

Отслеживание контура лабораторной мыши в видеопоследовательности на основе метода ветвей и границ

Михаил Синдеев¹, Антон Конушин²

¹Институт прикладной математики им. М.В.Келдыша РАН

²Лаборатория компьютерной графики и мультимедиа ВМК МГУ им. М.В. Ломоносова

E-mail: m_sindeev@mail.ru, ktosh@graphics.cs.msu.ru

Аннотация

В данной статье рассматривается задача отслеживания контура лабораторных животных в видеопоследовательности на основе заданной модели формы и предлагается алгоритм для решения данной задачи, основанный на методе ветвей и границ.

Ключевые слова: видеорегистрация, слежение, сегментация, вычитание фона

1. ВВЕДЕНИЕ

При проведении экспериментальных исследований на лабораторных животных важной задачей является количественная оценка поведения. Для этого поведение регистрируется с помощью видеокамеры, затем по полученной видеопоследовательности отслеживается траектория движения животного и для каждого момента времени определяется ряд характеристик тела животного, такие как поворот и изгиб. На основе этих данных проводится анализ поведения, например, поведение разбивается на отдельные поведенческие акты, вычисляется общая длина пути, пройденного животным, скорость движения животного, время нахождения в выделенных функциональных зонах и т.д.

Стандартная экспериментальная установка состоит из огороженной арены и жестко закрепленной над ней видеокамеры. В рассматриваемом семействе экспериментов одновременно на арене может находиться не более одного животного.

В этой статье рассматривается задача отслеживания контура лабораторного животного по видеопоследовательности, полученной со стандартной экспериментальной установки. Для каждого кадра видеозаписи необходимо построить маску животного – бинарное изображение, в каждом пикселе которого метка «0» соответствует фону, а «1» – объекту. Также необходимо найти определенные ключевые точки ([1]), которые используются при анализе поведения животного.

Алгоритмы отслеживания лабораторных животных можно разделить на две группы – онлайн и оффлайн трекинг. При онлайн-трекинге построение маски требуется в реальном времени, т.е. скорость работы программы должна соответствовать скорости поступления кадров видео. Онлайн-трекинг требуется, например, в случае, когда программа должна управлять механизмами арены в зависимости от действий животного (например, открывать/закрывать дверцы в лабиринте, когда животное совершает нужное действие).

При оффлайн-трекинге обрабатывается заранее полученная видеопоследовательность и нет жестких ограничений на скорость работы, но требуется хранить всю видеопоследовательность на внешних запоминающих устройствах.

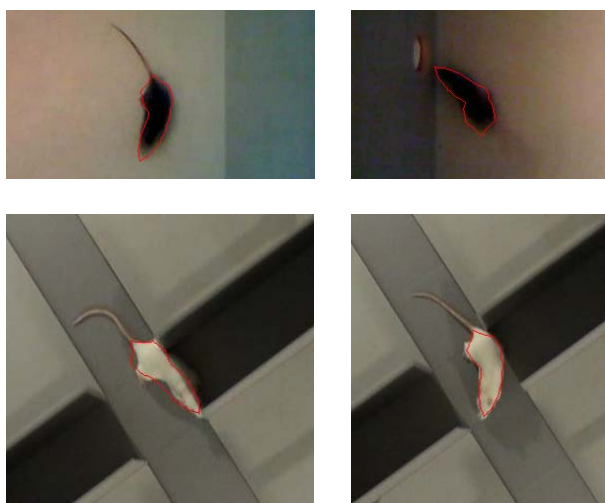


Рисунок 1. Результат работы алгоритма. Показаны фрагменты кадров видеопоследовательностей с наложением найденного контура мыши.

В данной статье предлагается алгоритм для задачи высокоточного оффлайн-трекинга. Он также может также быть применен для генерации эталонных данных для тестирования других алгоритмов онлайн- и оффлайн-трекинга.

2. ОБЗОР СУЩЕСТВУЮЩИХ МЕТОДОВ

Стандартный подход при отслеживании лабораторных животных – попиксельное вычитание фона ([2]). Такой метод применяется в коммерческих системах [1], [3].

Существуют также расширения этого метода для динамического фона, например [4], где для повышения устойчивости к динамическому фону используется моделирование окрестности пикселя (гистограмма цветов).

Попиксельные методы дают шумную маску и не устойчивы к теням (которые, по условию задачи, алгоритм должен относить к фону, но часто не справляется с этим, так как тень имеет отличный от фона цвет). Для фильтрации которой обычно применяются эвристические методы на основе математической морфологии, которые требуют ручной подгонки радиуса фильтра для каждой видеопоследовательности (или даже для ее фрагментов).

Для нахождения опорных точек на контуре животного (нос, основание хвоста) и постобработки маски может быть использован метод на основе активных моделей формы, но он оперирует с бинарной маской, поэтому часто ошибочно присоединяет тень к контуру животного (**Error! Reference source not found.**). Аналогичные проблемы при использова-

нии кинематических моделей [6]. Был предложен ряд алгоритмов, использующих глобальную оптимизацию и учитывающих контраст (например, [7]), но они по-прежнему не устойчивы к теням, т.к. граница тени может быть довольно резкой.

Для более устойчивого трекинга необходимо на этапе построения маски использовать информацию о форме животного. Для сегментации объектов с использованием априорной информации о форме в статичных изображениях был предложен метод ветвей и границ. Этот метод уже применялся для сегментации изображений [8] при помощи разрезов графов, однако в таком варианте не подходит для видеопоследовательностей, так как работает слишком долго.

3. АЛГОРИТМ

Мышь моделируется контуром заданной формы с четырьмя степенями свободы: две компоненты сдвига (в плоскости), поворот, изгиб. Такой набор параметров соответствует реальным степеням свободы рассматриваемых животных (при этом

остальные степени свободы совершают пренебрежимо малый вклад в видимое положение животного), а также позволяет получить необходимые данные для анализа – положение центра масс, плюс ключевые точки на контуре (для мыши – точка носа и точка «крепления» хвоста).

Нахождение положения маски сводится к минимизации функционала

$$E = \iint_{\Omega} L(\omega) d\omega, \quad (1)$$

где $\Omega = \Omega(cx, cy, angle, bend)$ – внутренняя область контура при конкретных значениях параметров центра (cx, cy), угла ($angle$) и изгиба ($bend$), а $L(\omega)$ – потенциал фона в данной точке ω на изображении. Если внутри некоторой области потенциал фона минимален, то данная область, скорее всего, соответствует искомому объекту.

3.1 Вычитание фона

Для нахождения потенциала фона используется вероятностная модель. Цвет каждого пикселя на изображении фона моделируется нормальным распределением. Параметры распределения – математическое ожидание и дисперсия/матрица ковариации – оцениваются на этапе захвата фона, который осуществляется перед помещением животного на арену. В процессе трекинга фон обновляется на основе алгоритма бегущего среднего, см. [2]. Условием применимости данной модели является разделение цветов объекта и фона, а также отсутствие переходных цветов (например, теней). Для выполнения этого условия цвет арены выбирается контрастным по отношению к цвету животного. Например, черные мыши снимаются на белом фоне, а белые – на темно-сером или черном. Освещение должно быть рассеянным и равномерным с разных сторон, чтобы снизить затенение областей вокруг животного. Ввиду статичности фона более сложные модели на основе смесей нормальных распределений [2] и гистограмм [4] не используются, т.к. на этих данных дают аналогичные результаты, но с большими затратами времени и памяти.

Положив потенциал равным логарифму вероятности:

$$L = \log(p) + c, \quad (2)$$

получим в формуле (1), в предположении о независимости меток всех пикселей, плотность вероятности нахождения маски животного в заданном положении (определяемом параметрами $cx, cy, angle, bend$) при $c = 0$. Выберем константу c так, чтобы все значения L были неположительными:

$$c = -\max \log(p), L \leq 0. \quad (3)$$

3.2 Поиск положения модели животного

Глобальный минимум функционала (1), очевидно, может быть найден с заданной точностью перебором всех значений параметров на четырехмерной дискретной сетки, шаг которой определяется требуемой точностью.

Однако такой перебор имеет экспоненциальную сложность и не может быть выполнен за приемлемое время.

Локальные алгоритмы трекинга используют небольшое пространство поиска в окрестности результата, найденного на предыдущем кадре (можно считать, что на первом кадре положение животного задается пользователем). Основным недостатком таких алгоритмов является низкая стабильность: ошибки отслеживания накапливаются, а сбивание алгоритма хотя бы на одном кадре приводит к неверному результату и на всех последующих кадрах.

В данной статье предлагается использовать для глобальной оптимизации функционала (1) метод ветвей и границ.

Пусть некоторая область Ω_1 вложена в область Ω_2 . Тогда

$$E_1 = \iint_{\Omega_1} L(\omega) d\omega \geq \iint_{\Omega_2} L(\omega) d\omega = E_2 \quad (4)$$

при условии неположительности значений L (3).

Здесь E_1 и E_2 не обязательно соответствуют значениям функционала (1), т.к. области Ω_1, Ω_2 не обязательно соответствуют возможным маскам для заданной модели животного. Однако данное неравенство позволяет получать нижние оценки значений функционала (1).

Рассмотрим пространство поиска для значений $cx, cy, angle, bend$: $[0, W] \times [0, H] \times [0^\circ, 360^\circ] \times [-bend_limit, bend_limit]$, где W, H – ширина и высота изображения, а $bend_limit$ – максимальный угол изгиба животного.

Пусть Ω_2 – объединение масок для некоторого пространства поиска, вложенного в исходное или совпадающее с ним (объединение масок – маска, в которой пиксель имеет метку «1», если хотя бы в одной из объединяемых масок данный пиксель имеет эту метку, и «0» в противном случае, т.е. это попиксельная дизъюнкция масок). Тогда значение функционала (1) в каждой маске Ω_1 из этого пространства удовлетворяет неравенству (4).

Метод ветвей и границ можно применить следующим образом. Поместим исходное пространство поиска в буфер. Разбивая пространство поиска пополам вдоль одной из координат $cx, cy, angle, bend$ и вычисляя верхнюю оценку по объему пространства поиска, будем выбирать подпространство с наименьшей нижней оценкой энергии для дальнейшего разбиения, а остальные помещать в буфер вместо разбиваемого

подпространства. Выбор подпространства осуществляется среди всех подпространств в буфере. Когда выбранное подпространство будет достаточно малым, остановим разбиение и положим результат всего алгоритма равным некоторой точке этого подпространства. Таким образом, алгоритм будет спускаться от верхней оценки значения функционала (1) к глобальному минимуму этого функционала.

На реальных данных число шагов такого алгоритма полиномиальное, против экспоненциального при полном переборе.

В отличие от алгоритма **Error! Reference source not found.**, поиск положения животного осуществляется на основе карты вероятностей L , без построения предварительной грубой маски, что дает более стабильный результат (за счет того, что эта карта придает пикселям разный вес).

3.3 Уточнение поворота

При некоторых позах рассматриваемое животное (мышь) имеет симметричную форму относительно плоскости, разделяющей переднюю и заднюю половинки. Из-за этого примерно в 15% случаев найденный угол поворота имеет ошибку 180° . При малых размерах мыши ошибка проявляется ещё чаще, так как мышь становится почти округлой. Хвост нарушает симметрию, но его обнаружение затруднено из-за малой толщины при низком качестве видео (веб-камера), сильным размытием при движении хвоста из стороны в сторону и большим числом степеней свободы в рамках рассматриваемой модели на основе масок (много возможных точек изгиба).

Для исправления таких ошибок предлагается алгоритм, основанный на том, что мышь чаще движется вперед, чем назад, и движется вперед быстрее, чем назад. Математически это можно смоделировать как максимизацию скалярного произведения вектора скорости на вектор направления мыши.

Сопоставив каждому кадру t булеву переменную $f(t)$ со значениями «0» – направление выбрано правильно и «1» – направление следует изменить на 180° (и изменить знак угла изгиба $bend$) можно сформулировать оценочную функцию

$$C = \sum_{t=1}^T V(t) \cdot D(t, f(t)) + \lambda \cdot \sum_{t=2}^T D(t-1, f(t-1)) \cdot D(t, f(t)), \quad (5)$$

где $V(t)$ – вектор скорости (вычисляемый как разность координат центра (cx, cy) в соседних кадрах), $D(t, f)$ – единичный вектор направления в кадре t , в предположении, что угол равен $angle$, если $f = 0$ и $angle + 180^\circ$, если $f = 1$. T – общее число кадров в видеопоследовательности. Второе слагаемое отвечает за согласованность направлений мыши между кадрами. Такую оценочную функцию можно максимизировать методом динамического программирования [9].

Данный алгоритм применим также и к онлайн-методам трекинга, страдающим от 180° -градусных ошибок на при вычисления поворота. Суммирование в формуле (5) в этом случае осуществляется до текущего кадра, а хранение значений C только для предыдущего и текущего кадров (всего 4 значения: для $f = 0, 1$ в кадре $t - 1$ и для $f = 0, 1$ в кадре t). В онлайн-варианте возможны ошибки в течение первых нескольких кадров, но затем алгоритм стабилизируется.

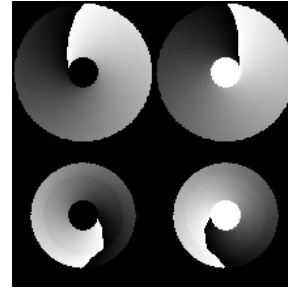


Рисунок 2. Карта поворота половинок животного (мыши). Верхний ряд: передняя часть, нижний: задняя. Карты слева определяют угол появления метки «1» в заданном пикселе, карты справа – угол исчезновения этой метки. Черный цвет в шаблонах соответствует углу 0° , белый – 360° . Черная область вне шаблонов задается специальным значением -1 .

4. РЕАЛИЗАЦИЯ

В реализации пространство поиска является дискретным с шагом в 1 пиксель по координатам и 1° по углу поворота и углу изгиба. Максимальный угол изгиба $bend_limit$ равен 35° и соответствует независимому повороту верхней и нижней половинки животного в разных направлениях. Форма животного моделируется масками этих двух половинок. Маски обеих половинок имеют в точке центра круг («шарнир»), который гарантирует, что объединение половинок будет цельным объектом, достаточно гладким в месте соединения (гладкости в математическом смысле не требуется – достаточно лишь схожести с реальным контуром животного на изображении).

Разбиение пространства поиска производится по координате, имеющей наибольший диапазон. Для параметров разной размерности используется соотношение 1 пиксель $\sim 1^\circ$.

Объединение масок вычисляется сначала по углу поворота и изгиба: для каждой половинки вычисляются границы диапазона углов $[angle - bend, angle + bend]$, где $angle$ и $bend$ – также диапазоны для текущего пространства поиска. Для получения самой маски используются заранее посчитанные карты поворота каждой из половинок (Рисунок 2).

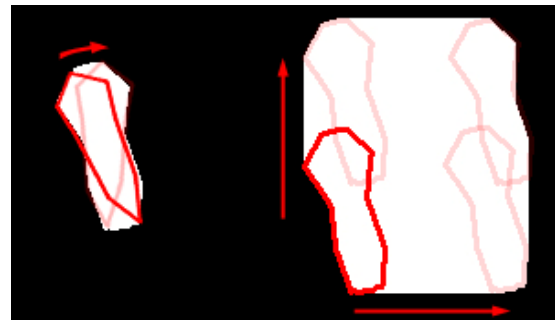


Рисунок 3. Построение объединения масок по диапазону параметров. Слева: объединение по углу, справа: последующее объединение по диапазону координат. Изгиб отсутствует ($bend = 0$).

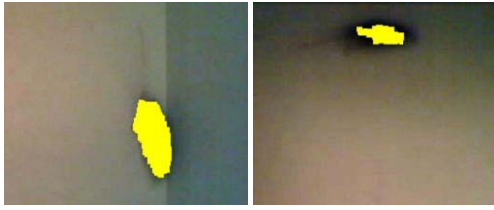


Рисунок 4. Алгоритм порога (на основе вычитания фона) может давать маску неправильной формы, т.к. в нее попадают тени и шум. Кроме того, не всегда возможно выбрать значение порога, которое давало бы правильный результат в разных областях арены и при разных условиях съемки.

Затем строится объединение маски по диапазонам координат x , y путем расположения масок, объединенных по повороту и изгибу, в углах прямоугольника и их соединения (Рисунок 3).

Предложенный алгоритм был сравнен с алгоритмом порога на основе вычитания фона (Рисунок 4) на наборе видеопоследовательностей, предоставленных ООО «НПК Открытая Наука» [10] и НИИ Нормальной физиологии им. П.К. Анохина РАМН [11]. Результат оказался более точным и стабильным, так как использует модель формы животного и находит наиболее подходящую область, в то время как пороговый алгоритм может находить область большего или меньшего размера, а также неправильной формы. Кроме того, предложенный алгоритм находит глобальный минимум функционала энергии, без какой-либо привязки к значению этого функционала. Пороговый алгоритм требует точного указания значения порога, по которому происходит классификация пикселей, однако такое значение не всегда можно выбрать оптимальным образом: например, в крестообразном лабиринте с высокими стенками вдоль двух «рукавов» лабиринта (Рисунок 5) мышь может одновременно находиться в областях разной освещенности.

5. ЗАКЛЮЧЕНИЕ

В данной статье предложен алгоритм оффлайн-трекинга лабораторных животных по видеопоследовательности, устойчивый к ошибкам обнаружения и изменению освещенности. Состояние животного в каждый момент времени описывается в виде вектора из четырех параметров: координат (x , y) центра модели животного и двух углов – поворота и изгиба. Задача сформулирована в виде поиска глобального оптимума функции и решается методом ветвей и границ.

5.1 Дальнейшая работа

Предложенный алгоритм оффлайн-трекинга на основе метода ветвей и границ планируется ускорить, сделав его применимым для онлайн-трекинга. Этого можно достичь иерархической обработкой изображения, уменьшением пространства поиска (рассматривать только небольшую окрестность результата предыдущего кадра), а также ускорением вычислительных процедур алгоритма. В частности, сумму по заданному объединению масок можно вычислять на основе интегральных изображений, учитывая, что значительную часть такого объединения составляет прямоугольник (см. Рисунок 3, справа); оставшуюся часть объединения также можно представить набором прямоугольников и горизонтальных/вертикальных отрезков.

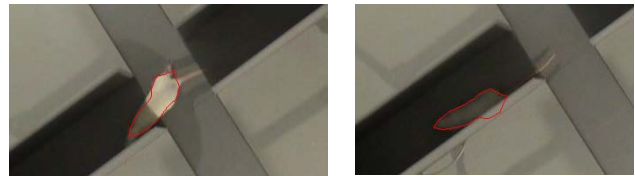


Рисунок 5. Результаты работы предлагаемого алгоритма в сложных случаях на примере крестообразного лабиринта (показан сверху). Слева: разные условия освещенности. Справа: частичное перекрытие стенкой лабиринта.

6. БЛАГОДАРНОСТЬ

Работа выполнена при поддержке федеральной целевой программы "Научные и научно-педагогические кадры инновационной России" на 2009 - 2013 годы", проект №П264 от 23 июля 2009 г.

7. ЛИТЕРАТУРА

- [1] L. Noldus, A. Spink. EthoVision: a versatile video tracking system for automation of behavioral experiments. *Behavior Research Methods, Instruments & Computers*, 33, pp. 398–414.
- [2] M. Piccardi. Background subtraction techniques: a review. In *IEEE International Conference on Systems, Man and Cybernetics*, vol. 4. IEEE, October 2004, pp. 3099–3104.
- [3] Stoelting ANY-maze, <http://www.anymaze.com/>
- [4] T. Ko, S. Soatto, D. Estrin. Background Subtraction on Distributions. In *ECCV 2008, Part III, LNCS 5304*, pp. 276–289.
- [5] Конушин А, Ломакина-Румянцева Е, Ветров Д, Воронин П, Синдеев М. Система видеонаблюдения за поведением лабораторных животных с автоматической сегментацией на поведенческие акты. In: *GraphiCon, 2008*. pp. 199-205.
- [6] Т. Мухина, И. Зарайская, К. Анохин. Отслеживание движений животных с использованием 2D модели. *Математические методы распознавания образов (ММРО-11): Сборник докладов 11-й Всероссийской конференции*, сс. 378–380.
- [7] J. Sun, W. Zhang, X. Tang, H. Shum. Background cut. In *ECCV, 2006*, pp. 628–641.
- [8] V. Lempitsky, A. Blake, C. Rother. Image Segmentation by Branch-and-Mincut. In *ECCV 2008, Vol. 5305*, pp. 15–29.
- [9] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.
- [10] ООО «НПК Открытая Наука», <http://openscience.ru/>
- [11] НИИ Нормальной физиологии им. П.К. Анохина, РАМН, <http://nphys.ru>