feature point tracking



Slide credit: Yonsei Univ.

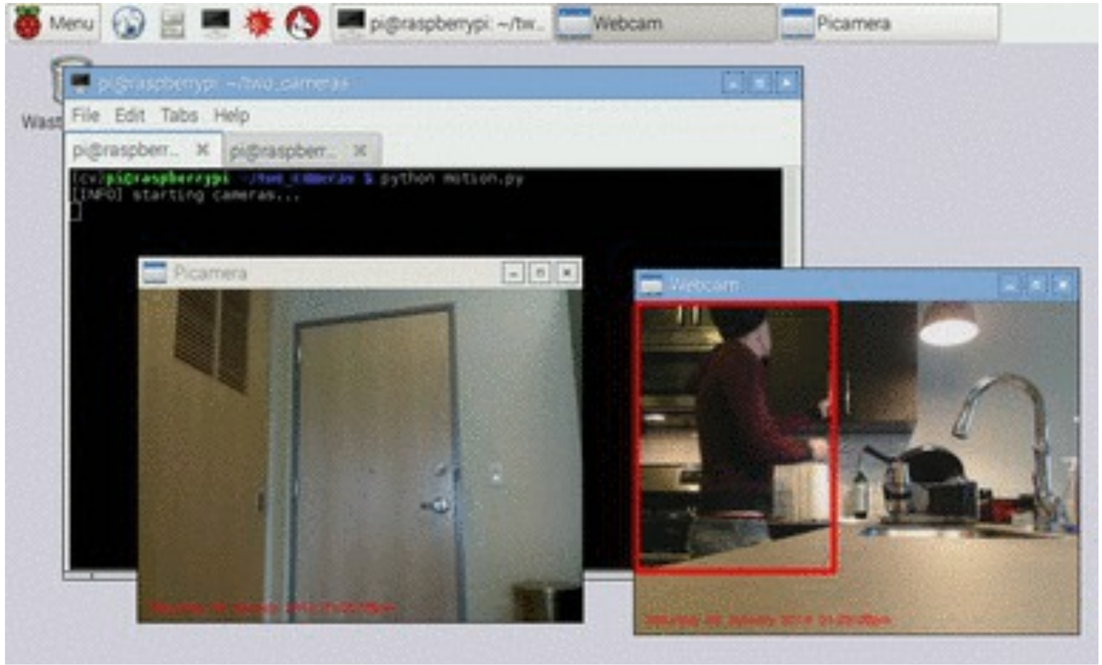
Single object tracking

Multiple object tracking

Tracking with a fixed camera

Tracking with a moving camera

Tracking with multiple cameras



Проблемы в задаче сопровождения

- Определить за чем следить
- Некоторые точки могут меняться во времени
  - То есть повороты, изменения яркости и тд
- При длительном наблюдении могут накапливаться ошибки
- Точки могут появляться или исчезать.
  - необходимо иметь возможность добавлять/удалять отслеживаемые точки

Какие особенности применимы для сопровождения?

- Интуитивно мы хотим избежать гладких областей и краев. Но есть ли еще принципиальный способ определить хорошие черты?

- Какие области изображения можно легко и последовательно обнаружить?

Какие особенности применимы для сопровождения?

- Можно измерять "качество" функций с одного изображения

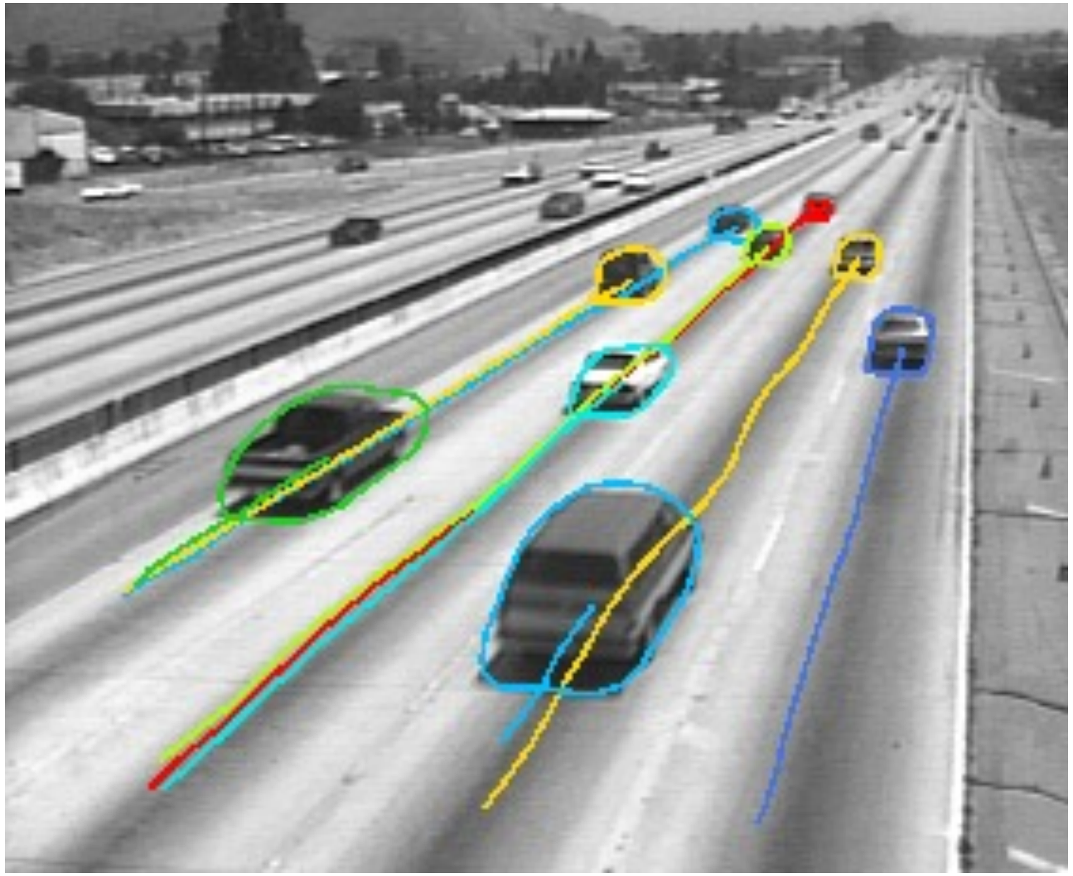
- Следовательно: отслеживание углов Харриса гарантирует малую чувствительность к ошибкам

Оценка движения

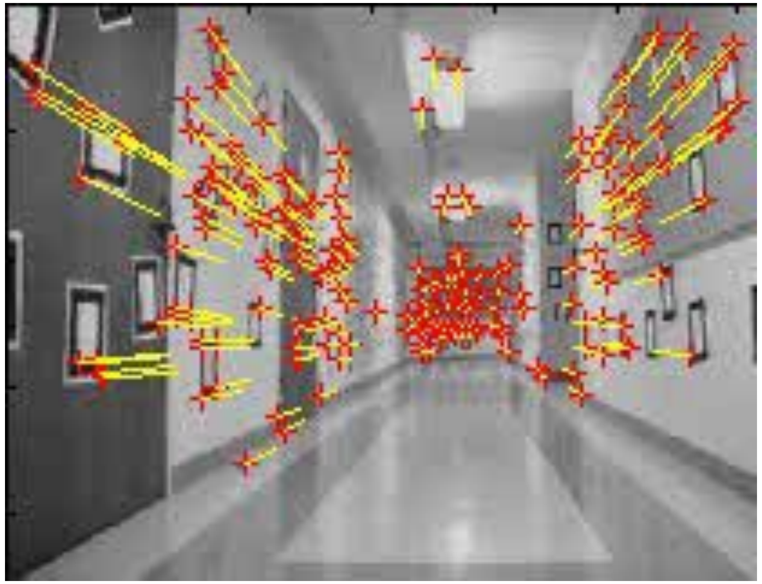
- Optical flow
  - Восстановление движения изображения на каждом пикселе из пространственно-временных изменений яркости изображения
- Feature-tracking
  - Извлекайте визуальные элементы (углы, текстурные области) и отслеживать их по нескольким кадрам

Оптический поток может помочь отслеживать особенности

Как только у нас появятся особенности, которые мы хотим отслеживать, Lucas-Kanade или другие алгоритмы оптического потока могут помочь в отслеживании этих характеристик

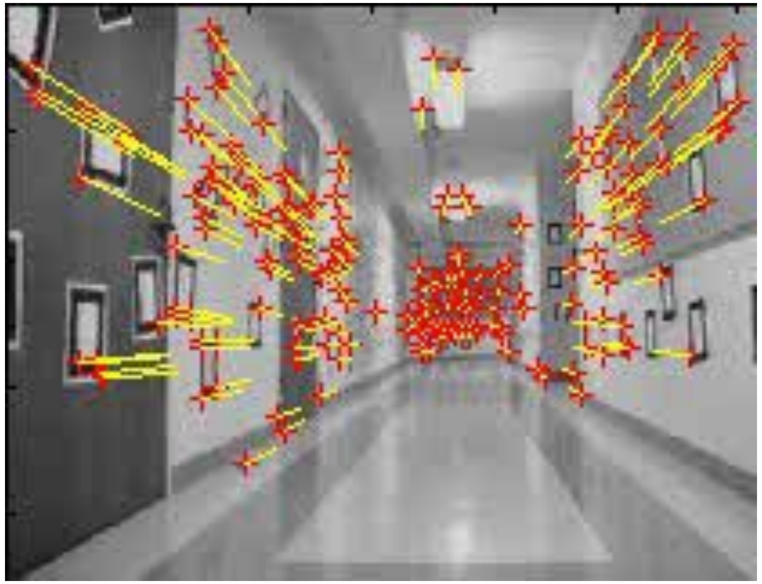


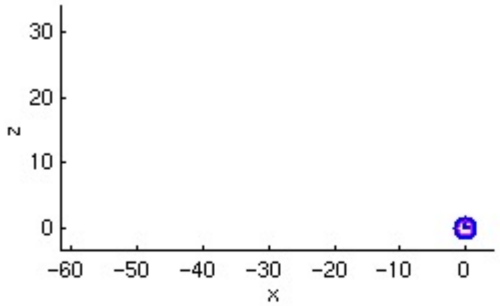
Feature-tracking



Courtesy of Jean-Yves Bouguet – Vision Lab, California Institute of Technology

Feature-tracking





Courtesy of Jean-Yves Bouguet – Vision Lab, California Institute of Technology

Что будем изучать сегодня

- Feature Tracking
- Simple KLT tracker
- 2D transformations
- Iterative KLT tracker
- Multi-target tracking

Simple Kanade–Lucas–Tomasi (KLT) tracker

1. Найти хорошую точку для отслеживания (угол Харрис)
2. Для каждого угла Харриса вычислить движение между последовательными кадрами
3. Соединить векторы движения в последовательных кадрах, чтобы получить дорожку для каждой точки
4. Ввести новые точки, применяя детектор Харриса через каждые кадров
5. Повторить шаги 1-3

KLT tracker for fish

Video credit: Kanade

Tracking cars

Video credit: Kanade

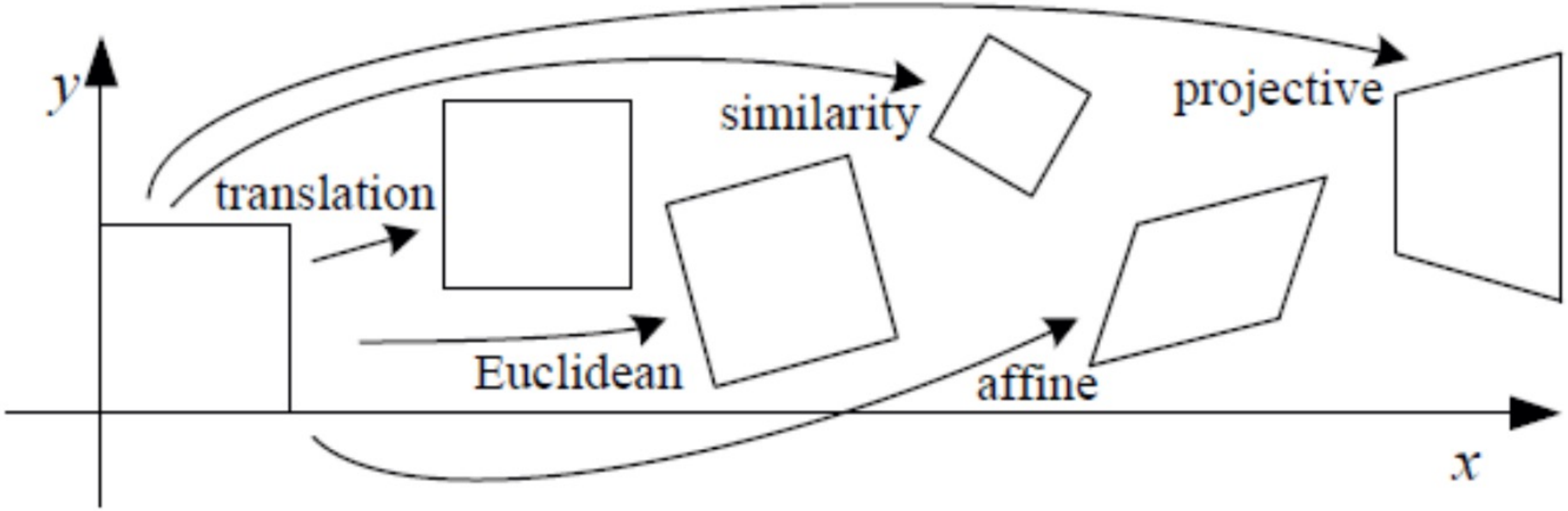
Tracking movement

Video credit: Kanade

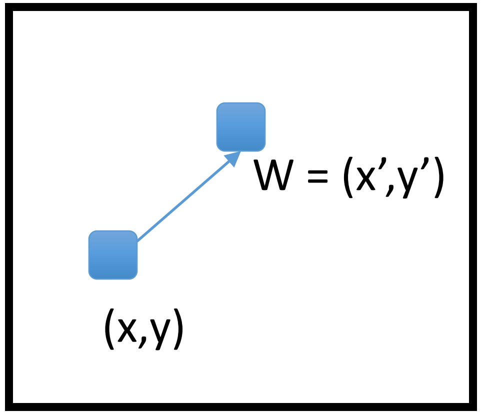
Что будем изучать сегодня

- Feature Tracking
- Simple KLT tracker
- 2D transformations
- Iterative KLT tracker
- Multi-target tracking

Типы 2D преобразований



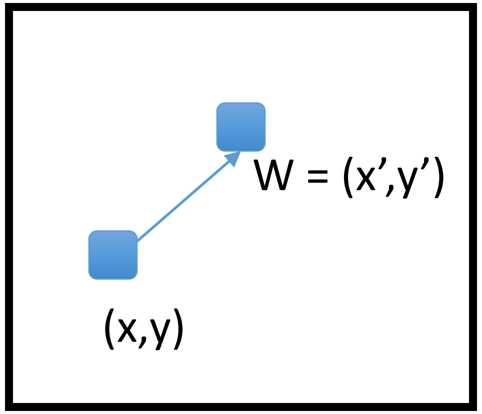
Смещение



- Пусть начальная функция будет расположена по (x, y).
- В следующем кадре она переводится в (x', y').
- Мы можем записать преобразование как:

$$x' = x + b_1$$
$$y' = y + b_2$$

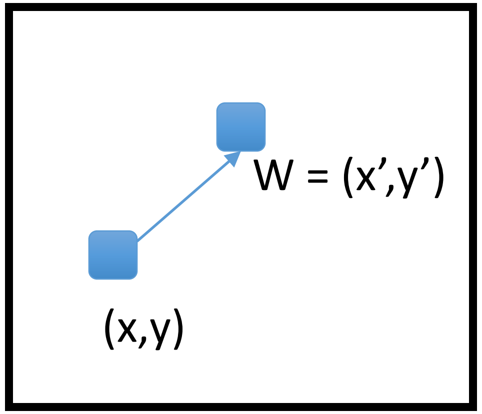
Смещение



$$x' = x + b_1$$
$$y' = y + b_2$$

- Запишем в гомогенных координатах:
- $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 & b_1 \\ 0 & 1 & b_2 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

Смещение

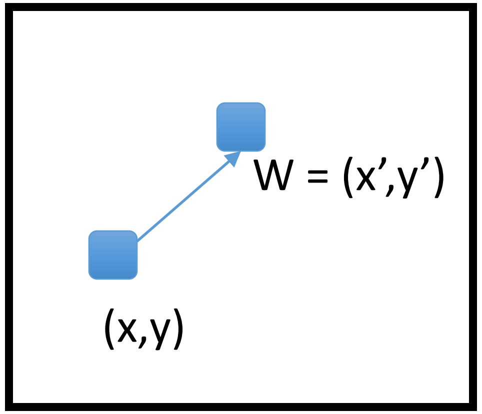


- $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 & b_1 \\ 0 & 1 & b_2 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

Обозначим:

- $W(\boldsymbol{x}; \boldsymbol{p}) = \begin{bmatrix} 1 & 0 & b_1 \\ 0 & 1 & b_2 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

Модель перемещения для преобразования



- $W(\boldsymbol{x};\boldsymbol{p}) = \begin{bmatrix} 1 & 0 & b_1 \\ 0 & 1 & b_2 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

Получаем 2 параметра:

$$\boldsymbol{p} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

Тогда производна по параметрам **p**:

$$\frac{\partial W}{\partial \boldsymbol{p}}(\boldsymbol{x};\boldsymbol{p}) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Получили якобиан.

Similarity motion

- Жесткое движение включает в себя масштабирование + перевод.
- Мы можем записать трансформации как:

$$x' = ax + b_1$$
$$y' = ay + b_2$$

- $W(\boldsymbol{x};\boldsymbol{p}) = \begin{bmatrix} \mathrm{a} & 0 & b_1 \\ 0 & \mathrm{a} & b_2 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$
- $\boldsymbol{p} = \begin{bmatrix} a & \mathrm{b}_1 & \mathrm{b}_2 \end{bmatrix}^T$
- $\frac{\partial W}{\partial \boldsymbol{p}}(\boldsymbol{x};\boldsymbol{p}) = \begin{bmatrix} x & 1 & 0 \\ y & 0 & 1 \end{bmatrix}$

Affine motion

- Аффинное движение включает в себя масштабирование + вращение + перевод

$$x' = a_1x + a_2y + b_1$$
$$y' = a_3x + a_4y + b_2$$

- $W(\boldsymbol{x};\boldsymbol{p}) = \begin{bmatrix} a_1 & a_2 & b_1 \\ a_3 & a_4 & b_2 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$
- $\boldsymbol{p} = \begin{bmatrix} a_1 & \mathrm{a}_2 & \mathrm{b}_1 & a_3 & a_4 & b_2 \end{bmatrix}^T$
- $\frac{\partial W}{\partial \boldsymbol{p}}(\boldsymbol{x};\boldsymbol{p}) = \begin{bmatrix} x & y & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & y & 1 \end{bmatrix}$

Что будем изучать сегодня

- Feature Tracking
- Simple KLT tracker
- 2D transformations
- Iterative KLT tracker
- Multi-target tracking

Постановка задачи

- Учитывая последовательность видеокадров, найдем особенности и отследим их по всему видео
- Воспользуемся обнаружением углов, чтобы найти особенности и их местоположение $\boldsymbol{x}$
- Для каждой особенности $\boldsymbol{x} = [x \;\; y]^T$:
  - Выберем дескриптор, создадим исходный шаблон для этих особенностей: $T(\boldsymbol{x})$

Цель KLT

- Наша цель - найти $\boldsymbol{p}$, который минимизирует разницу между шаблоном $T(\boldsymbol{x})$ и описанием нового местоположения $\boldsymbol{x}$ после прохождения трансформации.

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

- Для всех особенностей $\boldsymbol{x}$ на изображении $\boldsymbol{I}$,
  - $I(W(\boldsymbol{x};\boldsymbol{p})))$ - это оценка того, куда переходят особенности в следующем кадре после преобразования, определенного $W(\boldsymbol{x};\boldsymbol{p})$. Напомним, что $\boldsymbol{p}$ - это наш вектор параметров.

Цель KLT

- Так как $\boldsymbol{p}$ может быть большим, минимизация этой функции может быть затруднена:

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

- Вместо этого мы разобьем $\boldsymbol{p} = \boldsymbol{p_0} + \Delta\boldsymbol{p}$
  - Большое + маленькое/постоянное движение
  - Где $\boldsymbol{p_0}$ будет исправлен и мы решим для $\Delta\boldsymbol{p}$, что является небольшим значением.
  - Мы можем инициализировать $\boldsymbol{p_0}$ с нашим лучшим предположением о том, что такое движение, и инициализировать $\Delta\boldsymbol{p}$ как ноль.

Немного математики: ряд Тейлора

- Разложение по Тейлору:

$$f(x+\ \Delta x) = f(x) + \Delta x\,\frac{\partial f}{\partial x} + \Delta x^2\,\frac{\partial^2 f}{\partial x^2} + \ldots$$

- Предположим, что Δx маленькая.
- Мы можем применить это разложение к KLT и ограничиться только первыми двумя членами разложения:

Расширенная цель KLT

$$\sum_{x} [I(W(\boldsymbol{x}; \boldsymbol{p_0} + \Delta \boldsymbol{p})) - T(x)]^2$$

$$\approx \sum_{x} \left[I(W(\boldsymbol{x}; \boldsymbol{p_0})) + \nabla I \frac{\partial W}{\partial \boldsymbol{p}} \Delta \boldsymbol{p} - T(x) \right]^2$$

Хорошо, что мы уже подсчитали, как будет выглядеть $\frac{\partial W}{\partial \boldsymbol{p}}$ для аффинных, переводов и других преобразований

Расширенная цель KLT

- Поэтому наша цель - найти $\Delta \boldsymbol{p}$, которая сводит к минимуму следующее:

$$\underset{\Delta \boldsymbol{p}}{\operatorname{argmin}} \sum_{x} \left[I(W(\boldsymbol{x}; \boldsymbol{p_0})) + \nabla I \frac{\partial W}{\partial \boldsymbol{p}} \Delta \boldsymbol{p} - T(x) \right]^2$$

- Где $\nabla I = \begin{bmatrix} I_x & I_y \end{bmatrix}$
- Берем производную ($\Delta \boldsymbol{p}$) и приравниваем к нулю:

$$\sum_{x} \left[\nabla I \frac{\partial W}{\partial \boldsymbol{p}} \right]^T \left[I(W(\boldsymbol{x}; \boldsymbol{p_0})) + \nabla I \frac{\partial W}{\partial \boldsymbol{p}} \Delta \boldsymbol{p} - T(x) \right] = 0$$

Разрешая $\Delta\boldsymbol{p}$

- Решая $\Delta\boldsymbol{p}$:

$$\sum_x \left[\nabla I \frac{\partial W}{\partial \boldsymbol{p}}\right]^T \left[I(W(\boldsymbol{x};\boldsymbol{p_0})) + \nabla I \frac{\partial W}{\partial \boldsymbol{p}} \Delta\boldsymbol{p} - T(x)\right] = 0$$

- Получим:

$$\Delta\boldsymbol{p} = H^{-1} \sum_x \left[\nabla I \frac{\partial W}{\partial \boldsymbol{p}}\right]^T [T(x) - I(W(\boldsymbol{x};\boldsymbol{p_0}))]$$

где $H = \sum_x \left[\nabla I \frac{\partial W}{\partial \boldsymbol{p}}\right]^T \left[\nabla I \frac{\partial W}{\partial \boldsymbol{p}}\right]$

Интерпретация матрицы H для преобразований смещения

$$H = \sum_x \left[\nabla I \frac{\partial W}{\partial \boldsymbol{p}}\right]^T \left[\nabla I \frac{\partial W}{\partial \boldsymbol{p}}\right]$$

Вспомним:

1. $\nabla I = \begin{bmatrix} I_x & I_y \end{bmatrix}$
2. Для смещения, $\frac{\partial W}{\partial \boldsymbol{p}}(\boldsymbol{x}; \boldsymbol{p}) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

Поэтому,

$$H = \sum_x \left[\begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right]^T \left[\begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right]$$

$$= \boxed{\sum_x \begin{bmatrix} {I_x}^2 & I_x I_y \\ I_x I_y & {I_y}^2 \end{bmatrix}}$$

Это детектор угла Харриса

Интерпретация H-матрицы для аффинных преобразований

$$H = \sum_{\mathbf{x}} \begin{bmatrix} I_x^2 & I_x I_y & x I_x^2 & y I_x I_y & x I_x I_y & y I_x I_y \\ I_x I_y & I_y^2 & x I_x I_y & y I_y^2 & x I_y^2 & y I_y^2 \\ x I_x^2 & y I_x I_y & x^2 I_x^2 & y^2 I_x I_y & xy I_x I_y & y^2 I_x I_y \\ y I_x I_y & y I_y^2 & xy I_x I_y & y^2 I_y^2 & xy I_y^2 & y^2 I_y^2 \\ x I_x I_y & x I_y^2 & x^2 I_x I_y & xy I_y^2 & x^2 I_y^2 & xy I_y^2 \\ y I_x I_y & y I_y^2 & xy I_x I_y & y^2 I_y^2 & xy I_y^2 & y^2 I_y^2 \end{bmatrix}$$

Общий алгоритм итеративного KLT

Учитывая особенности детектора Harris:

$$\Delta \boldsymbol{p} = H^{-1} \sum_{x} \left[\nabla I \frac{\partial W}{\partial \boldsymbol{p}} \right]^{T} [T(x) - I(W(\boldsymbol{x}; \boldsymbol{p_0}))]$$

1. Инициализация $\boldsymbol{p_0}$ и $\Delta\boldsymbol{p}$.
2. Вычисление начальных шаблонов $T(x)$ для каждой особенности.
3. Трансформация особенностей изображении I с $W(\boldsymbol{x}; \boldsymbol{p_0})$.
4. Вычисление ошибки: $I(W(\boldsymbol{x}; \boldsymbol{p_0})) - T(x)$.
5. Вычисление градиента изображения $\nabla I = [I_x \quad I_y]$.
6. Вычисление якобиана преобразования $\frac{\partial W}{\partial \boldsymbol{p}}$.
7. Вычисление спуска по градиенту $\nabla I \frac{\partial W}{\partial \boldsymbol{p}}$.
8. Вычисление обратную матрицу Hessian H^{-1}
9. Вычисление изменения параметра $\Delta\boldsymbol{p}$
10. Обновление параметра $\boldsymbol{p_0} = \boldsymbol{p_0} + \Delta\boldsymbol{p}$
11. Повторить от 2 до 10 пока$\Delta\boldsymbol{p}$ маленькое.

KLT на нескольких кадрах

- Как только вы найдете преобразование для двух кадров, вы повторите этот процесс для каждой пары кадров
- Запустите детектор Харриса через каждые 15-20 кадров, чтобы найти новые особенности.

Проблемы

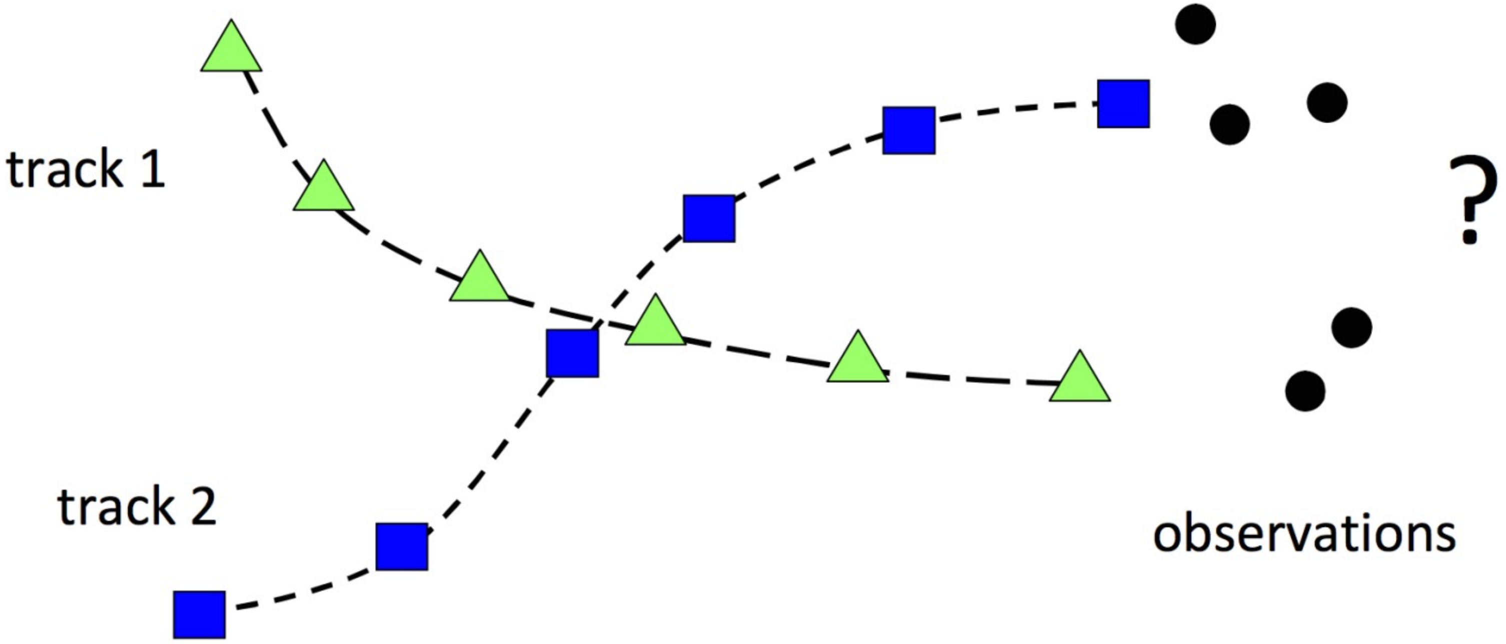
- Вопросы применения:
  - Размер окна
  - Маленькое окно более чувствительно к шуму и может пропускать большие движения (без пирамиды).
  - Большое окно с большей вероятностью пересекает границу окклюзии (и оно медленнее). 15x15 - 31x31 кажется типичным
- Взвешивание окна
  - Обычно применяют веса, чтобы центр был важнее (например, гауссово взвешивание)

Что будем изучать сегодня

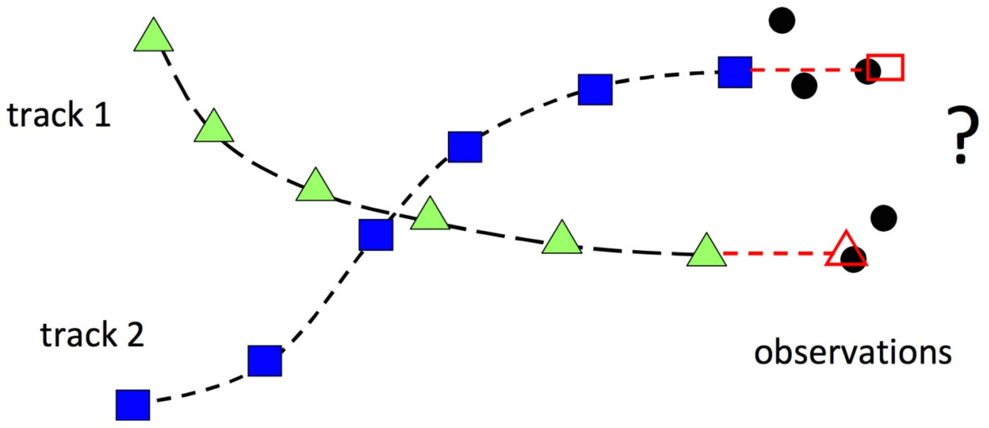
- Feature Tracking
- Simple KLT tracker
- 2D transformations
- Iterative KLT tracker
- Multi-target tracking

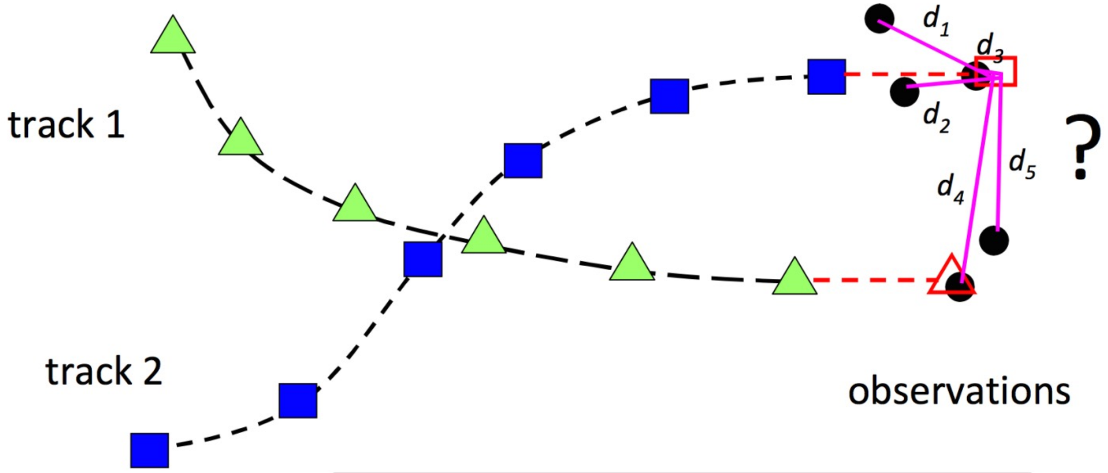
О чем задача multi-target tracking?

- Сопоставление данных
- Проблемы с назначением



О чем задача multi-target tracking?





Hungarian algorithm

- **Mathematical definition**

maximize: $$\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}x_{ij}$$

subject to:
$$\left.\begin{array}{l} \sum_j x_{ij} = 1; \quad i = 1,2,\ldots,n \\ \sum_i x_{ij} = 1; \quad j = 1,2,\ldots,n \\ x_{ij} \in \{0,1\} \end{array}\right\}$$

constraints that say
X is a permutation matrix

Where w is the affinity matrix and x is the assignments

Hungarian algorithm
finds the optimal assignment

Slide from Collins, PSU

Hungarian algorithm

	1	2	3	4	5
1	$\boxed{0.95}$	0.76	0.62	0.41	0.06
2	0.23	0.46	0.79	$\boxed{0.94}$	0.35
3	0.61	0.02	$\boxed{0.92}$	0.92	0.81
4	0.49	$\boxed{0.82}$	0.74	0.41	0.01
5	0.89	0.44	0.18	0.89	$\boxed{0.14}$

Greedy Solution

Score=3.77

	1	2	3	4	5
1	$\boxed{0.95}$	0.76	0.62	0.41	0.06
2	0.23	0.46	$\boxed{0.79}$	0.94	0.35
3	0.61	0.02	0.92	0.92	$\boxed{0.81}$
4	0.49	$\boxed{0.82}$	0.74	0.41	0.01
5	0.89	0.44	0.18	$\boxed{0.89}$	0.14

Optimal Solution

Score=4.26

Заключение

- Feature Tracking
- Simple KLT tracker
- 2D transformations
- Iterative KLT tracker
- Multi-target tracking