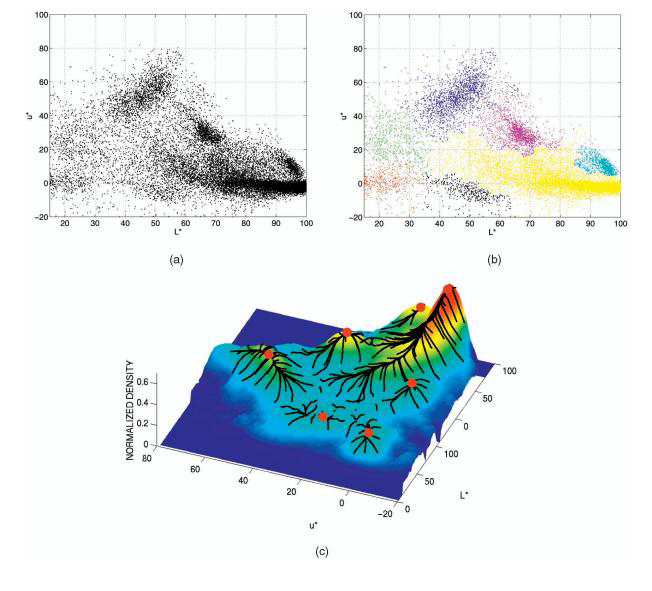
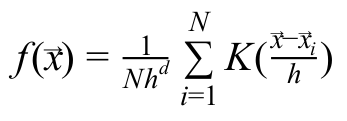
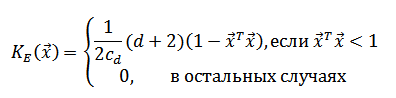
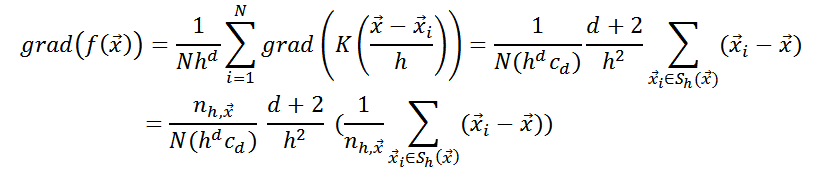
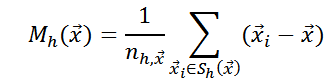
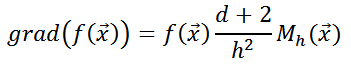
Алгоритм сегментации **MeanShift**

MeanShift группирует объекты с близкими признаками. Пиксели со схожими признаками объединяются в один сегмент, на выходе получаем изображение с однородными областями.   
  
   
  
Например, в качестве координат***в пространстве признаков*** можно выбрать координаты пикселя ***(x, y)*** и компоненты ***RGB*** пикселя. Изобразив пиксели в пространстве признаков, можно заметить сгущения в определенных местах.



*Рис. 7. (a) Пиксели в двухмерном пространстве признаков. (b) Пиксели, пришедшие в один локальный максимум, окрашены в один цвет. (с) https://habrastorage.org/files/451/946/6bf/4519466bfe504e92aa4bb5ee3a90d304.png — функция плотности, максимумы соответствуют местам наибольшей концентрации пикселей. Рисунок взят из статьи [3].*   
  
Чтобы легче было описывать сгущения точек, вводится ***функция плотности***:   
https://habrastorage.org/files/bd7/53a/f56/bd753af56ff94f5895b4d9643c216f2c.png– вектор признаков***i***-ого пикселя,***d*** — количество признаков, ***N*** — число пикселей,***h***— параметр, отвечающий за гладкость, https://habrastorage.org/files/1d1/cd3/21a/1d1cd321acfd4b50abd418400a59bacb.png — ядро. Максимумы функции https://habrastorage.org/files/451/946/6bf/4519466bfe504e92aa4bb5ee3a90d304.png расположены в точках сгущения пикселей изображения в пространстве признаков. Пиксели, принадлежащие одному локальному максимуму, объединяются в один сегмент. Получается, чтобы найти к какому из центров сгущения относится пиксель, надо шагать по градиенту https://habrastorage.org/files/451/946/6bf/4519466bfe504e92aa4bb5ee3a90d304.png для нахождения ближайшего локального максимума. 

Оценка градиента от функции плотности

Для оценки градиента функции плотности можно использовать вектор среднего сдвига https://habrastorage.org/files/6dd/c9f/a1c/6ddc9fa1c02b4cd3b160e6cf94f6f155.png   
В качестве ядра https://habrastorage.org/files/1d1/cd3/21a/1d1cd321acfd4b50abd418400a59bacb.png в OpenCV используется ядро Епанечникова [4]:  
  
https://habrastorage.org/files/5f8/dd0/4bb/5f8dd04bb2a04eb68da390c5e72a7b4e.png — это объем ***d***-мерной сферы c единичным радиусом.  
  
https://habrastorage.org/files/f34/2e2/005/f342e20058274b209bf447417208584d.png означает, что сумма идет не по всем пикселям, а только по тем, которые попали в сферу радиусом***h*** с центром в точке, куда указывает вектор https://habrastorage.org/files/c61/32b/715/c6132b7158674a1ab80f546515af5878.png в пространстве признаков [4]. Это вводится специально, чтобы уменьшить количество вычислений. https://habrastorage.org/files/d88/e63/9d5/d88e639d57cd4886b6f75dad55258876.png — объем ***d***-мерной сферы с радиусом h, Можно отдельно задавать радиус для пространственных координат и отдельно радиус в пространстве цветов. https://habrastorage.org/files/856/fa7/67e/856fa767ebb64997b210f6e2c645a869.png — число пикселей, попавших в сферу. Величину https://habrastorage.org/files/5ab/db9/16a/5abdb916a58a4b96baaf2ac398fb680d.png можно рассматривать как оценку значения https://habrastorage.org/files/451/946/6bf/4519466bfe504e92aa4bb5ee3a90d304.png в области https://habrastorage.org/files/182/cab/e24/182cabe2481f4a20b14947a07b414624.png.  
  
  
Поэтому, чтобы шагать по градиенту, достаточно вычислить значение https://habrastorage.org/files/6dd/c9f/a1c/6ddc9fa1c02b4cd3b160e6cf94f6f155.png — вектора среднего сдвига. Следует помнить, что при выборе другого ядра вектор среднего сдвига будет выглядеть иначе.

При выборе в качестве признаков координат пикселей и интенсивностей по цветам в один сегмент будут объединяться пиксели с близкими цветами и расположенные недалеко друг от друга. Соответственно, если выбрать другой вектор признаков, то объединение пикселей в сегменты уже будет идти по нему. Например, если убрать из признаков координаты, то небо и озеро будут считаться одним сегментом, так как пиксели этих объектов в пространстве признаков попали бы в один локальный максимум.  
  
Если объект, который хотим выделить, состоит из областей, сильно различающихся по цвету, то **MeanShift** не сможет объединить эти регионы в один, и наш объект будет состоять из нескольких сегментов. Но зато хорошо справиться с однородным по цвету предметом на пестром фоне. Ещё **MeanShift** используют при реализации алгоритма слежения за движущимися объектами [5].

Пример кода для запуска алгоритма:

Mat image = imread("strawberry.jpg", CV\_LOAD\_IMAGE\_COLOR);

Mat imageSegment;

int spatialRadius = 35;

int colorRadius = 60;

int pyramidLevels = 3;

pyrMeanShiftFiltering(image, imageSegment, spatialRadius, colorRadius, pyramidLevels);

imshow("MeanShift", imageSegment);

waitKey(0);

Результат:  
  
*Рис. 8. Исходное изображение*  
  
  
*Рис. 9. После сегментации алгоритмом****MeanShift***