1. Что означает термин «история движения»? Объясните назначение параметров в функции motionHistoryImage().

Изображение истории движения (MHI) - это шаблон статического изображения, который помогает понять местоположение и путь движения по мере его продвижения.

В MHI информация о временном движении сворачивается в единый шаблон изображения, где интенсивность является функцией недавности движения. Таким образом, интенсивность пикселей MHI является функцией истории движения в этом месте, где более яркие значения соответствуют более недавнему движению. Используя MHI, движущиеся части видеопоследовательности могут быть выгравированы одним изображением, из которого можно предсказать поток движения, а также движущиеся части видео действия.

Параметры:

* Силуэт - маска силуэта с ненулевыми пикселями, в которых происходит движение.
* Mhi - изображение истории движения, которое обновляется функцией (одноканальное, 32-битное с плавающей запятой).
* Timestamp - Текущее время в миллисекундах или других единицах.
* Duration - максимальная длительность дорожки движения в тех же единицах, что и временная метка.

1. Если мы хотим узнать на сколько тот или иной объект объект сместился по отношению к его же положению на предыдущем кадре за то время, которое прошло между фиксацией кадров, то скорее всего в первую очередь мы вспомним про оптический поток (optical flow).

Оптический поток (Optical flow) – технология, использующаяся в различных областях computer vision для определения сдвигов, сегментации, выделения объектов, компрессии видео.

Оптический поток (ОП) – изображение видимого движения, представляющее собой сдвиг каждой точки между двумя изображениями.

Алгоритм Лукаса — Канаде — широко используемый в компьютерном зрении дифференциальный локальный метод вычисления оптического потока.

Метод Лукаса-Канаде  
В основе всех дальнейших рассуждений лежит одно очень важное и не очень справедливое предположение: **Предположим, что значения пикселей переходят из одного кадра в следующий без изменений**. Таким образом, мы делаем допущение, что пиксели, относящиеся к одному и тому же объекту, могут сместиться в какую либо сторону, но их значение останется неизменным. Конечно же это предположение имеет мало общего с реальностью, потому что от кадра к кадру могут меняться глобальные условия освещения и освещенность самого движущегося объекта. Масса проблем связана с этим допущением, но, как ни странно, вопреки всему оно достаточно хорошо работает на практике.

Основное уравнение оптического потока содержит две независимые переменные и не может быть однозначно решенной. Алгоритм Лукаса — Канаде решает неоднозначность за счет использования информации о соседних пикселей на в каждой точке. Метод основан на предположении, что в локальном окрестности каждого пикселя значение оптического потока одинаково, таким образом таким образом можно записать уравнение оптического потока для всех точек в окрестности и решить систему уравнений методом наименьших квадратов.

<https://habr.com/ru/post/169055/>

**Алгоритм Лукаса — Канаде** — широко используемый в [компьютерном зрении](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%BE%D0%B5_%D0%B7%D1%80%D0%B5%D0%BD%D0%B8%D0%B5) дифференциальный локальный метод вычисления [оптического потока](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%BF%D0%BE%D1%82%D0%BE%D0%BA).

Основное уравнение оптического потока содержит две неизвестных и не может быть однозначно разрешено. Алгоритм Лукаса — Канаде обходит неоднозначность за счет использования информации о соседних [пикселях](https://ru.wikipedia.org/wiki/%D0%9F%D0%B8%D0%BA%D1%81%D0%B5%D0%BB%D1%8C) в каждой точке. Метод основан на предположении, что в локальной окрестности каждого пикселя значение оптического потока одинаково, таким образом можно записать основное уравнение оптического потока для всех пикселей окрестности и решить полученную систему уравнений [методом наименьших квадратов](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%BD%D0%B0%D0%B8%D0%BC%D0%B5%D0%BD%D1%8C%D1%88%D0%B8%D1%85_%D0%BA%D0%B2%D0%B0%D0%B4%D1%80%D0%B0%D1%82%D0%BE%D0%B2).[[1]](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%9B%D1%83%D0%BA%D0%B0%D1%81%D0%B0_%E2%80%94_%D0%9A%D0%B0%D0%BD%D0%B0%D0%B4%D0%B5#cite_note-LKproc-1)[[2]](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%9B%D1%83%D0%BA%D0%B0%D1%81%D0%B0_%E2%80%94_%D0%9A%D0%B0%D0%BD%D0%B0%D0%B4%D0%B5#cite_note-Lthesis-2)

Алгоритм Лукаса — Канаде менее чувствителен к шуму на изображениях, чем поточечные методы, однако является сугубо локальным и не может определить направление движения пикселей внутри однородных областей.

CalcOptycalFlowPyrLK():

Вычисляет оптический поток для разреженного набора функций с помощью итеративного метода Лукаса-Канаде с пирамидами.

Параметры

PrevImgfirst 8-битное входное изображение или пирамида, построенная buildOpticalFlowPyramid.

NextImg - второе входное изображение или пирамида того же размера и того же типа, что и prevImg.

PrevPtsvector 2D точек, для которых нужно найти поток; Координаты точки должны быть числами с плавающей запятой одинарной точности.

NextPtsoutput вектор двумерных точек (с координатами с плавающей запятой одинарной точности), содержащий вычисленные новые положения входных объектов во втором изображении; Когда передается флаг OPTFLOW\_USE\_INITIAL\_FLOW, вектор должен иметь тот же размер, что и на входе.

Statusoutput вектор состояния (беззнаковых символов); Каждый элемент вектора устанавливается в 1, если поток для соответствующих функций был найден, в противном случае он устанавливается в 0.

Erroutput вектор ошибок; Каждый элемент вектора устанавливается на ошибку для соответствующего признака, тип меры ошибки может быть установлен в параметре flags; Если поток не найден, ошибка не определена (для поиска таких случаев используйте параметр status).

WinSizesize окна поиска на каждом уровне пирамиды.

Максимальный номер уровня пирамиды на основе maxLevel0; Если установлено значение 0, пирамиды не используются (одноуровневые), если установлено значение 1, используются два уровня и так далее; Если на вход передаются пирамиды, алгоритм будет использовать столько же уровней, сколько и пирамид, но не более maxLevel.

Критерийпараметр, определяющий критерии завершения алгоритма итеративного поиска (после указанного максимального числа итераций критериев. maxCount или когда окно поиска перемещается меньше чем на критерии. epsilon.

Флаги операций:

OPTFLOW\_USE\_INITIAL\_FLOW использует начальные оценки, хранящиеся в nextPts; Если флаг не установлен, то prevPts копируется в nextPts и считается начальной оценкой.

OPTFLOW\_LK\_GET\_MIN\_EIGENVALS использовать минимальные собственные значения в качестве меры ошибки (см. Описание minEigThreshold); Если флаг не установлен, то расстояние L1 между фрагментами вокруг исходной и перемещенной точки, деленное на количество пикселей в окне, используется в качестве меры ошибки.

MinEigThreshold алгоритм вычисляет минимальное собственное значение нормальной матрицы 2x2 уравнений оптического потока (эта матрица называется матрицей пространственного градиента в [24]), деленное на количество пикселей в окне; Если это значение меньше minEigThreshold, то соответствующая функция отфильтровывается, и ее поток не обрабатывается, что позволяет удалить плохие точки и повысить производительность.

Функция реализует разреженную итерационную версию оптического потока Лукаса-Канаде в пирамидах. См. [24]. Функция распараллеливается с библиотекой TBB.

3. Trackers

**1.BOOTTING**Это отслеживание объектов в режиме реального времени на основе алгоритма AdaBoost.

**2.MIL**генерирует классификатор в онлайн-режиме, чтобы отделить объект от фона.

**3.KCF**-это новая система отслеживания, которая использует свойства циркулянтной матрицы для повышения скорости обработки

**4.MEDIANFLOW**подходит для очень плавных и предсказуемых движений, когда объект виден на протяжении всей последовательности.

**5.TLD**отслеживание, обучение и обнаружение. Трекерследует за объектом от кадра к кадру.Детектор локализует все видимые явления, которые наблюдались до сих пор, и при необходимости корректирует трекер.Изучение оценивает ошибки детекторов и обновляет их, чтобы избежать этих ошибок в будущем.

**6.MOSSE**Отслеживание визуальных объектов с использованием адаптивных корреляционных фильтров

**7.GOTURN**является своего рода трекерамина основе сверточныхнейронных сетей (CNN). Текущий метод GOTURN не обрабатывает окклюзии;однако он довольно устойчив к изменениям зрения, изменениям освещения и деформациям.

Camshift:

Meanshift:

Интуиция, лежащая в основе этого средства, проста. Считайте, что у вас есть набор очков. (Это может быть распределение пикселей, такое как обратная проекция гистограммы). Вам предоставляется небольшое окно (может быть кружком), и вы должны переместить это окно в область максимальной плотности пикселей (или максимального количества точек).

Внимательно следили за последним результатом? Существует проблема. Наше окно всегда имеет одинаковый размер, независимо от того, находится ли автомобиль очень далеко или очень близко от камеры. Это не хорошо. Нам нужно адаптировать размер окна к размеру и повороту цели. И снова решение пришло из «OpenCV Labs» и называется CAMshift (Continuously Adaptive Meanshift), опубликованным Гэри Брадски в его статье «Отслеживание лиц с компьютерным зрением для использования в перцептивном пользовательском интерфейсе» в 1998 году [28]. Сначала применяется метод переключения средств. Как только средство смещения сходится, он обновляет размер окна как s = 2 × M00256 −−− √. Он также вычисляет ориентацию наиболее подходящего к нему эллипса. Снова он применяет смещение средних значений с новым масштабированным окном поиска и предыдущим местоположением окна. Процесс продолжается до тех пор, пока не будет достигнута требуемая точность.

YOLO

Они применяют модель к изображению в разных местах и ​​в разных масштабах. Области изображения с высокими оценками считаются обнаружениями.  
  
Мы используем совершенно другой подход. Мы применяем единую нейронную сеть к полному изображению. Эта сеть делит изображение на области и предсказывает ограничивающие рамки и вероятности для каждой области. Эти ограничивающие прямоугольники взвешиваются по прогнозируемым вероятностям.

Он смотрит на все изображение во время тестирования, поэтому его прогнозы основываются на глобальном контексте изображения. Он также делает прогнозы с одной оценкой сети, в отличие от таких систем, как R-CNN, которые требуют тысячи для одного изображения.