

# Динамическая сегментация последовательности кадров\*

*Хашин С. И.*

khash2@mail.ru, <http://math.ivanovo.ac.ru/dalgebra/Khashin/>

<sup>1</sup>Ивановский Государственный Университет

Описывается алгоритм сегментации, основанный не на одном кадре, а на паре соседних кадров из видеопоследовательности. По сравнению с обычной, статической сегментацией каждого кадра по-отдельности, качество значительно повышается. При сохранении той же погрешности, количество сегментов удается сократить в несколько десятков раз. Типичное количество сегментов, нужных для получения приемлемой погрешности уменьшается с 200 – 500 до 5 – 15.

## Dynamic segmentation of frames sequences\*

*Khashin S. I.*<sup>1</sup>

<sup>1</sup>Ivanovo State University

Describes the image segmentation algorithm, based not on a single frame, but on the pair of frames from a video sequence. Compared with usual, static segmentation of each frame individually, the quality improves drastically. While maintaining the same error, the number of segments can be reduced in tens times. Typical number of segments needed to produce an acceptable error is reduced from 200 – 500 to 5 – 15.

### 1 Введение

2 В основе алгоритмов сжатия видеоинформации лежит идея построения прогноза те-  
3 кущего кадра на основе предыдущего (или предыдущих). Во всех применяемых сегодня  
4 алгоритмах [2, 3, 4] для построения прогноза текущего кадра на основе предыдущего, он  
5 разбивается на квадраты или прямоугольники небольшого размера (от 32\*32 до 4\*4) и  
6 для каждого из них находится *вектор движения*, с помощью этих векторов строится про-  
7 гноз текущего кадра, находится *дифференциальный кадр*, *displaced frame difference (DFD)*,  
8 который тем или иным образом кодируется. Эти методы позволяют в несколько раз уве-  
9 личить коэффициент сжатия без потери качества по сравнению со сжатием отдельных  
10 статических изображений.

11 Недостатком такого подхода является то, что движущиеся объекты плохо аппрокси-  
12 мируются прямоугольниками и для получения приемлемого результата приходится раз-  
13 бивать кадр на очень мелкие блоки.

14 В работах [8, 9, 10] предложен новый, объектно-ориентированный алгоритм сжатия.  
15 Его основа – сегментация предыдущего кадра, то есть разбиение его не на прямоугольники,  
16 а на сегменты произвольной формы.

17 Различных алгоритмов сегментации существует довольно много, см., например, [7, 11].  
18 Но экспериментальная проверка показывает, что с нашей точки зрения различия меж-  
19 ду ними не так велики: погрешность сегментации (Определение 9) различается на 10-15  
20 процентов. Вряд ли можно ожидать появления новых методов сегментации, существенно  
21 уменьшающих эту погрешность.

22 Для наших целей алгоритм сегментации должен быть управляемым, то есть должна  
 23 быть возможность строить сегментацию с заранее заданным (ориентировочным) количе-  
 24 ством сегментов. Практика показала, что для кадров размера  $720 \times 480$  для построения  
 25 кадра-прогноза с той же погрешностью, какую дает стандарт *h264* при  $PSNR = 40$  обыч-  
 26 но требуется сегментация на 100..500 сегментов. Причем, это количество слабо зависит  
 27 от размера кадра. При переходе от размера  $320 \times 240$  к  $1920 \times 1080$  требуемое количество  
 28 сегментов увеличивается лишь в 2 – 4 раза.

29 Оказывается алгоритм можно существенно улучшить, если сегментировать предыду-  
 30 щий кадр не сам по себе, а на основе целой цепочки из нескольких предыдущих кадров.  
 31 В простейшем случае можно сегментировать пару предыдущих кадров. В этом случае  
 32 многие визуально различные сегменты обладающие одинаковым (аффинным) движе-  
 33 нием будут объединены в один сегмент. В результате, для получения сегментации того же  
 34 качества оказывается достаточным строить не несколько сотен сегментов, а не более од-  
 35 ного десятка. Отметим, что *качество* сегментации в нашем случае – это не субъективная  
 36 оценка, а вполне конкретное число (см. Определение 9).

37 Алгоритму нахождения таких сегментаций и посвящена настоящая статья.

## 38 Основные определения

39 В настоящей работы мы будем рассматривать прямоугольные полноцветные  
 40 (TrueColor) изображения.

41 **Определение 1.** Изображением (кадром) размера  $tx \times ty$  будем называть набор из  
 42 трех матриц (RGB) размера  $tx \times ty$ . Обычно, но не обязательно, элементы этих матриц  
 43 являются целыми числами из отрезка  $[0..255]$ . Эти три матрицы мы будем рассматри-  
 44 вать как одно отображение  $F : U \rightarrow \mathbb{R}^3$ , где  $U$  – подмножество в  $\mathbb{Z}^2$  состоящее из точек  
 45  $(x, y) : 0 \leq x < tx, 0 \leq y < ty$ . Отображение  $F$  естественным образом продолжается  
 46 до отображения  $\mathbb{Z}^2 \rightarrow \mathbb{R}^3$ . Кроме того, используя некоторую интерполяционную формулу  
 47 (билинейную, бикубическую, сплайны, и т.д.) функцию  $F$  можно продолжить до непре-  
 48 рывной кусочно-полиномиальной функции  $\mathbb{R}^2 \rightarrow \mathbb{R}^3$ , которую мы также будем обозначать  
 49 буквой  $F$ .

50 **Определение 2.** Сегментацией  $U = \cup U_i$  изображения будем называть разбиение обла-  
 51 сти  $U$  на произвольные части  $U_1, \dots, U_N$ , сегменты.

52 **Определение 3.** Размером  $k(U_i)$  сегмента  $U_i$  будем называть количество точек в нем.

53 **Определение 4.** Точку  $(x, y) \in U$  будем называть граничной для данной сегментации,  
 54 если она принадлежит одному сегменту, а одна из ее четырех соседних точек – другому.

55 **Определение 5.** Два сегмента называются соседними, если в первом сегменте суще-  
 56 ствует точка, одна из соседних к которой принадлежит второму сегменту.

57 **Определение 6.** Аффинной сегментацией  $S = \{U_i, A_i, i = 1, \dots, N\}$  будем называть  
 58 обычную сегментацию  $U = \cup U_i$  снабженную набором аффинных преобразований  $A_i : U_i \rightarrow \mathbb{R}^2$ :

$$60 \quad A_i : \begin{pmatrix} x \\ y \end{pmatrix} \rightarrow \begin{pmatrix} a_0 + a_1x + a_2y \\ a_3 + a_4x + a_5y \end{pmatrix} \quad (1)$$

61 Для каждой целочисленной точки  $(x, y)$  из области  $U$  обозначим через  $S^*(x, y) \in \mathbb{R}^2$   
 62 точку  $A_i(x, y)$ , где  $i$  – номер сегмента, к которому принадлежит точка  $(x, y)$ .

63 **Определение 7.** Пусть  $(F_1, F_2)$  – два изображения одинакового размера и  $A$  – аффинное  
 64 преобразование плоскости. Через  $A^*(F_2)$  обозначим функцию, принимающую в точке  $v$   
 65 значение  $F_2(S^*(v))$  (напомним, что функцию  $F_2$  можно считать продолженной на всю  
 66 плоскость  $\mathbb{R}^2$ ). Для каждой точке  $v \in U$  длину вектора  $|F_1(v) - A^*(F_2)(v)|$  будем называть  
 67 погрешностью преобразования  $A$  для пары кадров  $(F_1, F_2)$ .

68 **Определение 8.** Пусть  $S$  – аффинная сегментация области  $U$ . Тогда для произвольно-  
 69 го изображения  $F$  через  $S^*(F)$  обозначим новое изображение, строящееся по формулам  
 70  $S^*(F)(x, y) = F(S^*(x, y))$

71 В работе [9] предложен некоторый метод численной оценки качества сегментации. В  
 72 случае, когда нас интересуют не отдельные изображения, а видеопоследовательности, це-  
 73 почки изображений, мы можем предложить гораздо более естественное в данной ситуации  
 74 понятие качества сегментации.

75 **Определение 9.** Пусть  $(F_1, F_2)$  – два изображения одинакового размера и  $S$  – аффинная  
 76 сегментация области  $U$ . Изображение  $F_1 - S^*(F_2)$  будем называть разностным изображе-  
 77 нием. Среднеквадратичное значение длины вектора  $|F_1 - S^*(F_2)|$  по всем точкам области  
 78  $U$  будем называть погрешностью сегментации.

79 Всюду в дальнейшем мы будем предполагать фиксированной пару кадров  $(F_1, F_2)$  оди-  
 80 накового размера  $tx \times ty$ .

## 81 Алгоритм “А”

82 Обычно при обработке видеоданных [2, 3, 4, 5] используются те или иные методы  
 83 нахождения движения. Все они являются различными вариантами одного и того же алго-  
 84 ритма Лукаса-Канады [6, 1] и находят движения лишь в виде сдвигов. Эти методы очень  
 85 эффективны, но лишь для случая малых областей, сегментов. Во всех реализованных на  
 86 сегодняшний день способах видеообработки это так и есть. Но в случае больших сегмен-  
 87 тов, занимающих значительную долю всего кадра мы должны искать движения более  
 88 общего вида, например, аффинных преобразований вида (1).

89 В работе [10] приведен алгоритм поиска таких преобразований, минимизирующих по-  
 90 грешность для данной пары кадров при фиксированной сегментации. Этот алгоритм за-  
 91 ключается в следующем. Для каждого сегмента  $U_i$  рассмотрим аффинное преобразование  
 92  $A_i$  вида (1). Для него на сегменте  $U_i$  вычисляем величину

$$93 \quad S = \sum_{(x,y) \in U_i} |F_1(x, y) - A_i^*(F_2)(x, y)|^2$$

94 и находим значения  $(a_0, \dots, a_5)$  при которых  $S$  достигаем минимума. Описанный алгоритм  
 95 является естественным обобщением стандартного алгоритма Лукаса-Канады на случай  
 96 аффинных преобразований и практически так же эффективен и надежен. Разумеется,  
 97 так как нам надо определить не две координаты вектора сдвига, а все шесть параметров  
 98 аффинного преобразования, он будет работать медленнее обычного алгоритма Лукаса-  
 99 Канады. Однако, это будет замедление в разы, а не на порядки. Будем называть этот  
 100 процесс алгоритмом “А”.

101 В задаче, исследуемой в настоящей работе мы имеем возможность менять не только  
 102 аффинные преобразования, но и саму сегментацию.

## Клеточный автомат

Сегменты, получающиеся в процессе построения иногда могут иметь очень хаотичную структуру. Для из сглаживания будем использовать клеточный автомат. При его описании “соседними” для данной целочисленной точки считаются ее восемь соседей. Таким образом, у угловой точки области  $U$  три соседних, у точки на границе, кроме угловых – 5 соседних, у внутренних точек – по 8 соседей.

Один шаг работы клеточного автомата заключается в следующем. Перебираем все точки области  $U$  в псевдослучайном порядке. Если очередная точка принадлежит сегменту  $i$ , а более половины ее соседей – к сегменту  $j \neq i$ , то относим эту точку к сегменту  $j$ , в противном случае – ничего не делаем. Таким образом, за один шаг мы перебираем все точки кадра в псевдослучайном порядке.

Работа алгоритма зависит от выбранного способа псевдослучайного обхода всех  $m \times n$  точек области  $U$ . Метод рандомизации, применяемый в нашей работе основан на выборе простого числа  $p$  несколько большего, чем  $m \times n$  и первообразного корня  $a$  по модулю  $p$ . Как известно, в этом случае последовательность  $1, a, a^2, \dots, a^{p-1}$  является перестановкой чисел  $1, \dots, p-1$ .

Применением клеточного автомата к данной сегментации будем называть последовательного выполнения описанных выше шагов клеточного автомата. Работа алгоритма прекращается, если на очередном шаге сегментация не изменилась.

Хорошо известно, что описанный выше клеточный автомат закончит свою работу за конечное количество шагов.

Практика показывает, что описанный алгоритм дает очень хорошие результаты по сглаживанию сегментов.

## Разделение сегмента

В этом разделе используется стандартный алгоритм “сглаживания” матрицы. Опишем его более подробно.

**Определение 10.** Сглаживанием матрицы  $R$  с радиусом  $k$  назовем ее свертку с ядром  $Q(x, y) = q(|x|) \cdot q(|y|)$ , где

$$q(x) = \begin{cases} (k+1-x)/(k+1)^2 & x \leq k+1 \\ 0 & x > k+1 \end{cases},$$

то есть

$$R'(x, y) = \sum_{dx, dy=-k}^k Q(dx, dy) \cdot R(x+dx, y+dy)$$

В частности, при сглаживании с радиусом 0 мы получаем исходную матрицу, при сглаживании с радиусом 1 – свертку с ядром:

$$\frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

Можно использовать и более сложные методы сглаживания, например гауссова [7, 11], но это вряд ли существенно изменит результаты.

Пусть  $(F_1, F_2)$  – два изображения одинакового размера и  $U_1$  – некоторый сегмент на  $U$ . Выделим из него подсегмент  $U_2$  по следующему алгоритму. Вначале зафиксируем некоторую величину ожидаемого шума  $e$ , для типичных последовательностей кадров удобно будет взять  $e = 2$ .

- 143 1. Строим матрицу  $R$ , принимающую значения 1 в точках сегмента и 0 в остальных  
144 точках.
- 145 2. Сглаживаем матрицу  $R$  с радиусом  $(mx + my)/10$ .
- 146 3. Находим точку  $(x_S, y_S)$  в которой эта сглаженная матрица принимает наибольшее зна-  
147 чение. Если таких точек несколько, берем первую из них в лексикографическом по-  
148 рядке.
- 149 4. Обходим по спирали точки вокруг  $(x_S, y_S)$ . Точки, принадлежащие сегменту  $U_1$  добав-  
150 ляем в сегмент  $U_2$  до тех пор, пока не получим 100 точек.
- 151 5. С помощью алгоритма “А” находим аффинное преобразование  $A_2$ , оптимальное для  
152 сегмента  $U_2$ . Величину погрешности преобразования  $A_2$  на сегменте  $U_2$  обозначим  $e_2$ .  
153 Если  $e_2 < e$ , положим  $e_2 = e$ .
- 154 6. Находим  $E$  – матрицу погрешностей преобразования  $A_2$  на всей области  $U$ .
- 155 7. Сглаживаем матрицу  $E$  с радиусом 10.
- 156 8. В сегмент  $U_2$  теперь отбираем те точки из исходного сегмента  $U_1$ , для которых величина  
157 сглаженной погрешности не превышает  $e_2$ .
- 158 9. Сглаживаем построенную область с помощью клеточного автомата.
- 159 10. Повторяем шаги (5, 6, 7, 8, 9) до тех пор, пока сегмент  $U_2$  не стабилизируется. Числен-  
160 ные эксперименты показывают, что требуемое количество итераций от 3 до 6, причем  
161 более четырех повторений требуется крайне редко.

## 162 Улучшение сегментции

163 Пусть  $(F_1, F_2)$  – два изображения одинакового размера и  $S = \{U_i, A_i, i = 1, \dots, N\}$  –  
164 аффинная сегментация области  $U$ . Для ее улучшения применяем следующий алгоритм.

- 165 1. Для всех аффинных преобразований  $A_i$  строим  $E_i$  – матрицы погрешностей преобра-  
166 зований  $A_i$  на всей области  $U$ .
- 167 2. Сглаживаем матрицы  $E_i$  с радиусом 10.
- 168 3. В сегмент  $U_i$  отбираем те точки из  $U$ , для которых величина сглаженной погрешности  
169 наименьшая среди всех  $E_i$ .
- 170 4. Сглаживаем построенные сегменты с помощью клеточного автомата.
- 171 5. Удаляем пустые сегменты.
- 172 6. С помощью алгоритма “А” находим оптимальные аффинные преобразования для каж-  
173 дого вновь полученного сегмента  $U_i$ .
- 174 7. Повторяем шаги (1 – 6) до тех пор, пока сегменты  $U_i$  почти не стабилизируются. Прак-  
175 тика показывает, что четырех повторений практически всегда достаточно.

## 176 Динамическая сегментация

177 Пусть дана пара последовательных кадров  $(F_1, F_2)$  и пусть мы хотим разбить второй  
178 кадр  $F_2$  на  $N$  сегментов.

179 Алгоритм динамической сегментации состоит из следующих шагов.

- 180 1. Начинаем с одного сегмента  $U_0$ , состоящего из всей области  $U$  и тождественного аф-  
181 финного преобразования  $A_0$ .
- 182 2. На каждом шаге выделяем из  $U_0$  один сегмент с помощью алгоритма, описанного выше,  
183 до тех пор, пока не получим требуемое количество сегментов.
- 184 3. Улучшаем полученную сегментацию.

## Полученные результаты

В нашей ситуации для оценки качества получающейся сегментации имеется уже готовый числовой параметр – погрешность сегментации (см. Определение 9).

### Один сегмент, сдвиг

Для проверки работоспособности алгоритма берем следующую пару кадров:  $f1$  – левые 500 столбцов стандартного изображения `lena.bmp` размера  $512 \times 512$ ,  $f2$  – правые 500 столбцов. Таким образом,  $f2$  получается из  $f1$  в сдвиге на 12 точек по горизонтали. В результате применения описанного алгоритма получаются два сегмента. Один содержит почти весь кадр, точнее говоря 99.66% от площади кадра с аффинным преобразованием

$$\begin{aligned} x' &= 11.998994 + 1.000002x + .000003y \\ y' &= 0.000539 - 0.000001x + .999998y \end{aligned}$$

Фактически, оно совпадает со сдвигом на 12 точек влево. Второй сегмент появляется только из-за краевых эффектов. Общая погрешность преобразования оказывается равной 2.7. На примере этой, практически идеальной ситуации мы видим, каких показателей стоит ожидать от “хорошей” аффинной сегментации.

### Несколько сегментов со спрайтами

Здесь мы опять строим пару искусственных кадров. Каждый кадр размера  $640 \times 480$  состоит из фона и четырех небольших спрайтов (эллипсы с полуосями 50..70). И для фона, и для каждого спрайта задано свое аффинное преобразование.

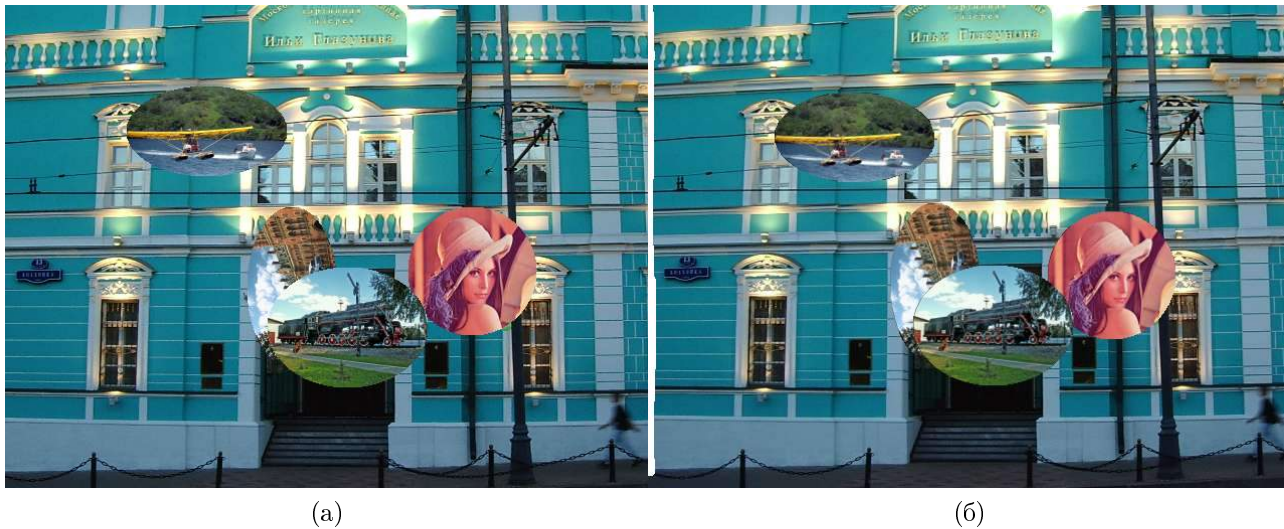


Рис. 1. Два кадра со спрайтами

Алгоритм построил 8 сегментов, на рисунке 2(а) разными цветами показаны разные сегменты, (б) – прорисованы их границы на исходном изображении. Погрешность сегментации оказывается равной 8.6. Более подробное изучение дифференциального кадра показывает, что, как и следовало ожидать, наибольшая погрешность – в областях, которым нет соответствия на втором кадре, “uncovered background“. Это случай тоже можно рассматривать как близкий к идеальному, все сегменты и аффинные преобразования оказались найдены верно.

### Реальные видеопоследовательности

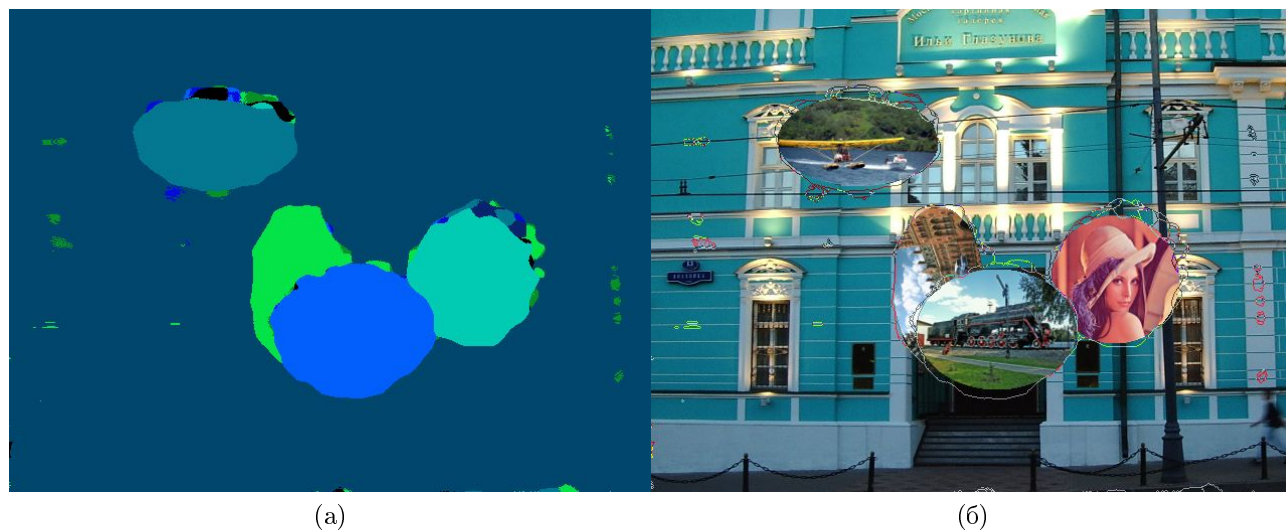


Рис. 2. Сегментация, 8 сегментов

211 Описанные алгоритмы были применены к нескольким реальным видеопоследователь-  
212 ностям. Среди них стандартная видеопоследовательность *Foreman* размерности  $352 \times 288$   
213 и мультфильм *Factory* размерности  $1920 \times 1080$ .

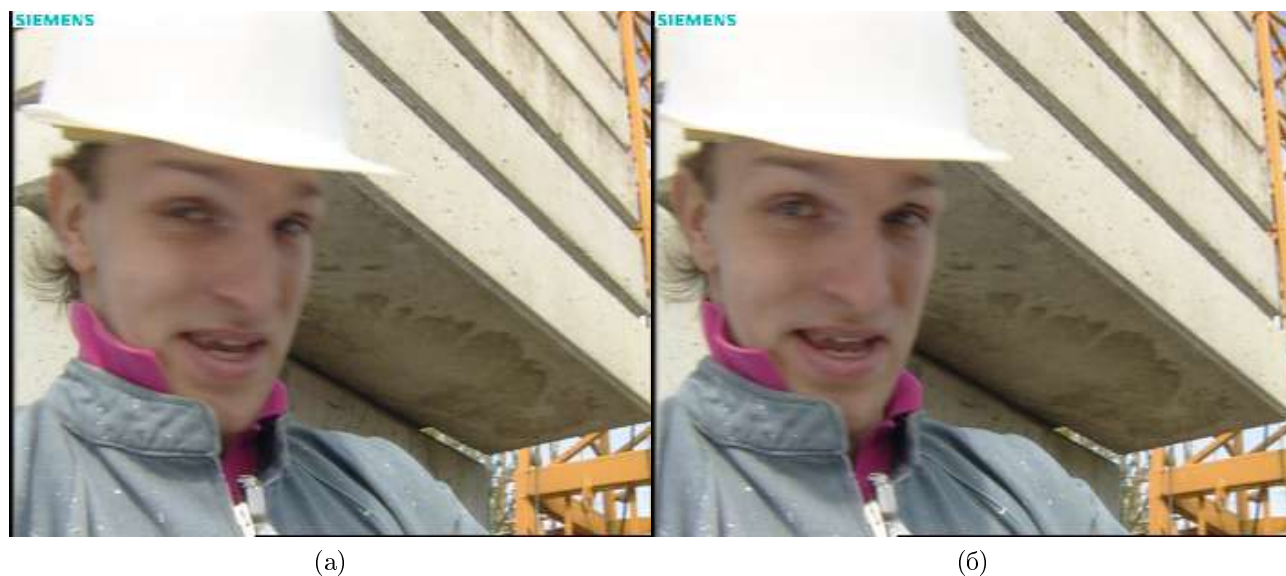


Рис. 3. Исходные кадры, номера (172, 173)

214 В последовательности *Foreman* получается от 2 до 11 сегментов (на рисунке 4, кадр  
215 номер 173, – 10 сегментов. Слева разные сегменты показаны разным цветом, справа –  
216 на исходном изображении прорисованы границы сегментов), погрешность сегментации –  
217 от 5.6 до 9.8. Видно, что сегментация связана именно с различными движениями частей  
218 кадра. Например, если глаза двигаются синхронно со всем лицом, то они не выделяются  
219 в отдельные сегменты.

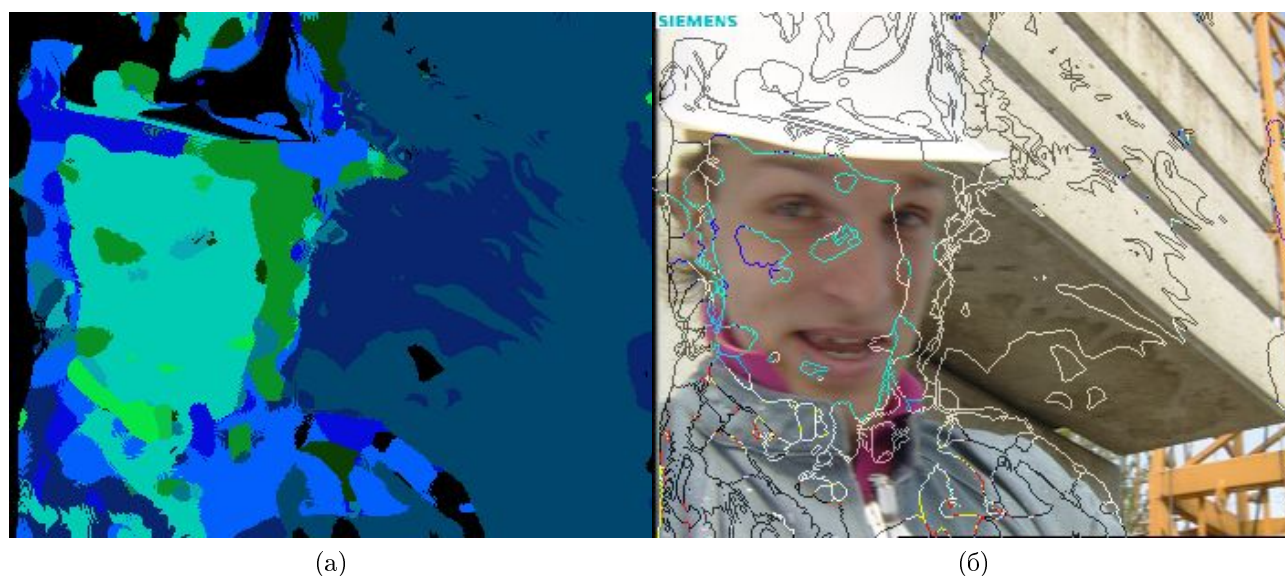


Рис. 4. Сегментация на 10 сегментов

220 В последовательности **Factory** получается от 2 до 15 сегментов, погрешность сегмен-  
 221 тации – от 2.6 до 11.2. То есть при огромном увеличении площади кадра, количество  
 222 сегментов увеличивается незначительно.

## 223 Заключение

- 224 1. Предложенный метод дает весьма эффективный алгоритм сегментации: по сравнению  
 225 со статической сегментацией, при той же погрешности количество сегментов можно  
 226 уменьшить в несколько десятков раз. Здесь требуется более детальное сравнение, за-  
 227 планированное на ближайшее время.
- 228 2. Алгоритм надежен: визуальное изучение получающихся сегментов показывает, что нет  
 229 практически ни одной существенной ошибки.
- 230 3. Количество сегментов меняется в пределах от 2 до 20, причем оно мало меняется с  
 231 ростом размера кадра.
- 232 4. Погрешность сегментации – от 1 до 14, причем она не изменяется с ростом размера  
 233 кадра.
- 234 5. В реальных кадрах основной составляющей погрешности является не ошибки сегмен-  
 235 тации и нахождения движения, а “uncovered background“ и цветовое различие между  
 236 кадрами, то есть причины, которые невозможно устранить выбором сегментации и  
 237 аффинных преобразований.

## 238 Литература

- 239 [1] *S. Baker, R. Gross, I. Matthews* Lucas-Kanade 20 Years On: A Unifying Framework // *Int.J.of*  
 240 *Computer Vision*, 2002,— Vol. 56, Pp. 111–122.
- 241 [2] *ITU-T and ISO/IEC JTC 1* Generic coding of moving pictures and associated audio information.  
 242 Part 2: Video // *ITU-T Recommendation H.262 – ISO/IEC 13818-2 (MPEG-2)*, Nov. 1994.
- 243 [3] *ITU-T* Video coding for low bit rate communication // *ITU-T Recommendation H.263; version 1*,  
 244 *Nov. 1995; version 2, Jan. 1998; version 3, Nov. 2000.*
- 245 [4] *ITU-T Rec. H.264 / ISO/IEC 11496-10*. Advanced Video Coding // *Final Committee Draft*,  
 246 *Document JVT-E022*, September 2002.



- 247 [5] Draft ITU-T Recommendation and Final Draft International Standard of Joint Video  
248 Specification // (ITU-T Rec. H.264/ISO/IEC 14496-10 AVC) Joint Video Team (JVT), Mar. 2003,  
249 Doc. JVT-G050.
- 250 [6] *B. D. Lucas, T. Kanade* An iterative image registration technique with an application to stereo  
251 vision // Proc. of Imaging Understanding Workshop. 1981. — Pp. 121–130
- 252 [7] *Гонсалес Р., Вудс Р.* Цифровая обработка изображений // М., Техносфера, 2006, 1072 с.
- 253 [8] *Хашин С. И.* Применение методов распознавания образов для сжатия видеoinформации //  
254 Докл. всеросс. конф. ММРО-13. — М.: МАКС Пресс, 2007. — С. 420-424.
- 255 [9] *Хашин С. И.* Оценка качества сегментации изображения // Вестник ИвГУ. — Иваново, 2010,  
256 вып. 2. — С. 112-118.
- 257 [10] *Хашин С. И.* Аффинная версия алгоритма Лукаса-Канады // Докл. всеросс. конф. ММРО-  
258 13. — М.: МАКС Пресс, 2011. — С. 459-462.
- 259 [11] *Яне Б.* Цифровая обработка изображений // М., Техносфера, 2007, 583 с.