

Алгоритм вычитания фона, основанный на поблочных классификаторах

Евгений Шальнов, Владимир Кононов, Вадим Конушин

Лаборатория Компьютерной Графики и Мультимедиа,

Московский Государственный Университет имени М.В.Ломоносова, Москва, Россия

{eshalnov, vkononov, vadim}@graphics.cs.msu.ru

Аннотация

Алгоритмы вычитания фона широко применяются в задачах видеонаблюдения и активно изучаются в последнее время. Большинство существующих алгоритмов вычитания фона основываются на сравнении цвета пикселей или блоков очередных кадров со своими цветовыми моделями. При этом значительные перепады в освещении обычно приводят к большому числу ошибок, а иногда и к полной деградации работы системы.

В методе, предлагаемом в данной статье, весь кадр видео разбивается на блоки, для каждого из которых на основе бустинга строится отдельный классификатор, определяющий перекрыт данный блок или нет. Благодаря использованию признаков, основанных на сравнении цветов разных пикселей в блоке, достигается большая устойчивость к изменениям освещенности и дрожанию камеры. Требуя значительного времени при обучении под новую сцену, на этапе выполнения алгоритм демонстрирует высокую скорость, сравнимую со скоростью простейших алгоритмов вычитания фона.

Было проведено сравнение точности сегментации видео с помощью данного алгоритма и нескольких существующих аналогов на выложенных в открытый доступ видеороликах. Сравнение показало высокую точность работы предложенного алгоритма.

Ключевые слова: вычитание фона, машинное обучение, бустинг.

1. ВВЕДЕНИЕ

Эта статья посвящена проблеме выделения в видео, снятом статичной камерой, объектов переднего плана в реальном времени. Одним из наиболее распространенных применений такой задачи являются алгоритмы слежения за людьми, которые используют вычитание фона для поиска объектов. Такие алгоритмы используются в охранном видеонаблюдении, освещении спортивных трансляций. Также алгоритмы вычитания фона используются для создания дополнительных эффектов во время проведения видеоконференций, например для замены фона.

Подобные приложения накладывают следующие ограничения на алгоритмы вычитания фона: (1) устойчивость к небольшим изменениям освещения и дрожаниям камеры; (2) вычислительная простота для возможности работы в реальном времени.

Стандартные методы вычитания фона основаны на построении модели фона по правилам, не зависящим от



Рисунок – 1. Задача вычитания фона. Слева направо: фон, кадр с объектом переднего плана, искомая маска объекта.

рассматриваемой области изображения. Из-за этого для описания сложных случаев изменения освещения или дрожания камеры необходимо либо использовать сложную модель фона, требующую больших вычислительных затрат, либо обновлять модель фона динамически во время работы алгоритма, что приводит к существенным ошибкам в работе на тех кадрах, на которых происходит эволюция модели.

В этой статье предлагается алгоритм вычитания фона с использованием поблочных классификаторов для сегментирования блоков на передний и задний план. С помощью машинного обучения происходит выявление наиболее информативных признаков для каждого конкретного блока. Это позволяет с одной стороны быстро обрабатывать кадры видеопоследовательности, так как используются простые признаки, а с другой стороны достигать высокой точности сегментации пикселей, так как набор простых признаков уникален для каждого блока.

Статья построена следующим образом. В секции 2 приводится обзор текущих методов решения задачи, в секции 3 описывается предложенный алгоритм. В секции 4 приводятся результаты экспериментов и сравнения, а в секции 5 подводится заключение.

2. СУЩЕСТВУЮЩИЕ МЕТОДЫ

Большинство методов вычитания фона можно разбить на 3 категории:

- Попиксельные [1,2,11]
- Поблочные [11]
- Алгоритмы, основанные на Марковских или Условных случайных полях [4,5]

Методы первой категории обрабатывают все пиксели очередного кадра независимо. Цвет каждого пикселя на текущем кадре сравнивается с его цветовой моделью. Типичными цветовыми моделями являются нормальное распределение [11], смесь нормальных распределений [1] и непараметрические модели [2]. Метод [1] предполагает, что значение цвета пикселя фона – случайная величина, чье распределение можно аппроксимировать смесью нормальных

распределений. Среди недостатков данного метода можно выделить необходимость использования большого числа гауссиан в смеси для моделирования фона, что приводит к существенным вычислительным затратам. Кроме того, данный метод чувствителен к небольшим дрожаниям камеры.

Методы второй категории независимо обрабатывают не пиксели, а целые блоки. То есть весь кадр разбивается на блоки, и для каждого из них принимается независимое решение. Несмотря на то, что потенциально такие методы не могут быть идеально точными, зачастую они дают более приемлемый результат, т.к. используют для принятия решения информацию с целой области. В эту категорию входят, например, метод на основе построения LBP-гистограмм [10] и метод на основе метрики SSD. Последний метод хранит модель фона в виде одного изображения. Результат на каждом кадре производится с помощью сравнения значения метрики близости с порогом. Метрика считается поблочно и равна сумме квадратов разностей между цветами соответствующих пикселей фона и текущего кадра. Предложенный метод также попадает в эту категорию.

В последнее время были предложены методы вычитания фона, проводящие сегментацию пикселей изображения, используя модели Марковского и Условного случайного поля для учета пространственных зависимостей между пикселями [4-5]. С помощью данных методов можно дополнительно требовать, чтобы границы областей объектов переднего плана проходили преимущественно по краям на входном изображении. Но, к сожалению, методы данной категории являются более медленными, и почти не используются на практике в системах видеонаблюдения.

В статье [5] авторы предлагают алгоритм вычитания фона для использования в видеоконференциях. Поэтому, они используют несколько иную постановку задачи: в качестве переднего плана выступает лишь человек, ближайший к видеокамере. Люди на заднем плане, считаются частью фона. Для решения данной задачи предлагается использовать информацию о наиболее вероятном движении объектов переднего плана. С помощью данного метода можно уменьшить ошибки сегментации из-за движений в фоновом изображении, но с другой стороны необходимо заранее знать наиболее вероятные модели движения для объектов переднего плана, что возможно не в каждом случае.

Все алгоритмы вычитания фона условно можно разделить на быстрые, но слабоустойчивые к изменениям освещения и дрожаниям камеры, и качественные, но требующие существенных временных затрат на обработку каждого кадра. Мы предлагаем метод, который может одновременно быстро и качественно решать задачу, основываясь на классификаторах, натренированных для каждого блока.

3. ПРЕДЛОЖЕННЫЙ МЕТОД

Мы предлагаем алгоритм вычитания фона, работающий с блоками кадра, который устойчив к изменениям освещения и способен работать в приложениях реального времени. Этот эффект достигается за счет использования простых признаков для принятия решения о принадлежности блока к переднему или заднему планам, причем для каждого блока выбираются наиболее информативные для него признаки. Использование простых признаков не требует больших вычислительных затрат, что позволяет работать в реальном времени. Наиболее

информативные признаки для каждого блока кадра удается выявлять с помощью алгоритма машинного обучения.

Для тренировки алгоритма машинного обучения необходима выборка положительных и отрицательных примеров. В общем случае для ее создания необходимо много данных, размеченных вручную. Эта процедура требует больших временных затрат и не гарантирует получение репрезентативной выборки. Поэтому мы генерируем выборку без объектов переднего плана, имитируя возможные изменения в освещении и настройках видеокамеры, а также выборку, имитирующую возможные перекрытия фона. Для такой генерации достаточно всего одного кадра сцены без объектов переднего плана.

Таким образом, в предложенном алгоритме можно выделить следующие шаги:

1. Генерация положительных и отрицательных примеров для обучения классификаторов.
2. Построение классификаторов.
3. Использование обученных классификаторов для получения маски объектов переднего плана в видеопоследовательности.

3.1 Построение обучающей выборки

На первом шаге алгоритма происходит построение обучающей выборки для тренировки классификаторов. Данная выборка состоит из положительных и отрицательных примеров для фоновое изображение (Рис. 2).



Рисунок – 2. Кадры с положительными примерами из обучающей выборки.

Так как предлагаемый алгоритм должен быть устойчив к изменениям в освещении, то в положительные примеры добавляются кадры с различными вариантами освещения. Кроме того используются примеры, полученные при изменении контрастности и небольшим размытием исходных кадров из обучающего видео.

Для описания метода построения отрицательных примеров необходимо сначала рассмотреть признаки, используемые при обучении. В качестве признаков классификатора используется разность соответствующих цветовых каналов пар случайных пикселей блока. Так как при этом используется не значение цвета, а его значение относительно цвета другого пикселя, то данный классификатор более устойчив к изменениям освещения сцены.

При построении отрицательных примеров мы моделируем ситуацию, когда один или оба пикселя из пары, используемой при построении значения признака, перекрыты объектом. Таким образом в используется 3 способа построения значения признака отрицательного примера: (1) цвет первого пикселя пары берется из изображений с положительными примерами, цвет второго — равномерно распределенная случайная

величина; (2) цвет первого пикселя — случайная величина, а цвет второго — из изображения; (3) цвета обоих задаются случайно.

3.2 Построение классификаторов

На следующем этапе работы алгоритма происходит обучение классификаторов. В предлагаемом алгоритме используется бустинг в качестве алгоритма машинного обучения [8-9]. С его помощью удастся выявлять наиболее информативные признаки для каждого конкретного блока.

3.3 Использование обученных классификаторов

На третьем этапе работы алгоритма применяются обученные классификаторы для сегментации тестового видео.

Так как используется бустинг в качестве алгоритма машинного обучения, то в результате каждый обученный классификатор представляет собой взвешенную сумму простых правил. Правилами являются сравнение конкретного признака с порогом. Это означает, что сложность классификации отдельного блока линейно зависит от сложности применения отдельного простого правила. За счет этого удастся достичь быстрой обработки блоков кадра.

Для уменьшения ошибок классификации на данном этапе применяется медиана по результатам сегментации на нескольких последовательных кадрах.

4. ЭКСПЕРИМЕНТЫ И СРАВНЕНИЕ

Предложенный метод был протестирован на наборе видеопоследовательностей [6], снятых внутри помещения, в каждой из которых происходили существенные изменения освещения. Предложенный алгоритм вычитания фона показал хорошие результаты работы на данной базе (рис. 3).

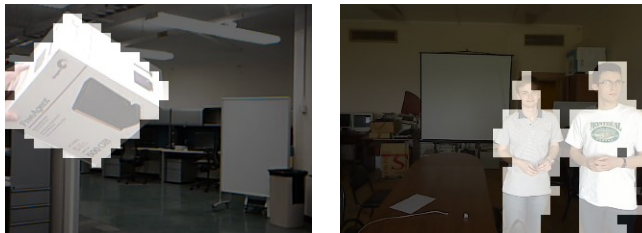


Рисунок – 3. Примеры работы предложенного метода.

В текущей версии нашего алгоритма при существенных изменениях освещения происходят ошибки при сегментации на объекты переднего и заднего плана (рис. 4). Данное явление связано с выбранным методом построения обучающей выборки. В настоящее время используется простое приближение алгоритма настройки камеры при изменении освещения. Одним из основных направлений последующей работы будет устранение этого недостатка.



Рисунок – 4. Примеры неудачной работы метода.

Было проведено экспериментальное сравнение нашего метода с алгоритмами, основанными на использовании смеси гауссиан и сравнении блоков по метрике SSD.

Использовалась встроенная в MATLAB реализация смеси гауссиан из Computer Vision System Toolbox. Полученные результаты продемонстрированы на рис. 5. Для сравнения использовались попиксельные метрики точности (precision) и полноты (recall). Графики на рис. 5 создавались изменением порога на значение метрики или выхода классификатора.

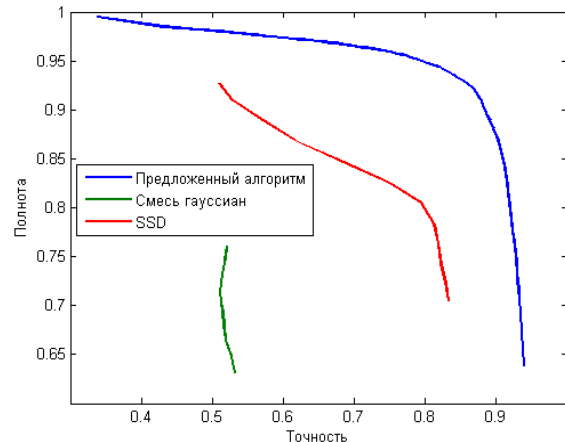


Рисунок – 5. Сравнение алгоритмов.

В первую очередь стоит отметить нетипичный график алгоритма, использующего смесь гауссиан. Так как изменяемый при построении соответствующего графика параметр (минимальная априорная вероятность фона) подается на вход встроенной в MATLAB функции, мы не можем точно объяснить данный факт. Видимо, этот параметр влияет не только на выход алгоритма (как порог на вероятностный выход), но и изменяет внутреннее состояние алгоритма (например, влияет на обновление цветовой модели в пикселях).

Также стоит разъяснить низкую итоговую точность данного алгоритма. В случае стабильной работы, когда не происходит сильного изменения освещения или настроек камеры, точность алгоритма, основанного на смеси гауссиан, превосходит точность двух других сравниваемых методов (см. рис. 6). Однако, как только значительно изменяется освещение, число ошибок резко возрастает (см. рис. 7). Во многих системах встраивают отдельный шаг для обработки таких ситуаций. Например, по резкому изменению гистограммы яркости пикселей кадра определяют момент изменения освещения, после чего вносятся изменения в модель фона (или наоборот, изменяют цвета пикселей очередного кадра видео). Мы использовали данный алгоритм из MATLAB без изменений, не встраивая подобную обработку. При её встраивании, мы допускаем, что алгоритм, основанный на смеси гауссиан смог бы показать на тестовых видеороликах наибольшую точность. Однако обычно такие обработки являются эвристиками, использующими ряд вручную подбираемых параметров, из-за чего они являются неустойчивыми. Предложенный алгоритм справляется с данной проблемой без каких-либо дополнительных модификаций.

Алгоритм, основанный на сравнении блоков по метрике SSD, справляется с изменением освещения лучше. Объяснить это

можно тем, что используя информацию сразу по некоторой окрестности, при классификации блоков он может использовать менее жесткие пороги, чем попиксельные методы. Тем не менее, метрика SSD не является оптимальной метрикой для сравнения блоков, из-за чего итоговая точность сегментации данным алгоритмом оказалась ниже, чем у предложенного подхода.

Помимо устойчивости к изменениям освещения среди достоинств предложенного метода можно выделить быструю обработку кадров за счет использования простых признаков. За секунду алгоритм обрабатывает 6 кадров размера 320x240 с использованием блоков размером 10x10 при неоптимизированной реализации на MATLAB на компьютере Intel Core 2Duo 2.53 GHz, 2048 Mb RAM. При этом алгоритм, основанный на использовании смеси гауссиан, обрабатывает 4-5 кадров в секунду, а алгоритм, использующий сравнение блоков по метрике SSD – 28 кадров в секунду.

Реализация на C++ может существенно ускорить предложенный алгоритм. Также он допускает простое и эффективное распараллеливание.

Стоит отметить, что предлагаемый алгоритм вычитания фона требует длительного процесса обучения. В среднем на обучение одного классификатора требуется около 2 минут. Таким образом, на обучение классификаторов для всех блоков потребовалось 17 часов.

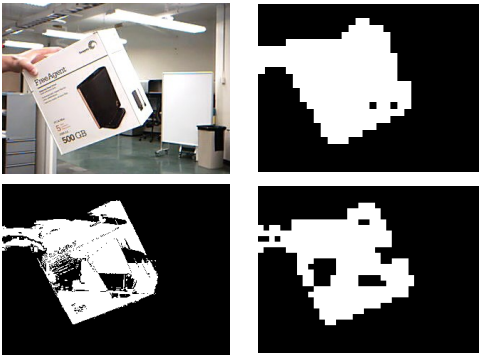


Рисунок – 6. Результаты работы алгоритмов при несильном изменении освещения. Верхний ряд: кадр видео с объектом переднего плана, результат работы предполагаемого метода. Нижний ряд: Результат работы GMM, результат работы SSD.

5. ЗАКЛЮЧЕНИЕ

В данной статье предложен алгоритм вычитания фона в видеопоследовательности на основе построения поблочных классификаторов с использованием алгоритма машинного обучения. Алгоритм показал устойчивость к изменениям освещения. При этом низкая вычислительная сложность алгоритма позволяет после обучения классификаторов обрабатывать видео в реальном времени.

Алгоритм, предложенный в данной статье, находится на стадии разработки. В дальнейшем мы планируем улучшить моделирование положительных и отрицательных примеров для создания выборки. Это позволит правильно решать задачу в случаях сильного изменения освещения и параметров камеры, а также избавит от существующих недостатков.

6. БЛАГОДАРНОСТИ

Работа была выполнена при поддержке гранта РФФИ №11-01-00957-а.

7. ССЫЛКИ

- [1] Chris Stauffer, W.E.L. Grimson, Adaptive background mixture models for real-time tracking, CVPR 1999
- [2] Ahmed Elgammal, David Harwood, Larry Davis, Non-parametric model for background subtraction, ICCV, 2000
- [3] Antoine Monnet, Anurag Mittal, Nikos Paragios, Visvanathan Ramesh, Background modeling and subtraction of dynamic Scenes, ICCV, 2003
- [4] Jian Sun, Weiwei Zhang, Xiaoou Tang, Heung-Yeung Shum, Background cut, CVPR, 2006
- [5] A. Criminisi, G. Cross, A. Blake, V. Kolmogorov, Bilayer segmentation of live video, CVPR, 2006
- [6] <http://www.cs.utexas.edu/~changhai/icra10-datasets/datasets.html>
- [7] V. Kolmogorov, A. Criminisi, A. Blake, G. Cross, C. Rother, Bi-layer segmentation of binocular stereo video, CVPR, 2005
- [8] J. Friedman, T. Hastie, and R. Tibshirani, Additive logistic regression: a statistical view of boosting. Annals of statistics, 38:337–374, 2000.
- [9] <http://graphics.cs.msu.ru/science/research/machinelearning/daboosttoolbox>
- [10] Marko Heikkila and Matti Pietika, A texture-based method for modeling background and detecting moving objects, IEEE, 2006
- [11] C. Wren, A. Azarbayejani, T. Darrel, and A. Pentland, Pfinder: Real time tracking of the human body. In IEEE Trans. on Pattern Analysis and Machine Intelligence, 19 (7), pp. 780–785, 1997

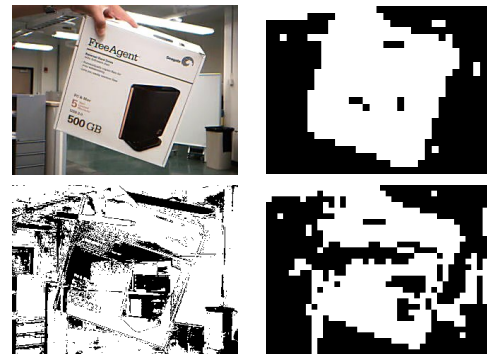


Рисунок – 7. Результаты работы алгоритмов при значительном изменении освещения.