

Возможность трекинга камеры с помощью дополнительной соосной широкоугольной камеры

А.А. Савенко, Б.Б. Морозов, И.Г. Таранцев
Институт Автоматики и Электрометрии СО РАН,
Новосибирск, Россия
egor@sl.iae.nsk.su

Аннотация

Рассматривается возможность реализации трекинга камеры в реальном масштабе времени с помощью дополнительной соосной камеры с широким углом обзора. Предлагается формат трекингового изображения в виде штрих-кода, облегчающий процесс его выделения на исходном изображении. Рассматриваются проблемы реализации и влияние шумов.

Ключевые слова: Трекинг камеры, виртуальная студия.

1. ВВЕДЕНИЕ

При построении виртуальных студий вопрос трекинга камер приобретает первостепенное значение. Для достижения правдоподобности система синтеза виртуальных декораций должна изменять положение виртуальной камеры синхронно с изменением положения реальной камеры. Вопрос синхронности получения информации о положении камеры в пространстве наиболее успешно решается при использовании оптических систем трекинга, когда для определения положения камеры используется некоторое изображение, поступающее синхронно с изображением основной камеры.

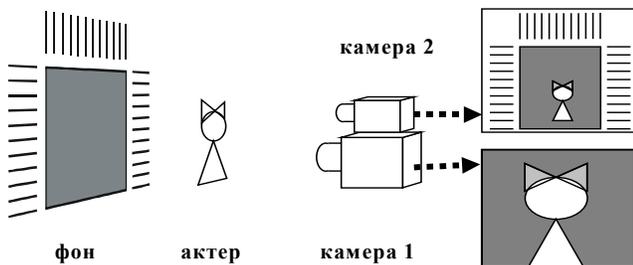


Рис. 1. Внешний вид студии с системой трекинга камеры

В системе Orad Pattern Recognition [1] используется трекинговое изображение, нанесенное непосредственно на одноцветный фон за актером, что приводит к заметному ухудшению качества рирпроекции. В системе Radamec EPO [2] используется дополнительная камера, жестко закрепленная с основной камерой, направленная на потолок, на который нанесено трекинговое изображение. Эта система не работает в небольших студиях с низкими потолками, либо при использовании оборудования, закрепленного на потолке (например, светового оборудования). Трекинговое изображение можно расположить непосредственно вокруг фона, где, как правило, не располагается дополнительного оборудования. В этом случае дополнительная камера, жестко закрепленная на основной камере, соосная с ней, направленная на фон, с широким углом обзора будет постоянно «видеть» трекинговое изображение, расположенное вокруг фона. В этом случае для определения текущего фокусного расстояния основной камеры можно использовать простые механические датчики,

обеспечивающие достаточную точность. Перед нами стоит задача оценить возможность построения системы трекинга в данном случае для достижения приемлемой точности совмещения реального и виртуального изображения, а именно половины пикселя изображения основной камеры.

2. РЕАЛИЗАЦИЯ

Рассмотрим математическую модель камеры (рис 2), где O – точка фокуса камеры, F – фокусное расстояние камеры, D – размер светочувствительной матрицы (ширина), α – угол обзора камеры, L – расстояние до сцены, W – ширина сцены.

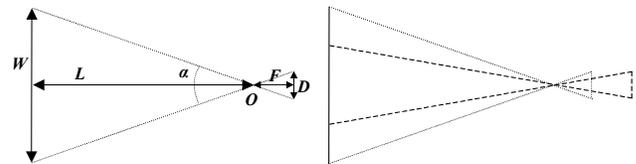


Рис. 2. Математические модели одной камеры и двух соосных камер с общей точкой фокуса

Для двух соосных камер можно выбрать следующую модель (рис 3.), объединив точки фокуса двух камер (штрихованной линией нарисована основная камера, а точками – дополнительная камера с большим углом обзора):

Из соотношений подобия получаем $\frac{W}{L} = \frac{D}{F}$.

Соответственно, K пикселей соответствует участку сцены шириной $L_{px} = W \cdot \frac{K}{N} = \frac{K \cdot L \cdot D}{N \cdot F}$, где N –

горизонтальное разрешение матрицы (обычно 720 пикселей). Отсюда видно, что при смещении камеры относительно сцены по горизонтали (и аналогично по вертикали) изображения с основной и дополнительной камер будут смещаться на разное количество пикселей в соответствии с

отношением фокусных расстояний камер $\frac{K_{осн}}{K_{доп}} = \frac{F_{осн}}{F_{доп}}$.

Рассмотрев смещения камеры к сцене или от нее, а также рассмотрев поворот вокруг каждой из осей камеры можно аналогично показать, что с высокой точностью (до третьего порядка) смещения изображений основной и дополнительной камер соответствуют отношению фокусных расстояний камер.

На практике фокусное расстояние основной камеры колеблется в пределах от 6,5мм до 65мм, что соответствует углу обзора от 50° до 5°. Таким образом, для совмещения виртуальных декораций с актером с точностью до половины пикселя необходимо на порядок повысить точность получения координат камеры в 3D-пространстве по сравнению с обычной камерой. Для этого можно

использовать высококонтрастное трекинговое изображение с большой избыточностью, чтобы статистическими методами повысить точность определения координат основной камеры. Например, можно использовать аналог штрих-кода из большого количества тонких линий, расположенных на определенном расстоянии друг от друга.

Задача построения системы трекинга разбивается на множество более мелких и уже известных задач. Первый этап – выделение из изображения элементов (feature detection) на основании которых определяется положение видеокамеры. Второй этап – получение собственно шести координат видеокамеры на основании известных координат элементов (3D/6DOF Pose Estimation). На третьем этапе точность полученных координат повышается за счет использования сглаживающих фильтров, учитывающих инерционность реальных камер. В свою очередь выделение трекинговых элементов разбивается на следующие этапы, каждый из которых распадается на подэтапы:

- 1) Выделение контурных линий на изображении.
- 2) Выделение групп объектов с заданными параметрами.
- 3) Привязка объектов в этих группах к известным нам объектам в 3D-пространстве.
- 4) Уточнение координат линий на изображении.

2.1 Выделение контурных линий на изображении

Во-первых, изображение сглаживается оператором Гаусса. Затем используется фильтр выделения контуров. Было рассмотрено несколько различных фильтров, например, известный алгоритм Кенни, а так же классические фильтры Перевитт и Собеля. В результате выбор был остановлен на операторе Лапласа. Далее используется простой алгоритм восстановления контуров, объединяющий полученные точки в произвольные контуры. На следующем этапе с помощью метода наименьших квадратов из полученных контуров выбираются контуры, наиболее похожие на прямые линии. А именно, если ошибка приближения контура одной прямой не должна превышать 2-3 пикселей.

2.2 Выделение групп объектов с заданными параметрами

После предыдущего этапа остается очень много разных линий, из которых необходимо выделить трекинговые элементы. Если исходно трекинговым элементам задать некоторые свойства, то по этим свойствам их можно будет отличить от всех других линий, встречающихся в изображении. Мы разбили трекинговые элементы на три группы, расположенные слева, сверху и справа от фона. Внутри каждой группы отрезки параллельны, имеют одинаковую длину и расположены под одним углом к осевой линии, проходящей через центры всех отрезков. Количество элементов каждой группе очень велико (несколько десятков отрезков). Это позволяет последовательно группировать все линии на изображении и отбрасывать группы с числом элементов менее десяти линий. Для группировки используется разработанный нами простой итеративный алгоритм. На каждой итерации все элементы разбиваются на группы по заданному параметру с некоторой допустимой погрешностью. Затем группы с близкими границами объединяются. На следующей итерации допустимая

погрешность уменьшается. Алгоритм останавливается, если группы стабилизируются или допустимая погрешность становится слишком малой. Вначале делается группировка всех линий по углу наклона (т.е. параллельные линии объединяются в одну группу), затем делается группировка по длине, затем выделяются группы, лежащие на одной прямой (выделяется осевая линия группы), и, наконец, сравнивается угол наклона отрезков относительно осевой линии группы. По окончании каждой из группировок из рассмотрения выбрасываются группы с малым числом элементов.

2.3 Привязка объектов в этих группах к известным нам объектам в 3D-пространстве

Для однозначного сопоставления найденных линий к идеальным изображением было решено использовать неповторяющиеся последовательности линий, чтобы даже по небольшому числу найденных линий можно было точно понять, какую часть трекингового изображения мы обрабатываем. Для этого был выработан специальный код.

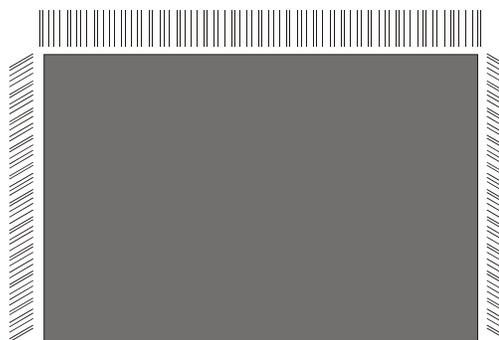


Рис. 3. Пример трекингового изображения

Для каждой линии предусмотрено знакоместо, в пределах которого линия располагается в центре (код 0), немного влево (-1) или немного вправо (+1). Т.е. каждая линия соответствует одному из трех значений -1, 0 или +1. Группирование нескольких линий (например, трех) в один символ позволят создать большое количество символов, достаточных для однозначного различия этих символов. В процессе экспериментов было обнаружено значительное влияние перспективных искажений на точность узнавания разных символов. Также для однозначной идентификации начала символа было решено каждый символ начинать парой линий, расположенных максимально близко друг к другу (т.е. использовать последовательность +1 и -1). Соответственно при кодировании символов было запрещено использовать последовательность +1,-1. Для корректного разделения соседних линий ширина знакоместа одной линии должна быть не менее 10 пикселей. При ширине строки в 720 пикселей это означает не более 70 разных линий. В итоге был выбран код из пяти линий на один символ: первые две линии всегда +1, -1, а следующие три передают идентификатор символа. При распознавании трекинговых элементов в выделенных группах вначале ищется самая близкая пара линий, которая характеризует масштаб текущего изображения. Затем выделяются все начала символов, а затем в интервалах между началами символов ищется корреляция существующих линий со всеми возможными символами нашего кода. По максимуму корреляции выбирается идентификатор конкретного кода.

2.4 Уточнение координат линий на изображении

После того, как все линии сопоставлены элементам кода, делается обратное сопоставление и в том месте изображения, где располагается или должен располагаться трекинговый элемент запускается высокоточный алгоритм поиска координат трекинговых элементов. Для этого производится построчное или постолбцовое сканирование в пределах нескольких пикселей вокруг искомой линии. Для каждого сканирования методом интерполяции полиномом находится максимум. Координаты максимумов для всех строк аппроксимируются линией (методом наименьших квадратов). Вдоль осевой линии группы в трекинговом изображении нарисована дополнительная линия, поэтому следующим этапом алгоритм находит эту линию и также определяет ее координаты методом наименьших квадратов. Затем вычисляются пересечения с этой прямой каждой из найденных линий кода и именно эти точки пересечений и сопоставляются с идеальными координатами трекинговых изображений.

2.5 Определение координат камеры

Для получения параметров видеокамер по некоторому множеству точек с известными координатами в 3D-пространстве и на изображении, существует множество алгоритмов. Их можно разбить на три группы: алгебраические (быстрые, но неустойчивые к шумам), итеративные (требуют примерной начальной оценки, от которой зависит скорость схождения, в нашей ситуации это плюс, поскольку в качестве начальной оценки можно использовать предыдущее значение координат) и гибридные. После рассмотрения существующих вариантов был выбран линейный алгоритм, предложенный Long Quan и Zhongdan Lan в 1999 году. Известно, что задача не имеет единственного решения при знании 3-х точек, поэтому одно из требований к трекинговому изображению является отсутствие “средних” точек, т.е. тех точек, которые находятся на одной прямой. Очевидно, что в нашем случае таких точек очень много, поэтому для работы алгоритма используется множество последовательностей из 4-х точек, две из которых взяты из одной группы, а две – из другой группы. Это накладывает ограничение, чтобы дополнительная камера всегда видела верхний левый или верхний правый угол сцены, где сходятся пара групп: верхняя и левая или верхняя и правая.

2.6 Учет инерционности реальных видеокамер

Видеокамера обладает большой инерционностью, поэтому положение камеры в 3D-пространстве не может изменяться очень быстро. Задав механические характеристики (массу и момент инерции) можно ограничивать изменения восстановленного положения камеры от кадра к кадру. Также хорошие результаты показало использование фильтра Кальмана для сглаживания изменений в положении камеры.

3. ХАРАКТЕРНЫЕ ШУМЫ

Существует два типа шумов, которые необходимо учитывать при обработке видеоизображений: яркостные шумы и построчное смещение изображения. Яркостные шумы приводят к ошибке в определении положения трекингового отрезка. Характерная величина яркостного шума не превышает нескольких процентов. Причем можно выделить

два типа яркостных ошибок – построчные и попиксельные. Построчные ошибки приводят к синхронному изменению яркости вдоль всей строки изображения. Такие ошибки не будут влиять на определение максимума яркости вдоль строки. Попиксельные ошибки приводят к независимому изменению яркости соседних пикселей. Ошибка при смещении строки в целом может достигать половины пиксела, что приводит к гораздо более серьезным ошибкам при определении координат трекинговых отрезков. Для уменьшения влияния такого класса ошибок необходимо использовать большое число трекинговых элементов, либо эти элементы должны быть достаточно протяженными.

4. БИБЛИОГРАФИЯ

- [1] Orad's Pattern Recognition tracking system. http://orad.tv/page.aspx?page_id=49.
- [2] A Real-Time Infrared Tracking System for Virtual Environments http://www.ercim.org/publication/Ercim_News/enw53/foursa.htm.

Об авторах

Игорь Геннадьевич Таранцев – научный сотрудник Института Автоматики и Электрометрии СО РАН. Адрес: Новосибирск, 630090, пр-т Лаврентьева, 1а. Телефон: (383) 33-99-220
E-mail: egor@sl.iae.nsk.su

Борис Борисович Морозов – научный сотрудник Института Автоматики и Электрометрии СО РАН.
E-mail: bbm@sl.iae.nsk.su

Александр Андреевич Савенко – магистрант физического факультета Новосибирского Государственного Университета

Possibility of camera tracking with the help an additional coaxial camera

Abstract

Discuss possibility of camera tracking with the help an additional coaxial camera. The camera should have fixed focus length. Format of tracking image is like bar code. It allow to detect tracking image easily and fine. Describe realization and noise problem.

Keywords: Camera tracking, Virtual Studio.

About the authors

Igor Tarantsev is research officer at Institute of Automation and Electrometry. His contact email is egor@sl.iae.nsk.su.

Boris Morozov is research officer at Institute of Automation and Electrometry. His contact email is bbm@sl.iae.nsk.su.

Alexander Savenko is student at Novosibirsk State University, Department of Physics.