

Динамическая сегментация последовательности кадров

16 Конференция «Математические методы
распознавания образов»
Казань 6-12 октября 2013 г.

Хашин С.И.

<http://math.ivanovo.ac.ru/dalgebra/Khashin/index.html>

ИвГУ, Иваново
25 сентября 2013 г.

Введение

В основе алгоритмов сжатия видеoinформации лежит идея построения прогноза текущего кадра на основе предыдущего (или предыдущих). Во всех применяемых сегодня алгоритмах (вплоть до H264) для построения прогноза текущего кадра на основе предыдущего, он разбивается на квадраты или прямоугольники небольшого размера (от 32×32 до 4×4) и для каждого из них находится *вектор движения*, с помощью этих векторов строится прогноз текущего кадра, находится *дифференциальный кадр, displaced frame difference (DFD)*, который тем или иным образом кодируется. Эти методы позволяют в несколько раз увеличить коэффициент сжатия без потери качества по сравнению со сжатием отдельных статических изображений.

В работе

Хашин С. И., Применение методов распознавания образов для сжатия видеоинформации. Докл. всеросс. конф. ММРО-13. — М.: МАКС Пресс, 2007. — С. 420-424.

(и некоторых других) предложен новый, объектно-ориентированный алгоритм сжатия. Его основа – сегментация предыдущего кадра, то есть разбиение его не на прямоугольники, а на сегменты произвольной формы.

Различных алгоритмов сегментации существует довольно много. Но экспериментальная проверка показывает, что с нашей точки зрения различия между ними не так велики: погрешность сегментации различается на 10-15 процентов. Вряд ли можно ожидать появления новых методов сегментации, существенно уменьшающих эту погрешность. Оказывается сегментацию можно существенно улучшить, если сегментировать предыдущий кадр не сам по себе, а на основе целой цепочки из нескольких предыдущих кадров. В этом случае многие визуально различные сегменты обладающие одинаковым (аффинным) движением будут объединены в один сегмент. В результате, для получения сегментации того же качества оказывается достаточным строить не несколько сотен сегментов, а не более одного десятка.

Определение. Изображением (кадром) размера $m_x \times m_y$ будем называть набор из трех матриц (RGB) размера $m_x \times m_y$. Эти три матрицы мы будем рассматривать как одно отображение $F : U \rightarrow \mathbb{R}^3$, где U – подмножество в \mathbb{Z}^2 состоящее из точек $0 \leq x < m_x$, $0 \leq y < m_y$. Отображение F естественным образом продолжается до отображения $\mathbb{Z}^2 \rightarrow \mathbb{R}^3$. Кроме того, используя некоторую интерполяционную формулу функцию F можно продолжить до непрерывной кусочно-полиномиальной функции $\mathbb{R}^2 \rightarrow \mathbb{R}^3$, которую мы также будем обозначать буквой F .

Определение. Сегментацией $U = \cup U_i$ изображения будем называть разбиение области U на произвольные части U_1, \dots, U_N , сегменты.

$A^*(F)$

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

$$A_i : \begin{pmatrix} x \\ y \end{pmatrix} \rightarrow \begin{pmatrix} a_0 + a_1x + a_2y \\ a_3 + a_4x + a_5y \end{pmatrix}$$

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

$$A : U \longrightarrow \mathbb{R}^2 \supset U$$

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

Определение. Пусть S – аффинная сегментация области U . Тогда для произвольного изображения F через $S^*(F)$ обозначим новое изображение, строящееся по формулам $S^*(F)(x, y) = F(S^*(x, y))$

$$\begin{array}{ccccc}
 A^*(F) & & F & & F \\
 & A & & & \\
 U & \longrightarrow & \mathbb{R}^2 & \supset & U
 \end{array}$$

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

Всюду в дальнейшем мы будем предполагать фиксированной пару кадров (F_1, F_2) одинакового размера $m_x \times m_y$.

Обычно при обработке видеоданных используются те или иные методы нахождения движения. Все они являются различными вариантами одного и того же алгоритма Лукаса-Канады и находят движения лишь в виде сдвигов. В нашем случае мы должны искать движения в виде аффинных преобразований. Для каждого сегмента U_i рассмотрим аффинное преобразование A_i вида. Для него на сегменте U_i вычисляем величину

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

и находим значения (a_0, \dots, a_5) при которых S достигаем минимума. Будем называть этот процесс алгоритмом «А».

Идея алгоритма

А) По данной сегментации U_i находим аффинные преобразования A_i . Каждое из них задает свою функцию шума E_i на всей области U .

Б) По набору аффинных преобразований A_i находим сегментацию: каждую точку относим к тому сегменту, для которого шум наименьший.

Клеточный автомат

Для сглаживания сегментов будем использовать клеточный автомат. При его описании “соседними” для данной целочисленной точки считаются ее восемь соседей.

Один шаг: 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 прекращается, если на очередном шаге сегментация не изменилась.

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

1. Для всех афф. преобразований A_i строим E_i – матрицы погрешностей преобразований A_i на всей области U .
2. Сглаживаем матрицы E_i с радиусом 10.
3. В сегмент U_i отбираем точки из U , для которых величина сглаженной погрешности наименьшая среди всех E_i .
4. Сглаживаем построенные сегменты с помощью клеточного автомата.
5. Удаляем пустые сегменты.
6. С помощью алгоритма “А” находим оптимальные аффинные преобразования для каждого вновь полученного сегмента U_i .
7. Повторяем шаги (1 – 6) до тех пор, пока сегменты U_i не стабилизируются.

Полученные результаты, Один сегмент

В нашей ситуации для оценки качества получающейся сегментации имеется уже готовый числовой параметр – погрешность сегментации.

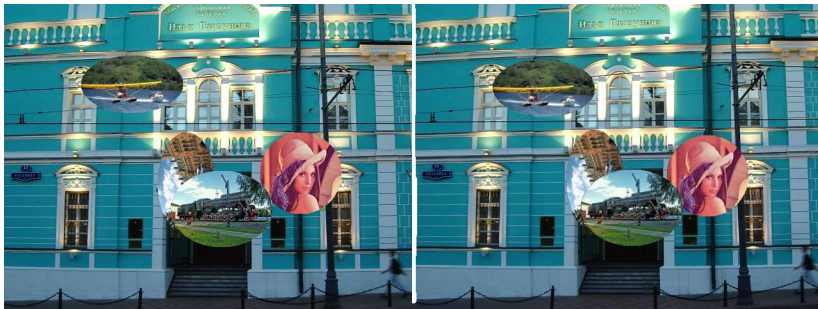
Для проверки берем пару кадров: f_1 – левые 500 столбцов стандартного изображения `lena.bmp` размера 512×512 , f_2 – правые 500 столбцов. В результате получаются два сегмента. Один содержит почти весь кадр с аффинным преобразованием

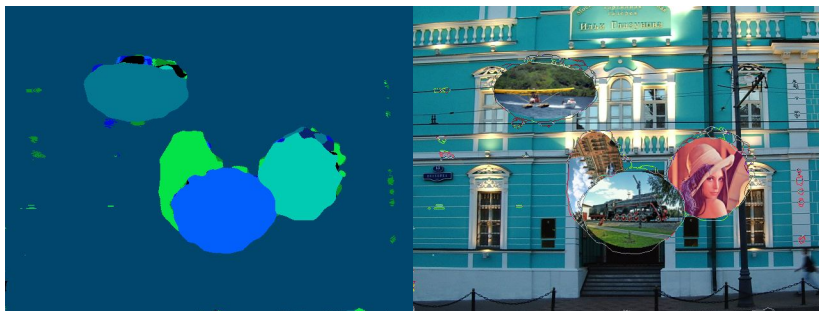
$$\begin{aligned}x' &= 11.998994 + 1.000002x + .000003y \\y' &= 0.000539 - 0.000001x + .999998y\end{aligned}$$

Второй сегмент появляется только из-за краевых эффектов. Общая погрешность равна 2.7. На примере этой, практически идеальной ситуации мы видим, каких показателей стоит ожидать от “хорошей” аффинной сегментации.

Несколько сегментов со спрайтами

Строим пару искусственных кадров. Каждый кадр размера 640×480 состоит из фона и четырех небольших спрайтов (эллипсы с полуосями 50..70). И для фона, и для каждого спрайта задано свое аффинное преобразование.

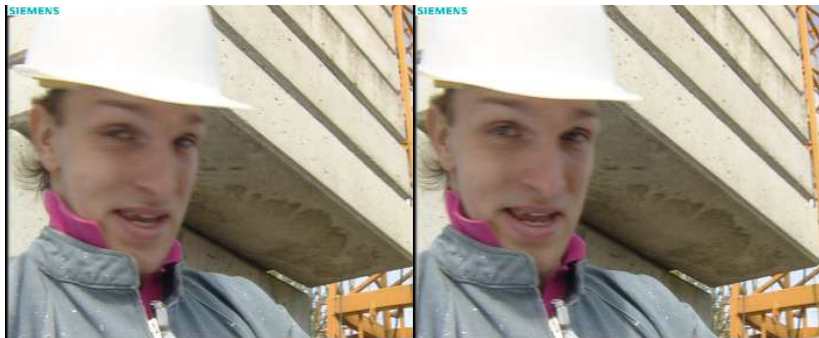


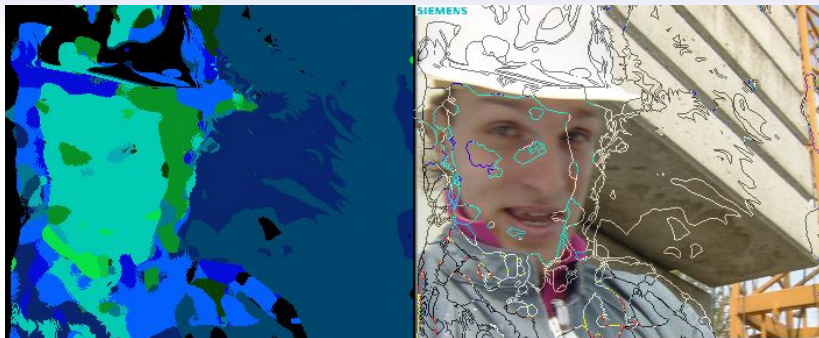


Алгоритм построил 8 сегментов. Погрешность равна 8.6. Наибольшая погрешность – в областях, которым нет соответствия на втором кадре, “uncovered background“. Это случай тоже можно рассматривать как близкий к идеальному, все сегменты и аффинные преобразования оказались найдены верно.

Реальные видеопоследовательности

Foreman размерности 352×288 и мультфильм Factory размерности 1920×1080 .





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

Заклучение

1. Предложенный метод дает весьма эффективный алгоритм сегментации: по сравнению со статической сегментацией, при той же погрешности количество сегментов можно уменьшить в несколько десятков раз.
2. Алгоритм надежен: визуальное изучение получающихся сегментов показывает, что нет практически ни одной существенной ошибки.
3. Количество сегментов меняется в пределах от 2 до 20, причем оно мало меняется с ростом размера кадра.
4. Погрешность сегментации – от 1 до 14, причем она не изменяется с ростом размера кадра.
5. В реальных кадрах основной составляющей погрешности является не ошибки сегментации и нахождения движения, а “uncovered background” и цветовое различие между кадрами, то есть причины, которые невозможно устранить выбором сегментации и аффинных преобразований.

