

А.В. Козырева

ОПРЕДЕЛЕНИЕ КООРДИНАТ МОБИЛЬНОГО УСТРОЙСТВА В ПРОСТРАНСТВЕ НА ОСНОВЕ ИЗОБРАЖЕНИЙ, ПОЛУЧАЕМЫХ ОТ ЕГО ВИДЕОКАМЕРЫ

ВВЕДЕНИЕ

В статье описаны алгоритмы, необходимые для решения задачи позиционирования мобильного устройства в пространстве относительно окружающих объектов. Под мобильными устройствами будем понимать смартфоны (мобильные телефоны, обладающие операционной системой) и КПК (карманные компьютеры), обладающие «зрением», т.е. видеокамерой. Видеокамера — наш единственный источник восприятия окружающего пространства. На основе поступающих с камеры снимков будет проходить анализ изменения положения в пространстве.

Нам с камеры в постоянном потоке поступают изображения. Будем сравнивать изображения попарно и на основе различий между ними вычислять на сколько и как сдвинулось мобильное устройство.

Таким образом в каждый момент времени имеем два изображения. Нам совершенно не нужно сравнивать два полученных изображения полностью. Нам достаточно рассмотреть лишь некоторую их часть. Т.е. сначала мы должны найти на изображении точки, следя за которыми мы поймем как изменилось положение. Такие точки называют «опорными» или «особыми». Найдя особые точки на первом изображении нам необходимо отыскать их на втором. Здесь нам потребуются алгоритмы слежения за особенностями. Следует отметить, что если на втором изображении «исчезли» все «опорные» точки первого изображения, то мы не сможем решить поставленную задачу. Чтобы этого избежать нам необходимо либо очень часто делать снимки, либо следить за большим числом точек. Для решения задачи нам требуется минимум четыре точки на одном изображении и соответствующие четыре точки на втором. Максимальное число точек ограничено всем изображением. Так как наши «опорные» точки на получаемых изображениях имеют две координаты (изображение является плоским), а «опорная» точка в пространстве имеет 3 координаты, то нам необходимо знать правила по которым происходит данное преобразование. Данные правила зависят от внутренних параметров фотокамеры, т.е. изначально

нам необходимо провести так называемую калибровку камеры. Данную процедуру можно проводить различными способами, которые отличаются лишь точностью. В процессе разработки нашей программной системы необходимо будет отыскать наиболее оптимальное сочетание качества и скорости работы. Само определение изменилось ли положение в пространстве или нет, производится путем поиска неизвестных коэффициентов в матрице вращения и векторе сдвига.

Таким образом задача определения изменения положения в пространстве мобильного устройства разбивается на три части, при этом внутренние параметры камеры ищутся лишь один раз:

- Поиск внутренних параметров камеры
- Найти соответствующие «особые» точки на изображениях
- Посчитать матрицу поворотов и вектор сдвига.

1. ВНУТРЕННИЕ ПАРАМЕТРЫ КАМЕРЫ

1.1. Модель регистрирующей камеры

Строго говоря, различные точки пространства предметов отображаются оптической системой камеры в пространстве изображений на различных расстояниях от фокальной плоскости. Однако, если расстояние между камерой и наблюдаемой сценой значительно превышает фокусное расстояние оптической системы, можно считать, что изображение строится в ее фокальной плоскости. В этом случае можно воспользоваться проективной моделью камеры, в которой изображение трехмерного объекта получается проектированием его в фокальную плоскость (плоскость изображения) через единственную точку, называемую оптическим центром. Прямая линия, перпендикулярная плоскости изображения и проходящая через эту точку, называется оптической осью камеры, а точка пересечения оптической оси с плоскостью изображения — главной точкой.

Определим в трехмерном пространстве ортогональную правую систему координат $OXYZ$, начало которой совпадает с оптическим центром, ось OZ — с оптической осью камеры. Такая система называется стандартной системой координат камеры. Пусть плоскость изображения находится на расстоянии f от оптического центра. В этой плоскости зададим систему координат oxy с началом в главной точке и осями ox и oy , параллельными осям OX и OY соответственно (рис. 1). Легко убедиться, что в стандартной

системе координат проекцией точки трехмерного пространства M с координатами (X, Y, Z) является точка m в плоскости изображения с координатами (x, y) , причем $x = fX / Z$, $y = fY / Z$.

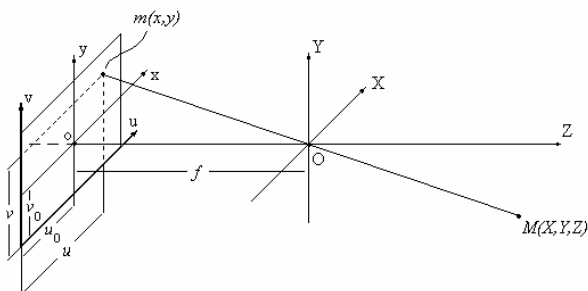


Рис. 1. Система координат проективной камеры

Для полного описания камеры следует учесть, что для регистрации изображения в плоскости изображения камеры помещается какой-либо фотоприемник. В общем случае измерение координат в фотоприемнике осуществляется в единицах, отличных от единиц, задающих координаты в стандартной системе. Поэтому для полного описания камеры необходимо выразить координаты точки m в естественных единицах фотоприемника. В достаточно общем для любых фотоприемников виде (рис. 1) это может выглядеть как

$$u = x / w + u_0, \quad v = y / h + v_0,$$

где (u_0, v_0) — координаты главной точки относительно начала координат фотоприемника (в естественных координатах фотоприемника); w и h — масштабы вдоль осей ox и oy (например, расстояния между ячейками матричного фотоприемника вдоль строк и столбцов).

В новой системе координаты проекции точки m примут вид:

$$u = \frac{fX}{wZ} + u_0, \quad v = \frac{fY}{hZ} + v_0. \quad (1.1)$$

Для последующего изложения введем трехмерный вектор $\mathbf{M} = (X, Y, Z)^T$, соответствующий точке M , и двумерный вектор

$\mathbf{m} = (x, y)^T$, соответствующий точке m . Определим также вектор однородных внутренних координат камеры $\mathbf{v} = (u, v, 1)^T$. Используя эти обозначения, соотношения (1.1) можно представить в компактной векторно-матричной записи:

$$Z\mathbf{v} = \mathbf{A}\mathbf{M}, \quad (1.2)$$

где $\mathbf{A} = \begin{bmatrix} f/w & 0 & u_0 \\ 0 & f/h & v_0 \\ 0 & 0 & 1 \end{bmatrix}$ — матрица, известная под названием матрицы

внутренних параметров камеры, поскольку она содержит только параметры оптической системы и фотоприемника камеры.

1.2. Калибровка камеры

Внутренние параметры камеры находим самокалибровкой камеры по плоскому шаблону.

Пусть у нас есть точка на плоскости $m = [u, v]^T$, которая является проекцией некоторой точки $M = [X, Y, Z]^T$ в пространстве. Соотношение между этими двумя точками можно представить в следующем виде:

$$sm = \mathbf{A}[\mathbf{R} \ t]M \quad (1.3)$$

где s — некоторый скаляр, пара $(\mathbf{R} \ t)$ — так называемые внешние параметры камеры, и \mathbf{A} — матрица, представляющая собой внутренние параметры

камеры: $A = \begin{bmatrix} \alpha & \gamma & x_0 \\ 0 & \beta & y_0 \\ 0 & 0 & 1 \end{bmatrix}$, где (x_0, y_0) — координаты основной точки, α и

β — коэффициенты сжатия по осям Ox и Oy , соответственно, и γ — коэффициент асимметрии между осями изображения.

Без потери общности можем принять $Z = 0$. Обозначим через r_i i -й столбец матрицы вращения \mathbf{R} . Тогда из (1.3) для точки на объекте $\tilde{M} = [X, Y, 1]^T$ и соответствующей точки на изображении \tilde{m} имеет место следующее соотношение:

$$s\tilde{m} = \mathbf{H}\tilde{M}, \quad (1.4)$$

где \mathbf{H} — матрица гомографии изображения, $\mathbf{H} = \mathbf{A} [r_1 \ r_2 \ t]$.

В [1] приводится способ расчета матрицы гомографии. Мы предположим, что данная матрица нами уже вычислена.

Положим, что $H = [h_1 \ h_2 \ h_3]$, тогда из (3.4) мы получим:

$$[h_1 \ h_2 \ h_3] = \lambda A [r_1 \ r_2 \ t],$$

где λ — скаляр.

Используя, что r_1 и r_2 ортогональны, получаем

$$h_1^T A^{-T} A^{-1} h_2 = 0 \quad (1.5)$$

$$h_1^T A^{-T} A^{-1} h_1 = h_2^T A^{-T} A^{-1} h_2 \quad (1.6)$$

Положим

$v_{ij} = [h_{i1}h_{j1}, h_{i1}h_{j2} + h_{i2}h_{j1}, h_{i2}h_{j2}, h_{i3}h_{j1} + h_{i1}h_{j3}, h_{i3}h_{j2} + h_{i2}h_{j3}, h_{i3}h_{j3}]^T$, тогда (1.5) и (1.6) можно переписать в следующем виде:

$$\begin{bmatrix} v_{12}^T \\ (v_{11} - v_{22})^T \end{bmatrix} b = 0. \quad (1.7)$$

В (1.7) b — это 6D вектор, равный $[B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33}]$, где B_{ij} — элементы матрицы $B = A^{-T} A^{-1}$. Используя n изображений шаблона и ниже указанные уравнения (1.8), мы можем вычислить внутренние параметры камеры (матрицу A) при помощи (1.7).

$$\begin{aligned} v_0 &= (B_{12}B_{13} - B_{11}B_{23}) / (B_{11}B_{22} - B_{12}^2) \\ \lambda &= B_{33} - [B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})] / B_{11} \\ \alpha &= \sqrt{\lambda / B_{11}} \\ \beta &= \sqrt{\lambda B_{11} / (B_{11}B_{22} - B_{12}^2)} \\ \gamma &= -B_{12}\alpha^2\beta / \lambda \\ u_0 &= cv_0 / \alpha - B_{13}\alpha^2 / \lambda \end{aligned} \quad (1.8)$$

Более подробно процесс расчета матрицы внутренних параметров камеры и матриц гомографии изображений можно прочитать в [1] и [2]. Там же приведены результаты работы алгоритмов на реальных объектах.

2. «ОСОБЕННОСТИ» ИЗОБРАЖЕНИЙ

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 $o_2(m)$. Точечной особенностью сцены должна соответствовать точечная особенность изображений. Обратное неверно: существуют такие особые точки изображения, которым не соответствует никакие особые точки сцены. Такие точки называются ложными особенностями сцены.

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.

Методы построения таких моделей в последнее десятилетия бурно развивались. 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 последовательности 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, обычная 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.

2.1. Слежение за точечными особенностями

В общем случае под слежением за точечными особенностями сцены 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 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.

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).

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 производятся 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.

2.2. 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
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.

2.3. 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 [3]. Для каждого пикселя изображения вычисляется значение особой функции отклика угла, 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 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.

2.4. 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*.

- *Lucas—Kanade* — особенность считается только смещающейся, без искажений [5];
- *Tomasi—Kanade* — переформулирование *Lucas—Kanade*. Движение считается смещением, и рассчитывается путем итеративного решения построенной системы линейных уравнений [6];
- *Shi—Tomasi—Kanade* — учитывает аффинные искажения особенности [4];
- *Jin—Favaro—Soatto* — модификация *Shi—Tomasi—Kanade* с учетом аффинных изменений освещенности особенности [8].

2.5. Алгоритм «Lucas—Kanade»

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: $E = \sum_{x \in R} [F(x+h) - G(x)]^2$, где $F(x)$, $G(x)$ — две функции.

Функцию $F(x)$ с помощью разложения в ряд Тейлора можно приближенно представить в виде: $F(x+h) \approx F(x) + h \frac{\partial F}{\partial x}$, где

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

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

$$0 = \frac{\partial}{\partial h} E \approx \frac{\partial}{\partial h} \sum_x [F(x) + h \frac{\partial F}{\partial x} - G(x)]^2 = \sum_x 2 \frac{\partial F}{\partial x} [F(x) + h \frac{\partial F}{\partial x} - G(x)].$$

Отсюда смещение h можно получить как

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

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.

2.6. Алгоритм «Tomasi—Kanade»

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: $\zeta = \sum_W [I(x+d, t+\tau) - I(x, t)]^2$

Опять же аналогично функция изображения раскладывается с помощью ряда Тейлора:

$$I(x+d, t+\tau) \approx I(x, t) + \nabla I(x, t)^T d + I_t(x, t)\tau, \text{ где} \\ \nabla I^T = [I_u, I_v] = [\partial I / \partial u, \partial I / \partial v], \quad I_t = \partial I / \partial t.$$

Тогда разницу между окнами по мере можно переписать в виде:

$$\zeta \approx \sum_W (\nabla I(x, t)^T d + I_t(x, t)\tau)^2.$$

Продифференцировав это выражение по d и приравняв производную к нулю, получаем линейную систему относительно d : $Cd=g$, где:

$$C = \sum_W \begin{bmatrix} I_u^2 & I_u I_v \\ I_u I_v & I_v^2 \end{bmatrix} \text{ и } g = -\tau \sum_W I_t [I_u I_v]^T.$$

Из этой системы d получается как: $d_k = C^{-1}g$.

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:

$$\begin{cases} d_0 = 0 \\ d_{k+1} = d_k + C^{-1} \sum_W [(I(x, t) - I(x + d_k, t + 1)) \nabla I(x, t)] \end{cases}$$

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.

2.7. Алгоритм «Shi—Tomasi—Kanade»

В этом алгоритме впервые учитываются аффинные искажения изображения окрестности особых точек, поэтому движение пикселей окна особенности описывается в виде $Ax + d$, где A — матрица (2×2) , а d — смещение (2×1) .

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

$$\zeta = \iint_W [J(\Delta x + d) - I(x)]^2 w(x) dx$$
, где W — окно особенности, а w — весовая функция (может использовать, а может и быть равна 1 во всем окне), $J(x)$ и $I(x)$ — два изображения.

Выражение дифференцируется относительно параметров движения, и производная приравнивается к 0. Затем система линеаризуется с помощью разложения функции изображения в ряд Тейлора: $J(\Delta x + d) = J(x) + g^T(u)$. Это дает нам линейную 6×6 систему: $Tz = a$, где в векторе z объединены все искомые параметры: $z^T = [d_{xx} d_{yx} d_{xy} d_{yy} d_x d_y]$.

Вектор ошибки a записывается в виде:

$$a = \iint_W [I(x) - J(x)] \begin{bmatrix} xg_x \\ xg_y \\ yg_x \\ yg_y \\ g_x \\ g_y \end{bmatrix} w dx.$$

А матрицу размерности 6×6 T можно представить следующим образом:

$$T = \iint_W \begin{bmatrix} U & V \\ V^T & Z \end{bmatrix} w dx,$$

$$U = \begin{bmatrix} x^3 g_x^3 & x^3 g_x g_y & xy g_x^3 & xy g_x g_y \\ x^3 g_x g_y & x^3 g_y^3 & xy g_x g_y & xy g_y^3 \\ xy g_x^3 & xy g_x g_y & y^3 g_x^3 & y^3 g_x g_y \\ xy g_x g_y & xy g_y^3 & y^3 g_x g_y & y^3 g_y^3 \end{bmatrix},$$

$$V^T = \begin{bmatrix} xg_x^3 & xg_x g_y & yg_x^3 & yg_x g_y \\ xg_x g_y & xg_y^3 & yg_x g_y & yg_y^3 \end{bmatrix},$$

$$Z = \begin{bmatrix} g_x^3 & g_x g_y \\ g_x g_y & g_y^3 \end{bmatrix}.$$

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*.

2.8. Автоматический отбор некачественных особенностей

Авторы алгоритма *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 [5–7], 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): $I(\delta(x), t) - I(x, 0) \approx \eta(0, 1)$.

Т.к. квадрат нормального распределения представляет собой распределение хи-квадрат, мы получаем: $[I(\delta(x), t) - I(x, 0)]^2 \approx \chi^2(1)$.

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 $N \times N$ W имеет вид: $\zeta = \sum_w [I(\delta(x), t) - I(x, 0)]^2 \approx \chi^2(N^2)$.

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: $\zeta \approx \eta(N^2, 2N^2)$.

Поэтому, если два окна, которые мы сравниваем, принадлежат плохой особенности, то разница должна быть выбросом 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 (Median Absolute Deviation — MAD):

$$MAD = \underset{i}{mad} \{ | \zeta_i - \underset{j}{mad} \zeta_j | \},$$

где ζ — 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 $[u-3.5\sigma, u+3.5\sigma)$ содержит более 99.9% значений нормального распределения.

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

3. МАТРИЦА ПОВОРОТОВ И ВЕКТОР СДВИГА

Рассмотрим общий случай, когда оптические оси камер не параллельны, и направление смещения оптического центра одной камеры относительно оптического центра другой произвольно (рис. 2). Введем для каждой камеры свою стандартную систему координат. Пусть первой камере соответствует система координат $O'X'Y'Z'$, а второй — $O''X''Y''Z''$ (рис. 2). Пусть вектор $\mathbf{M}' = (X', Y', Z')^T$ характеризует координаты некоторой точки M трехмерного пространства в системе первой камеры, а вектор $\mathbf{M}'' = (X'', Y'', Z'')^T$ — в системе второй. Переход от глобальной системы координат к стандартным системам первой и второй камер осуществляется с помощью преобразований $\mathbf{M}' = \mathbf{R}\mathbf{M} + \mathbf{t}'$ и $\mathbf{M}'' = \mathbf{R}''\mathbf{M} + \mathbf{t}''$ соответственно. Учитывая это, легко показать, что связь между векторами \mathbf{M}' и \mathbf{M}'' задается соотношением

$$\mathbf{M}'' = \mathbf{R}\mathbf{M}' + \mathbf{t}, \tag{3.1}$$

где $\mathbf{R} = \mathbf{R}''\mathbf{R}'^T$ — ортогональная матрица, описывающая ориентацию системы координат второй камеры относительно первой, а $\mathbf{t} = -\mathbf{R}''\mathbf{R}'^T\mathbf{t}' + \mathbf{t}''$ — вектор трансляции, определяющий положение оптического центра второй камеры в системе координат первой. Матрицу \mathbf{R} и вектор \mathbf{t} принято называть внешними параметрами системы регистрации.

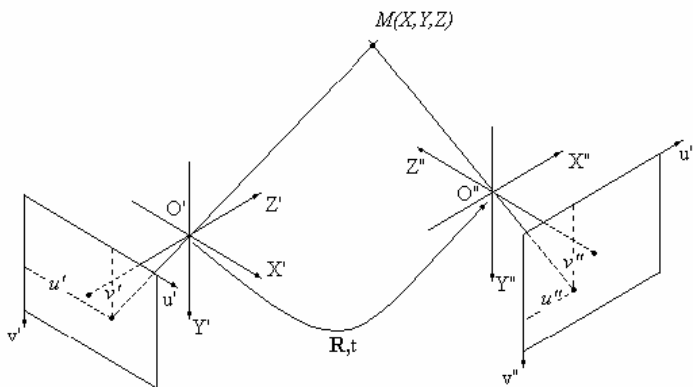


Рис. 2. Система двух произвольно ориентированных камер

В общем случае трехмерные координаты точки могут быть заданы в системе, не совпадающей со стандартной системой координат камеры (назовем ее глобальной). Пусть $OXYZ$ — глобальная система координат, а $O'X'Y'Z'$ — стандартная система координат камеры. Переход от системы $OXYZ$ к системе $O'X'Y'Z'$ можно осуществить поворотом координатных осей к системе $OX''Y''Z''$ и последующим смещением начала координат. Тогда связь между координатами точки M в глобальной и стандартной системе может быть представлена как

$$\mathbf{M}' = \mathbf{R}\mathbf{M} + \mathbf{t}, \quad (3.2)$$

где \mathbf{M} и \mathbf{M}' — векторы пространственных координат точки M в глобальной и стандартной системах, соответственно; \mathbf{R} — матрица размерности 3×3 , описывающая поворот стандартной системы координат относи-

тельно глобальной; компонентами матрицы являются направляющие косинусы осей глобальной системы в стандартной системе координат ; \mathbf{t} — трехмерный вектор смещения начала координат глобальной системы относительно начала координат стандартной.

Теперь у нас есть как минимум четыре «особые» точки (см. ранее) и мы знаем как они связаны ((3.2)). Найдем матрицу поворота и вектор сдвига. У нас есть уравнения вида:

$$\begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} = R \begin{pmatrix} U_i \\ V_i \\ W_i \end{pmatrix} + t$$

$$\begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} = A^{-1} \begin{pmatrix} x_i \\ y_i \end{pmatrix}, \quad i=1..4,$$

$$\begin{pmatrix} U_i \\ V_i \\ W_i \end{pmatrix} = A^{-1} \begin{pmatrix} u_i \\ v_i \end{pmatrix}$$

где $(x_i, y_i)^T$ — точка в плоскости первого изображения, являющаяся проекцией точки в пространстве $(X_i, Y_i, Z_i)^T$, $(u_i, v_i)^T$ — это точка в плоскости второго изображения, являющаяся проекцией точки $(U_i, V_i, W_i)^T$ в пространстве, A^{-1} — матрица внутренних параметров камеры, R и t — искомые матрица вращения и вектор сдвига соответственно.

Каждое уравнение указано в «своих» координатах: первое в глобальной системе, второе в системе камеры в первом положении, третье в системе камеры, находящейся во втором положении.

С учетом того, что нам известны двухмерные координаты точки на каждом изображении и матрица внутренних параметров, мы можем считать, что и трехмерные координаты нам также известны. Таким образом, получа-

ем четыре уравнения вида $\begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} = R \begin{pmatrix} U_i \\ V_i \\ W_i \end{pmatrix} + t$. Распишем их более подробно:

$$\begin{cases} X_1 = r_{11}U_1 + r_{12}V_1 + r_{13}W_1 + t_1 \\ X_2 = r_{11}U_2 + r_{12}V_2 + r_{13}W_2 + t_1 \\ X_3 = r_{11}U_3 + r_{12}V_3 + r_{13}W_3 + t_1 \\ X_4 = r_{11}U_4 + r_{12}V_4 + r_{13}W_4 + t_1 \end{cases}, \begin{cases} Y_1 = r_{21}U_1 + r_{22}V_1 + r_{23}W_1 + t_2 \\ Y_2 = r_{21}U_2 + r_{22}V_2 + r_{23}W_2 + t_2 \\ Y_3 = r_{21}U_3 + r_{22}V_3 + r_{23}W_3 + t_2 \\ Y_4 = r_{21}U_4 + r_{22}V_4 + r_{23}W_4 + t_2 \end{cases},$$

$$\begin{cases} Z_1 = r_{31}U_1 + r_{32}V_1 + r_{33}W_1 + t_3 \\ Z_2 = r_{31}U_2 + r_{32}V_2 + r_{33}W_2 + t_3 \\ Z_3 = r_{31}U_3 + r_{32}V_3 + r_{33}W_3 + t_3 \\ Z_4 = r_{31}U_4 + r_{32}V_4 + r_{33}W_4 + t_3 \end{cases}.$$

Мы видим, что неизвестные каждой системы не зависят от неизвестных других систем. В каждой системе 4 неизвестных: три из матрицы вращения и одна неизвестная из вектора сдвига. Поэтому нам и требовалось лишь четыре «особых» точки. Каждая система представляет линейные уравнения. Распишем более подробно первую систему:

$$\begin{cases} X_1 = r_{11}U_1 + r_{12}V_1 + r_{13}W_1 + t_1 \\ X_2 = r_{11}U_2 + r_{12}V_2 + r_{13}W_2 + t_1 \\ X_3 = r_{11}U_3 + r_{12}V_3 + r_{13}W_3 + t_1 \\ X_4 = r_{11}U_4 + r_{12}V_4 + r_{13}W_4 + t_1 \end{cases} \Rightarrow \begin{pmatrix} U_1 & V_1 & W_1 & 1 \\ U_2 & V_2 & W_2 & 1 \\ U_3 & V_3 & W_3 & 1 \\ U_4 & V_4 & W_4 & 1 \end{pmatrix} \begin{pmatrix} r_{11} \\ r_{12} \\ r_{13} \\ t_1 \end{pmatrix} = \begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{pmatrix} \Rightarrow A \begin{pmatrix} r_{11} \\ r_{12} \\ r_{13} \\ t_1 \\ -1 \end{pmatrix} = 0,$$

$$\text{где } A = \begin{pmatrix} U_1 & V_1 & W_1 & 1 & X_1 \\ U_2 & V_2 & W_2 & 1 & X_2 \\ U_3 & V_3 & W_3 & 1 & X_3 \\ U_4 & V_4 & W_4 & 1 & X_4 \end{pmatrix}.$$

Приведем A к верхнетреугольному виду и получим решение:

$$\begin{cases} t_1 = -A_{45} \\ r_{13} = -A_{35} - t_1 \\ r_{12} = -A_{23}r_{13} - t_1 - A_{25} \\ r_{11} = -A_{12}r_{12} - A_{13}r_{13} - t_1 - A_{15} \end{cases}.$$

Или в общем виде:

$$\begin{aligned}
 t_i &= -A_{45} \\
 r_{i3} &= -A_{35} - t_i \\
 r_{i2} &= -A_{23}r_{i3} - t_i - A_{25} \\
 r_{i1} &= -A_{12}r_{i2} - A_{13}r_{i3} - t_i - A_{15}
 \end{aligned}
 \quad i=1,2,3,$$

где A — это верхнетреугольная матрица, полученная для каждой системы уравнений.

Таким образом, мы нашли как изменилось положение камеры в пространстве на основе двух снимков.

СПИСОК ЛИТЕРАТУРЫ

1. **Harker M., O'Leary P.** Computation of Homographies // Department of Product Engineering University of Leoben, Австрия.
2. **Zhang Z.** A Flexible New Technique for Camera Calibration // Microsoft Research, One Microsoft Way, США, Рэдмонд — 1998.
3. **Pollefeý M.** Tutorial on 3d reconstruction // 2000 <http://www.esat.kuleuven.ac.be/~pollefeý/tutorial/>
4. **Shi J., Tomasi C.** Good features to track. // IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'94), IEEE Computer Society, Сиэтл — 1994. <http://citeseer.ist.psu.edu/shi94good.html>
5. **Lucas B., Kanade T.** An Iterative Image Registration Technique with an Application to Stereo Vision // Proc. of 7th International Joint Conference on Artificial Intelligence (IJCAI), 1981
6. **Kanade T.** Detection and tracking of point features // TR Carnegie-Melon University, 1991
7. **Smith P., Sinclair D., Cipolla R., Wood K.** Effective Corner Matching. // In Proc. of BMVC'98, V.2 — P.545–556, Великобритания — 1998.
8. **Jin H., Favaro P., Soatto S.** Real-time Feature Tracking and Outlier Rejection with Changes in Illumination. // Intl. Conf. on Computer Vision, 2001
9. **Конущин А.** Слежение за точечными особенностями сцены, <http://ict.edu.ru/ft/002409/num4pntrac.pdf>