Термины и определения

Трекер – программная реализация процедуры отслеживания объекта.

Вступление

/\*

Общие слова про отслеживание объектов, где оно используется и почему актуально.

\*/

Визуальное отслеживание заключается в последовательном определении местоположения целевого объекта на каждом кадре видеопотока. Оно по-прежнему остается открытой проблемой во многих областях, так или иначе связанных с обработкой видеозаписей и получением из них некоторой информации. В качестве примеров можно привести системы человеко-машинного взаимодействия, системы видеонаблюдения, системы анализа спортивных матчей, появившиеся в недавнее время системы «умного дома», и многие-многие другие.

Несмотря на значительное количество научных работ и исследований, проводимых в данной области, точное и устойчивое к ошибкам отслеживание объектов на видео остается сложной проблемой. Небольшие размеры объектов, меняющийся фон, резкие ускорения и смены траектории движения объектов, частичные или полные перекрытия, изменение масштаба отслеживаемой цели при ее приближении или удалении от камеры являются основными трудностями на пути к построению точной траектории перемещения объекта на кадрах видеозаписи.

В последнее время достаточно широкое распространение получила технология «фильтра частиц» (particle filter), предоставляющая широкие возможности и гибкий инструментарий для отслеживания деформируемых объектов в видеопоследовательностях со сложным изменяющимся фоном. В данной статье речь пойдет о конкретном примере применения фильтра частиц для покадрового определения местоположения объекта. В первой главе будет дан краткий обзор существующих на сегодняшний день подходов к отслеживанию объектов в видеопотоке. Далее будут рассмотрены основные принципы метода фильтрации частиц. Третья глава будет посвящена реализации данного метода в рамках задачи отслеживания объекта на видеозаписи. В последующих главах будет произведен анализ полученных с помощью фильтра частиц результатов.

# Подходы к отслеживанию объектов в видеопотоке

Результаты визуального отслеживания сильно зависят от эффективности определения положения объекта на отдельно взятом кадре, что делает задачи отслеживания и распознавания объекта тесно связанными между собой. На этой связи основаны два общих подхода к отслеживанию объектов: отслеживание с помощью распознавания и распознавание с помощью отслеживания.

В первом подходе ключевую роль играют алгоритмы распознавания. С их помощью определяются новые объекты, появляющиеся в сцене, и инициализируются трекеры. Также они могут предоставлять точные оценки положения цели, увеличивая тем самым точность результатов работы трекера. В этом случае процесс отслеживания может быть сведен к задаче сопоставления объектов, распознанных на текущем кадре, с объектами, выявленными на предыдущем кадре.

В качестве примера алгоритма, реализующего данный подход, можно выделить AdaBoost [1]. Основная идея данного алгоритма – представить проблему отслеживания в виде задачи бинарной классификации. Принцип работы заключается в следующем: зная положение объекта на текущем кадре , классификатор вычисляет как можно больше возможных положений объекта на следующем кадре заданной области поиска, после чего анализируется полученная в результате вычислений карта достоверности (confidence map) с целью найти наиболее вероятное положение объекта, и, наконец, происходит обновление классификатора. Для построения используемых классификатором гипотез применяется процесс сопоставления с эталоном (template matching). Для сравнения используются признаки Хаара (Haar-like features), гистограммы направленных градиентов (Histogram of Oriented Gradients, HOG) и локальные бинарные шаблоны (Local Binary Patterns, LBP).

Другим алгоритмом, использующим отслеживание с помощью распознавания, является алгоритм, основанный на методе опорных векторов (Support Vector Machine, SVM) [2]. Выделение объектов происходит путем определения пикселей фона с помощью смешанной Гауссовой модели (Gaussian Mixture Model, GMM). При этом могут быть выделены некоторые посторонние объекты, не представляющие интереса для трекера. Для разбиения выделенных объектов на классы интересных и неинтересных с точки зрения отслеживания используется классификатор, обученный с помощью метода опорных векторов.

Необходимо отметить, что на алгоритмы, реализующие отслеживание путем распознавания, накладываются два основных ограничения: во-первых, отслеживаемый объект должен иметь отличительные внешние характеристики (цвет, текстура и т.п.), а во-вторых, не должен претерпевать значительных изменений на соседних кадрах. Таким образом, эти алгоритмы неэффективны в тех случаях, когда нельзя заранее предсказать визуальные характеристики цели, например, при быстрых перемещениях объекта, когда его внешний вид также меняется достаточно быстро. Данный класс алгоритмов не подходит также для отслеживания целей небольших размеров и целей, не имеющих достаточных отличительных признаков.

С описанными ограничениями справляется второй подход, распознавание на основе отслеживания. В этом подходе используется слабая низкоуровневая модель внешнего представления объекта. На каждом шаге вычисляется функция плотности распределения вероятности положения объекта. Алгоритмы, реализующие данный подход, делятся на два основных класса: детерминированные и стохастические (вероятностные).

Наиболее известными представителями класса детерминированных алгоритмов отслеживания являются сдвиг среднего (Mean Shift) и непрерывно адаптивный сдвиг среднего (Continuously Adaptive Mean-shift, CAM-shift). Они определяют для каждого кадра весовую функцию, такую, что положение объекта соотносится с ее минимальным (или максимальным) значением. В этом случае задача визуального отслеживания сводится к оптимизации весовой функции. Для поиска минимума (максимума) алгоритмы сдвига среднего используют метод градиентного спуска. Проблемой в данном случае является выбор начальной точки для спуска. Таким образом, необходимо решать вопрос о нахождении локального оптимума, который, в конечном итоге, может оказаться достаточно далеко от реального положения объекта. По этой причине детерминированные алгоритмы показывают неудовлетворительные результаты на достаточно долгих видеопоследовательностях, так как внешний вид объекта может коренным образом меняться на протяжении видеозаписи, вследствие чего весовая функция скорее всего будет мультимодальна, и определение его положения станет слишком непредсказуемо.

Стохастические (вероятностные) алгоритмы отслеживания основаны на теории байесовского оценивания. В них задача отслеживания сводится к итеративной оценке функции плотности распределения вероятности вектора состояния объекта, производимой на основе измерения его значений на последовательных кадрах. Вектор состояния объекта может описывать его положение, размер, скорость или другие характеристики. Основные трудности в таком подходе связаны с процессом измерения значения вектора состояния. Этот процесс чувствителен к шумовым воздействиям, что может негативно повлиять на получаемые значения вектора. Чтобы получить фактическое значение вектора состояния объекта по зашумленным наблюдениям, необходимо применить процесс фильтрации.

Широко известным представителем данного класса отслеживателей является фильтр Кальмана (Kalman Filter). Он использует предположение, что функция плотности распределения вероятности вектора состояний является Гауссовой, что позволяет получить аналитическое решение ее оценки. К сожалению, данная гипотеза во многих случаях не верна. Кроме того, фильтр Кальмана в чистом виде не применим к нелинейным системам. Для решения этой проблемы были разработаны модификации данного метода: расширенный фильтр Кальмана (Extended Kalman Filter, EKF), применяющий процесс линеаризации, и ансцентный фильтр Кальмана[[1]](#footnote-1) (Unscented Kalman Filter, UKF), использующий детерминированные выборки. Однако в большинстве своем эти решения показывают удовлетворительные результаты только при задании строгих ограничений на область применения алгоритма. Обычно они требуют задания большого количества параметров, настройка которых поможет адаптировать алгоритм для решения конкретной задачи. Но и в этом случае остается проблема обработки длинных видеопоследовательностей и ситуаций с резкими изменениями формы и внешнего вида объекта.

Значительный интерес в области вероятностного отслеживания вызывают методы Монте-Карло, в частности, фильтр частиц. Эти методы работают с несколькими гипотезами одновременно, что обуславливает их естественную способность адаптироваться к изменениям, происходящим с отслеживаемым объектом. Более того, фильтр частиц способен справляться с нелинейными процессами измерения, оставаясь при этом простым и эффективным для реализации. Но и он предполагает наложение ряда ограничений, особенно в тех случаях, когда целевой объект не имеет характерных отличительных внешних признаков. В следующей главе этот алгоритм будет рассмотрен подробнее.

# Метод фильтра частиц

2.1. Принцип вероятностного отслеживания.

Вероятностное отслеживание основано на байесовском подходе, который заключается в том, чтобы построить функцию плотности распределения вероятности (probability density function, pdf) вектора состояния, используя всю доступную к данному моменту информацию. Эта функция характеризует текущую степень знания о векторе состояния, и с ее помощью можно определить оптимальный (с точки зрения выбранной пользователем функции стоимости) порядок действий.

Проблему отслеживания можно сформулировать с помощью следующего математического представления пространства состояний объекта:

(2.1)

(2.2)

Уравнение описывает динамику системы, то есть изменение ее состояния при переходе от кадра к кадру . Фактически состояние зависит от предыдущего состояния объекта и стохастической ошибки , представляющей собой погрешность в обновлении состояния. Поскольку ошибка является случайной величиной с известной статистикой, уравнение неявно задает функцию плотности распределения вероятности . Уравнение (2.2) описывает зависимость наблюдения от текущего состояния и вектора ошибки , который задает погрешность процесса измерения состояния объекта. Аналогично (2.1), поскольку является стохастической переменной, уравнение (2.2) неявно задает функцию правдоподобия .

С точки зрения байесовского подхода проблема отслеживания заключается в рекурсивном вычислении некоторой степени доверия к состоянию объекта на кадре с учетом всех имеющихся к текущему моменту наблюдений. Таким образом, необходимо вычислить функцию плотности распределения вероятности ), называемую постериорной. Вычисление данной функции проводится в два этапа: предсказание и обновление (коррекция). Полагая известным значение искомой функции на предыдущем кадре, с помощью уравнения Чепмена-Колмогорова можно вычислить априорную функцию плотности распределения вероятности состояния объекта на кадре :

(2.3)

В этом заключается этап предсказания. На кадре становится известным наблюдение , которое может быть использовано для обновления априорной функции по правилу Байеса:

(2.4)

где – нормирующий множитель, зависящий от функции правдоподобия.

Рекурсивные соотношения (2.3) и (2.4) формируют базис для нахождения оптимального байесовского решения. Однако данный подход является концептуальным с той точки зрения, что в общем случае это решение не может быть получено аналитическим путем. Примером частного случая, когда аналитическое решение поставленной задачи все-таки существует, является упомянутый в предыдущей главе фильтр Кальмана. В более сложных случаях искомое решение можно получить с помощью аппроксимации. Одним из возможных алгоритмов для этого является фильтр частиц [3].

* 1. Фильтр частиц

В основе алгоритма лежит аппроксимация функции плотности распределения вероятности состояния объекта коллекцией взвешенных выборок – частиц, поведение которых регулируется с помощью их весов. Фактически, частица является представлением некоторого возможного состояния объекта. Множество (2.5) задает приближенную функцию плотности распределения вероятности для состояния при заданном наборе наблюдений .

(2.5)

Алгоритм воспроизведения условной плотности (CONditional DENSity propAGATION, CONDENSATION) – базовый алгоритм фильтрации частиц, на основании которого строится большинство алгоритмов данной группы, применяемых в компьютерном зрении. Задача состоит в том, чтобы построить метод восстановления множества на основании . Согласно [4] алгоритм состоит из следующих этапов:

1. Пусть построена коллекция взвешенных выборок для кадра :

.

Необходимо вычислить интегральные веса согласно .

(2.6)

1. Определим экземпляр выборки . Для этого выберем случайное число и вычислим . Таким образом, частица попадает в множество с вероятностью, равной ее весовому коэффициенту.
2. Предсказание следующего состояния согласно уравнению динамики (2.1):

(2.7)

Алгоритм SIR и его адаптивный вариант.

Описание алгоритмов, которые я уже реализовала.

1. Оценка качества работы алгоритма отслеживания.

Кратко какие есть способы оценки, и какую я сама применяла.

1. Сравнение результатов работы простого и адаптивного алгоритма

Используя выбранную в предыдущем пункте оценку сравнить, что получилось для простого и для адаптивного алгоритмов. Здесь будут графики этой оценки для разных видео, какие-нибудь сравнительные таблицы…

Заключение

Краткие выводы, направление дальнейшего исследования

Литература

[1] Helmut Grabner, Michael Grabner, and Horst Bischof. Real-time tracking

via on-line boosting. InBritish Machine Vision Conference, 2006

[2] Guangyu Zhu; Changsheng Xu; Qingming Huang; Wen Gao, "Automatic Multi-Player Detection and Tracking in Broadcast Sports Video using Support Vector Machine and Particle Filter," *Multimedia and Expo, 2006 IEEE International Conference on* , vol., no., pp.1629,1632, 9-12 July 2006

[3] Arulampalam, M.S.; Maskell, S.; Gordon, N.; Clapp, T., "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *Signal Processing, IEEE Transactions on* , vol.50, no.2, pp.174,188, Feb 2002

[4] Кустикова В.Д. Отслеживание движения и алгоритмы сопровождения ключевых точек [лекция].

1. В названии алгоритма приведена транслитерация оригинального названия, поскольку устоявшегося термина в русском языке пока нет, а дословный перевод не имеет смысла (unscented – «без запаха») [↑](#footnote-ref-1)