

Правительство Российской Федерации
Федеральное государственное автономное образовательное
учреждение высшего профессионального образования

"Национальный исследовательский университет
"Высшая школа экономики"

Отделение программной инженерии
Кафедра Управления разработкой программного обеспечения

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

На тему: Программа динамической идентификации объекта на основе алгоритма
TLD

Студента группы № 471ПИ

(Ф.И.О.)

Научный руководитель

(должность, звание)

(Ф.И.О.)

Консультант

(должность, звание)

(Ф.И.О.)

Москва, 2013 г.

Содержание

Аннотация.....	3
Введение.....	4
1. Обзор трекеров и постановка задачи.....	5
1.1. Трекеры.....	5
1.2. Постановка задачи.....	7
2. Алгоритм TLD.....	8
2.1. Структура алгоритма.....	8
2.2. Детектор.....	8
2.3. Обучение.....	9
3. Особенности реализации.....	11
4. Исследование.....	12
4.1. Исследование воздействия шумов.....	12
4.2. Исследование минимального размера объекта.....	13
4.3. Исследование максимального перемещения объекта внутри кадра в пикселях.....	13
4.4. Исследование устойчивости к увеличению и уменьшению размера объекта.....	13
4.5. План дальнейших работ.....	14
Библиография.....	15

Аннотация

Дипломная работа посвящена алгоритму TLD (Tracking Learning Detection). Приводится его описание и области применения. Кроме того в рамках дипломной работы было осуществлено исследование алгоритма. В тексте описана методология и результаты, позволяющие установить границы применимости алгоритма.

Введение

Целью данной работы является реализация и исследование алгоритма TLD. Данный алгоритм, разработанный Зденеком Калалом, предназначен для долговременного отслеживания объектов на видео.

Отслеживание (или трекинг) объектов используется в таких практических сферах как:

- Автономная навигация — трекинг позволяет предсказать положение объекта, обеспечивая динамическое планирование маршрута.
- Системы внешнего наблюдения — трекинг позволяет отслеживать заданного человека с возможным исчезновением человека на видео.
- Человеко-компьютерный интерфейс — отслеживание определённой части тела позволяет использовать движения человека как ввод для компьютера.
- Дополненная реальность — знание положения объекта на видео позволяет накладывать эффекты дополненной реальности.

В настоящее время все описанные области применения достаточно новы и актуальны.

Тема трекеров освещена в литературе, однако алгоритм TLD дополняет классический подход к трекингу, добавляя элемент обучения в схему алгоритма. Сам алгоритм TLD является достаточно новым и на данный момент неполностью изученным.

В ходе данной работы осуществлены реализация и исследование области применимости данного алгоритма.

1. Обзор трекеров и постановка задачи

1.1. Трекеры

В этой главе представлен обзор способов, решающих задачу отслеживания объекта. Рассмотрены основные принципы алгоритмов.

В первую очередь дадим определение трекера: Трекер — алгоритм отслеживающий положение целевого объекта на видеопотоке.

При отслеживании объекта трекер должен знать его положение. Для этого он представляет положение объекта в каком-либо виде. Существует несколько способов представления объекта:

- Точки. При данном подходе объект представляет из себя набор из одной (в данном случае называемой центроидом) или нескольких точек. Трекер отслеживает перемещение каждой из точек.
- Прimitives геометрические фигуры. В данном случае перемещения объекта обычно моделируются как аффинное или проективное преобразование.
- Силует объекта или контур. Здесь надо дать определение силуэта и контура. Определения взяты из исследования Омара Джаведа и Мубарака Шах: Контур — граница объекта. Площадь внутри контура — силуэт объекта.
- Сочленённые геометрические фигуры. В этой модели части объекта связаны вместе. Как пример — человеческое тело можно представить как набор соединённых геометрических фигур, представляющих различные части тела.
- Скелетонная модель. При этом подходе извлекается скелетон. Модель применима и для твёрдых и для подвижных объектов.

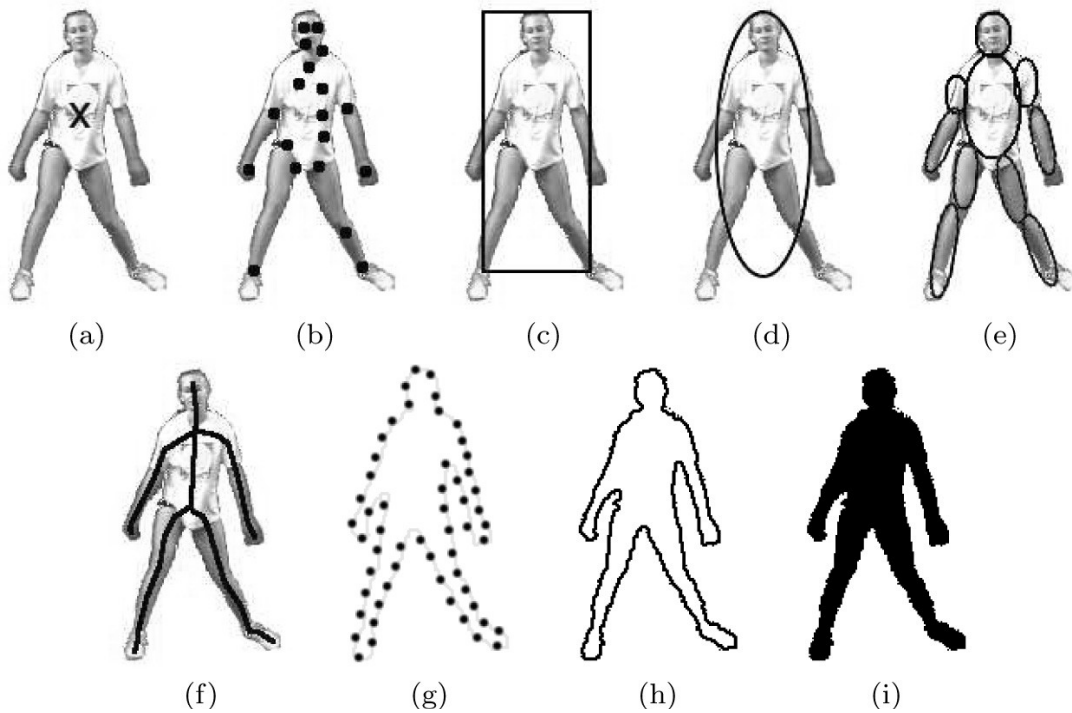


Рис. 1. Пример различных способов представления модели. (a) Центроид, (b) множество точек, (c) прямоугольный патч, (d) эллиптический патч, (e) сочленённые фигуры, (f) скелетон объекта, (g) контрольные точки на контуре, (h) полный контур, (i) силуэт объекта

Другой важный аспект для трекера — выбор характеристики. Наиболее частые характеристики:

- Цвет. Здесь за характеристикой объекта принимается цвет, однако данная характеристика чувствительна к шумам.
- Грань. Этот подход определяет изменения границ объекта, которые обычно меняются значительно. Наиболее популярный алгоритм определения границ — алгоритм Кэнни.
- Оптический поток. Оптический поток — это изображение видимого движения объектов, поверхностей или краев сцены, получаемое в результате перемещения наблюдателя (глаз или камеры) относительно сцены. Способы расчёта оптического потока включают метод Хорна-Шунка и метод Лукаса-

Канаде.

1.2. Постановка задачи

Задачами данной дипломной работы являются изучение, реализация и исследование алгоритма TLD.

2. Алгоритм TLD

2.1. Структура алгоритма

Алгоритм TLD использует подход Tracking-Modeling-Detection. Это означает, что алгоритм, в дополнение к основной задаче трекера, выполняет детектирование и моделирование объекта, тем самым повышая качество отслеживания. Так же это позволяет алгоритму редетектировать объект в случае его исчезновения на видеопотоке. Схема взаимодействия показана на рисунке 2.

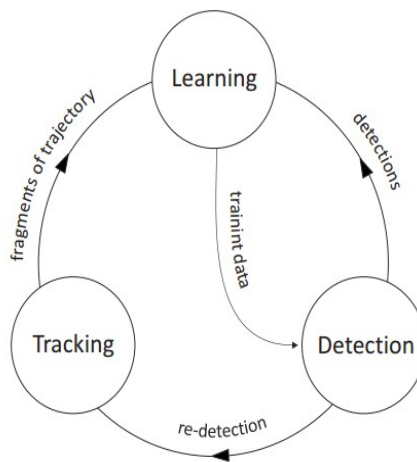


Рис. 2. Схема алгоритма TLD.

2.2. Детектор

Детектор — алгоритм, позволяющий найти заданный объект на изображении. Для этого алгоритм использует модель объекта, заданную в каком-либо виде. TLD использует модель, представленную набором патчей на основе 2 битного двоичного паттерна. В алгоритме TLD используется для редетектирования объекта и обучение модели.

Стратегия детектора, используемого в алгоритме TLD — стратегия сканирующего окна. Входное изображение сканируется окном и для каждого подизображения бинарный классификатор решает присутствует ли там объект или нет.

Классификатор состоит из набора характеристик. По каждой происходит

сравнение, подходит ли подизображение под характеристику или нет. Конечный результат вычисляется средним ответом. Если подизображение классифицируется положительно более чем 50% характеристик, то классификатор даёт положительный ответ. В противном случае отрицательный. Схема работы классификатора представлена на рисунке 3.



Рис. 3. Схема работы классификатора.

2.3. Обучение

Алгоритм является самообучающимся, используя для обучения непомеченные данные. В качестве данных о положении объекта выступают результаты работы трекера. В случае, если расстояние между положением, указанным детектором близко к положению, указанным трекером, то патч определяется как положительный. В противном случае как отрицательный.

Модель в алгоритме TLD представляется в виде рандомизированного леса. Обучение происходит путём двух событий: расширяющего (при определении патча как положительного) и урезающего (при определении патча как отрицательного). Расширение модели необходимо для добавления новых правильных возможных состояний модели, и происходит добавлением и расширением веток. Урезание модели необходимо для удаление неверных сэмплов, и происходит путём удаление веток из леса. Схема изменения модели

приведена на рисунке 4.

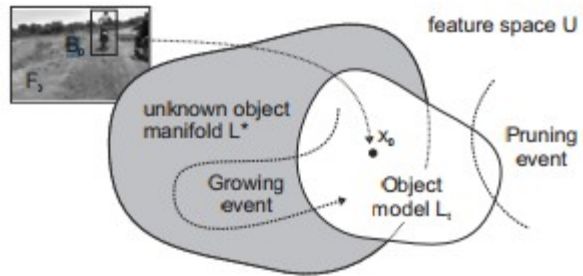


Рис. 4. Схема изменения модели.

3. Особенности реализации

Алгоритм реализован на языке C++. Программа состоит из контроллера, алгоритма и системы ввода/вывода и контроллера (см. рисунок 5), таким образом реализуя шаблон проектирования MVC (Модель-представление-контроллер).



Рис. 5. Схема программы.

Положение объекта программа указывает посредством прямоугольника зелёного цвета. Пример можно увидеть на рисунке 6.

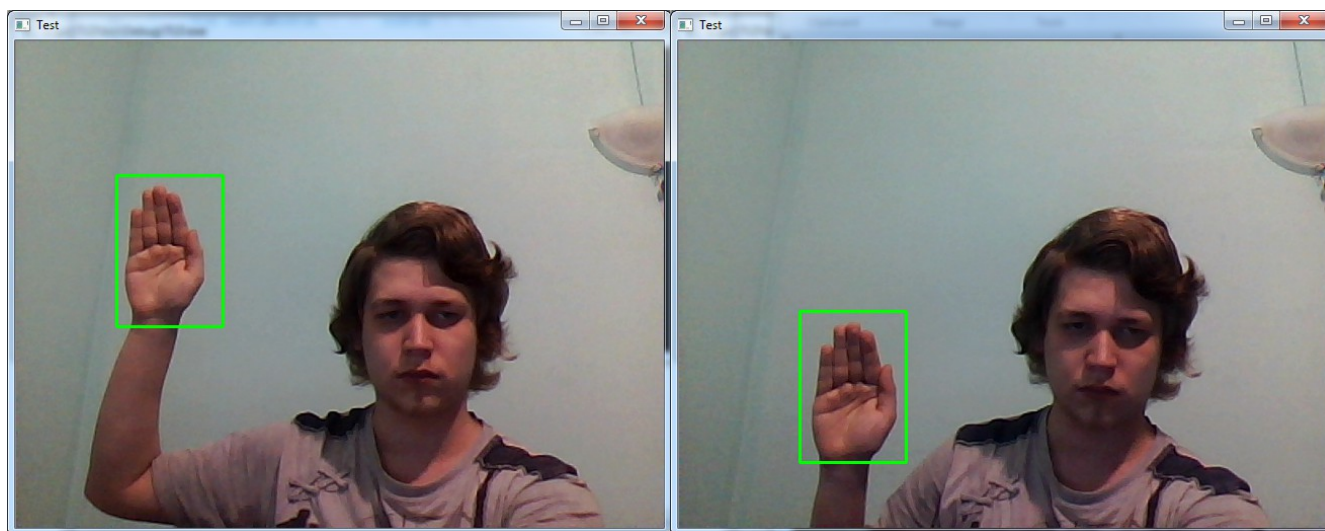


Рис. 6. Программа отслеживает положение ладони.

4. Исследование

4.1. Исследование воздействия шумов

Для исследования воздействия шумов взята последовательность из 200 изображений. На оригинальную последовательность накладывался Гауссовский цветной шум. Математическое ожидание шума — серый цвет со значением (128, 128, 128). Стандартное отклонение увеличивалось с шагом 5 до тех пор, пока алгоритм не перестал следить за объектом. В последовательности есть ряд фреймов где объект исчезает из последовательности. Таким образом проверяются критические значения не только когда алгоритм перестаёт следить за объектом, но и когда он не может возобновить слежение за объектом после его пропажи.

В результате исследования было выявлено, что алгоритм не может продолжить отслеживания объекта с шумом, стандартное отклонение равно 125 и больше. Алгоритм не в состоянии следить за объектом при шуме большем или равном 200. Вид одного из изображений последовательности с шумами с указанными отклонениями показаны на рисунке 7.



Рис. 7. Изображение из последовательности (оригинальное, с шумом с отклонением 125 и 200 соответственно)

В случаях когда отслеживание объекта происходило была проверена корректность отслеживания объекта. Объект во фрейме считался определённым

правильно если прямоугольник, обозначающий объект, выдаваемый алгоритмом при данном уровне шума имеет пересечение с прямоугольником на соответствующем фрейме в оригинальной последовательности. Для всех случаев вывод соответствовал критериям.

4.2. Исследование минимального размера объекта

Для выяснения минимального размера были подготовлены последовательности фреймов, где задавался различный размер объекта. Изначальный ограничивающий прямоугольник задавался минимально возможного размера, способного вместить объект. Было выяснено, что алгоритм не способен обрабатывать объекты, минимальная ограничивающая рамка которого имеет размер меньше 18 пикселей по одному из измерений.

4.3. Исследование максимального перемещения объекта внутри кадра в пикселях

Для исследования максимального перемещения в пикселях были подготовлены последовательности фреймов. Внутри каждой последовательности расстояние перемещение объекта в смежных фреймах являлось постоянным. В первой последовательности скорость объекта составляла 5 пикселей. В последующих последовательностях скорость увеличивалась с шагом 5 пикселей. Было выяснено, что максимальная дальность перемещения объекта между фреймами составляет 15 пикселей. Корректность отслеживания проводилась по тому же критерию, что и при исследовании шумов. Вывод соответствовал критериям.

4.4. Исследование устойчивости к увеличению и уменьшению размера объекта

При исследовании данного показателя предполагалось, что размер прямоугольника, обозначающего объект должен изменяться в зависимости от

размера объекта. Оптимальным прямоугольником будем считать минимальный прямоугольник, что полностью охватывает объект. Критерий допустимого размера прямоугольника: если он отклонялись от оптимального размера не более чем на 30%. Были подготовлены две последовательности фреймов — для тестирования увеличения и уменьшения. В них размер объекта изменялся на 2 пикселя по каждому направлению в каждом фрейме. Было выяснено, что алгоритм перестаёт удовлетворять критерию на двадцать шестом фрейме при увеличении. К уменьшению размера алгоритм оказался более устойчив, перестав удовлетворять критерию на сто пятом фрейме.

Примечание: размеры объекта в обеих последовательностях находились не ниже минимально допустимого, который определён выше.

4.5. План дальнейших работ

План при дальнейших исследованиях включает в себя:

- Исследование быстродействия в зависимости от размера фрейма.
- Исследование быстродействия в зависимости от размера объекта.
- Устойчивость к аффинным преобразованиям.
- Изучение влияния цвета.

Библиография

- [1] Omar Javed, Mubarak Shah (2006) Object Tracking: A Survey. ACM Computing Surveys, Vol. 38, No. 4, Article 13
- [2] Random Forests Leo Breiman and Adele Cutler
http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm. Получено 14 февраля 2013
- [3] RANDOM FORESTS--RANDOM FEATURES Leo Breiman
<http://www.stat.berkeley.edu/~breiman/random-forests.pdf>. Получено 14 февраля 2013
- [4] Zdenek Kalal, Jiri Matas, Krystian Mikolajczyk (2009) Online learning of robust object detectors during unstable tracking. Computer Vision Workshop 2009, Киото, Япония, IEEE CS.
- [5] Zdenek Kalal, Jiri Matas, Krystian Mikolajczyk (2010) P-N Learning: Bootstrapping Binary Classifiers by Structural Constraints. Conference on Computer Vision and Pattern Recognition, CVPR, Июнь 13-18, Сан-Франциско, США, 2010
- [6] Zdenek Kalal, Krystian Mikolajczyk, Jiri Matas (2010) Forward-Backward Error: Automatic Detection of Tracking Failures. International Conference on Pattern Recognition, 23-26 Августа 2010, Стамбул, Турция
- [7] Donald Knuth (1969) The Art of Computer Programming. Volume 2. — Addison-Wesley.