

---

# Слежение за точечными особенностями сцены (Point feature tracking)

Автор: Антон Конушин [ktosh@zmail.ru](mailto:ktosh@zmail.ru)

[1 Введение](#)

[1 Особенности или особые точки](#)

[2 Слежение за точечными особенностями](#)

[3 Простая и расширенная схема работы системы слежения](#)

[4 Нахождение набора особенностей](#)

[5 Математическая формулировка задачи слежения](#)

[6 Развитие алгоритмов слежения](#)

[7 Lucas-Kanade tracker](#)

[8 Tomasi-Kanade tracker](#)

[9 Shi-Tomasi-Kanade tracker](#)

[10 Модификация трекера для случая переменного освещения](#)

[11 Нормализация окрестностей изображения для учета изменений освещения](#)

[12 Автоматический отбор некачественных особенностей](#)

[13 Пример трекинга особенностей](#)

[14 Ссылки](#)

## 0. Введение

Любое изображение в отдельности, несмотря на кажущуюся простую структуру - двумерную матрицу чисел, содержит очень сложную и комплексную - интегральную информацию о наблюдаемой сцене. Извлечь из изображений какую-нибудь структурированную и осмысленную информацию о наблюдаемой сцене невероятно сложно. Видеопоследовательность, полученная с движущейся камеры, особенно если на ней запечатлена динамическая сцена, анализировать еще сложнее. Поэтому просто необходимы технологии, которые позволили бы извлекать из видеопоследовательностей некоторую, пускай очень неполную, но осмысленную и достаточно просто структурированную информацию об объектах сцены и ее динамике.

Одной из таких технологий является слежение за точечными особенностями в потоке изображений. В самом деле, точка - это наиболее простой и понятный геометрический элемент, с которым, как кажется, проще всего работать. Точка может рассматриваться как элемент дискретного представления некоторой функции. Если мы выделим в сцене некоторое количество "особых" точек, и определим их положение на каждом кадре, то полученные данные смогут дать огромное количество информации о структуре сцены, а также о параметрах камеры, с которой были получены изображения.

Некоторые методы используют слежение за особенностями для определения траектории наблюдаемых объектов, для вычисления структуры наблюдаемой динамической сцены, выделения отдельных движений и движущихся объектов. Если известны некоторые ограничения на наблюдаемую сцену (например, что она статическая, а камера движется по сцене), то можно определить параметры внутренней и внешней калибровки камеры. Если в сцене отслеживается большое количество особых точек, то даже может быть построена грубая трехмерная модель всей сцены.

## 1. Особенности или особые точки

Что же такое точечная особенность или особая точка сцены? Поскольку почти вся информация, известная нам о сцене, содержится в ее изображениях, то и особая точка сцены должна определяться через изображения.

*Особая точка сцены или точечная особенность (point feature) - это такая точка сцены, изображение которой можно отличить от изображений всех соседних с ней точек сцены.*

Сразу встает множество вопросов - а что понимается под изображением точки и как изображение одной точки можно отличить от изображения другой точки, если у нас есть только дискретизированное представление изображения сцены (цифровое изображение - digital image)?

Для сравнения и описания точек можно использовать ее окрестность. Тогда предыдущее определение уточняется следующим образом:

*Под точечной особенностью понимается такая точка сцены  $M$ , лежащая на плоском участке поверхности сцены  $PlaneSegment$ , изображение окрестности  $I(PlaneSegment)$  которой можно отличить от изображений окрестностей всех других точек сцены  $N$  из некоторой другой окрестности этой точки  $O(M)$ .*

Воспользовавшись последним определением можно дать определение точечной особенности изображения:

*Точечная особенность 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.  $o_2(m)$*

Из этих определений хорошо видно, что между особенностями сцены и особенностями изображения есть соответствие: точечной особенности сцены должна соответствовать точечная особенность изображений. Обратное неверно: существуют такие особые точки изображения, которым не соответствует никакие особые точки сцены. Такие точки называется ложными особенностями сцены (false feature point).

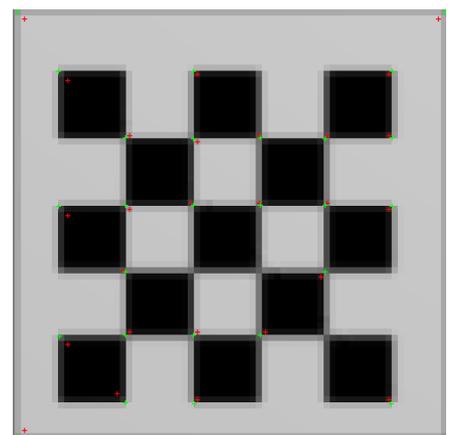
Отделение ложных особенностей от настоящих является одной из основных проблем при отслеживании движения камеры или при реконструкции трехмерной сцены. Усугубляется она тем, что надежно решить ее, работая только с одним изображением, невозможно. Необходимо устанавливать соответствие между особенностями нескольких изображений, а затем построить некоторую модель, которой эти соответствия будут удовлетворять наилучшим образом. Особые точки, соответствия которых не будут удовлетворять такой модели, будут признаны ложными, или выбросами (outlier).

Методы построения таких моделей в последнее десятилетие бурно развивались, но даже краткое их описание выходит далеко за рамки этой статьи. Алгоритмы, о которых рассказывается далее, применимы для всех точечных особенностей изображений. Однако необходимо дополнительно отметить, что при слежении за ложной точечной особенностью изображения, на разных кадрах видеопоследовательности ей будут соответствовать разные точки сцены.

Для простоты в качестве окрестности точки изображения берется прямоугольное окно небольшого размера. Для сравнения таких прямоугольных окон могут использоваться различные меры на изображениях (например, обычное SSD - Sum of Squared Difference или кросс-корреляция).

Большинство детекторов точечных особенностей работают сходным образом: для каждой точки изображения вычисляется некоторая функция от ее окрестности. Точки, в которых эта функция достигает локального максимума, очевидно можно отличить от всех точек из некоторой ее окрестности.

Существует целый набор функций, которые можно использовать для обнаружения точечных особенностей. Чаще всего для задач отслеживания точек сцены применяются функции, находящие в изображении структуры, похожие на угол - уголки. (corners). Детекторы, использующие такие функции, называются детекторами углов. Именно они чаще всего применяются для решения задач отслеживания точечных особенностей сцены.



## 2. Слежение за точечными особенностями

С точки зрения "черного ящика", без рассмотрения принципов работы, задача слежения за особенностями (feature tracking) формулируется так:

Дана последовательность изображений ImageSequence некоторой сцены  $S$ , полученная с движущейся или неподвижной камеры. Необходимо получить набор как можно более точных последовательностей координат проекции некоторых точек сцены в каждом кадре, т.е. набор SceneFeatureImage[i] = {x(t), y(t)}, где x(t), y(t) - координаты проекции особенности  $i$  в кадре, полученном в момент времени  $t$ .

Другими словами, под слежением за точечными особенностями сцены в общем случае понимается определение координат проекции точки сцены в текущем кадре, если известны координаты ее проекции в предыдущем, и неизвестно ничего о камере, с которых получены изображения.

Однако ничего о точечных особенностях сцены нам не известно. В изображениях мы можем выделить только набор особенностей изображения, которые могут как соответствовать, так и не соответствовать каким-то особенностям сцены. Поэтому, работая исключительно в пространстве изображений мы можем отслеживать только точечные особенности изображений, а не сцены.

Разумеется, предполагаемая природа точечных особенностей изображения должна учитываться. Например, т.к. точечная особенность считается лежащей на плоском сегменте сцены, то ее изображение может претерпевать перспективные искажения, которые в свою очередь можно приблизить аффинными. Учитывая это обстоятельство, задачу слежения за особенностями формулируется следующим образом:

Дана последовательность изображений ImageSequence одной и той же сцены  $S$ , полученная с движущейся или неподвижной камеры, и набор точечных особенностей  $\{N\}$ , выделенных в первом кадре последовательности. Для каждой точечной особенности  $n$  из  $\{N\}$  найти такие точки  $n(t)$  на всех изображениях, что их окрестности будут максимально близки к окрестности  $n(0)$ , с учетом предполагаемой природы искажения ее окрестности и движения точки.

Очевидно, что изображение окрестности любой особой точки с разных точек обзора будет разным, и чем более удалены точки обзора друг от друга, тем больше будут отличаться изображения. Поэтому от кадра к кадру изображение окрестности отслеживаемой точки обычно изменяться. Современные методы позволяют отслеживать движение точек только при условии малых искажений изображения окрестности, которых достигаются при условии незначительных изменений сцены и малых смещений точки обзора. В этом случае можно утверждать, что искомая точка не может далеко сместиться в новом кадре от своего предыдущего положения.

В простейшем случае в новом кадре находится ближайшая к предыдущему положению точка с наиболее близкой в используемой мере окрестностью. Такое слежение за особенностями сцены может рассматриваться как дискретизация оптического потока, т.е. определение его величины не на всем изображении, а в нескольких отдельных точках. Для каждой точки вычисляется только ее смещение от кадра к кадру (2 параметра).

Если учесть, что окрестность точечной особенности считается плоским сегментом сцены, тогда искажения ее изображения будут достаточно точно описываться с помощью аффинных трансформаций изображения. В этом случае при слежении вычисляется шесть параметров изменения окрестности точки.

При изменении точки зрения меняется и освещенность окрестности точки сцены, а значит, меняется и яркость пикселей изображения этой окрестности. Эти изменения можно минимизировать, если воспользоваться нормализацией изображения окрестности особенности. С другой стороны, изменения освещенности при небольших смещениях камеры от кадра к кадру можно описать в форме аффинных искажений  $a * I + b$ . При поиске нового положения особенности в некоторых алгоритмах рассчитывается и параметры изменения освещения ее окрестности  $a$  и  $b$ .

Чаще всего параметры искажения окрестности особых точек в дальнейшем совершенно не используются. Вычисление этих параметров производится только для того, чтобы получить как можно

более точные и истинные положения проекций особых точек на кадре.

Необходимо упомянуть, что любой алгоритм слежения за особенностями может быть применен к любым точкам изображения, а не обязательно только к особенностям. Однако в этом случае невозможно корректно определить ее положение в новом кадре. Например, если применить слежения за точкой на некоторой границе изображения, то скорее всего будет найдена тоже точка на границе, но она может на самом деле и не соответствовать той же точке сцены.

### 3. Простая и расширенная схема работы системы слежения

Схема работы любой системы слежения за особенностями состоит из двух основных этапов:

#### **Схема А.**

##### **Этап 1 (Детектирование):**

Определить в первом кадре особенности изображения

##### **Этап 2 (Слежение):**

Для каждого последующего кадра:

- Для каждой особенности  $Feature(i)$ :
  - Найти новое положение особенности в кадре  $t$

Особенности обычно выделяются в первом кадре последовательности и отслеживаются на протяжении всех остальных. Но при движении камеры или сцены часть особенностей может время от времени пропадать из вида, или изображения их окрестностей будут искажаться настолько, что перестанут быть особенностями. В этом случае слежение за такими точками становится невозможным.

В реальных практических системах слежения за особенностями постоянно вычисляется качество отслеживаемых особенностей (степень их особенности). Если оно падает ниже некоторого порога, то такие особенности отбрасываются. Вместо них, а также вместо пропавших из вида, в текущем изображении ищутся новые особые точки, которые в дальнейшем будут отслеживаться вместе со старыми, еще качественными особенностями.

С учетом оценки качества особенностей, и дополнения множества отслеживаемых особенностей схема А расширяется до:

#### **Схема В.**

##### **Этап 1 (Детектирование и оценка)**

1. Найти набор особенностей  $\{Features\}$
2. Определить качество всех особенностей -  $Quality(\{Features\})$
3. Оставить только особенности, чье качество выше некоторого заранее или динамически определенного порога, получив множество  $\{GoodFeatures\}$

##### **Этап 2 (Слежение и оценка)**

Для каждого последующего кадра:

1. Найти в текущем кадре новое положение всех особенностей из  $\{GoodFeature\}$  - слежение (tracking)
2. Определить текущее качество всех  $\{GoodFeatures\}$
3. Оставить только те особенности, чье качество удовлетворяет некоторому критерию
4. Если число отслеживаемых точек падает ниже требуемого, применить детектор к текущему изображению и добавить в  $\{GoodFeatures\}$  новые точки.

#### 4. Нахождение набора особенностей

За последние 20 лет было создано громадное количество различных детекторов точечных особенностей изображений. Все они используют разные подходы к формированию функции оценки пикселей для нахождения особенностей. В нашем случае, в системах слежения за особенностями, определяющим параметром качества детектора будет качество последующего отслеживания.

Чаще всего используется детектор Харриса [1]. Для каждого пикселя изображения вычисляется значение особой функции отклика угла (corner response function), оценивающая степень похожести изображения окрестности точки на угол.

Для этого вначале рассчитывается матрица:

$$M = \begin{bmatrix} \left(\frac{\partial I}{\partial x}\right)^2 & \left(\frac{\partial I}{\partial x}\right)\left(\frac{\partial I}{\partial y}\right) \\ \left(\frac{\partial I}{\partial x}\right)\left(\frac{\partial I}{\partial y}\right) & \left(\frac{\partial I}{\partial y}\right)^2 \end{bmatrix},$$

где  $I(x,y)$  - яркость изображения в точке  $(x,y)$

Если оба ее собственных значения велики, то даже небольшое смещение точки  $(x,y)$  в сторону вызывает значительные изменения в яркости. Что и соответствует особенности изображения. Функция отклика угла записывается в следующем виде:

$$R = \det M - k(\text{trace}M)^2$$

Параметр  $k$  обычно полагается 0.04 (предложено Харрисом). Точки изображения, соответствующие локальным максимумам этой функции и признаются особенностями. Для достижения субпиксельной точности может использоваться квадратичная (quadratic) интерполяция.

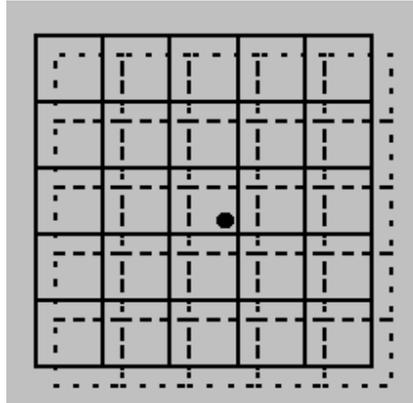
Для снижения влияния шумов на найденные особенности используется сглаживание по Гауссу, но не в самом изображении, а в картах частных производных:

$$\left(\frac{\partial I}{\partial x}\right)^2, \left(\frac{\partial I}{\partial y}\right)^2, \left(\frac{\partial I}{\partial x}\right)\left(\frac{\partial I}{\partial y}\right)$$

Во многих случаях находится чересчур большое количество углов, из-за чего в дальнейшем их будет сложно отслеживать. Поэтому вводится ограничение на минимальное расстояние между найденными особенностями, и все лишние отбрасываются.

##### **Окрестность с субпиксельной точностью**

Найденные особенности могут уточняться с помощью субпиксельной коррекции. Но окрестность точки, в не зависимости от ее расположения относительно пикселя берется одинаковой. Некоторые исследователи утверждают, что в этом случае снижается качество последующего сопоставления. В [5] было предложено с помощью билинейной интерполяции построить более точную окрестность точки таким образом, чтобы особенность всегда находилась в самом центре своего окна.



### GoodFeaturesToTrack:

Для повышения качества слежения за особенностями детектор Харриса был в последующем модифицирован [4]. Главное отличие заключается в суммировании матриц по окну  $W$  (потенциальной особенности)

Для каждого пикселя изображения вычисляется матрица  $H$ :

$$H = \sum_W \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Точка считается особенностью, если минимальное собственное значение (eigenvalue) больше некоторого заданного порога:

$$\min(\lambda_1, \lambda_2) > \lambda$$

## 5. Математическая формулировка задачи слежения

Пусть  $I(x, t)$  - яркость изображения-кадра со временем  $t$  в точке  $x$ , где  $x$  - вектор. Движение изображения (image motion), вдали от границ видимости (occluding boundaries), описывается с помощью уравнения вида:  $I(x, t) = I(\text{delta}(x), t+t1)$  (\*), где  $\text{delta}(x)$  - движение точки  $x$  при переходе от кадра ( $t$ ) к ( $t+t1$ ).

Перемещение особенности от кадра к кадру описывается этим уравнением для всех точек  $x$  из окрестности особенности  $W$ .

Отметим, что в этом случае полагается, что освещение точки сцены, соответствующей особенности, остается постоянным.

При малых изменениях изображения от кадра к кадру можно считать, что окно особенности просто смещается, и движение  $\text{delta}(x)$  принимает вид  $\text{delta}(x) = x + d$ . Однако при увеличении длительности слежения, изображение точки сцены искажается. Это искажение может быть приближенно описано аффинной трансформацией, поэтому движение точек описывается аффинным преобразованием  $\text{delta}(x) = Ax + d$ , где  $A$  - матрица размерности  $2 \times 2$ .

Задача трекера заключается в отыскании значения движения  $\text{delta}(x)$  для всех точек окна особенности  $W$ . Т.к. в реальных условиях (\*) никогда строго не выполняется, то ищется такое движение, при котором минимизируется разница между окнами при текущем и будущем положении особенности, т.е. такое

$\delta(x)$ , при котором достигается минимум

$$e = |I(\delta(x), t+1) - I(x, t)|,$$

или

$$e = \min \sum_W (I(\delta(x), t+1) - I(x, t))^2$$

если норма разности изображений  $L_2$

## 6. Развитие алгоритмов слежения

Все современные алгоритмы слежения за особенностями опираются на работу 1981 году Лукаса и Канаде. В 1991 году математическая формулировка этого алгоритма была изменена, и стала основой для всех последующих обобщений с учетом аффинных искажений окрестности и освещенности. Путем замены соответствующих переменных на константы любой из них превращается в обычный алгоритм Lucas-Kanade.

1. **Lucas-Kanade** - особенность считается только смещающейся, без искажений
2. **Tomasi-Kanade** - переформулирование Lucas-Kanade. Движение считается смещением, и рассчитывается путем итеративного решения построенной системы линейных уравнений.
3. **Shi-Tomasi-Kanade** - учитывает аффинные искажения особенности
4. **Jin-Favaro-Soatto** - модификация Shi-Tomasi-Kanade с учетом аффинных изменений освещенности особенности

## 7. Lucas-Kanade tracker [2]

Этот алгоритм в принципе применим для функций любой размерности  $n$ . Пусть  $x$  - особенность первой функции  $F$ , необходимо найти такую точку  $x+h$  функции  $G$ , что разность окрестностей этих точек по мере - минимальна.

Расстояние между окрестностями записывается в виде:

$$E = \sum_{x \in \mathbb{R}^n} [F(x+h) - G(x)]^2$$

где  $F(x)$ ,  $G(x)$  - две функции

Функцию  $F(x)$  с помощью разложения в ряд Тейлора можно приближенно представить в виде:

$$F(x+h) \approx F(x) + h \frac{\partial}{\partial x} F(x)$$

где

$$\frac{\partial}{\partial x} = \left[ \frac{\partial}{\partial x_1} \frac{\partial}{\partial x_2} \dots \frac{\partial}{\partial x_n} \right]^T$$

- градиент.

Используя это приближение, ищется минимум  $E$  путем дифференцирования и приравнивания производной к нулю:

$$\begin{aligned}
0 &= \frac{\partial}{\partial h} E \\
&\approx \frac{\partial}{\partial h} \sum_x [F(x) + h \frac{\partial F}{\partial x} - G(x)]^2 \\
&= \sum_x 2 \frac{\partial F}{\partial x} [F(x) + h \frac{\partial F}{\partial x} - G(x)],
\end{aligned}$$

Отсюда смещение  $h$  можно получить как

$$h = \left[ \sum_x \left( \frac{\partial F}{\partial x} \right)^T [G(x) - F(x)] \left[ \sum_x \left( \frac{\partial F}{\partial x} \right)^T \left( \frac{\partial F}{\partial x} \right) \right]^{-1} \right]$$

Как было указано ранее, задача слежения за особенностями без учета аффинных искажений является поиском величины оптического потока в наборе точек. Поэтому метод Lucas-Kanade часто применяется для поиска оптического потока во всем изображении.

## 8. Tomasi-Kanade tracker [3]

В этом алгоритме движение особых точек также описывается смещением вида:  
 $\text{delta}(x) = x + d$ .

Как и в предыдущем алгоритме, задача заключается в поиске такого  $d$ , при котором минимизируется разность окон особенностей:

$$\epsilon = \sum_w [I(x + d, t + \tau) - I(x, t)]^2$$

Опять же аналогично функция изображения раскладывается с помощью ряда Тейлора:

$$I(x+d, t+\tau) \approx I(x, t) + \nabla I(x, t)^T d + I_t(x, t) \tau$$

где

$$\nabla I^T = [I_u, I_v] = [\partial I / \partial u, \partial I / \partial v] \text{ and } I_t = \partial I / \partial t.$$

Тогда разницу между окнами по мере можно переписать в виде:

$$\epsilon \approx \sum_w (\nabla I(x, t)^T d + I_t(x, t) \tau)^2$$

Дифференцировав это выражение по  $d$  и приравняв производную к нулю, получаем линейную систему относительно  $d$ :

$$C\mathbf{d} = \mathbf{g}$$

где:

$$C = \sum_w \begin{bmatrix} I_u^2 & I_u I_v \\ I_u I_v & I_v^2 \end{bmatrix}$$

$$\mathbf{g} = -\tau \sum_w I_u I_v \mathbf{J}^T$$

Из этой системы  $\mathbf{d}$  получается как:

$$\mathbf{d}_k = C^{-1}\mathbf{g}$$

С учетом приближения функции изображения с помощью ряда Тейлора, решение получается неточным. Для его уточнения удобно применить итеративную процедуру Ньютона-Рафсона. Т.е. полученное на первом шаге решение берется за новое первое приближение, уравнение снова и снова.

На каждом шаге рекомендуется пересчитывать окрестность особенности с помощью билинейной интерполяции для достижения субпиксельной точности нахождения положения особенности в новом кадре.

Если интервал времени между кадрами принять за 1, получается следующий алгоритм:

$$\begin{cases} \mathbf{d}_0 = \mathbf{0} \\ \mathbf{d}_{k+1} = \mathbf{d}_k + C^{-1} \sum_w \left[ (I(\mathbf{x}, t) - I(\mathbf{x} + \mathbf{d}_k, t + 1)) \nabla I(\mathbf{x}, t) \right] \end{cases}$$

Таким образом, этот алгоритм слежения фактически является поиском точки, в которой достигается минимум некоторой функции, методом градиентного спуска. Во время каждой итерации мы сдвигаемся вдоль направления градиента изображения в текущей точке.

## 9. Shi-Tomasi-Kanade tracker [4]

В этом алгоритме впервые учитываются аффинные искажения изображения окрестности особых точек, поэтому движение пикселей окна особенности описывается в виде  $A\mathbf{x} + \mathbf{d}$ , где  $A$  - матрица (2\*2), а  $\mathbf{d}$  - смещение (2\*1).

Задача слежения за особенностью сводится к проблеме определения параметров движения и искажения окна особенности, при которой минимизируется разность:

$$\varepsilon = \int \int_W [J(A\mathbf{x} + \mathbf{d}) - I(\mathbf{x})]^2 w(\mathbf{x}) d\mathbf{x}$$

где  $W$  - окно особенности, а  $w$  - весовая функция (может использовать, а может и быть равна 1 во всем окне),  $J(\mathbf{x})$  и  $I(\mathbf{x})$  - два изображения.

Выражение дифференцируется относительно параметров движения, и производная приравнивается к 0. Затем система линеаризуется с помощью разложения функции изображения в ряд Тейлора:

$$J(\lambda \mathbf{x} + \mathbf{d}) = J(\mathbf{x}) + \mathbf{g}^T(\mathbf{u})$$

Это дает нам линейную 6\*6 систему:

$$Tz = \mathbf{a}$$

, где в векторе z объединены все искомые параметры:

$$\mathbf{z}^T = [ d_{xx} \quad d_{yx} \quad d_{xy} \quad d_{yy} \quad d_x \quad d_y ]$$

Вектор ошибки a записывается в виде:

$$\mathbf{a} = \int \int_W [I(\mathbf{x}) - J(\mathbf{x})] \begin{bmatrix} xg_x \\ xg_y \\ yg_x \\ yg_y \\ g_x \\ g_y \end{bmatrix} w d\mathbf{x}$$

А матрицу размерности 6\*6 T можно представить следующим образом:

$$T = \int \int_W \begin{bmatrix} U & V \\ V^T & Z \end{bmatrix} w d\mathbf{x}$$

$$U = \begin{bmatrix} x^2 g_x^2 & x^2 g_x g_y & x y g_x^2 & x y g_x g_y \\ x^2 g_x g_y & x^2 g_y^2 & x y g_x g_y & x y g_y^2 \\ x y g_x^2 & x y g_x g_y & y^2 g_x^2 & y^2 g_x g_y \\ x y g_x g_y & x y g_y^2 & y^2 g_x g_y & y^2 g_y^2 \end{bmatrix}$$

$$V^T = \begin{bmatrix} x g_x^2 & x g_x g_y & y g_x^2 & y g_x g_y \\ x g_x g_y & x g_y^2 & y g_x g_y & y g_y^2 \end{bmatrix}$$

$$Z = \begin{bmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{bmatrix}$$

Полученная система решается также итеративно по методу Ньютона-Рафсона.

Если движение считается не аффинным, а просто смещением, то первые четыре элемента искомого вектора z обращаются в 0, и значимыми остаются только последние два. Алгоритм превращается в

алгоритм Tomasi-Kanade .

## 10. Модификация трекера для случая переменного освещения ([2],[10])

Пусть поверхность сцены, на которой найдена особенность сцены, является ламбертовой. Тогда интенсивность освещения точки  $x = P(X)$ , где  $X$  - точка сцены,  $P$  - оператор проектирование,  $x$  - точка на изображении может быть описана как:

$$I(\mathbf{x}, 0) = \nu_E E(X) + \xi_E \quad \forall \mathbf{x} \in W_U,$$

где  $E(X)$  - альbedo (отражающая способность) точки сцены,  $U$  - окрестность точки сцены,  $\nu$  &  $e$  - постоянные параметры, которые представляют изменения контраста и яркости изображения соответственно. При движении камеры эти параметры меняются, т.е. зависят от времени. Изменения освещенности во времени можно записать как:

$$I(\mathbf{x}, 0) = \nu(t)I(\mathbf{x}, t) + \xi(t) \quad \forall \mathbf{x} \in W_U,$$

где

$$\nu(t) = \frac{\nu_E(t)}{\nu_E(0)}, \quad \xi(t) = \xi_E(t) - \frac{\nu_E(t)}{\nu_E(0)}\xi_E(0)$$

при  $t > 0$ .

Объединив аффинное движение окрестности особенности с изменением освещенности, получим выражение:

$$I(\mathbf{x}, 0) = \nu(t)I(A\mathbf{x} + \mathbf{d}, t) + \xi(t) \quad \forall \mathbf{x} \in W$$

Из-за шума в изображении, а также из-за приближенного моделирования движения и изменения освещенности это уравнение в реальности никогда не будет выполняться, поэтому задача слежения состоит в минимизации разности между окрестностями текущего и нового положения особенности:

$$\epsilon = \int_W [I(\mathbf{x}, 0) - \nu I(A\mathbf{x} + \mathbf{d}, t) + \xi]^2 w(\mathbf{x}) d\mathbf{x} \quad (7)$$

где  $w(\mathbf{x})$  - весовая функция.

С помощью разложения в ряд Тейлора в окрестности  $\mathbf{d}=0$ ,  $\nu = 1$ ,  $(\xi) = 0$  получим:

$$\nu I(\mathbf{y}, t) + \xi \simeq \nu I(\mathbf{x}, t) + \xi + \nabla I \frac{\partial \mathbf{y}}{\partial \mathbf{u}} (\mathbf{u} - \mathbf{u}_0)$$

$$\mathbf{y} = A\mathbf{x} + \mathbf{d} \quad A = \{d_{ij}\}$$

$$A = \{d_{ij}\} \quad \mathbf{d} = [d_1 \ d_2]^T$$

$$\mathbf{u} = [d_{11} \ d_{12} \ d_{21} \ d_{22} \ d_1 \ d_2]^T$$

$$\mathbf{u}_0 = [1 \ 0 \ 0 \ 1 \ 0 \ 0]$$

Перепишем получившееся равенство в матричной форме, получим следующее уравнение:

$$I(\mathbf{x}, 0) = F(\mathbf{x}, t)^T z$$

где

$$F(\mathbf{x}, t) = [xI_x \ yI_x \ xI_y \ yI_y \ I_x \ I_y \ I \ 1]^T$$

$$z = [d_{11} \ d_{12} \ d_{21} \ d_{22} \ d_1 \ d_2 \ \nu \ \xi]^T$$

Домножив на  $F(\mathbf{x}, t)^T$ , и проинтегрировав по всему окну особенности  $W$  с весовой функцией  $w$  мы получим систему уравнений  $8 \times 8$ :

$$Sz = a$$

$$a = \int_W F(\mathbf{x}, t)^T I(\mathbf{x}, 0) w(\mathbf{x}) d\mathbf{x}$$

$$S = \int_W F(\mathbf{x}, t)^T F(\mathbf{x}, t) w(\mathbf{x}) d\mathbf{x}$$

Заменяя интегрирование на сумму по всем пикселям в окне  $W$ , мы приходим к следующей системе линейных уравнений:

$$Sz = a$$

$$S = \sum_{\mathbf{x} \in W} \begin{bmatrix} T & U \\ U^T & V \end{bmatrix} w(\mathbf{x})$$

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

$$U^T = \begin{bmatrix} x I_x I & y I_x I & x I_y I & y I_y I & I_x I & I_y I \\ x I_x & y I_x & x I_y & y I_y & I_x & I_y \end{bmatrix}$$

$$V = \begin{bmatrix} I^2 & I \\ I & 1 \end{bmatrix}$$

Если матрица S получается обратимой, то решение системы линейных уравнений можно записать в виде:

$$z = S^{-1}a$$

Как во всех алгоритмах слежения, система решается итеративно по методу Ньютона-Рафсона. Итерации происходят до тех пор, пока изменения решения не становятся пренебрежимо малы.

## 11. Нормализация окрестностей изображения для учета изменений освещения

Для учета изменения освещенности можно использовать вместо стандартной меры SSD нормализованную:

$$\epsilon = \sum_{\mathcal{W}} \left[ \frac{J(A\mathbf{x} + \mathbf{d}) - \mu_v}{\sigma_v} - \frac{I(\mathbf{x}) - \mu_f}{\sigma_f} \right]^2$$

где  $v$  - средний уровень яркости в окне, а  $\sigma$  - стандартная дисперсия уровня яркости в окне.

## 12. Автоматический отбор некачественных особенностей

Авторы Shi-Tomasi-Kanade в своей работе показывают, что непосредственно для слежения за особенностями больше всего подходит обычный Tomasi-Kanade алгоритм. Алгоритм с учетом аффинных искажений в этом случае можно использовать для определения текущего качества особенности, т.е. степени близости ее окна к окну детектированной особенности. Однако они не предлагают никаких автоматических методов определения порога, когда особенность признается плохой и перестает отслеживаться. Этот недостаток попытались исправить авторы [6]-[9], с помощью введения правила X84.

Предположим, что после точного вычисления движения особенности, яркость соответствующих пикселей будет совпадать с точностью до некоторой ошибки, распределенной по нормальному закону (гауссову шуму):

$$I(\delta(\mathbf{x}), t) - I(\mathbf{x}, 0) \simeq \eta(0, 1)$$

Т.к. квадрат нормального распределения представляет собой распределение хи-квадрат, мы получаем:

$$[I(\delta(\mathbf{x}), t) - I(\mathbf{x}, 0)]^2 \simeq \chi^2(1)$$

Сумма  $n$  переменных, с распределением по хи-квадрат с одной степенью свободы, распределена как хи-квадрат с  $n$  степенями свободы. Поэтому, разница между особенностями по окну  $N \times N$   $\mathcal{W}$  имеет вид:

$$\epsilon = \sum_{\mathcal{W}} [I(\delta(\mathbf{x}), t) - I(\mathbf{x}, 0)]^2 \simeq \chi^2(N^2).$$

При возрастании степени свободы, распределение хи-квадрат приближается к нормальному распределению. При степени свободы больше 30, нормальное распределение можно использовать как приближение хи-квадрат. Если размер окна особенности, по крайней мере, 7\*7, то можно спокойно утверждать, что:

$$\epsilon \simeq \eta(N^2, 2N^2)$$

Поэтому, если два окна, которые мы сравниваем, принадлежат плохой особенности, то разница должна быть выбросом (outlier) по отношению к нормальному распределению разницы между хорошими особенностями. Для определения таких выбросов предлагается использовать правило X84, являющееся устойчивым за счет использования медианы и медианной дисперсии, вместо обычного среднего и дисперсии.

Согласно правилу, все значения, отклоняющиеся от медианы на более чем k Медианных абсолютных дисперсий (МАД, Median Absolute Deviation - MAD):

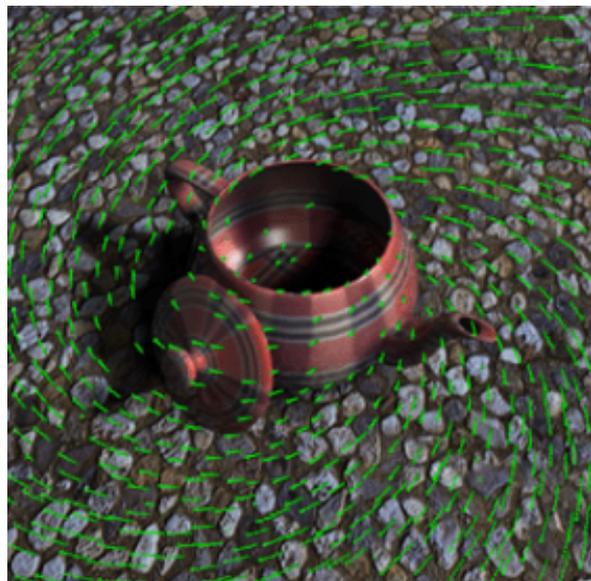
$$MAD = \underset{i}{\text{med}} \{ |\epsilon_i - \underset{j}{\text{med}} \epsilon_j| \},$$

где  $\epsilon$  - разница между текущим окном особенности и ее окном в первом кадре.

При значении  $k=5.2$  это соответствует примерно 3.5 стандартным дисперсиям, и интервал  $[\mu-3.5\sigma, \mu+3.5\sigma]$  содержит более 99.9% значений нормального распределения.

Как утверждается, правило позволяет эффективно определять уменьшение качества особенности и отсекал ее во всех случаях, при которых количество выбросов меньше 50%.

### 13. Пример трекинга особенностей



## 14. Ссылки:

1. Pollefev "Tutorial on 3d reconstruction" 2000  
<http://www.esat.kuleuven.ac.be/~pollefev/tutorial/>
2. Lucas, Kanade "An iterative image registration technique with an application to stereo vision" 1981
3. Tomasi, Kanade "Detection and tracking of point features" TR Carnegie-Melon University, 1991
4. Shi, Tommasini "Good features to track" 1994
5. Smith, Sinclair, Cipolla, Wood "Effective corner matching"
6. Tommasini, Fusiello, Roberto, Trucco "Robust feature tracking" 1998
7. Tommasini, Fusiello, Trucco, Roberto "Making good features track better" 1998
8. Fusiello, Trucco, Tommasini, Roberto "Improving feature tracking with robust statistics"
9. Tommasini, Fusiello, Roberto, Trucco "Robust feature tracking in underwater sequence"
10. Jin, Favaro, Soatto "Real-time tracking and outlier rejection with changes in illumination" 2001

---

(c) Graphics & Media lab ([webmaster@graphics.cs.msu.su](mailto:webmaster@graphics.cs.msu.su))

При использовании материалов в сети Интернет или бумажной прессе ссылка на сайт ([cgm.graphicon.ru](http://cgm.graphicon.ru)) обязательна.