

## **Инвариантные алгоритмы сопоставления точечных особенностей на изображениях.**

Автор:  
Гаганов Виктор  
vgaganov@graphics.cs.msu.ru

- [1 Введение](#)
- [2 Постановка задачи](#)
- [3 Простейший алгоритм сопоставления особенностей](#)
- [4 Scale-space детекторы особенностей](#)
  - [4.1 Понятие scale-space](#)
  - [4.2 Нормированные производные](#)
  - [4.3 Упрощенный детектор Харриса-Лапласа](#)
- [5 Дескрипторы особенностей](#)
  - [5.1 Инвариантность к изменению масштаба](#)
  - [5.2 Инвариантность к повороту](#)
  - [5.3 Инвариантность к изменению освещенности](#)
- [6 Алгоритм сопоставления](#)
- [7 Примеры работы](#)
- [8 Литература](#)

### **Введение:**

Обнаружения и сопоставление точечных особенностей на изображениях является важной задачей компьютерного зрения и находит применение в таких приложениях как 3D реконструкция, индексация в базах данных изображений и т.д. На первый взгляд наборы соответствующих точек на изображениях дают довольно мало информации об изображениях и наблюдаемой сцене, но на самом деле это не так. Например, если у нас есть несколько изображений одной сцены и наборы соответствующих точек на этих изображениях мы можем определить настройки и положение камеры для каждого изображения.

К сожалению, на данный момент нет универсального алгоритма качественно решающего данную задачу. Связано это с тем, что при съемке сцены изображения ее точек подвергаются различным искажениям, например проективные преобразования, связанные с перемещением камеры либо объектов сцены, изменения освещенности сцены и т.д.

В данной статье будут рассказано о некоторых принципах построения алгоритмов сопоставления точечных особенностей, инвариантных к таким искажениям как изменение масштаба, вызванное например оптическим или цифровым zoom'ом, поворот изображения и изменения освещенности сцены.

### **Постановка задачи:**

Рассмотрим задачу более детально. Пусть нам даны 2 изображения некоторой сцены. Требуется найти набор пар точек  $(x_{1,i}, y_{1,i}) (x_{2,i}, y_{2,i}) i = 1..N$  таких, что  $(x_{1,i}, y_{1,i}) (x_{2,i}, y_{2,i})$  являются изображениями одной и той же точки в 3D.

Заметим, что для сопоставления подходят далеко не все точки изображения (рис 1). Например, очень сложно найти соответствующую точку для некоторой точки изображения однородной поверхности. Поэтому для сопоставления используются т.н. особые точки. Особая точка изображения  $m$  - это такая точка изображения, окрестность которой  $o(m)$  можно отличить от окрестности любой другой точки изображения  $o(n)$  в некоторой другой окрестности особой точки.  $o_2(m)$ . Более детальное описание особых точек изображения приводится, например в [2].



Рис.1 Слева точка, которая является особой, справа точка, которая не является особой

Важным свойством любого алгоритма сопоставления особенностей является набор искажений изображения особой точки, с которыми он способен справиться. Обычно выделяют следующие виды искажений:

- Изменение освещенности особой точки
- Изменение масштаба (цифровой или оптический zoom, перемещение камеры по направлению к особенности)
- Поворот изображения особой точки (поворот камеры вокруг оптической оси, поворот объекта на котором находится особенность вокруг оптической оси камеры)
- Проективные искажения изображения особенности, например связанные с поворотом и перемещением камеры в пространстве (в статье не рассматриваются)

#### **Простейший алгоритм сопоставления особенностей:**

Для начала рассмотрим простейший, уже ставший классическим, алгоритм сопоставления особенностей на изображениях и его свойства. Алгоритм работает по следующей схеме:

1. Используя детектор Харриса [2,7] выделить особенности на паре изображений. Вместо детектора Харриса может быть использован любой другой подобный детектор особенностей, но обычно используют именно детектор Харриса. Обозначим наборы найденных особенностей  $A = \{a_i, i = 1..n\}$   $B = \{b_i, i = 1..m\}$
2. Для каждой точки первого изображения  $a_i$  найти точку второго изображения  $b_j$  такую что  $j = \arg \min_k (f(w(a_i), w(b_k)))$ , где  $w(x)$ - некоторая окрестность точки  $x$ , а  $f(w_1, w_2)$ - мера, используемая для сравнения окрестностей особых точек. Обычно  $f$  строится с использованием мер SSD или NCC. В силу того, что данный алгоритм обычно применяется только при незначительных изменениях положения камеры обычно соответствия для  $a_i$  ищутся в её небольшой окрестности  $w(a_i)$
3. Выбрать из полученных пар  $N$  наиболее близких в смысле меры  $f$

Приведенный выше алгоритм подходит разве что для сопоставления особенностей в кадрах видеопоследовательности т.к. он крайне неустойчив к любым видам искажения изображения особых точек. Например, при повороте камеры изображения окрестностей особых точек изменяются настолько сильно что алгоритм практически не дает верных соответствий. Аналогичная ситуация наблюдается и с изменением масштаба изображения. Единственный вид искажений, которым данный алгоритм способен противостоять – изменения освещенности (для этого надо проводить нормировку окрестностей особенностей).

Другая проблема алгоритма связана с повторяемостью обнаружения особенностей детектора Харриса и ему подобных детекторов. Например, при условии существенного изменения масштаба для пары изображений наборы особых точек которые выдает детектор Харриса имеют крайне мало пересечений т.е. то что являлось особой точкой на первом изображении в большинстве случаев не является особой точкой на втором изображении.

Указанные причины делают практически невозможным применение данного алгоритма на практике при наличии таких искажений как изменение масштаба и поворот. О том, как эти трудности преодолеть, и пойдет речь далее.

**Scale-space детекторы особенностей:**

Для того чтобы решить проблему изменения масштаба применяются т.н. scale-space детекторы особенностей. Такие детекторы помимо пространственного положения особенности  $(x, y)$  оценивают также локальный масштаб особенностей  $s$ , который отражает «размер» особенности на изображении. Существует множество таких детекторов особенностей, описания некоторых из них можно найти в [1,3,4,6]. В статье мы опишем упрощенный детектор Харриса-Лапласа т.к. это один из лучших scale-space детекторов особенностей, и при этом он может быть крайне эффективно реализован.

**Понятие scale-space:**

Введем понятие scale-space представления изображения.

**Опр:** Пусть  $I = I(x, y)$ - функция двух аргументов, тогда scale-space представлением функции  $I = I(x, y)$  называется функция трех аргументов

$$L(x, y, \sigma) = (g(\cdot, \sigma) * I(\cdot))(x, y), \text{ где } g(x, y, \sigma) = e^{-x^2+y^2/2\sigma^2} / 2\sigma^2\pi,$$

\* - символ, обозначающий свертку, а  $L(\cdot, 0) = I(\cdot)$

Таким образом scale-space представление изображения есть размытие изображения при помощи свертки Гауссианом  $g$  с разными значениями параметра размытия  $\sigma$ . Заметим что понятие scale-space определено для полутонового изображения и если вы работаете с цветным изображением в формате RGB следует сначала привести его к полутоновому, например используя формулу  $I = 0.3 * R + 0.59 * G + 0.11 * B$ .

**Нормированные производные:**

При работе со scale-space представлением изображения используются т.н. нормированные производные, которые определяются следующим образом.

$L_{a, \text{norm}}(\cdot, \sigma) = \sigma(\partial_a L)(\cdot, \sigma) = \sigma(\partial_a (g * I))(\cdot, \sigma)$  - производная от  $L$  по  $a$  ( $a$  – направление взятия производной, вместо  $a$  можно подставить  $x$  или  $y$ ). Заметим что основное отличие нормированных производных от обычных – то что производная умножается на значение масштаба  $\sigma$ . При использовании производных порядка большего чем 1 следует возвести  $\sigma$  в степень равную порядку производной, например  $L_{xx, \text{norm}}(\cdot, \sigma) = \sigma^2(\partial_{xx} L)(\cdot, \sigma)$  и т.д.

Это крайне неполное описание понятия scale-space представления изображения, но для дальнейшего изложения нам этого вполне достаточно. Более подробное описание, которое включает в себя scale-space представление многомерное функции, более общий вид нормированных производных, их свойства и т.д. можно найти в [3,5].

**Упрощенный детектор Харриса-Лапласа:**

Теперь полностью опишем алгоритм работы упрощенного детектора Харриса-Лапласа.

1. Вычислить значения адаптированной к масштабированию функции Харриса для масштабов  $\sigma_n = \xi^n \cdot \sigma_0$

$$H(x, \sigma_I, \sigma_D) = \det(\mu(x, \sigma_I, \sigma_D)) + 0.04 \text{trace}^2(\mu(x, \sigma_I, \sigma_D)), \text{ где}$$

$$\mu(x, \sigma_I, \sigma_D) = g(\sigma_I) * \begin{bmatrix} L_{x, \text{norm}}^2(x, \sigma_D) & L_{xy, \text{norm}}(x, \sigma_D) \\ L_{xy, \text{norm}}(x, \sigma_D) & L_{y, \text{norm}}^2(x, \sigma_D) \end{bmatrix}, \sigma_I = \sigma_n, \sigma_D = s\sigma_I, s = 0.7$$

Количество слоев и значение шага масштаба  $\xi$  следует выбирать в зависимости от того, насколько большим может быть изменение масштаба между 2 изображениями. На приведенных примерах строилось 17 слоев со значениями  $\xi = 1.2, \sigma_0 = 1$

2. Для каждого уровня масштаба найти локальные максимумы вычисленной функции Харриса, это и есть особые точки для данного масштаба изображения. Обычно таким образом получается достаточно много точек и часть из них можно отбросить. Например, можно отбросить все точки, для которых значение функции Харриса не превосходит некоторого значения  $H_{thr}$  т.к. максимумы с небольшим значением функции Харриса менее устойчивы. Мы использовали  $H_{thr} = 1000$ .
3. Для каждой найденной таким образом особенности установить достигается ли в ней максимум функции  $LoG(x, \sigma_n) = \left| L_{xx, norm}(x, \sigma_n) + L_{yy, norm}(x, \sigma_n) \right|$  по переменной  $n$ , т.е.  $LoG(x, \sigma_{n-1}) < LoG(x, \sigma_n), LoG(x, \sigma_{n+1}) < LoG(x, \sigma_n)$ . Если локальный максимум не достигается, либо значение функции не превосходит порога  $LoG_{thr}$ , то точка отбрасывается. В данной работе использовалось  $LoG_{thr} = 10$
4. Все оставшиеся точки являются особенностями изображения, с каждой точкой ассоциирован масштаб  $\sigma_n$ , на котором она была обнаружена.

Заметим, что именно использование многих уровней масштаба решает проблему повторяемости обнаружения особенностей. Если при сильном изменении масштаба обычный детектор Харриса не мог обнаружить большую часть особенностей из первого изображения на втором изображении, то упрощенный детектор Харриса-Лапласа обнаружит их, просто они будут обнаружены на другом уровне масштаба.

Пример работы алгоритма можно увидеть на рисунке 2. Зелеными точками на изображении отмечены положения особенностей на изображении, а синие окружности отражают масштаб особенности, причем  $r = 3\delta$ , где  $r$  - радиус окружности, а  $\delta$  - масштаб особенности.



Рис.2 Пример работы упрощенного детектора Харриса-Лапласа

**Дескрипторы особенностей:**

Для сопоставления обнаруженных особенностей используются т.н. дескрипторы особенностей. Дескриптор особенности – вектор числовых характеристик окрестности особенности  $D(x) = [f_1(w(x))...f_n(w(x))]$ . При сопоставлении особенностей, для принятия решений о том соответствуют ли друг другу особенности или нет, сравниваются именно дескрипторы этих особенностей. Простейший пример дескриптора особенности – сама окрестность особенности, записанная в виде вектора. Далее будет рассказано, как добиться того, чтобы дескрипторы особенностей были инвариантны к некоторым видам искажений изображения окрестности особенности.

**Инвариантность к изменению масштаба:**

При использовании scale-space детектора особенностей добиться инвариантности к изменению масштаба очень просто. Для этого достаточно перед вычислением дескриптора провести нормировку в соответствии с локальным масштабом особенности, например если с особенностью ассоциирован масштаб 2, то окрестность особенности следует масштабировать с коэффициентом 1/2 и т.д. Если дескриптор состоит из выражений, в которых используются исключительно нормированные производные, то масштабировать окрестность не обязательно. Достаточно рассчитывать значения производных для значения масштаба  $\sigma$ , который ассоциирован с особенностью.

**Инвариантность к повороту:**

Самый простой метод добиться инвариантности к повороту при сопоставлении особенностей – использовать дескрипторы, компоненты которых инварианты к повороту. Далее приводится пример дескриптора из [1], каждая компонента которого инвариантна к повороту. Все производные в выражении – нормированные производные.

$$\left[ \begin{array}{c}
 L \\
 L_x L_x + L_y L_y \\
 L_{xx} + L_{yy} \\
 L_{xx} L_x L_x + 2L_{xy} L_x L_y + L_{yy} L_y L_y \\
 L_{xx} L_{xx} + 2L_{xy} L_{xy} + L_{yy} L_{yy} \\
 L_{xxx} L_y L_y L_y + 3L_{xyy} L_x L_x L_y - 3L_{xxy} L_x L_y L_y - L_{yyy} L_x L_x L_x \\
 L_{xxx} L_x L_y L_y + L_{xxy} (-2L_x L_x L_y + L_y L_y L_y) + L_{xyy} (-2L_x L_y L_y + L_x L_x L_x) + L_{yyy} L_x L_x L_y \\
 L_{xxy} (-L_x L_x L_x + 2L_x L_y L_y) + L_{xyy} (-2L_x L_x L_y + L_y L_y L_y) - L_{yyy} L_x L_y L_y + L_{xxx} L_x L_x L_y \\
 L_{xxx} L_x L_x L_x + 3L_{xxy} L_x L_x L_y + 3L_{xyy} L_x L_y L_y + L_{yyy} L_y L_y L_y
 \end{array} \right]$$

Серьезным недостатком этого метода является то, что в дескрипторе нельзя использовать компоненты, которые не инвариантны к повороту, а число операторов, которые инвариантны к повороту, и при этом применимы на практике, ограничено.

Еще один способ добиться инвариантности к повороту – предварительно нормировать окрестность точки особым образом, чтобы скомпенсировать поворот, и лишь потом вычислять дескрипторы для особенности. Для того чтобы нормировать окрестность по повороту требуется оценить т.н. ориентацию особенности (см. рис 3). Существует много методов оценки локальной ориентации особенности, все они так или иначе используют направление векторов градиентов в окрестности особенности. Мы опишем метод Lowe, предложенный им в [4].

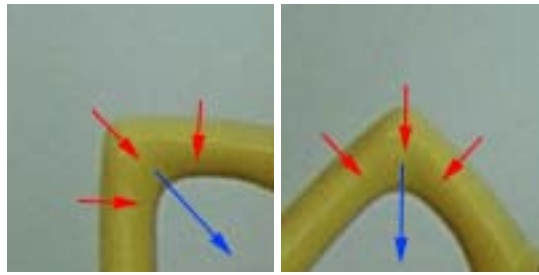


Рис. 3 Окрестность особой точки. Красным указаны направления градиентов, а синим показана ориентация особенности.

1. Разделим множество углов от 0 до 360 градусов на 36 одинаковых участков, каждый по 10 градусов. Каждому из участков сопоставим ячейку в гистограмме  $H$
2. Для каждой точки  $x_0$  из некоторой окрестности особенности  $a$  :
  - 1) Вычислить фазу и модуль вектора градиента  $\text{grad}(x_0, \partial) = (L_{x,\text{norm}}(x_0, \partial) \quad L_{y,\text{norm}}(x_0, \partial))$ ,  $\theta = L_{y,\text{norm}}(x_0, \partial) / L_{x,\text{norm}}(x_0, \partial)$ ,  $A = |\text{grad}(x_0, \partial)|$
  - 2)  $H[i_\theta] = H[i_\theta] + Aw$ , где  $i_\theta$  - индекс ячейки, которая соответствует фазе градиента, а  $w$  – вес точки  $x_0$ . В качестве веса можно использовать, например 1 (это простейший случай) но более качественных и устойчивых результатов оценки удастся добиться при использовании в качестве веса значения Гауссиана с центром в точке  $a$
3. После того как все сказанное выше проделано для каждой точки окрестности особенности в качестве ориентации особенности следует выбрать  $\varphi = i \cdot 10^\circ$ , где  $i$  - индекс элемента гистограммы с наибольшим значением.

После того, как была вычислена ориентация особенности, произвести нормировку очень просто – для этого достаточно повернуть особенность на угол  $-\varphi$  вокруг центра особенности. К сожалению, для некоторой части особенностей ориентация оценивается неверно (обычно 10-20%) и дескрипторы этих особенностей оказываются абсолютно непригодны к сопоставлению. Именно это является основным недостатком данного подхода.

**Инвариантность к изменению освещенности:**

Для того чтобы сделать дескриптор инвариантным к изменению освещенности нужно для начала ввести модель этого изменения. Обычно применяют т.н. аффинную модель, которая подразумевает, что значения пикселей окрестности при изменении освещенности меняется по следующему закону:  $\tilde{I} = a \cdot I(x) + b$ . Это модель не очень точно соответствует действительности и реально процессы, происходящие в пикселях при изменении освещенности намного сложнее, но в силу того, что особенности локальны и имеют небольшой размер такой модели более чем достаточно.

Учитывая принятую модель можно использовать следующий алгоритм, устраняющий влияние освещенности на значение пикселей в окрестности особенности:

$$I_{\text{mean}}(w(x)) = I(w(x)) - \text{mean}(I(w(x))), \quad I_{\text{result}}(w(x)) = I_{\text{mean}}(w(x)) / \text{std}(I(w(x))),$$

где  $\text{mean}(I(w(x)))$  и  $\text{std}(I(w(x)))$  обозначают выборочное среднее и среднее квадратичное отклонения в окрестности  $w$ ,  $I_{\text{mean}}(w(x))$  – окрестность, скомпенсированная по переносу, а  $I_{\text{result}}$  – результирующая окрестность, скомпенсированная по освещенности. Именно по особенности  $I_{\text{result}}$  следует вычислять дескриптор особенности, чтобы добиться инвариантности к изменениям освещенности.

Также можно составлять дескриптор из функций, которые инвариантны к аффинным изменениям освещенности, далее приводится инвариантный к повороту дескриптор из [1], преобразованный таким образом, чтобы его компоненты были инвариантны к изменениям освещенности. Стоит заметить, что для того чтобы сделать дескриптор инвариантным к изменениям освещенности пришлось удалить первые две его компоненты. Все производные в выражении – нормированные производные.

$$\left[ \begin{array}{c} (L_{xx} + L_{yy}) / (L_x L_x + L_y L_y)^{1/2} \\ (L_{xx} L_x L_x + 2L_{xy} L_x L_y + L_{yy} L_y L_y) / (L_x L_x + L_y L_y)^{3/2} \\ (L_{xx} L_{xx} + 2L_{xy} L_{xy} + L_{yy} L_{yy}) / (L_x L_x + L_y L_y) \\ (L_{xxx} L_y L_y L_y + 3L_{xyy} L_x L_x L_y - 3L_{xxy} L_x L_y L_y - L_{yyy} L_x L_x L_x) / (L_x L_x + L_y L_y)^2 \\ (L_{xxx} L_x L_y L_y + L_{xyy} (-2L_x L_x L_y + L_y L_y L_y) + L_{xxy} (-2L_x L_y L_y + L_x L_x L_x) + L_{yyy} L_x L_x L_y) / (L_x L_x + L_y L_y)^2 \\ (L_{xxy} (-L_x L_x L_x + 2L_x L_y L_y) + L_{xyy} (-2L_x L_x L_y + L_y L_y L_y) - L_{yyy} L_x L_y L_y + L_{xxx} L_x L_x L_y) / (L_x L_x + L_y L_y)^2 \\ (L_{xxx} L_x L_x L_x + 3L_{xyy} L_x L_x L_y + 3L_{xxy} L_x L_y L_y + L_{yyy} L_y L_y L_y) / (L_x L_x + L_y L_y)^2 \end{array} \right]$$

**Алгоритм сопоставления:**

После того как для каждой особенности на паре изображений рассчитаны дескрипторы можно непосредственно приступить к сопоставлению особенностей. Существует множество вариантов алгоритмов сопоставления особенностей по их дескрипторам, их описания можно найти, например в [1] и [6]. Мы предлагаем следующую схему, которая совмещает алгоритмы из [1] и [6].

Обозначим наборы особых точек первого и второго изображения  $A = \{a_i, i = 1..n\}$   $B = \{b_i, i = 1..m\}$  а их дескрипторы  $D_A = \{D_{A,i}, i = 1..n\}$   $D_B = \{D_{B,i}, i = 1..m\}$ . Будем считать что дескрипторы – вектора столбцы.

1. Для каждой точки первого изображения  $a_i$ 
  - 1) Найти точку второго изображения  $b_j$ , такую что  $j = \underset{k}{\operatorname{argmin}}(f(D_{A,i}, D_{B,k}))$   
 $f(D_1, D_2) = (D_1 - D_2) \Sigma^{-1} (D_1 - D_2)^T$ , где  $\Sigma$  - матрица ковариации дескриптора. Обычно считают что дескриптор – случайная величина с многомерным нормальным распределением. При этом мера  $f$  соответствует расстоянию Махаланобиса. О том как проводить оценку матрицы ковариации для дескрипторов подробно написано в [1].
  - 2) Если значение  $f(D_{A,i}, D_{B,j})$  превосходит значение некоторого порога  $f_{thr}$  следует отбросить данное соответствие. Значение порога сильно зависит от дескрипторов которые используются для сопоставления. О том как оценить значение порога подробно написано в [1].
2. Все пары соответствующих точек следует упорядочить, используя некоторую меру качества. Как меру качества мы использовали неоднозначность соответствия, предложенную в [6], которая имеет следующий вид:

$$Amb(D_{A,i}, D_{B,j}) = \frac{f(D_{A,i}, D_{B,j})}{\min_{k \neq j} (f(D_{A,i}, D_{B,k}))}$$

Чем меньше значение меры  $Amb$ , тем более качественным является соответствие.

3. Выбрать N первых (самых качественных) соответствий.

**Примеры работы:**

Далее приводятся примеры результатов сопоставления особенностей для нескольких различных пар кадров. Для обнаружения особенностей использовался упрощенный детектор Харриса-

Лапласа. В качестве дескрипторов при сопоставлении выступала сама окрестность особенности размером 11x11 пикселей с предварительной нормированная по масштабу, освещенности и повороту.

На рисунке 4 изображена пара изображений, которые связаны между собой поворотом вокруг оптической оси камеры и изменением масштаба. Для этой пары изображений были сначала найдены 100 лучших соответствий, по этим соответствиям была оценена гомография, после чего были заново посчитаны 100 лучших соответствия с учетом гомографии.





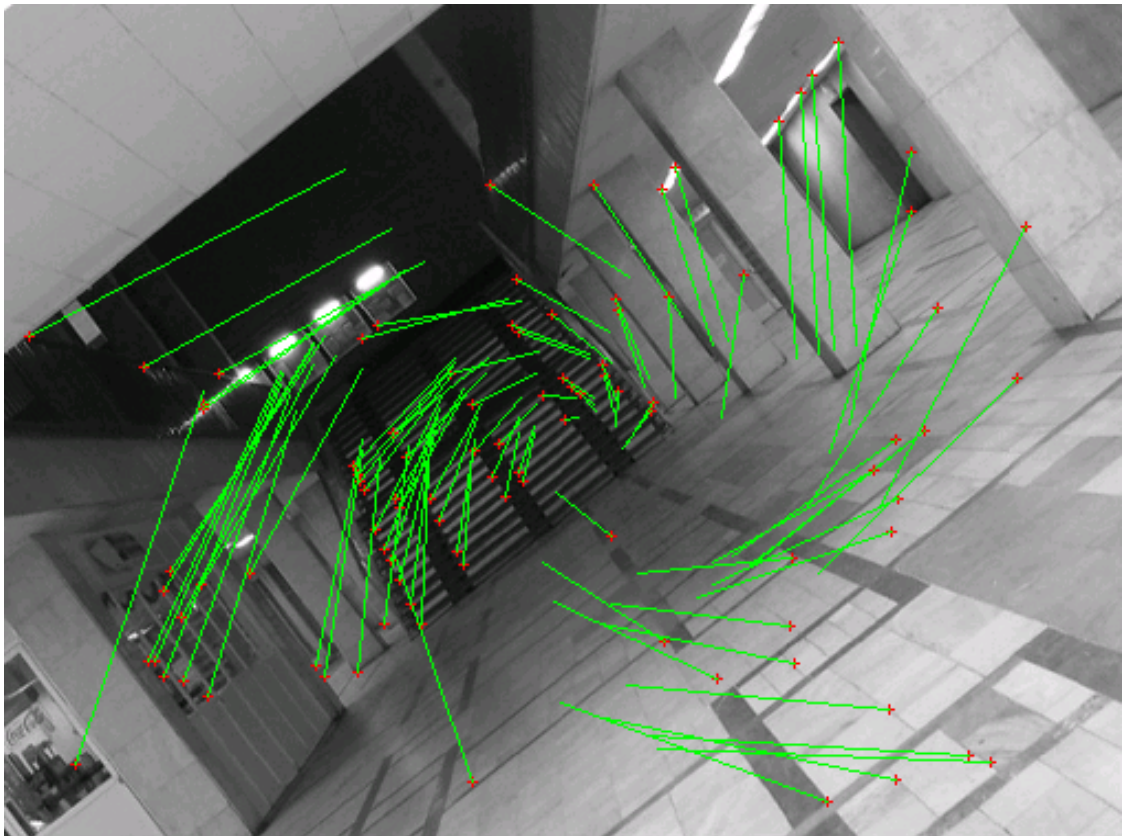


Рис. 4 Пара изображений с посчитанными точечными соответствиями. Красными точками отмечены особенности. Зеленые отрезки – т.н. «следы» особенностей

На рисунке 5 можно увидеть более сложную пару изображений. При съемке второго изображения камера была перенесена на несколько метров вперед. При движении камеры вперед окрестности точечных особенностей подвергаются проективным искажениям что осложняет задачу поиска соответствий. Для этой пары изображений были посчитаны 100 лучших соответствий, потом по ним была рассчитана фундаментальная матрица, после чего были посчитаны 100 лучших соответствий с учетом фундаментальной матрицы (выведены только 20 из них для того чтобы было проще воспринимать картинку).

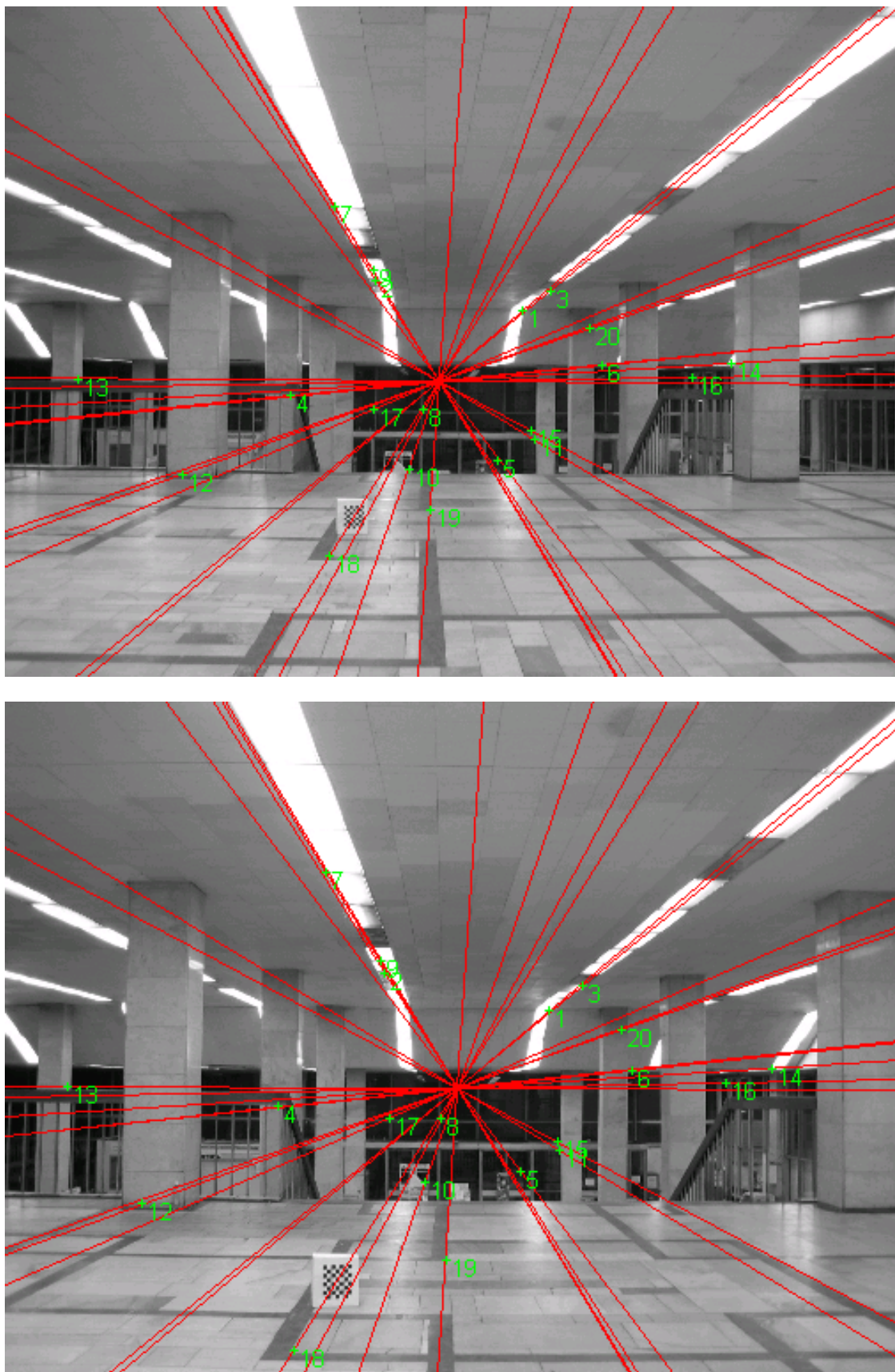


Рис. 5 Пара изображений с рассчитанными соответствиями и эпиллярной геометрией. Красным цветом изображены эпиллярные линии. Пронумерованные зеленые точки – соответствия.

На рисунке 6 изображена пара изображений, которые связаны между собой сильным поворотом камеры. Для этой пары изображений были сначала найдены 100 лучших соответствий, по этим

соответствиям была оценена гомография, после чего были заново посчитаны 100 лучших соответствия с учетом гомографии (выведены только 20 из них для того чтобы было проще воспринимать картинку).

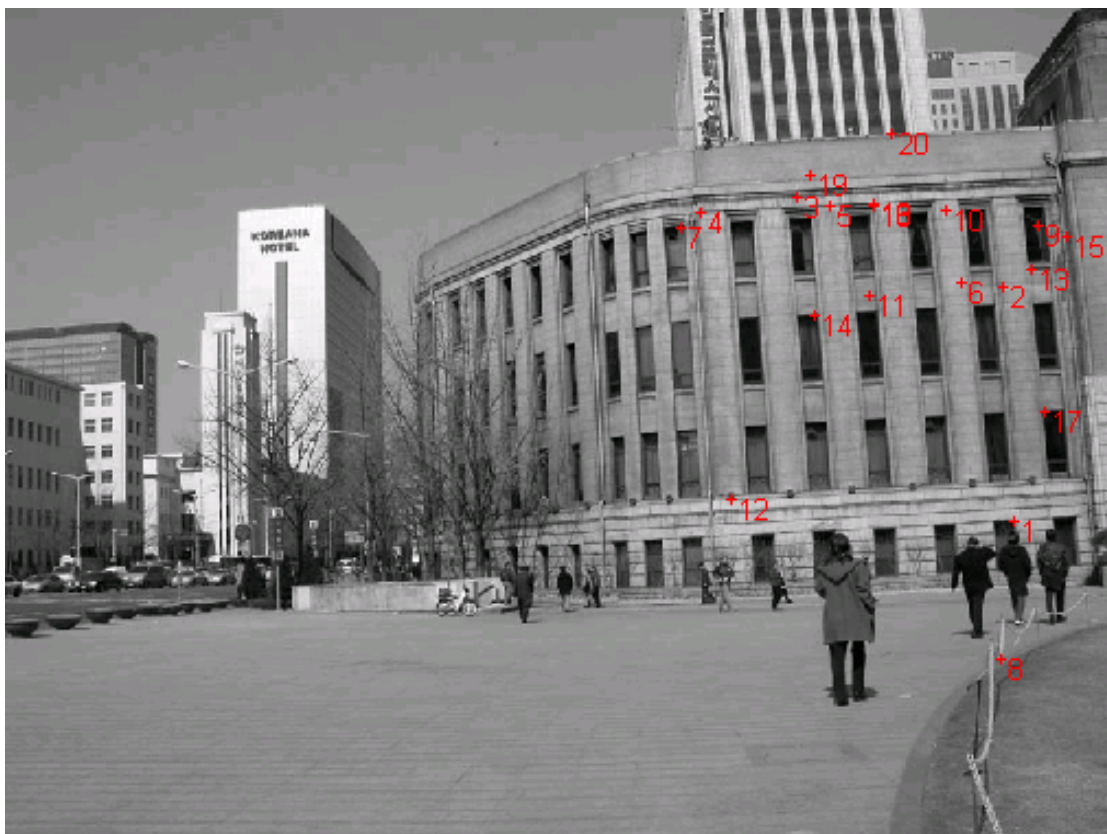


Рис. 6 Пара изображений с посчитанными точечными соответствиями. Красными точками отмечены особенности (они пронумерованы). Зеленые отрезки – «следы» особенностей.

**Литература:**

- [1] Krystian Mikolajczyk, "Detection of local features invariant to affine transformations", 2002
- [2] Антон Конушин, "Слежение за точечными особенностями сцены (Point feature tracking)", 2003
- [3] Tony Lindeberg, "Feature Detection with Automatic Scale Selection", 1998
- [4] David G. Lowe "Object Recognition from Local Scale-Invariant Features", 1999
- [5] Tony Lindeberg, "Scale-space theory: A basic tool for analysing structures at different scales", 1994
- [6] Adam Baumberg, "Reliable Feature Matching Across Widely Separated Views", 2000
- [7] Chris Harris, Mike Stephens, "A combined corner and edge detector", 1988