

Детекторы и дескрипторы ключевых точек

Алексей Спижевой
Летняя школа 2016, Itseez/ННГУ
Июль 2016

Содержание

1. Особые точки, детекторы особенностей

- a. Детектор угловых точек Харриса
- b. Детектор FAST

2. Детекторы особых точек

- a. Детектор/детектор SURF
- b. Детектор BRIEF/ORB

3. Сшивка изображений

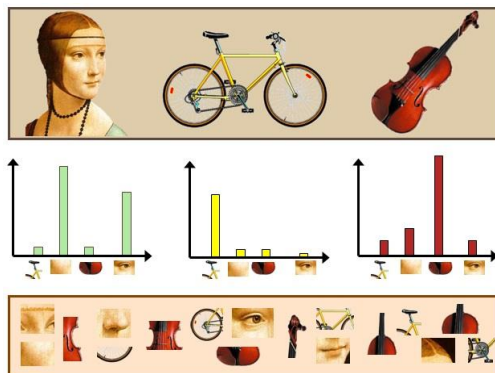
- a. Модель камеры, гомография

Зачем?

image mosaicing



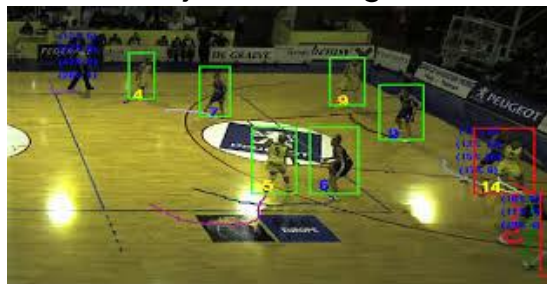
image classification



simultaneous localization and mapping (SLAM)



object tracking



3d reconstruction



Особые точки

Особая (ключевая) точка -- точка, обладающая высокой локальной информативностью.

Детектор -- это метод извлечения особых точек из изображения.

репрезентативность -- количество выдаваемых точек,

точность локализации,

повторяемость:

инвариантность к поворотам, сдвигам, изменениям яркости, ...

устойчивость к шумам, размытиям, ...

информативность окрестности,

вычислительная трудоемкость.

Дескриптор (описатель) -- вектор признаков, соответствующий особой точке и выделяющий её из остального множества точек.

размерность описателя,

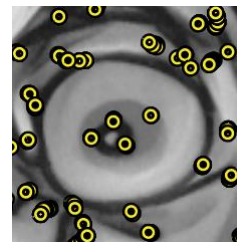
“тип” описателя, мера близости,

чувствительность дескриптора к преобразованиям:

инвариантность к поворотам, сдвигам, изменениям яркости, ...

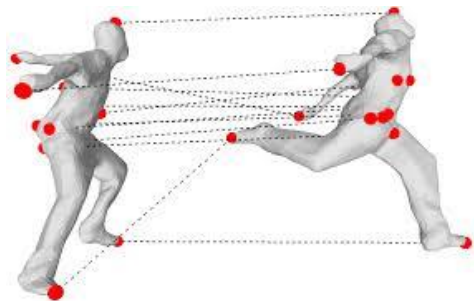
устойчивость к шумам, размытиям, ...

вычислительная трудоемкость.

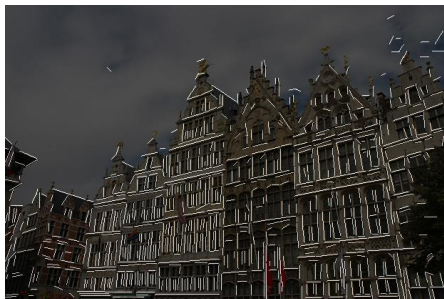


Особые точки

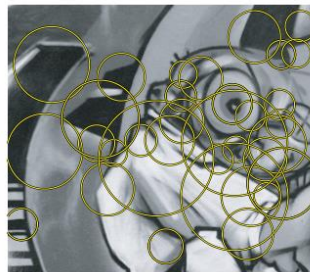
не только изображения,
но и объемные данные, видео



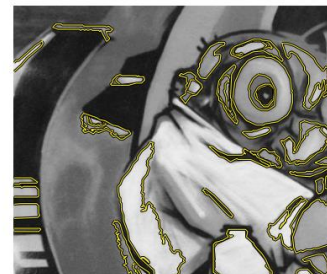
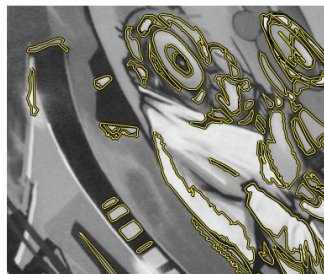
не только точки, но и ребра



дополнительно с точкам можно оценивать
масштаб, поворот, аффинное преобразование



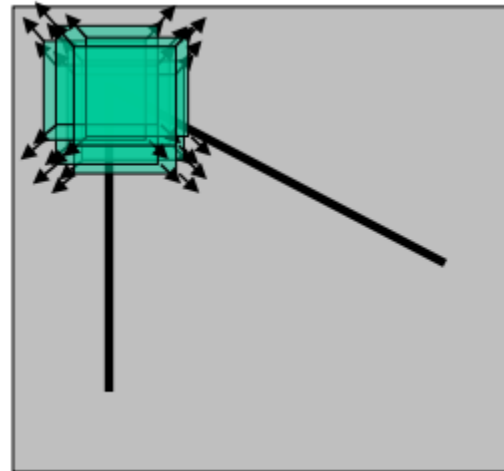
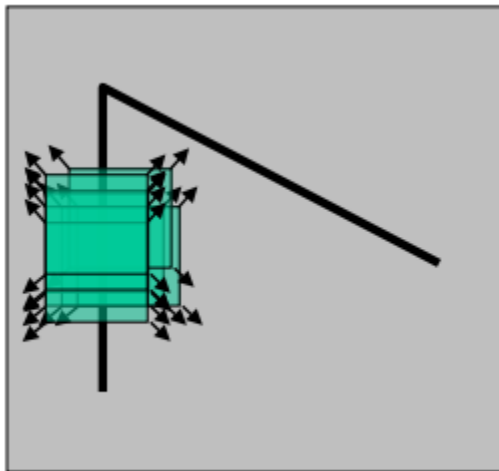
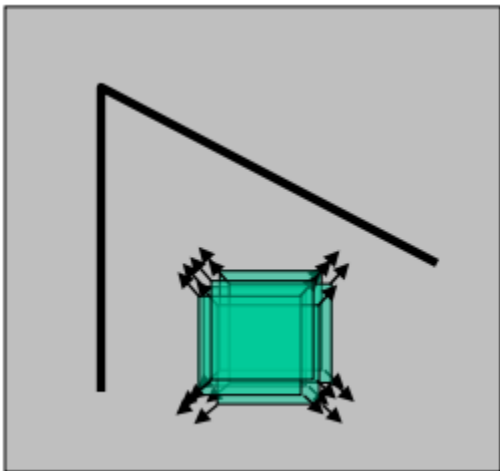
не только точки, но целые регионы



Детекторы особых точек

Детектор угловых точек Харриса

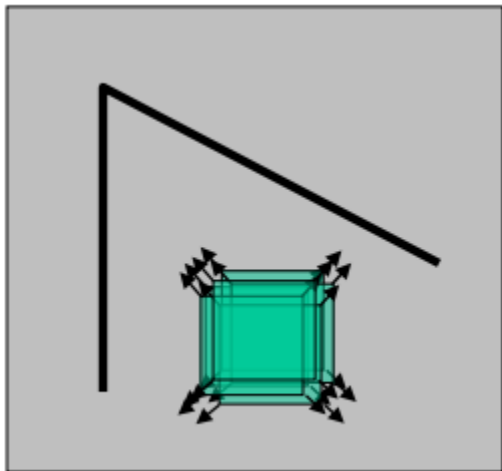
Попробуем найти **угловые точки**. То есть такие у которых окрестность сильно меняется, в какую бы сторону мы не сдвигались (локально).



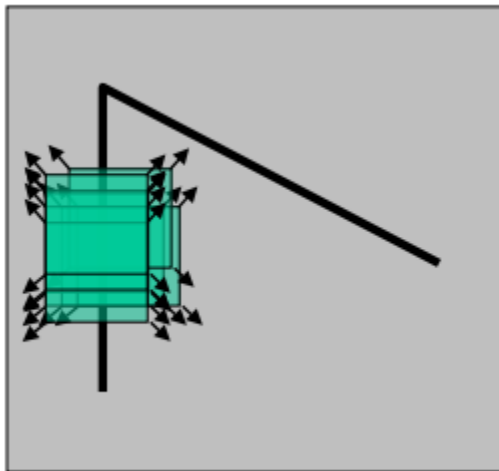
Детектор угловых точек Харриса

Попробуем найти **угловые точки**. То есть такие у которых окрестность сильно меняется, в какую бы сторону мы не сдвигались (локально).

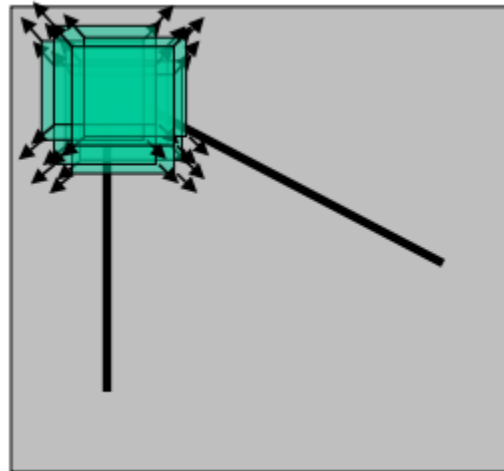
нет



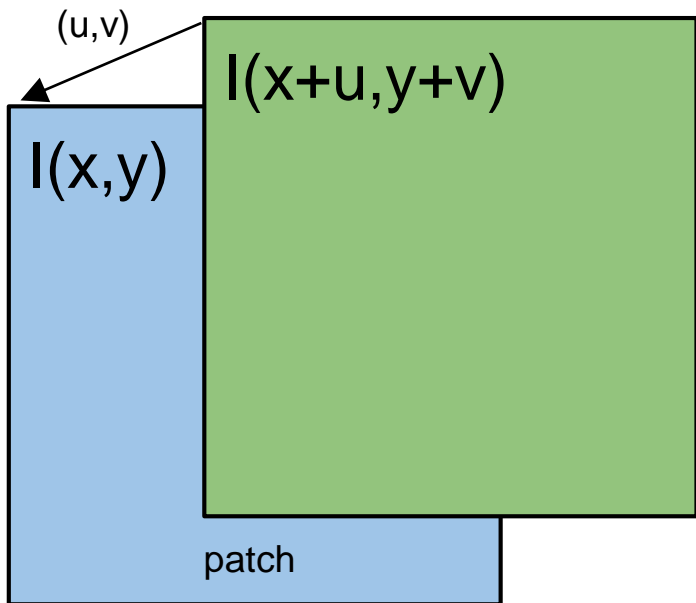
нет



да!



Детектор угловых точек Харриса



$$S(u, v) = \sum_{(x,y) \in \text{patch}} (I(x + u, y + v) - I(x, y))^2$$

-- непохожесть окон при сдвиге на (u,v)

$$I(x + u, y + v) \approx I(x, y) + I_x(x, y)u + I_y(x, y)v$$

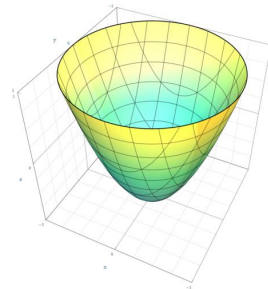
-- разложили в ряд Тейлора,
производные выше первой отбросили

Детектор угловых точек Харриса

$$S(u, v) = \sum_{x,y} (I_x(x, y)u + I_y(x, y)v)^2 = \sum_{x,y} \begin{pmatrix} u \\ v \end{pmatrix}^T \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}$$

$$S(u, v) = \mathbf{a}^T M \mathbf{a}$$

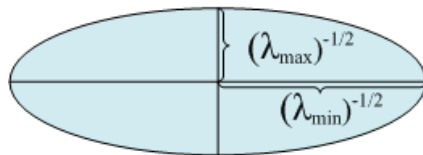
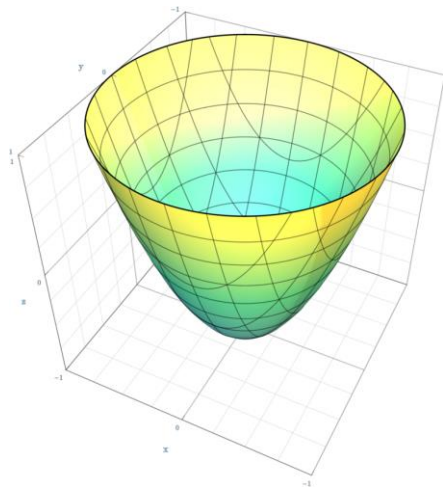
$$\mathbf{a} = \begin{pmatrix} u \\ v \end{pmatrix} \quad M = \sum_{x,y} \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix} \quad \text{-- структурный тензор}$$



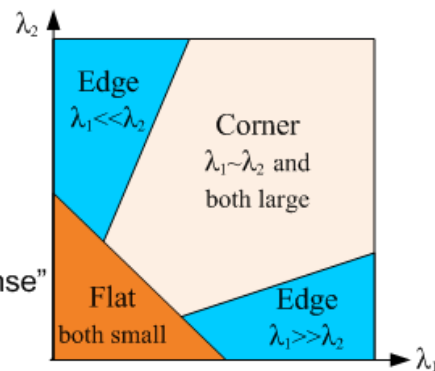
$$S(u, v) = \mathbf{a}^T M \mathbf{a} \in [\lambda_1, \lambda_2], \text{ if } |\mathbf{a}| = 1$$

- оба собственных числа близки к нулю -- однородная область
- одно большое, другое маленькое -- ребро
- **оба большие -- угловая точка**

Детектор угловых точек Харриса



Defines the
"Corner Response"



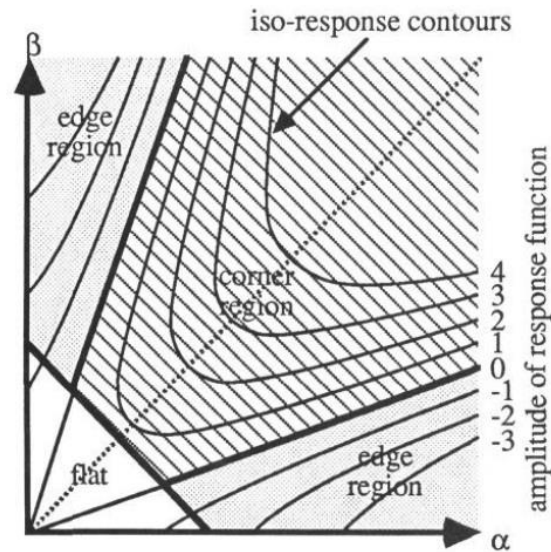
Т.о. нужно найти точки у которых **минимальное собственное число структурного тензора больше порога** (параметр метода).

Детектор угловых точек Харриса

Можно ли обойтись без нахождения собственных чисел, и определить является ли точка угловой быстрее? [Да! Harris score.](#)

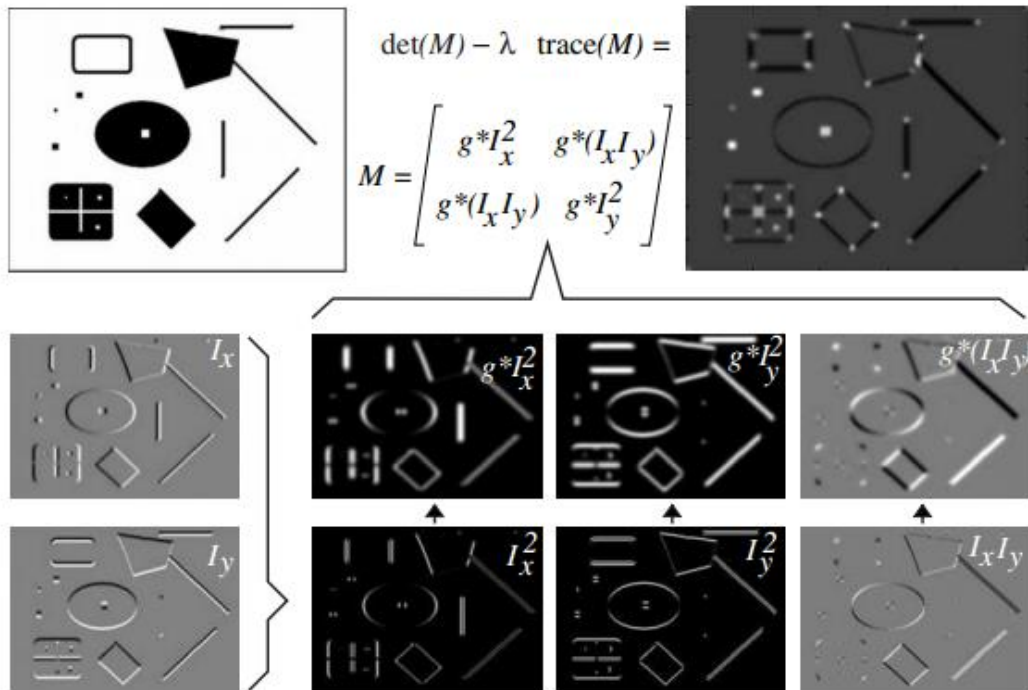
$$\text{cornerness}(x, y) = |M| - k * \text{trace}(M)^2$$

$$|M| - k * \text{trace}(M)^2 = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2$$



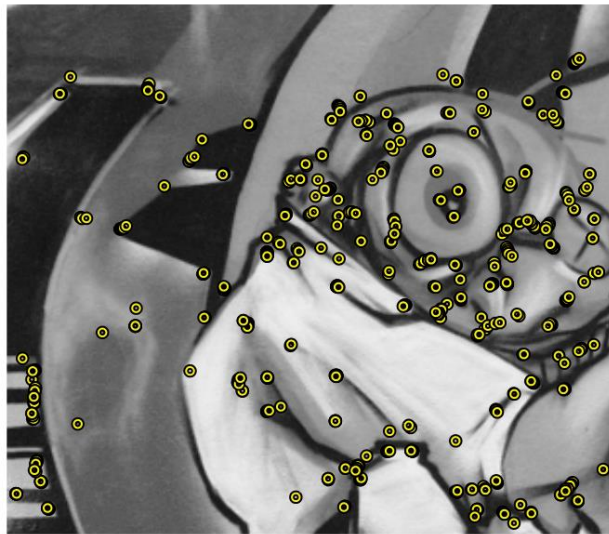
линии уровня

Детектор угловых точек Харриса

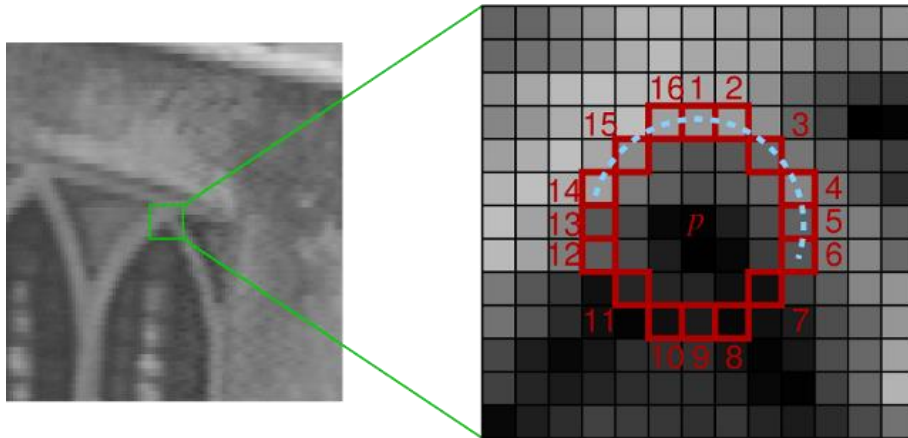


Детектор угловых точек Харриса

1. Вычислить производные (Sobel)
2. Подсчитать score (cornerness или мин. собств. число) для точек
3. Выбрать точки с высоким значением score
4. Non-maximum suppression: среди близко расположенных угловых точек оставить только наилучшую
5. Удалить слишком слабые точки (по сравнению с наилучшей)
6. Если точек слишком много, просто удалить слабейшие



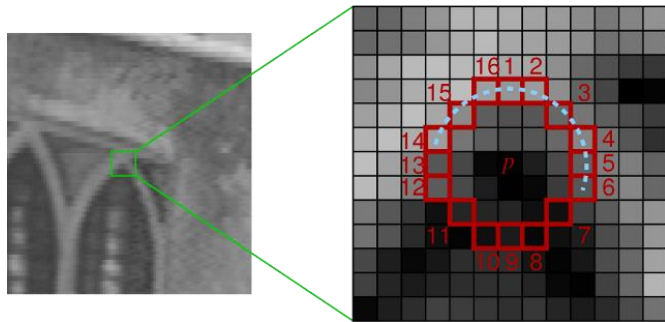
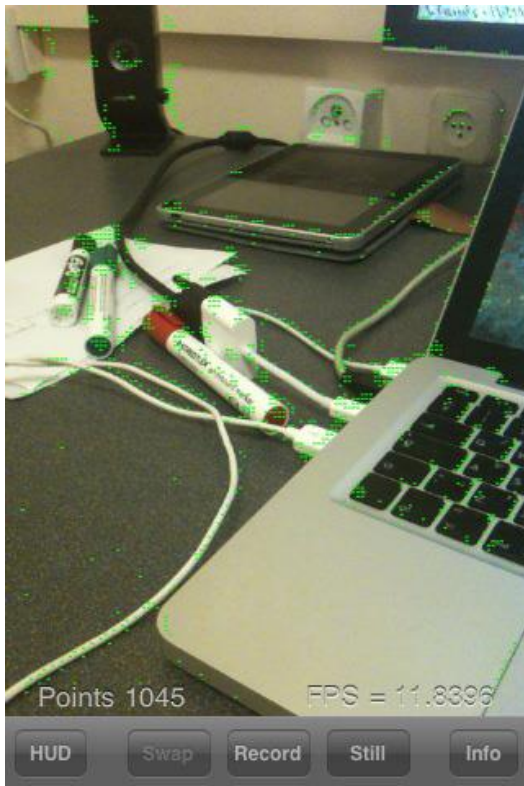
FAST детектор



$$S_{p \rightarrow x} = \begin{cases} d, & I_{p \rightarrow x} \leq I_p - t & \text{(darker)} \\ s, & I_p - t < I_{p \rightarrow x} < I_p + t & \text{(similar)} \\ b, & I_p + t \leq I_{p \rightarrow x} & \text{(brighter)} \end{cases}$$

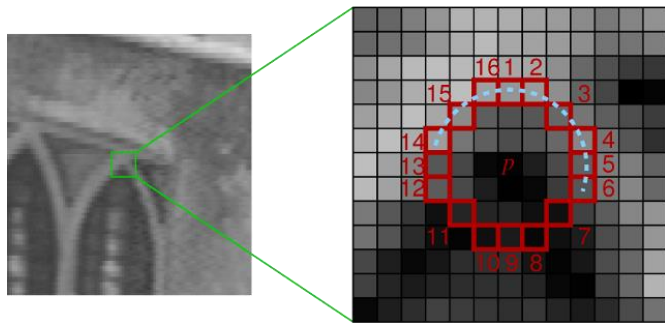
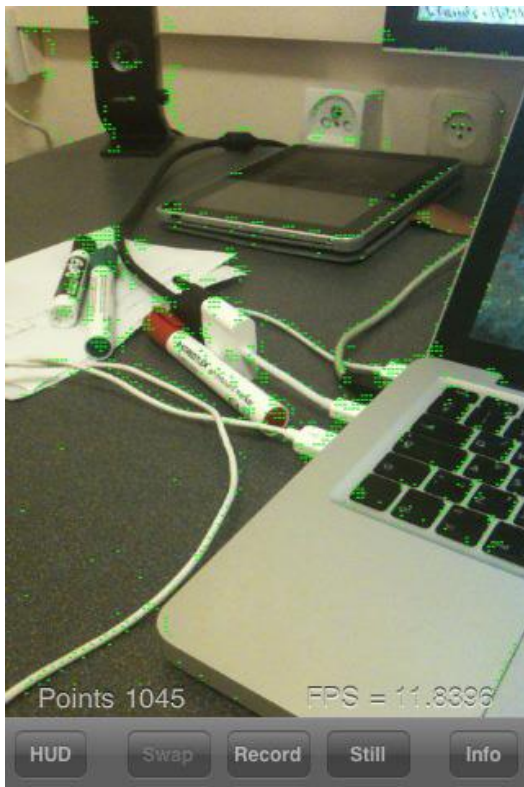
Если среди 16 соседей точки p , есть $\geq n$ смежных таких, что $I(x) > I(p)+t$ (или $I(x) < I(p)-t$), то точка считается угловой

FAST детектор



Необходимо ли проверять все $\geq n$ точек?

FAST детектор



Необходимо ли проверять все $\geq n$ точек? **Не всегда.**

High-speed test ($n \geq 12$)

- Можно сначала проверить точки 1,5,9,13. Из них три должны быть ярче (темнее) центральной.

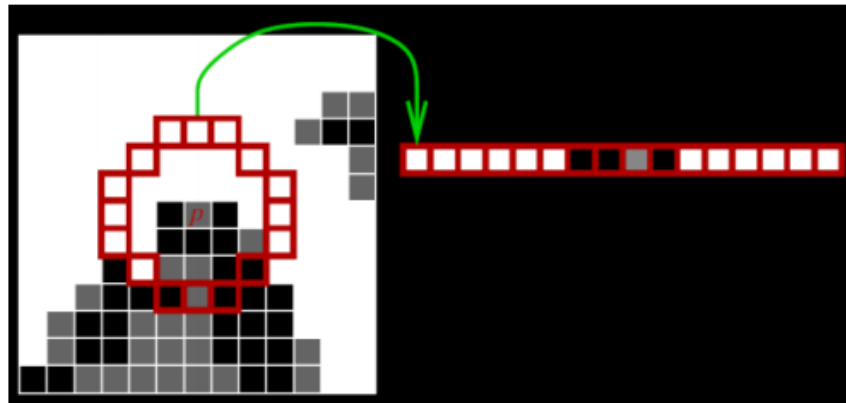
n влияет на повторяемость. Эксперименты показывают, что вариант с $n=9$ лучше.

FAST детектор

Можно попытаться найти последовательность сравнений с помощью методов машинного обучения

1. Собрать тренировочный набор изображений.
2. “Честно” найти ключевые точки.
3. Построить дерево решений (оно будет корректно классифицировать все точки тренировочной выборки).
4. Сконвертировать структуру дерева в C-код.

Если какое-то сравнение на пути классификации в дереве не встречается, то оно не делается. **Итого в среднем <3 сравнений с соседними точками на пиксель изображения**



Тренировочный набор данных: все пиксели всех изображений с метками классов (ключевая точка или нет).

Оценка ориентации

$$m_{pq} = \sum_{x,y} x^p y^q I(x, y)$$

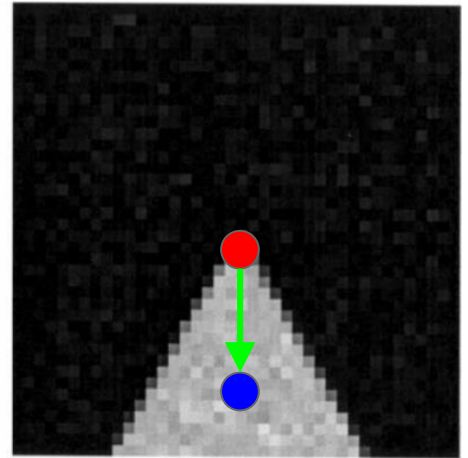
-- по патчу с центром в особой точке считаются моменты

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right)$$

-- находится "центр масс" патча

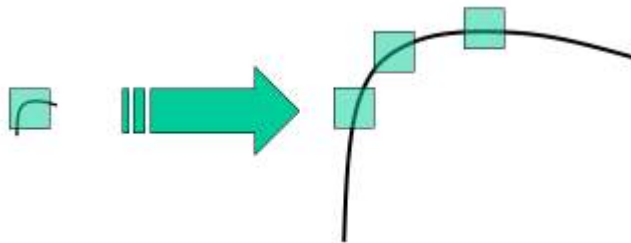
$$\theta = \text{atan2}(m_{01}, m_{10})$$

-- вычисляется угол от центра патча до центра масс



FAST + такой способ оценки в ориентации используется в детекторе/дескрипторе ORB

Изменение масштаба



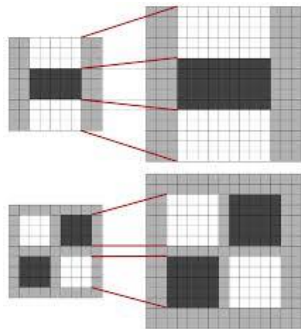
Детектор Харриса и FAST не инвариантны к масштабированию изображения!

Простейший вариант: детектировать ключевые точки на изображениях разного размера

Дескрипторы особых точек

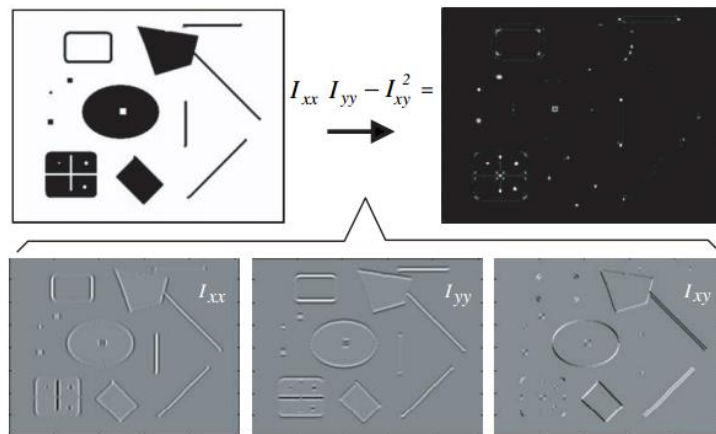
Детектор/дескриптор SURF

$$H = \begin{bmatrix} I_{xx}(\mathbf{x}, \sigma_D) & I_{xy}(\mathbf{x}, \sigma_D) \\ I_{xy}(\mathbf{x}, \sigma_D) & I_{yy}(\mathbf{x}, \sigma_D) \end{bmatrix} \quad \text{-- в точках } \mathbf{x}, \mathbf{y} \text{ изображения строится матрица Гессе для разных масштабов } \sigma$$

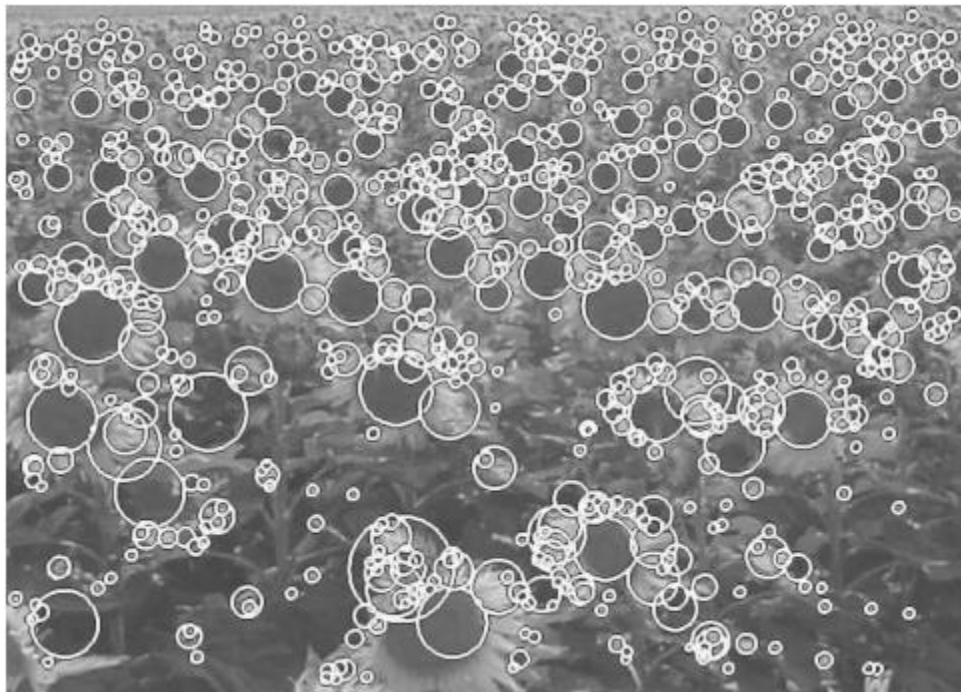


-- вторые производные аппроксимируются с помощью линейных фильтров следующего вида

ключевые точки соответствуют локальным максимумам Гессиана $\det(H(\mathbf{x}, \mathbf{y}, \sigma))$



Детектор/дескриптор SURF

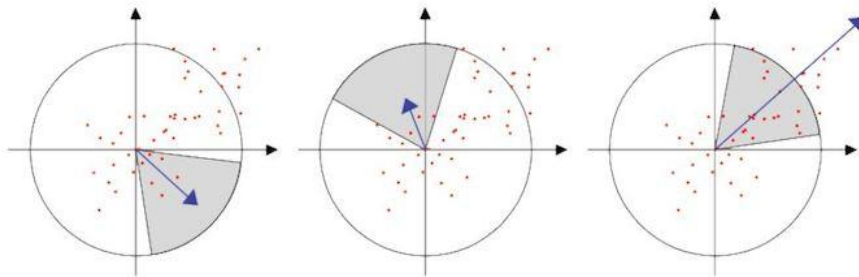


Детектор/дескриптор SURF

в окрестности особой точки в разных
позициях применяются вейвлеты Хаара

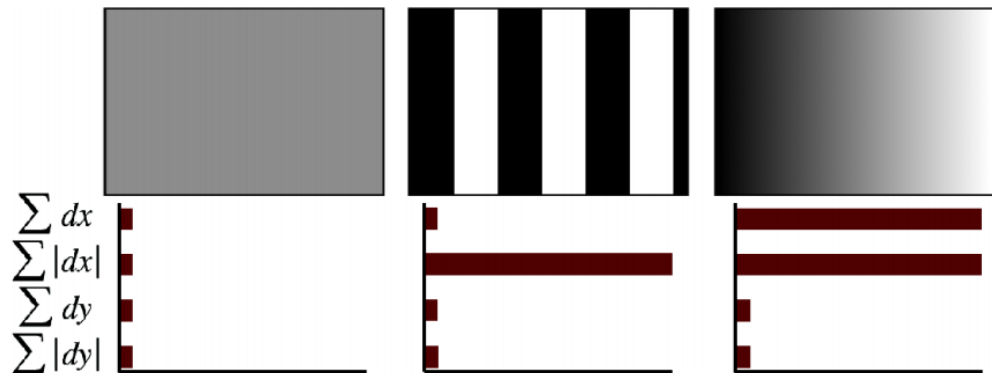
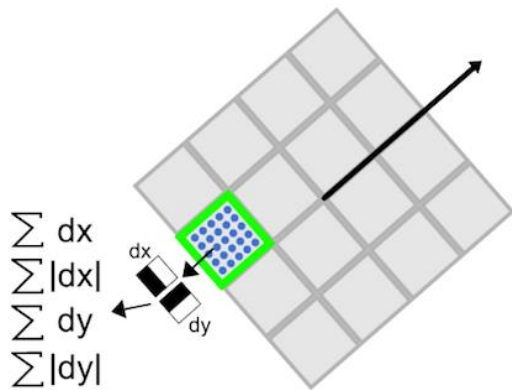


-- они вычисляют оценку
направления вектора градиента



-- доминирующее направление выбирается
с помощью сканирования сектором $\pi/3$

Детектор/дескриптор SURF



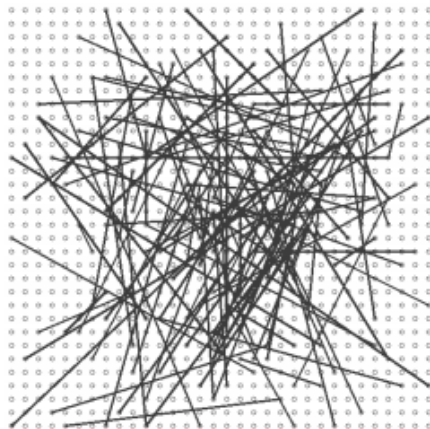
итого размерность дескриптора 64 (128) значений типа float

Дескриптор BRIEF

распределение бинарных тестов
(подобрано экспериментально,
распределение точек -- нормальное):

$$\tau(\mathbf{p}; \mathbf{x}, \mathbf{y}) := \begin{cases} 1 & \text{if } \mathbf{p}(\mathbf{x}) < \mathbf{p}(\mathbf{y}) \\ 0 & \text{otherwise} \end{cases}$$

$$f_{n_d}(\mathbf{p}) := \sum_{1 \leq i \leq n_d} 2^{i-1} \tau(\mathbf{p}; \mathbf{x}_i, \mathbf{y}_i)$$



- Типичный размер патча 32x32
- Изображение предварительно сглаживается (обычное усреднение или фильтр Гаусса)
- Типичное число тестов: 128, 256, 512 (т.е. 16, 32, 64 байта)
- Дескриптор бинарный, сравнение с помощью расстояния Хэмминга

Сшивка фотографий

Общая идея



(a) Image 1



(b) Image 2

Общая идея



(a) Image 1



(b) Image 2



(c) SIFT matches 1



(d) SIFT matches 2

Общая идея



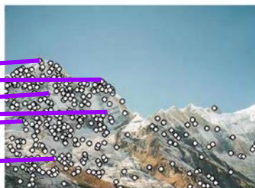
(a) Image 1



(b) Image 2



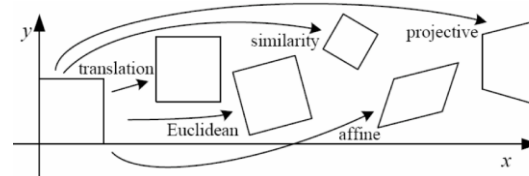
(c) SIFT matches 1



(d) SIFT matches 2

Общая идея

Базовые преобразования плоскости:



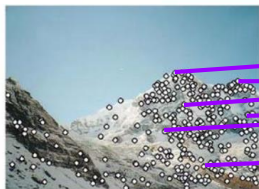
Name	Matrix	# D.O.F.	Preserves:	Icon
translation	$\begin{bmatrix} I & t \end{bmatrix}_{2 \times 3}$	2	orientation + ...	
rigid (Euclidean)	$\begin{bmatrix} R & t \end{bmatrix}_{2 \times 3}$	3	lengths + ...	
similarity	$\begin{bmatrix} sR & t \end{bmatrix}_{2 \times 3}$	4	angles + ...	
affine	$\begin{bmatrix} A \end{bmatrix}_{2 \times 3}$	6	parallelism + ...	
projective	$\begin{bmatrix} \tilde{H} \end{bmatrix}_{3 \times 3}$	8	straight lines	



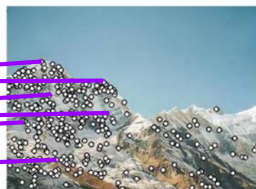
(a) Image 1



(b) Image 2



(c) SIFT matches 1



(d) SIFT matches 2



(e) Images aligned according to a homography

$$x' = \frac{h_{00}x + h_{01}y + h_{02}}{h_{20}x + h_{21}y + h_{22}} \quad \text{and} \quad y' = \frac{h_{10}x + h_{11}y + h_{12}}{h_{20}x + h_{21}y + h_{22}}$$

Сшивка двух изображений

Вход: две фотографии I_1, I_2

Выход: панорама

Алгоритм:

1. Найти ключевые точки на I_1, I_2
2. Вычислить описатели ключевых точек
3. Найти соответствия между точками
4. Оценить преобразование гомографии

H^{12} между двумя изображениями по соответствиям

$$\begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} \simeq H^{12} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} = \begin{pmatrix} H_{1\bullet}^{12} p_1 \\ H_{2\bullet}^{12} p_1 \\ H_{3\bullet}^{12} p_1 \end{pmatrix}$$

5. Наложить одно изображение на другое с учетом H^{12}

$$u_2 = \frac{H_{1\bullet}^{12} p_1}{H_{3\bullet}^{12} p_1} \quad v_2 = \frac{H_{2\bullet}^{12} p_1}{H_{3\bullet}^{12} p_1}$$

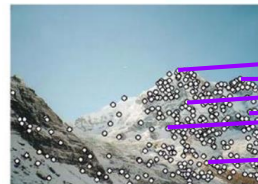
Сделанное предположение: гомография существует



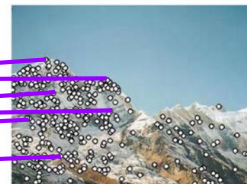
(a) Image 1



(b) Image 2



(c) SIFT matches 1



(d) SIFT matches 2



(e) Images aligned according to a homography

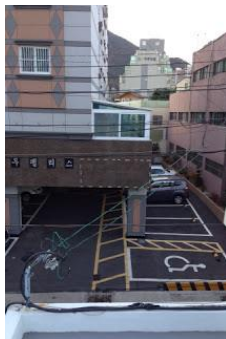
Результат сшивки двух изображений



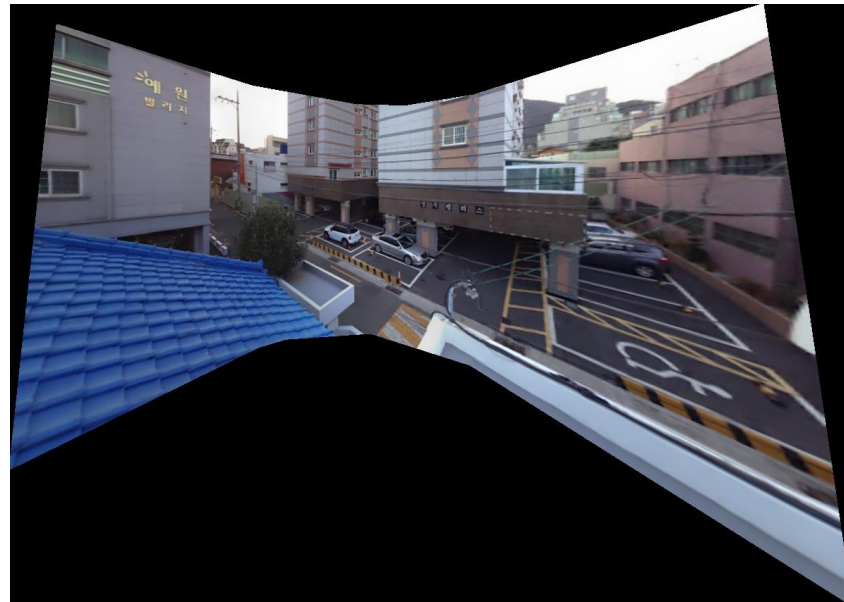
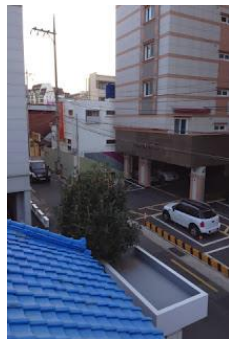
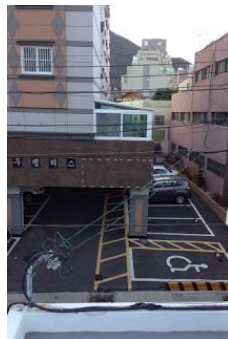
Результат сшивки двух изображений



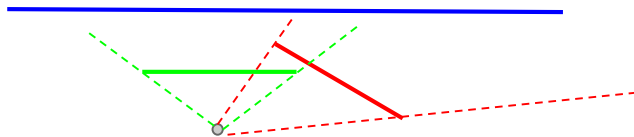
Результат сшивки >2 изображений



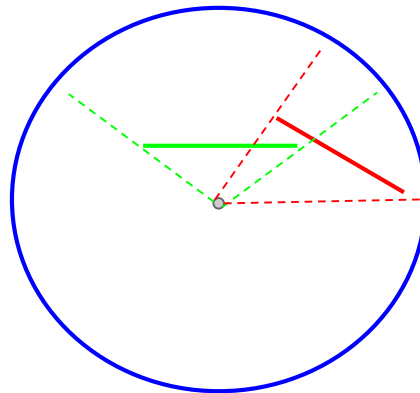
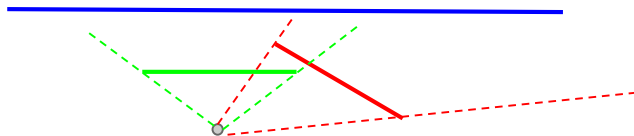
Результат сшивки >2 изображений



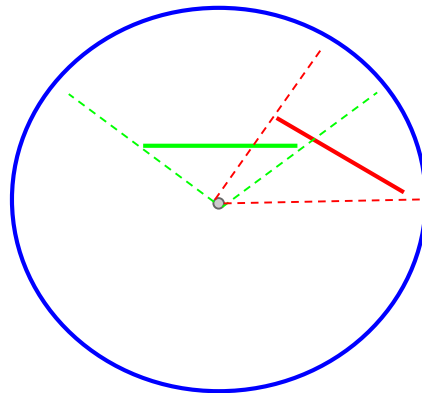
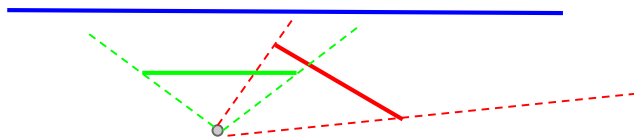
Можно лучше?



Можно лучше



Можно лучше



- Гомографии используются для извлечения поворотов между камерами (для этого надо знать параметры камер)

$$H_{12} = K_2 R_{12} K_1^{-1}$$

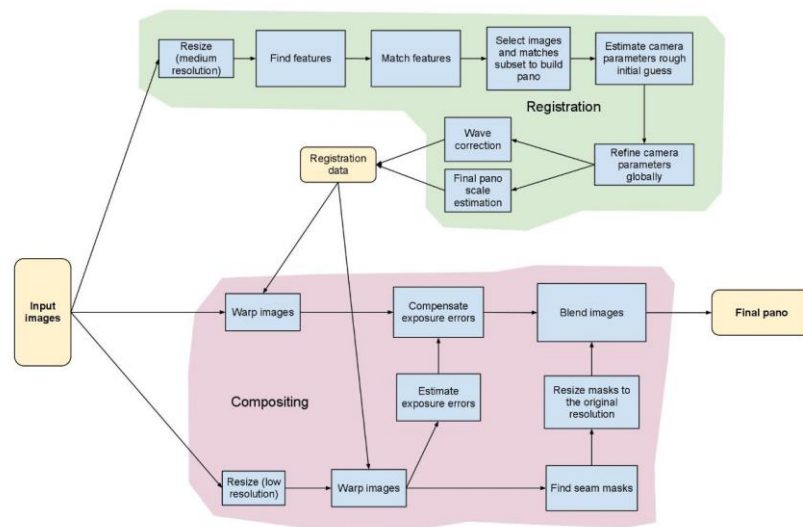
- Для проектирования используется не плоскость, а, например, поверхность сферы



Stitching модуль OpenCV

- Autocalibration
- Поиск швов
- Выравнивание яркости в изображениях
- Multi-band blending
- Поддержка различных детекторов и дескрипторов
- Planar, cylindrical, spherical warping
- Etc

Приложения **cpp-example_stitching** и **cpp-example-stitching_detailed**



Классификация изображений

Модель Bag-of-Words

Изначально модель BoW применялась для классификации текстов -- тексты кодируются гистограммой частот встречаемости слов из словаря

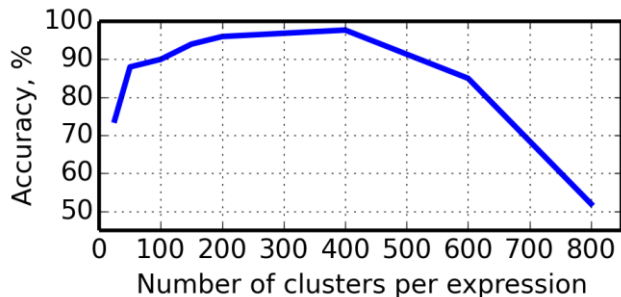
Модель используется для построения описателей изображений

Ключевая идея -- кодирование изображений на основе частот встречаемости “слов”

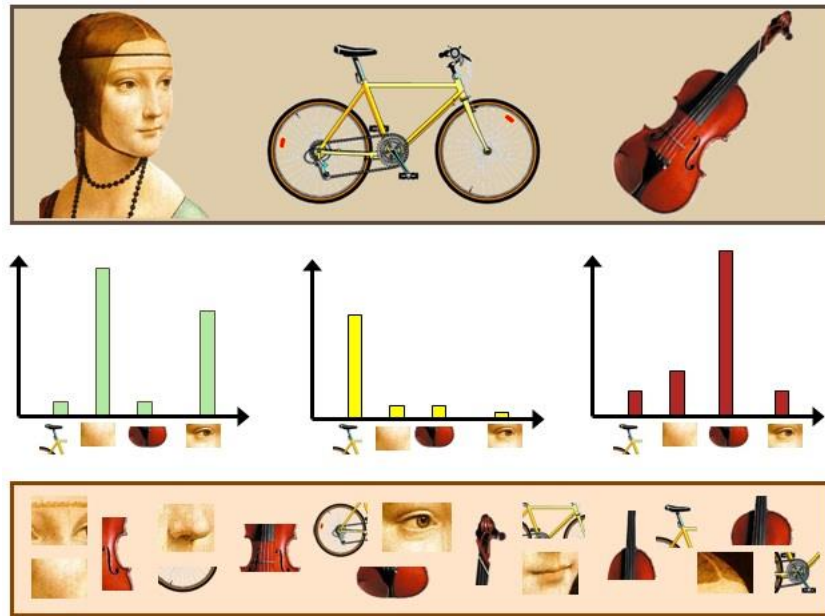
Модель Bag-of-Words

Алгоритм построения словаря:

1. Найти ключевые точки для всех изображений/видео тренировочной выборки (Harris, FAST, SURF, ...)
2. Вычислить описатели для всех ключевых точек (SURF, ORB, LBP, HOG, ...)
3. Среди всех описателей выделить кластеры (k-средних и его модификации)
4. Центры кластеров образуют словарь V



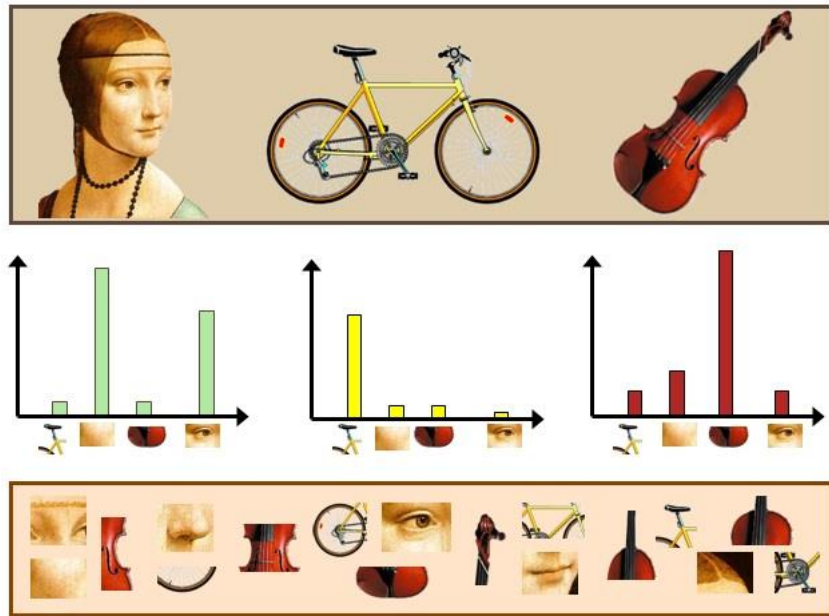
-- число кластеров имеет значение



Модель Bag-of-Words

Построение описателей:

1. Найти ключевые точки на входном изображении/видео
2. Вычислить описатели для найденных ключевых точек
3. Для описателя каждой ключевой точки **найти ближайшее "слово"** из словаря V (1-nn, k-d trees, LSH, ...)
4. Строится гистограмма встречаемости "слов"
5. Отнормированная гистограмма используется в качестве вектора признаков изображения/видео



размерность вектора признаков
фиксирована и равна $|V|$ -- удобно при
работе с видео переменной длины

пространственная информация при построении
гистограммы теряется -- гистограммы можно
строить для изображения по частям

Глубокое обучение

Крайне популярное направление машинного обучения, интенсивно развивается

SOTA результаты в классификации изображений, детектировании объектов

До определенной степени позволяет автоматизировать задачу выбора признаков (за счет их обучения)

Traditional Pattern Recognition: Fixed/Handcrafted Feature Extractor



Mainstream Modern Pattern Recognition: Unsupervised mid-level features



Deep Learning: Representations are hierarchical and trained



Литература

1. Bay, Herbert, Tinne Tuytelaars, and Luc Van Gool. "Surf: Speeded up robust features." *Computer vision–ECCV 2006*. Springer Berlin Heidelberg, 2006. 404-417.
2. Rosten, Edward, Reid Porter, and Tom Drummond. "Faster and better: A machine learning approach to corner detection." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 32.1 (2010): 105-119.
3. Rublee, Ethan, et al. "ORB: an efficient alternative to SIFT or SURF." *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011.
4. Tuytelaars, Tinne, and Krystian Mikolajczyk. "Local invariant feature detectors: a survey." *Foundations and Trends® in Computer Graphics and Vision* 3.3 (2008): 177-280.
5. Liu, Jialu. "Image retrieval based on bag-of-words model." *arXiv preprint arXiv:1304.5168* (2013).
6. Brown, Matthew, and David G. Lowe. "Automatic panoramic image stitching using invariant features." *International journal of computer vision* 74.1 (2007): 59-73.
7. Szeliski, Richard. "Image alignment and stitching: A tutorial." *Foundations and Trends® in Computer Graphics and Vision* 2.1 (2006): 1-104.
8. [Talks by Yann LeCun](#) -- overview/introductory deep learning materials
9. [Are we there yet?](#) -- list of state-of-the-art results for computer vision tasks

Спасибо за внимание!

контакты: alexey.spizhevoj@gmail.com

Доп. материалы

FAST детектор

Non-maximum suppression: среди близко расположенных угловых точек оставить только наилучшую

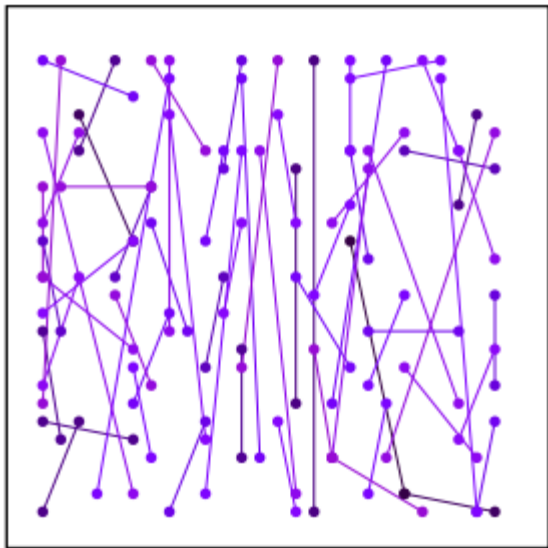
Для этого нужно ввести способ сравнения угловых точек:

$$V = \max \left(\sum_{x \in S_{\text{bright}}} |I_{p \rightarrow x} - I_p| - t, \sum_{x \in S_{\text{dark}}} |I_p - I_{p \rightarrow x}| - t \right)$$

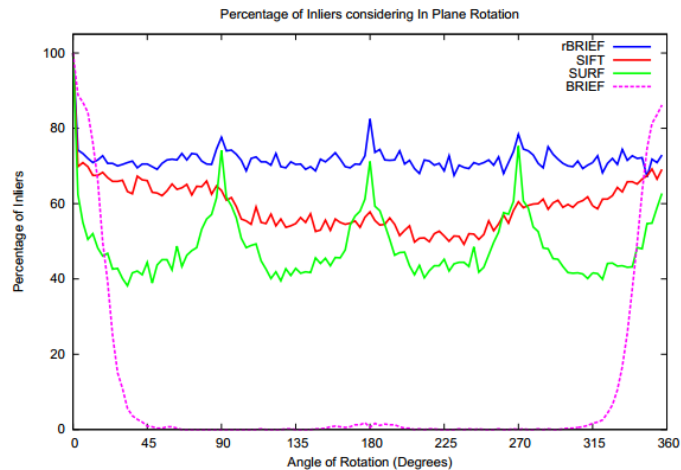


Дескриптор rBRIEF / ORB

набор бинарных тестов можно
“выучить”



1. Подготовить тренировочную выборку
2. Сгенерировать T -- все допустимые пары сравнений (всего порядка 200k)
3. $R := \{\}$
4. Выбрать 256 тестов:
 - а. Взять тест x из T с наибольшей выборочной дисперсией
 - б. Если x не коррелирует с тестами из R , то $R := R \cup \{x\}$



Гомография

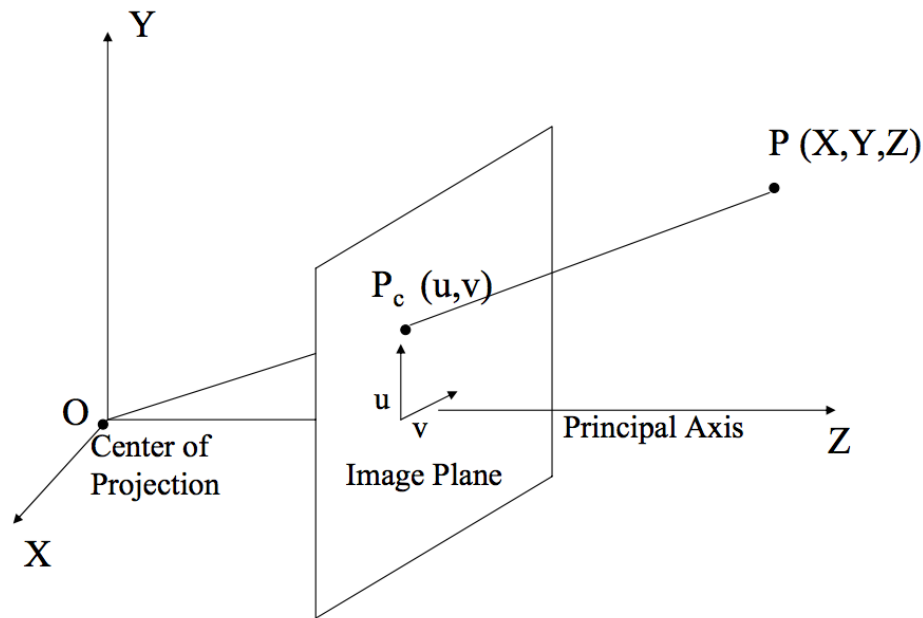
Проективное пространство P^n -- пространство лучей в R^{n+1} ,
т.е. $P^n = (R^{n+1} \setminus \{0\}) / \sim$

Луч задается любой точкой, лежащей на нем $(x_1, x_2, \dots, x_{n+1})$ -- **однородные координаты**

Гомография H -- обратимое преобразование проективного пространства, переводящее прямые в прямые.

В случае P^2 (проективная плоскость), H задается невырожденной матрицей 3×3 -- линейное преобразование в однородных координатах.

Геометрическая модель камеры



$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} f_x \frac{X}{Z} \\ f_y \frac{Y}{Z} \end{pmatrix} + \begin{pmatrix} c_x \\ c_y \end{pmatrix}$$

$$\lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

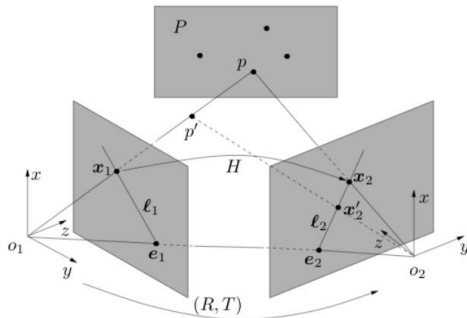
$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \simeq K \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

$$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \quad \text{-- матрица внутренних параметров камеры}$$

В каком случае гомография существует?

$$\begin{aligned}P_2 &= R_{12}P_1 \\p_1 &\simeq K_1P_1 \\p_2 &\simeq K_2P_2 \\p_2 &\simeq K_2R_{12}K_1^{-1}p_1 = H_{12}p_1\end{aligned}$$

-- если камеры связаны поворотом,
то существует гомография между
двумя проективными
пространствами!



В общем случае при произвольном
движении камеры гомография не
существует. Но существует в случае
плоской сцены!

Как искать гомографию по соответствиям?

метод Direct Linear Transformation

$$p_2^i \simeq H_{12} p_1^i$$

$$\lambda^i p_2^i = H_{12} p_1^i$$

Как искать гомографию по соответствиям?

метод Direct Linear Transformation

$$p_2^i \simeq H_{12} p_1^i$$

$$\lambda^i p_2^i = H_{12} p_1^i$$

-- домножим векторно на p_2^i

$$[a]_{\times} = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix}$$

$$[p_2^i]_{\times} H_{12} p_1^i = 0$$

-- система линейных уравнений относительно H_{12}
(достаточно 4-х соответствий)