Сопровождение людей в системах многокамерного видеонаблюдения для спортивных игр

Павел Батанов, Владимир Кононов, Антон Конушин
Факультет вычислительной математики и кибернетики
Московский Государственный Университет имени М.В.Ломоносова, Москва, Россия
{pbatanov,vkononov,ktosh}@graphics.cs.msu.ru

Аннотация

Видеонаблюдение в спорте применяется уже продолжительное время, в особенности, в профессиональных состязаниях. Но, в большинстве своем, задачи, решаемые в этой области, полагаются на высококачественное дорогостоящее оборудование, доступное только для лидеров в мире спорта или ручную разметку данных.

В данной работе описывается алгоритм для сопровождение спортсменов на поле с использованием более доступного оборудования. Описанный подход может быть легко масштабирован на использование произвольного числа камер, наблюдающих одно и то же игровое поле.

Ключевые слова: *видеонаблюдение, сопровождение людей, трекинг, спортивные игры.*

1. ВВЕДЕНИЕ

В современных командных видах спорта одной из проблем является сбор данных о передвижениях игроков во время спортивных мероприятий для последующего анализа или отображения положения игроков в реальном времени. Такие данные как график скорости передвижения игрока, статистика его положения на поле, расстояние, которое игрок преодолел с той или иной скоростью, являются ценной информацией, как во время тренировки, так и во время матча для повышения интереса зрителя. Также эти данные могут использоваться при найме новых спортсменов в коллектив в качестве характеризующих спортивных показателей

В данный момент описанная задача сопровождения решается либо с помощью системы дорогостоящего оборудования, либо через использование дешевого наемного ручного труда для разметки уже отснятого видео. Частично данную задачу можно решить использованием высокотехнологичной формы для спортсменов, например, датчиков в бутсах, которые подсоединяются к компьютеру после матча для выгрузки информации. Однако, такой подход не может быть использован для вычисления положения игрока на поле и, тем более, для получения и обработки данных в режиме реального времени. Поэтому задача сопровождения людей в видеозаписях спортивных игр является актуальной.

2. ПОСТАНОВКА ЗАДАЧИ

2.1 Входные данные

Видеопоследовательности y_i, содержащие запись спортивной трансляции с нескольких неподвижных и откалиброванных камер, с объектами площадью не менее 100 пикселей:

$$y(t) =< y_i(t), \quad i = 1..numCameras, t = 0..numFrames >$$

Объединение всех ракурсов видеопоследовательности покрывают полностью игровое поле. Видеозаписи также синхронизированы по времени

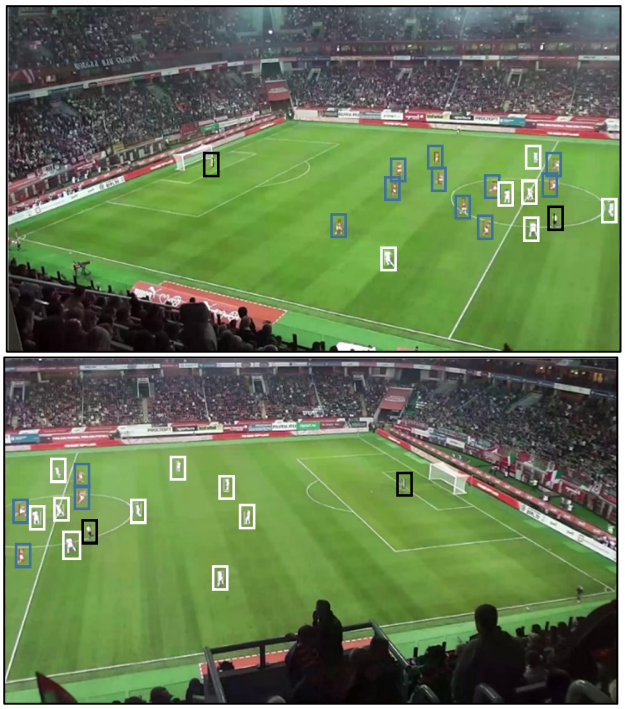

Рис 1: Примеры кадров из входных данных

2.2 Выходные данные

- Измеренные характеристики всех сопровождаемых объектов на всех кадрах последовательности
  $x(t) =< x_i(t), \quad i = 1..numObjects, t = 0..numFrames >$
  $x_i(t) =< u_i(t), v_i(t), w_i(t) >$
  - Положение ($u_i(t)$) на плоскости поля
  - Скорость ($v_i(t)$)
  - Размеры описывающей фигуры ($w_i(t)$)
- Индикация возможных ошибок работы алгоритма для дальнейшего исправления с участием оператора
  - Сопровождение объекта более чем одной меткой («склеивание меток»)
  - Потеря объекта сопровождения

2.3 Критерии качества

2.3.1 Точность алгоритма

В качестве основной метрики качества работы алгоритма будет выступать метрика MOTP [1]

$$MOTP = \frac{\sum_t \sum_i d_{i,t}}{\sum_t c_t}$$

- $d_{i,t}$ – метрическое расстояние между реальным и измеренным положением i-го объекта в момент времени t
- c_t – количество объектов в момент времени *t*

2.3.2 Время работы

Для использования алгоритма в режиме реального времени в спортивных состязаниях важна скорость обработки каждого поступившего кадра. Современные системы видеонаблюдения, в основном, используют в качестве рабочей частоты значения в 12-30 кадров в секунду. Таким образом, время обработки кадра на современном оборудовании должно производиться за 30-100 мс, в противном случае алгоритм будет работать с задержками либо с пониженной точностью (пропуском кадров). Также стоит отметить, что при более высокой частоте поступления кадров смещение сопровождаемых объектов между кадрами становится меньше, что позволяет регулировать быстродействие алгоритма за счет снижения размера пространства поиска возможных положений объекта.

3. ПРЕДЛОЖЕННЫЙ МЕТОД

3.1.1 Фильтр частиц

Фильтр частиц [2] представляет собой последовательный рандомизированный метод для оценки сложных Байесовских сетей. Под сложными сетями понимаются сети с непрерывными состояниями или сложными зависимостями между переменными (например, в линейных динамических системах). Цель этого метода – оценить скрытые значения (положение объекта) по наблюдаемым данным (изображениям с разных ракурсов) $p(x_t|y_0, \dots, y_t)$.

Для работы с фильтром частиц процесс движения объектов моделируется как Марковская цепь первого порядка следующим образом:

$x_0, x_1 \dots$ - Марковский процесс первого порядка такой, что $x_t|x_{t-1} \sim p_{x_t|x_{t-1}}(x|x_{t-1})$, где x_0 – известное начальное распределение. Для нашей задачи состоянием было выбрано положение объекта u, его скорость v и размеры ограничивающего цилиндра w.

$y_0, y_1 \dots$ - Наблюдения, условно независимы, при условии известности $x_0, x_1 \dots$, иными словами y_t зависит только от x_t: $y_t|x_t \sim p_{y|x}(y|x_t)$

В частности,

$$x_t = g(x_{t-1}) + m$$
$$y_t = h(x_t) + z$$

где m и z – взаимно независимые и одинаково распределенные случайные величины с известной плотностью вероятности, а g() и h() – известные функции.

Функция $x_t = g(x_{t-1}) + m_t$ представляет собой модель движения объекта. В качестве модели движения для нашей задачи была выбрана линейная модель со сглаживанием скоростей с параметром α:

$$u_t = u_{t-1} + v_{t-1} \cdot \Delta t + m_u$$
$$v_t = \alpha \cdot v_{t-1} + (1 - \alpha) \cdot (u_{t-1} - u_t) + m_v$$
$$w_t = w_{t-1} + m_w$$

где w_i – шум соответствующей компоненты состояния.

Модель движения дает априорную информацию о том, где может находиться объект в следующий момент времени. Для того, чтобы оценить $p(x_t|y_0, \dots, y_t)$, как и во многих рандомизированных методах (например, [3]), генерируется набор гипотез x_t^L (так же называемые частицами, отсюда название метода – «фильтр частиц»).

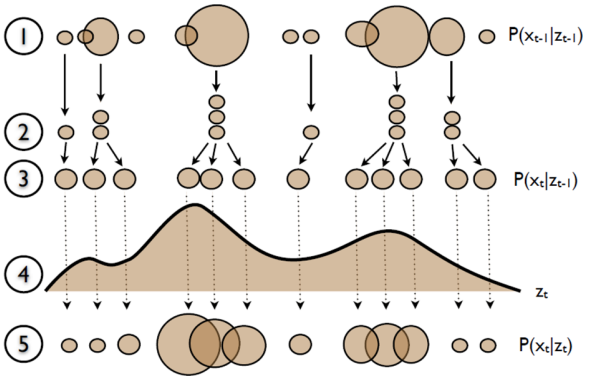

рис. 1 Схема фильтра частиц. Шаги фильтрации: (1) апостериорная функция t-1; (2) генерация новых частиц; (3) построение гипотез (априорная функция); (4) измерения; (5) апостериорная функция t; [8]

Тогда моменты могут быть оценены в соответствии с фильтрующим распределением следующим образом [4]:

$$\int f(x_k) p(x_k|y_0, \dots, y_k) dx_k \approx \frac{1}{P} \sum_{L=1}^{P} f(x_k^{(L)})$$

В генерации новых гипотез участвуют старые с вероятностью, пропорциональной их весу на предыдущем этапе [5]. Вес гипотезы оценивается из вероятности того, что гипотеза верна. В этом случае бо́льшая часть новых гипотез будет сгенерирована из частиц с бо́льшими весами. Но так же существует вероятность генерации и из частиц с малым весом.

После этапа генерации созданный набор частиц подлежит оценке с помощью модели фона и классификатора.

3.1.2 Моделирование фона

На данном этапе обработки новых поступивших кадров всех видеопоследовательностей происходит детектирование объектов путем моделирования фона.

Рис. 2 Пример пользовательской маски поля (слева) и выхода алгоритма ViBe (справа)

Моделирование фона выполняется с помощью алгоритма ViBe [6], который способен рассчитывать маску переднего плана последовательно для каждого нового кадра с поддержкой обновляемой модели фона. Дополнительно пользователь может задать глобальную маску, указывающую область кадра (Рис. 2), в которой детектирование производится не будет. Это требуется, если, например, в кадр попадают трибуны.

На выходе этапа моделирования фона получается маска переднего плана для каждого ракурса.

На приведенном примере (Рис. 3) видно, что кроме самих объектов выделяются также и другие динамические объекты, такие как, например, рекламные щиты вдоль поля. Это затрудняет работу при недостатке ракурсов вдали от камер, если положение камер таково, что сопровождаемые спортсмены могут пересекаться с этими объектами. Это выражается в неверной маске переднего плана для них.

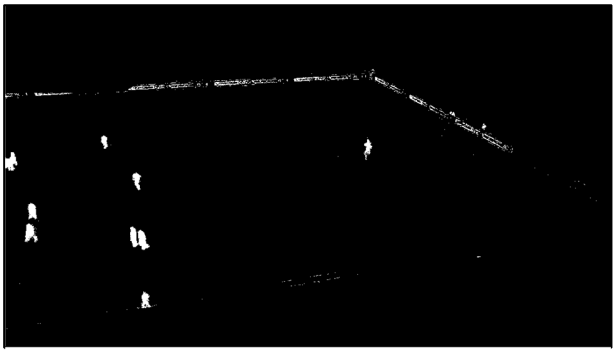

Рис. 3 Пример полученной маски переднего плана

3.1.3 Классификация гипотез

Для классификации каждой гипотезы фильтра частиц используется заранее обученный классификатор «случайный лес принятия решений» [7] (RF) и полученная на предыдущем шаге маска переднего плана. Для обучения классификатора используются фрагменты изображений, полученных в ходе первоначального выделения объектов пользователем.

В качестве признаков фрагментов строится трехмерная гистограмма цветов пикселей фрагмента в пространстве RGB. Размерность гистограммы – 8x8x8

Выходом классификатора является вектор голосов за принадлежность фрагмента изображения к определенному классу (Рис. 4):

- Игрок первой команды
- Игрок второй команды
- Вратарь первой команды
- Вратарь второй команды
- Судья
- Газон

Классы вратарей были добавлены в связи с тем, что в некоторых командных видах спорта форма вратарей сильно отличается от основной формы команды.

Рис. 4 Примеры объектов разных классов

Для классификации гипотеза pпроецируется на каждый ракурс c_i. После этого извлекается фрагмент кадра и участок маски переднего плана, соответствующий проекции гипотезы на данном ракурсе.

$$[patch(p, c_i), mask(p, c_i)] = projectParticle(p, c_i)$$

Если на некотором ракурсе не существует проекции гипотезы, то данный ракурс помечается для данной гипотезы как недействительный, чтобы в дальнейшем отбросить его при объединении данных.

Если для данного ракурса участок маски переднего плана, на которую проецируется гипотеза, содержит менее половины пикселей переднего плана, то данная гипотеза помечается как недействительная для всех ракурсов.

Для фрагмента изображения строится трехмерная гистограмма $hist$, которая подается на вход описанному классификатору. На выходе получается вектор голосов. Из этого вектора формируется значение, соответствующее доле голосов за целевой класс метки. Это значение становится весом гипотезы для данного вида.

$$votes = RF\big(hist(patch(p, c_i))\big)$$
$$votes = < v_j, j = 1..numClasses >$$
$$weight(p, c_i) = v_{objectClass}$$

Далее вес гипотезы умножается на процент маски переднего плана, попавшей во фрагмент изображения, соответствующего рассматриваемой гипотезе. Таким образом стимулируются гипотезы, покрывающие только объект.

$$weight(p, c_i) = weight(p, c_i) \cdot \sum mask(p, c_i) \backslash size(mask(p, c_i))$$

На выходе этапа измерения получается массив векторов с весами всех гипотез для каждого ракурса (или спец. значениями, обозначающих недействительность ракурса или гипотезы).

3.1.4 Объединение данных

После того, как для каждого вида был сформирован вектор весов гипотез, происходит этап объединения данных со всех ракурсов:

- Гипотезы, которые недействительны для всех ракурсов, получают вес 0.
- Гипотезы, которые, у которых есть хотя бы один действительный ракурс, получают вес, равный максимальному из весов для каждого ракурса этой гипотезы:

$$weight(p) = \max_{i=1..numCameras} weight(p, c_i)$$

Функция максимума была выбрана для того, чтобы обрабатывать перекрытия объектов при достаточном количестве ракурсов.

На данном шаге получается массив весов для каждой гипотезы:

$$weight(p), p = 1..numParticles$$

Из этого вектора берется подмножество гипотез с весами, не меньшими, чем $0.7 \cdot \max_p weights(p)$, на основе которых делается оценка путем усреднения их состояний.

4. ЭКСПЕРИМЕНТАЛЬНАЯ ОЦЕНКА

Алгоритм был протестирован на собранной тестовой выборке, которая представляет собой 300 размеченных кадров футбольного матча, снятого с двух ракурсов.

Алгоритм запускался с использованием 96 гипотез для каждой метки.

Алгоритм при данных параметрах показал метрику MOTP на размеченных данных равную 87.77 см

Время работы алгоритма составляет 4 секунды на кадр на процессоре Intel Core i7-3770K для версии в среде MATLAB с использованием встроенных средств многопоточности.

На C++ время работы алгоритма составило в среднем 80мс на один кадр, что позволяет обрабатывать видеопоследовательности с суммарной частотой до 24 кадров в секунду.

Предложенный методы был сравнен с методом [8], как наиболее схожим по области применения современным алгоритмом, на тех же данных с теми же параметрами.

Метрика MOTP для алгоритма [8] составила 224.11 см.

5. ЗАКЛЮЧЕНИЕ

В нашей работе мы предложили усовершенствованный алгоритм сопровождения людей в видеопоследовательностях, который применим для работы с видеозаписями спортивных трансляций.

6. БЛАГОДАРНОСТИ

Работа была выполнена при частичной поддержке гранта 10255р/16855 программы «УМНИК» фонда содействия развитию малых форм предприятий в научно-технической сфере и гранта Президента Российской Федерации для молодых ученых - кандидатов наук МК-4644.2012.9.

7. ССЫЛКИ

[1] K. Bernardin и R. Stiefelhagen, «Evaluating multiple object tracking performance: The CLEAR MOT metrics,» *Journal on Image and Video Processing - Regular,* 2008.

[2] M. Arulampalam, S. Maskell, N. Gordon и T. Clapp, «A tutorial on particle filters for online nonlinear/non-gaussian Bayesian tracking,» *IEEE Transactions on Signal Processing,* pp. 174-188, 2002.

[3] I. Murray, «Markov chain Monte Carlo,» 2009. [В Интернете]. Available: http://homepages.inf.ed.ac.uk/imurray2/teaching/09mlss/slides.pdf.

[4] Д. Ветров, «Лекция 11. Методы Монте-Карло. Фильтр частиц,» 2009. [В Интернете]. Available: http://courses.graphicon.ru/files/courses/smisa/2009/lectures/lecture10.pdf.

[5] A. Doucet, «Sequential Importance Sampling Resampling,» Departments of Statistics & Computer Science, 2010. [В Интернете]. Available: http://www.cs.berkeley.edu/~jordan/courses/260-spring10/readings/samsi_lec3.pdf. [Дата обращения: 2012].

[6] O. Barnich и M. Van Droogenbroeck, «ViBe: A universal background subtraction algorithm for video sequences,» *IEEE Transactions on Image Processing,* pp. 1709-1724, June 2011.

[7] L. Breiman, «Random Forests,» *Machine Learning,* т. 45, № 1, pp. 5-32, 2001.

[8] E. Morais, S. Goldenstein, A. Ferreira и A. Rocha, «Automatic tracking of indoor soccer players using videos from multiple cameras,» в *SIBGRAPI*, Ouro Preto, 2012.

Об авторах

Павел Батанов – выпускник факультета ВМК МГУ имени М.В. Ломоносова. Его адрес: pbatanov@graphics.cs.msu.ru.

Владимир Кононов – аспирант факультета ВМК МГУ имени М.В. Ломоносова. Его адрес: vkononov@graphics.cs.msu.ru

Антон Конушин – доцент факультета ВМК МГУ имени М.В. Ломоносова. Его адрес: ktosh@graphics.cs.msu.ru