



по набору эвристических критериев. Далее записи, уже прошедшие обработку, разделяются на группы, которые при необходимости импортируются в базу данных (batch update); если добавление не требуется, то они просто учитываются при формировании отчета.

Некоторые из использовавшихся принципов и эвристик:

- перебор сочетаний значений колонок, отвечающих за один атрибут, с дальнейшей перестановкой;
- отброс некоторых более приоритетных полей при наличии заполненной комбинации нескольких менее приоритетных;
- повторная проверка записей с целью перестройки конечного решения в случае появления новых дубликатов.

Система достаточно хорошо разделяет входные данные на классы (число ошибок близко к нулю), но быстродействие довольно невелико. Тем не менее, в первом приближении задачу можно считать выполненной. В дальнейшем, планируется применить нейронные сети при решении задачи классификации, что, теоретически, может позволить достигнуть большего быстродействия при той же точности классификации.

И.А. Лёзин, С.А. Кирьяков

МОРФИНГ РАСТРОВЫХ ИЗОБРАЖЕНИЙ

(Самарский государственный аэрокосмический университет имени академика С.П. Королёва (национальный исследовательский университет))

Морфинг (англ. *morphing*, трансформация) — технология в компьютерной анимации, визуальный эффект, создающий впечатление плавной трансформации одного объекта в другой. Используется в игровом и телевизионном кино, в телевизионной рекламе. Встречается втрёхмерной и двухмерной (как растровой, так и векторной) графике.

Для создания эффекта используются как минимум два изображения, на которых художник задаёт в зависимости от используемого программного обеспечения опорные фигуры или ключевые точки (т. н. маркеры, или метки), которые помогают компьютеру выполнить правильный морфинг, то есть создать изображения промежуточных состояний (интерполируя имеющиеся данные)[1].

Целью работы было написать программу морфинга человеческого лица, загруженного с фотографии, с мордой лошади взятой из заранее составленной базы.

Основные задачи:

1. Составить алгоритм нахождения опорных точек на человеческом лице;
2. Составить алгоритм построения полигональной сетки по имеющимся точкам;



3. Составить алгоритм трансформации полученных полигонов;
4. Составить алгоритм анимации.

Нахождение опорных точек

Для фильтрации цвета кожи использовалось цветовое пространство YCbCr так как оно лишено некоторых недостатков свойственных пространству RGB а именно из за большой глубины цвета в RGB нельзя использовать обычную пороговую фильтрацию. В YCbCr яркость хранится в компоненте Y, а цветовая составляющая (chroma) (сам цвет) — в компоненте Cb как разница по синему и в компоненте Cr как разница по красному. Цвет кожи в такой кодировке лежит в пределах $Y = [0, 1]$ $Cb = [-0.15, 0.05]$ $Cr = [0.05, 0.20]$.

Используя подобного рода пороговую фильтрацию можно отделить лицо человека от фона фотографии. В результате получится двоичное черно-белое изображение, которое маскирует цвет кожи и используется на следующем этапе[2].

Применив к такому изображению методы Собеля можно получить четкие контуры человеческого лица и основных черт(глаза, нос, рот, брови). Метод Собеля заключается в наложении маски на изображение вида[3]:

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

Построение полигональной сетки

Триангуляцией Делоне позволяет разбить множество точек S на треугольники так что для любого треугольник все точки из S, кроме точек являющиеся его вершинами, лежат вне окружности, описанной вокруг этого треугольника[4].

Трансформация полигонов

Имея полигон из 3 точек и его новые координаты можно составить систему из 6 линейных уравнение, решение которой позволит найти соответствие точек внутри нового полигона точкам старого полигона и установить соответствующий цвет.

$$\begin{cases} k_x^{(1)}x_1 + k_x^{(2)}y_1 + k_x^{(3)} = x'_1 \\ k_x^{(1)}x_2 + k_x^{(2)}y_2 + k_x^{(3)} = x'_2 \\ k_x^{(1)}x_3 + k_x^{(2)}y_3 + k_x^{(3)} = x'_3 \\ k_y^{(1)}x_1 + k_y^{(2)}y_1 + k_y^{(3)} = y'_1 \\ k_y^{(1)}x_2 + k_y^{(2)}y_2 + k_y^{(3)} = y'_2 \\ k_y^{(1)}x_3 + k_y^{(2)}y_3 + k_y^{(3)} = y'_3 \end{cases}$$

Анимация

Имея множество опорных точек на начальном и исходном изображении можно составить для каждой точки ее уравнение движения

$$\begin{aligned} x_1 &= a_x t_0 + b_x \\ x_2 &= a_x t_k + b_x \end{aligned}$$



$$y_1 = a_x t_0 + b_y$$

$$y_2 = a_y t_k + b_y$$

В результате трансформируя каждый полигон в зависимости от времени t можно получить плавную анимацию.

Результатом исследования является программное обеспечение, написанное на языке C#, реализующее морфинг одного изображения в другое по средством установки опорных точек.

Литература

1. Морфинг [Электронный ресурс]. – URL: <http://ru.wikipedia.org/wiki/Морфинг> (дата обращения: 14.05.2014).
2. FaceLight — распознавание лиц в реальном времени [Электронный ресурс]. – URL: <http://blogs.msdn.com/b/rucoding4fun/archive/2010/04/02/facelight-silverlight-4.aspx> (дата обращения: 14.05.2014).
3. Алгоритмы выделения контуров изображений [Электронный ресурс]. – URL: <http://habrahabr.ru/post/114452/> (дата обращения: 14.05.2014).
4. Триангуляция Делоне [Электронный ресурс]. – URL: http://ru.wikipedia.org/wiki/Триангуляция_Делоне (дата обращения: 14.05.2014).

И.В. Лезина, А.Ю. Заворотков

АППРОКСИМАЦИЯ ПЛОТНОСТИ ВЕРОЯТНОСТИ НЕЙРОННОЙ СЕТЬЮ ВОЛЬТЕРРИ

(Самарский государственный аэрокосмический университет)

При анализе и интерпретации большого числа массивов числовых и функциональных характеристик широко применяются аппроксимативные методы, суть которых заключается в нахождении подходящего аналитического выражения, которое бы описывало найденные экспериментальные результаты. Аппроксимативный подход оказывается эффективным и при обработке результатов имитационного моделирования (вычислительного эксперимента) [1].

Для решения задачи аппроксимации плотности вероятности была выбрана нейронная сеть Вольтерри, которая является динамической сетью для нелинейной обработки последовательности сигналов, задержанных относительно друг друга [2]. Возбуждением для сети в момент t служит вектор $x = [x_t, x_{t-1}, \dots, x_{t-L}]^T$, где L - количество единичных задержек, а $(L+1)$ означает длину вектора. В соответствии с определением ряда Вольтерри выходной сигнал генерируется по формуле:

$$y(t) = \sum_{i_1=1}^L w_{i_1} x(t - i_1) + \sum_{i_1=1}^L \sum_{i_2=1}^L w_{i_1 i_2} x(t - i_1) x(t - i_2) + \sum_{i_1=1}^L \dots \sum_{i_k=1}^L w_{i_1 i_2 \dots i_k} x(t - i_1) x(t - i_2) \dots x(t - i_k) \quad (1)$$