**Среднее фоновое изображение**

Вычислить среднее фоновое изображение путем осреднения 100 (уточняется при разработке) последних кадров с учетом их взаимных смещений. Совмещение выполнить с субпиксельной точностью.

Средний фон на текущем кадре получается из текущего кадра, на который с соответствующим смещением накладываем построенный по предыдущим кадрам средний фон. Смысл в том, что средний фон всегда смещается так, чтобы максимально соответствовать текущему кадру, а значит его надо постоянно двигать вслед за кадром.

Пусть B(i) - фон, накопленный к i-тому кадру с учетом его самого; F(i) - i-тый кадр;  
dr(i-1) - смещение F(i) кадра относительно F(i-1);  
M[F,dr] - оператор смещения кадра F на dr, то есть M[F(i-1),dr(i-1)] - есть приблизительно F(i).

Если написано С := M[F,dr] (:= - это оператор присваивания, как в PASCAL), то в C меняются только те пикселы, которые имеются в смещенном изображении, а в "черной зоне" остаются старые значения C.

Тогда алгоритм таков:

1. С := F(i)
2. С := M[B(i-1),dr(i-1)]
3. B(i) := (1-alf) \* C + alf \* F(i)

alf - коэффициент обновления, в нашем случае ~ 0.01.

В случае, если кадр не обновляется, а просто сдвигается, мы выполняем следующий алгоритм?

1. С := F(i)
2. С := M[B(i-1),dr(i-1)]
3. B(i) := C

**Реализовал алгоритм работы следующим образом.**

1. С := F(i)
2. С := M[B(i-1),dr(i-1)]
3. F(i) := Q[F(i), C]
4. B(i) := (1-alf) \* C + alf \* F(i)

Где Q[F, C] - оператор наложения изображений по следующему правилу:  
Q[F, C] = F, в области, где изменение НЕ СИЛЬНОЕ  
Q[F, C] = C, в области, где изменение СИЛЬНОЕ

**Субпиксельное смещение**

D(x+dx,y+dy) = S(x,y) \* (1-dx)\*(1-dy) + S(x+1,y) \*dx \* (1-dy) + S(x,y+1) \* (1-dx) \* dy + S(x+1,y+1) \* dx \* dy  
Здесь 0 <= dx <= 1 и 0 <= dy < 1 - нецелое смещение.

Эта формула выполняет субпиксельный сдвиг S и помещает результат в D.

**Изображение среднего абсолютного отклонения.**

Вычислить среднее отклонение по абсолютной величине от фонового изображения по тому же числу кадров в каждой точке (т.е. построить изображение средних абсолютных отклонений). Отобразить на экране после масштабирования по яркости.

Изображение среднего абсолютного отклонения должно двигаться от кадра к кадру так же, как и среднее фоновое изображение, поскольку они вместе образуют модель фона (т.е. его надо тоже двигать вслед за кадром, причем с субпиксельной точностью).

Изображение среднего абсолютного отклонения не должно обновляться там, где есть объект, поскольку является характеристикой средней изменчивости фона (т.е. его надо также обновлять по маске).

Модифицировать п.1, обновляя только те участки среднего фонового изображения, для которых на текущем кадре отклонение от среднего фонового изображения не превышает среднего отколнения по абсолютной величине, вычисленного в п.2, умноженного на некоторую константу (не обновлять фон участками изображения, где предположительно находится объект). Константа подбирается при разработке. Разумно взять где-то 2.

1. Смещаем среднее фоновое изображение и изображение среднего абсолютного отклонения
2. Вычисляем маску областей, где изменения сильные (движущийся объект)
3. Вычисляем среднее фоновое изображение  
   /\* Тут модель фона изменяется, и маска, вычисленная ранее, уже не столь хорошо взаимодействует с моделью фона, т.к. среднее фоновое изображение было вычислено после нахождения маски \*/
4. Вычисляем изображение среднего абсолютного отклонения (по старому среднему фону, а не по измененному)

Да, осталось поставить порог на вывод окончательного результата. Строится он абсолютно точно также, как и маска, только множитель надо другой поставить, видимо побольше.

**Фильтрация шумов**

перед выводом итогового (в 4 окне) изображения на экран прогонять его через глобальный фильтр, отсекая все точки, которые меньше определенного порога. Делать это надо, конечно, до бинаризации. Порог считаем по глобальной гистограмме \_histogram изображения так:

int DiffImageFilter::getBackgroundBound()

{

int startInd = 1;

if(\_histogram[0] <= \_histogram[1])

{

for(startInd = 1; startInd < 256; startInd++)

{

if(\_histogram[startInd+1] < \_histogram[startInd])

{

break;

}

}

}

int endInd = 1;

for(endInd = startInd; endInd < 256; endInd++)

{

if(\_histogram[endInd+1] >= \_histogram[endInd])

{

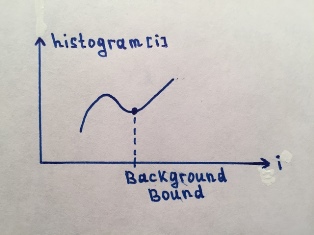
break;

}

}

return endInd;

}



**По какому изображению необходимо строить гистограмму?**

По тому, которое получается после вычитания из текущего кадра среднего фона.

То есть алгоритм работы получается следующим:

1. Нахожу изображение среднего абсолютного отклонения текущего кадра:  
   currentDeviationImg = abs(currentFrame - averageBackImg);
2. Нахожу маску неподвижной части изображения:  
   frameStaticPartMask = movingTargetFactor \* deviationImg - currentDeviationImg;
3. Нахожу порог яркостей фона (backgroundBound) по гистограмме, построенной на основе currentDeviationImg.
4. Пиксели currentDeviationImg, значения которых меньше порога backgroundBound обозначаю в маске frameStaticPartMask, как фон.

**Сегментация.**

Найти связные области, объединить те, которые близко лежат, окружить то, что получится, прямоугольниками.