

# Обзор методов сопровождения объектов на видео

Сергей Носов

Летняя школа Интел в ННГУ по компьютерному зрению

8 июля 2019

Internet of Things Group

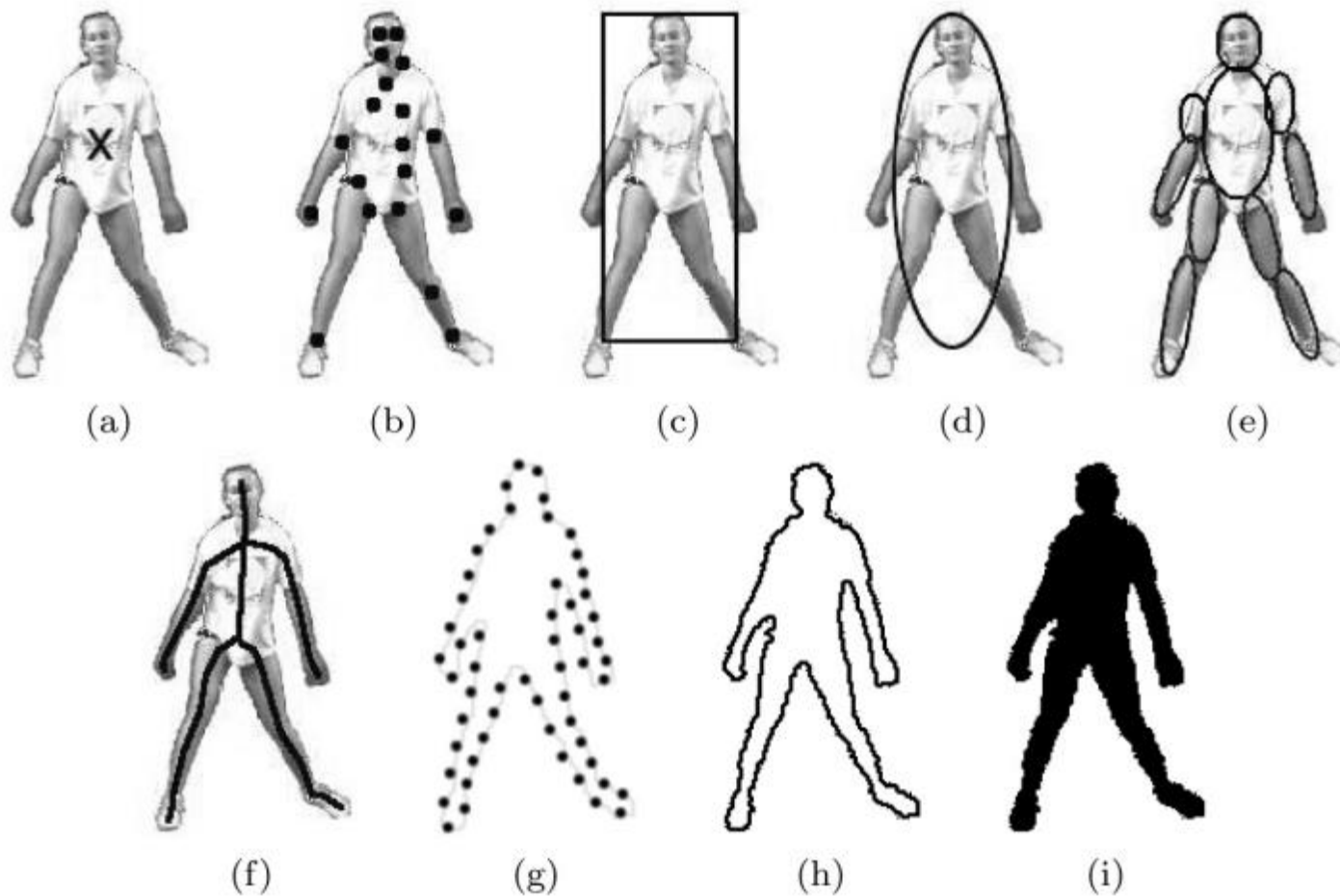
# Пример <https://www.youtube.com/watch?v=Tjx8BGoeZtI>



# Пишем свой трекер

1. Представление (модель) объекта
2. Поиск кандидатов и выбор лучшего
  - Генеративные
3. Сравнение кандидата с искомым объектом
  - Дискриминативные

# Представление объекта

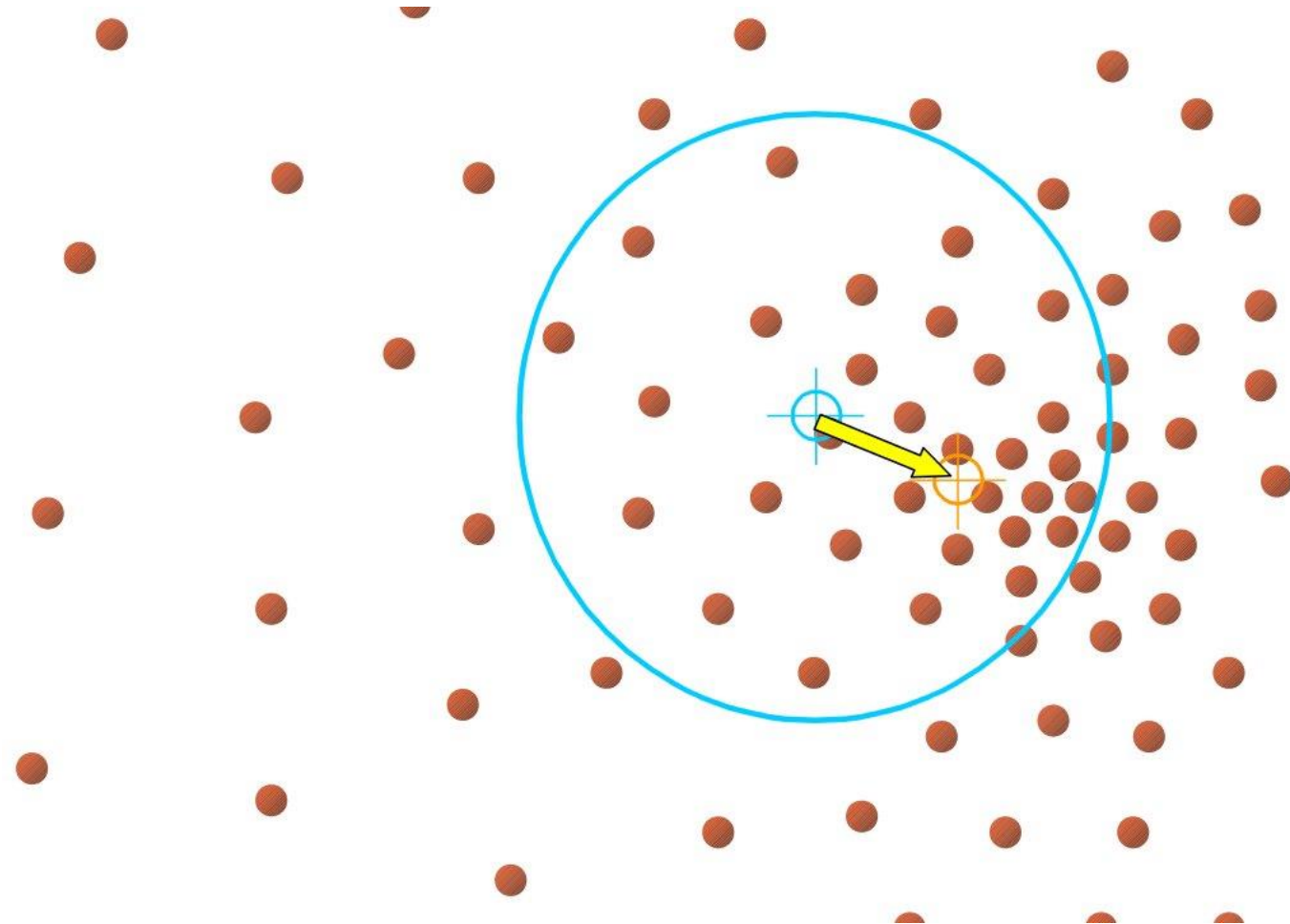


# Генеративные трекаеры

Предлагают процедуру поиска кандидатов и выбора среди них лучшего

- Mean-Shift
- Particle Filter
- Optical Flow
- etc.

# Mean-Shift



# Mean-Shift

$$m(x) = \frac{\sum_{x_i \in N(x)} K(x_i - x)x_i}{\sum_{x_i \in N(x)} K(x_i - x)}$$

$N(x)$  – множество «ближайших» к  $x$  кандидатов

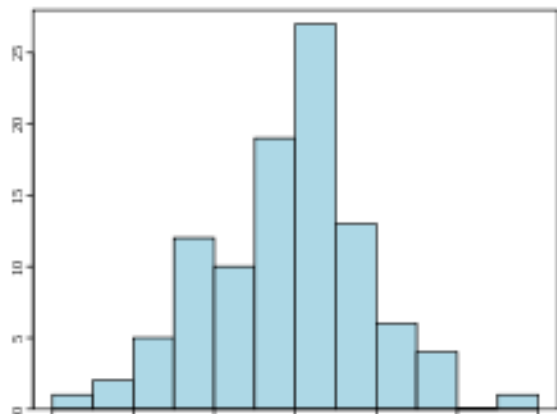
$K(x)$  – ядро, способ взвешивания кандидатов

Примеры ядер:

- Епанечникова  $K_e(x) = c (1 - |x|^*|x|)$ ,  $|x| \leq 1$
- Равномерное  $K_u(x) = c$ ,  $|x| \leq 1$
- Нормальное  $K_n(x) = c \exp(-0.5 |x|^*|x|)$

# BackProject и недостатки Mean Shift

- Классическая функция значения вероятности – BackProjection гистограммы



<https://ru.wikipedia.org/wiki/%D0%93%D0%B8%D1%81%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0>



[https://docs.opencv.org/master/d7/d00/tutorial\\_meanshift.html](https://docs.opencv.org/master/d7/d00/tutorial_meanshift.html)

- Недостатки

- Количество кандидатов растёт квадратично от стороны окна поиска
- Требуется несколько итераций для сходимости



# Particle filter

- Генерируются  $N$  кандидатов.
- Каждый кандидат может иметь нетривиальное состояние: координаты, размер (ширина/высота, scale), угол поворота, скорость
- При обработке очередного кадра, все кандидаты меняют состояние в соответствие с динамическим уравнением.
- К ним добавляется случайный гаусовский шум
- Обновленные кандидаты взвешиваются и выбирается результирующий кандидат (средний, лучший, и т.д.)
- "Плохие" кандидаты отбрасываются, вместо них добавляются "хорошие"

# Particle filter – пример

- Состояние кандидата ( $X, Y, VX, VY, W, H, s$ )
- Уравнения перехода  $X = X + VX, Y = Y + VY$
- Результирующим кандидатом объявляется средний.



# Systematic resampling

```
double draw = (double)rng / (double)particles_num;
for (size_t i = 0; i < particles_num; i++) {
    sum += weights_[i];
    while ( k / (double)particles_num + draw < sum)
        resampled_particles_[k++] = particles_[i];
}
assert(k == particles_num);
random_shuffle(resampled_particles_.begin(), resampled_particles_.end(), rng);
```

# Взвешивание кандидата

- Кросс-корреляция, разность квадратов и т.п.
- Сравнение гистограмм
- Любой классификатор, который можно себе позволить

# Разреженный оптический поток

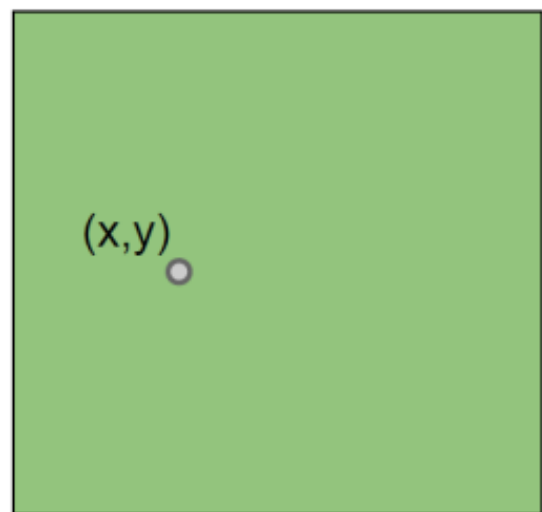


# Уравнение оптического потока

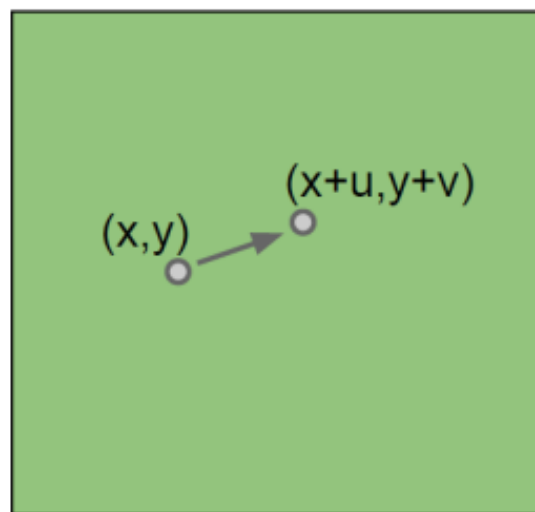
$I(x, y, t)$  – интенсивность точки  $(x, y)$  в момент времени  $t$

Предположим, что интенсивность точки в пространстве – *константа*, т.е.

$$I(x + u, y + v, t + 1) = I(x, y, t)$$



Frame  $i$



Frame  $i+1$

Используя разложение Тейлора  
 $I(x + u, y + v, t + 1) \sim I(x, y, t) + I_x u + I_y v + I_t$ ,  
получаем:

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

# Решение методом Лукаса-Канады

Для уравнения:

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

Предположим, что все точки в малой окрестности (например, 5x5), имеют один и тот же вектор смещения

Положим

$$A = \begin{pmatrix} I_x(x_1, y_1) & I_y(x_1, y_1) \\ \vdots & \vdots \\ I_x(x_n, y_n) & I_y(x_n, y_n) \end{pmatrix}, \quad b = \begin{pmatrix} -I_t(x_1, y_1) \\ \vdots \\ -I_t(x_n, y_n) \end{pmatrix}$$

Тогда можно решить системы методом наименьших квадратов

$$\left\| A \begin{pmatrix} u \\ v \end{pmatrix} - b \right\| \rightarrow \min, \quad \begin{pmatrix} u \\ v \end{pmatrix} = (A^T A)^{-1} (A^T b)$$

На практике метод вычисляется итеративно на пирамиде изображения

# Проблема малой/неподходящей апертуры

В определённых случаях система имеет множество решений

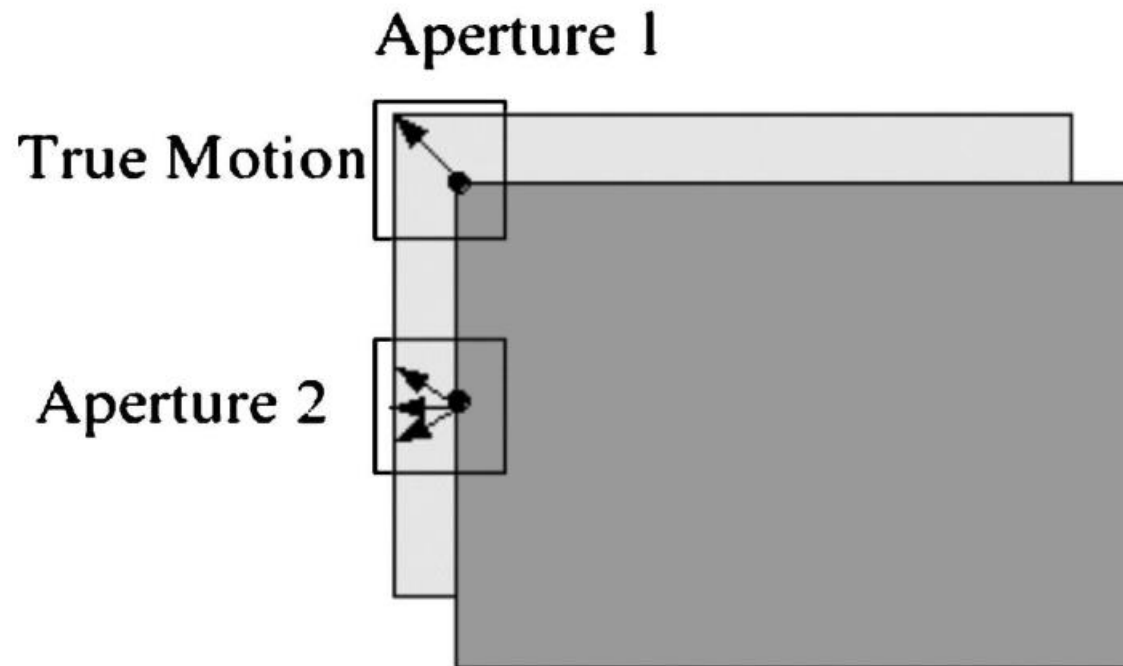
На каждой итерации LK вычисляет

$$\begin{pmatrix} u_{i+1} \\ v_{i+1} \end{pmatrix} = \left( A^T A \right)^{-1} A^T b_i$$

Проблемы возникают, когда

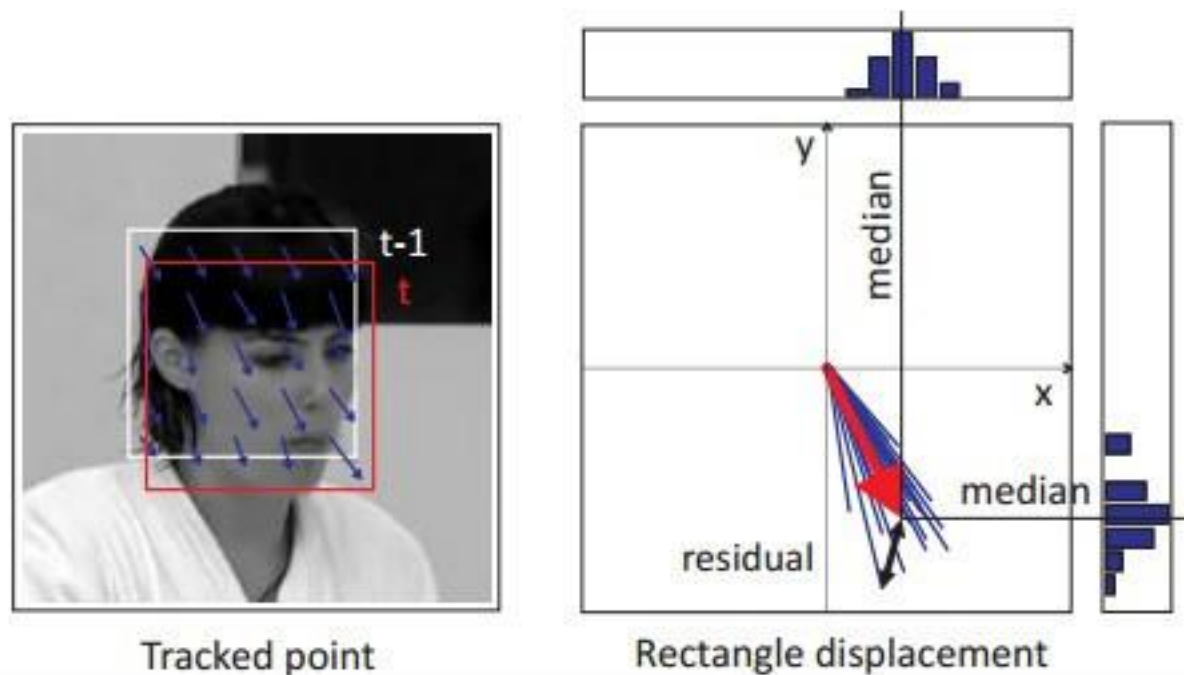
$$M = A^T A = \sum_{x,y} \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix}$$

плохо обусловлена





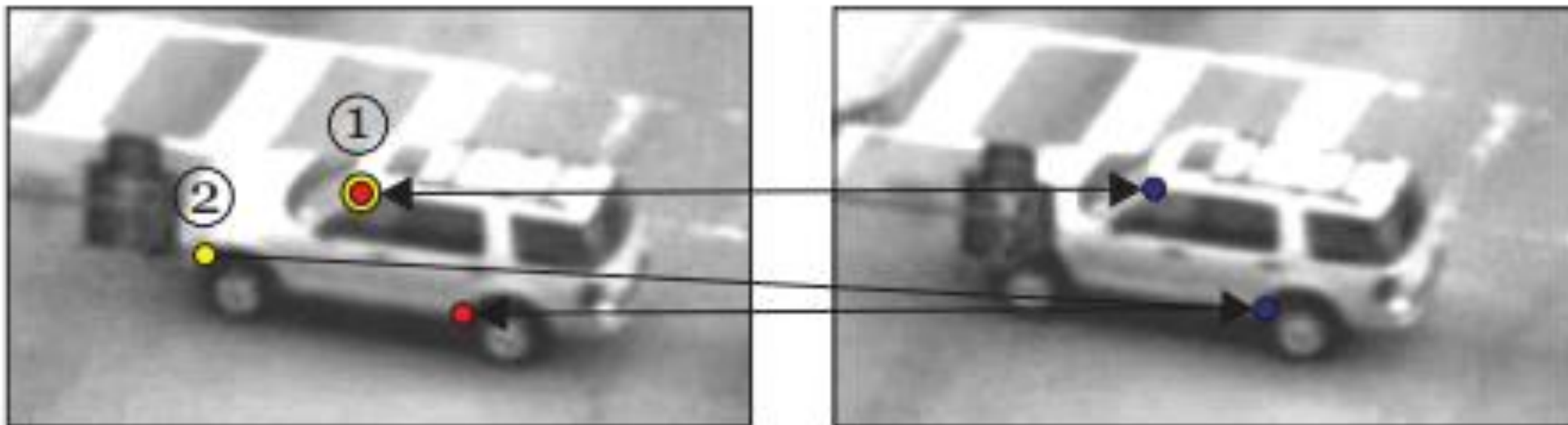
# Median Flow Tracker – первое приближение



zhou24388.blog.163.com

- Выбрать точки в прямоугольнике
- Вычислить для них optical flow (и отбросить "плохие")
- Взять медианные смещения по X и по Y

# Исчезновение точек

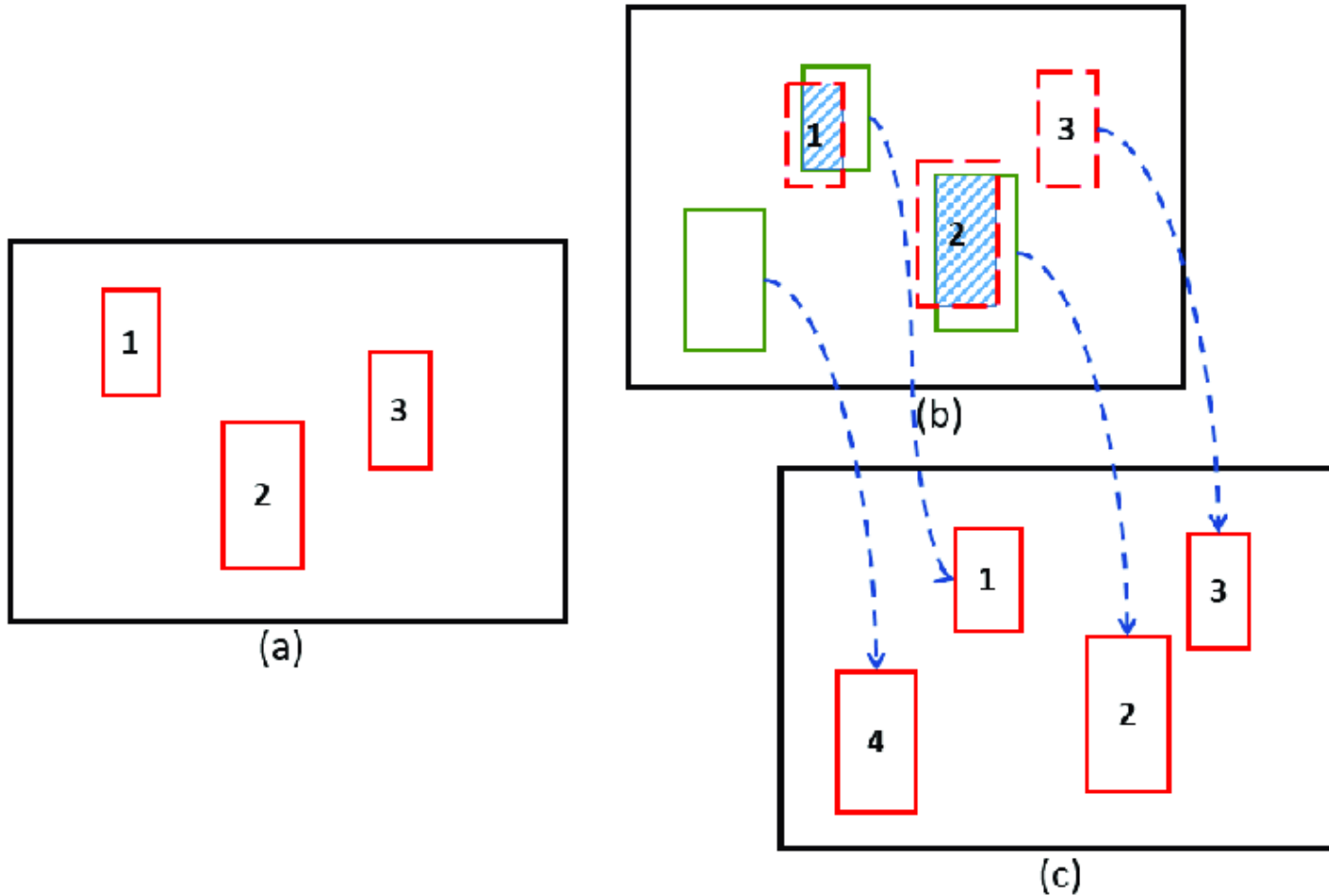


# Median Flow tracker - дополнение

- Можно использовать разные стратегии выбора точек для трекинга
- Можно фильтровать точки, для которых смещения от текущего кадра к следующему и от следующего к текущему не совпадают
- Можно фильтровать точки, для которых смещение «сильно» отличается от медианного
- Для оценки масштаба обычно оценивается отношение между всеми парами точек внутри объекта на текущем и следующем кадре
- Желательно иметь отдельную процедуру для детектирования срыва трекинга

А откуда вообще берётся  
объект для сопровождения?

# Tracking-by-detection



**Short name:** SRK\_ODESA  
**Benchmark:** [CVPR 2019 Tracking Challenge](#)  
**Description:** SRK\_ODESA follows tracking-by-detection approach, where special attention is paid to achieving better generalization at similarity estimation stage.

Prediction:

Faster R-CNN with ResNet101 backbone and 3x-Cascade, which was trained with MOT17 and CVPR19 detection sets, was turned into the prediction module by removing RPN block. The combination of predicts and public detects from the previous frame played the role of RPN output for the current frame. The output of regression and classification heads regarded as predicts was then mixed with the current frame public detects in order to get the final set of boundingboxes.

Similarity Estimation:

The object embedding employed in this solution was obtained by means of deep metric learning. The same model with embedding size of 128 was applied within SRK\_ODESA solution submitted to KITTI Pedestrian Tracking Benchmark. Neither MOT nor KITTI data were used for model training.

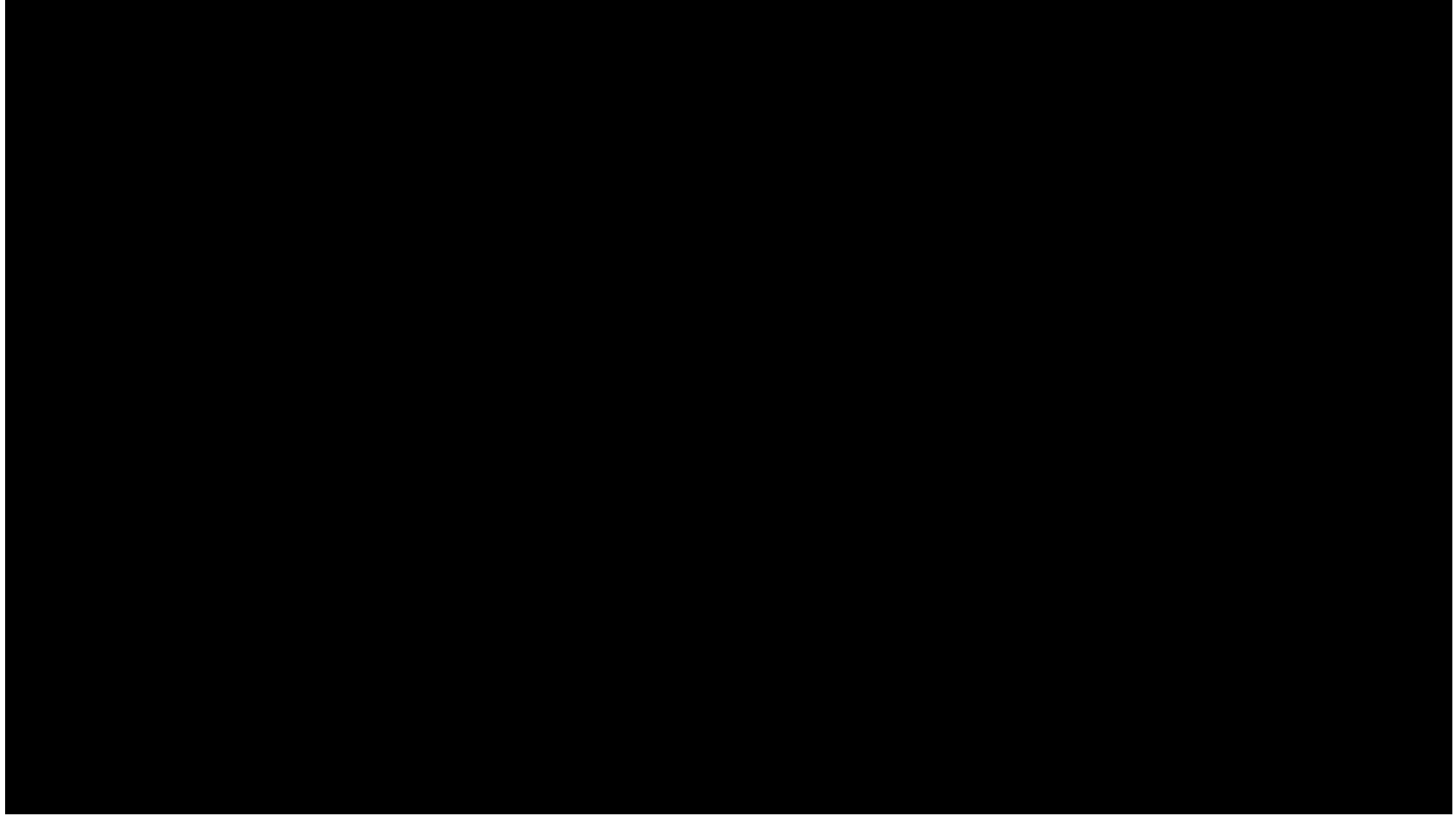
Matching:

Incoming detects corrected by Kalman filter were matched by means of Hungarian algorithm to the list of existing tracks. The history depth for the tracks was set to 100 frames.

Team Affiliation:

Samsung Research&Development Institute Ukraine (SRK)  
57 L'va Tolstogo Str., Kyiv 01032, Ukraine

<https://www.youtube.com/watch?v=1BY2CxiMYvQ>



# Пишем свой трекер – современный (2019) подход

## 1. Представление (модель) объекта

1. Прямоугольник
2. Маска объекта (см. [DAVIS challenge](#))

## 2. Поиск кандидатов и выбор лучшего

- DL детекторы (см. SSD, Faster-RCNN)
- Венгерский алгоритм, Multiple Hypothesis Tracking, и т.п.

## 3. Сравнение кандидата с искомым объектом

- DL метрики подобия



# Критерии оценки качества

- Visual Object Tracking (1 объект)
  - Precision = # правильных предсказаний / # предсказаний
  - Recall = # правильных предсказаний / # кадров с объектом
- Multiple Object Tracking (N объектов)

$$\text{MOTA} = 1 - \frac{\sum_t (m_t + f p_t + m m e_t)}{\sum_t g_t}, \quad (2)$$

where  $m_t$ ,  $f p_t$ , and  $m m e_t$  are the number of misses, of false positives, and of mismatches, respectively, for time  $t$ .

