Распознавание динамических жестов руки посредством обработки дальностных изображений человека

Ваагн Нагапетян
Факультет физико-математических и естественных наук
Российский университет дружбы народов, Москва, Россия

Аннотация

Рассматривается задача распознавания динамических жестов руки человека для создания человеко-машинного интерфейса, в котором взаимодействие осуществляется посредством естественных жестов руки без прикосновения к каким либо контроллерам и сенсорным экранам. Распознавание позиции и ориентации рук осуществляется посредством обработки каждого кадра видеоряда, полученного от трехмерного сенсора. Идентификация жеста осуществляется посредством сравнения траекторий центра ладони с траекториями эталонных жестов. Для сравнения траекторий жестов используется алгоритм динамической трансформации шкалы времени (Dynamic Time Warping - DTW).

Рассматриваются две системы, реализованные на основе предложенных алгоритмов, позволяющие бесконтактно рисовать на экране персонального компьютера с помощью жестов рук и пальцев, а также распознавать жесты, траектории которых представляют собой геометрические фигуры или цифры.

Ключевые слова: *Распознавание жестов, дальностное изображение, DTW*

1. ВВЕДЕНИЕ

Распознавание жестов рук является довольно актуальной задачей в таких приложениях, как например, автоматизированный сурдоперевод; управление компьютером, роботом или искусственной рукой; естественное взаимодействие с трехмерными компьютерными моделями объектов и т.д. Подходы к решению данной задачи отличаются друг от друга используемой аппаратурой и алгоритмами обработки данных о жесте руки. Например, в работе [1] для распознавания позиции руки и пальцев используется цветная камера. Изображение рук отделается от фона с учетом отличия цвета кожи человека от заднего фона. Полученное изображение сглаживается методом медианной фильтрации, контуры руки выделяются с применением алгоритма поиска контуров связанных компонент, а пальцы выделяются на основе анализа изгибов контура руки. В предложенной системе жесты руки используются для управления видеокамерой посредством взаимодействия с графическим интерфейсом пользователя. В работе [2] для распознавания динамических жестов руки система сначала обучается на моделях 20 жестов, используя скрытую Марковскую модель (СММ), где в качества дескрипторов жеста выступают коэффициенты Фурье. В работе приводится точность распознавания - 90%. Существуют также подходы, основанные на применении перчаток, оснащенных сенсорами [3], анализе дальностных изображений и распознавании позиции руки посредством использования деревьев решений [4].

В настоящей работе в качестве устройства ввода жестов был выбран трехмерный сенсор Asus Xtion Pro Live [5], который снабжен одной RGB камерой, излучателем структурированного инфракрасного света и приемником, который принимает отраженный свет от поверхностей объектов. В результате сенсор возвращает цветное изображение и дальностное изображение с разрешением 640x480 со скоростью 30 кадров в секунду, что вполне приемлемо для создания приложений, работающих в реальном времени.

2. ЗАДАЧА РАСПОЗНАВАНИЯ ДИНАМИЧЕСКИХ ЖЕСТОВ РУКИ

В зависимости от выбора человеко-машинного интерфейса, задачей распознавания жеста руки может быть:

1. Вычисление позиции руки/ладони.
2. Определение ориентации ладони.
3. Идентификация жеста по заданным эталонным образцам.

В системах, требующих непосредственного управления персональным компьютером, роботом или виртуальной кистью с помощью жестов рук, достаточным является вычисление позиции и ориентации руки. При этом должны учитываться такие факторы, как различие форм рук и цвета кожи у разных людей, изменение освещения и возможные изменения заднего фона наблюдаемого человека. Поскольку такие системы требуют мгновенного отклика от графического интерфейса пользователя, время обработки каждого кадра видеоряда не должно превышать 1/24 секунды.

В системах автоматизированного сурдоперевода, задачей распознавания является идентификация показанного жеста по заданным эталонным образцам. В данном случае должны учитываться разные скорости показа жеста и возможные отклонения распознаваемых жестов от эталонных образцов.

Далее рассмотрим основные этапы разработанного алгоритма для решения приведенных задач.

3. ВЫЧИСЛЕНИЕ ПОЗИЦИИ И ОРИЕНТАЦИИ ЛАДОНИ В КАДРЕ ВИДЕОРЯДА

Кадр видеоряда трехмерного сенсора представляет собой дальностное изображение, каждый пиксель которого характеризуется расстоянием до камеры наблюдения. На жестикулирующего человека ставится естественное ограничение – чтобы система рассматривала руку как управляющее устройство, расстояние руки до трехмерного сенсора должно быть не больше фиксированного значения d.

Для каждого кадра видеоряда выполняются следующие действия:

(1) Удаление всех точек, которые не входят в рассматриваемую зону (пороговая обработка);

(2) Поиск связанных компонентов;

(3) Вычисление центров компонентов;

(4) Фильтрация компонентов.

Первый шаг алгоритма осуществляется путем сравнение значения каждой точки кадра с заранее известным значением d. Если значение точки превышает d, то точке присваивается значение ноль. Обозначим полученное после пороговой обработки дальностное изображение буквой S.

Определение 1. В изображении *S* назовем точки (x, y) и (x', y') связанными, если существует последовательность точек (x_0, y_0), (x_1, y_1), …, (x_n, y_n) таких что, $(x_0, y_0) = (x, y)$, $(x_n, y_n) = (x', y')$, точки (x_{i-1}, y_{i-1}) и (x_i, y_i) являются 8-связаннимы соседями и выполняются условия $S(x_{i-1}, y_{i-1}) > 0, S(x_i, y_i) > 0$ для всех значений $i \in \{1,2,...,n\}$.

Определение 2. Связанной компонентой *C* в изображении *S* называется множество точек $C = \{(x, y) : S(x, y) > 0\}$, где любые две точки из *C* являются связанными друг с другом и все связанные точки (x', y') с точкой $(x, y) \in C$ принадлежат множеству *C*.

Поиск связанных компонентов осуществляется в два прохода, посредством алгоритма поиска связанных компонент в графе [6]. Во время первого прохода все точки изображения *S* помечаются временными метками, где метки представляют собой цифровые значения. Параллельно, создается множество эквивалентных меток. Например, на рисунке 1 метки 2 и 3 являются эквивалентными. Во втором проходе все временные метки меняются на метку с минимальным значением из числа эквивалентных меток. Например, на рисунке 1 метка 3 поменяется на метку 2. После второго прохода, множество точек, помоченных эквивалентными метками, будет представлять собой связанную компоненту в изображении S. Например, на рисунке 1 множество точек, помоченных меткой “1” и меткой “2” являются связанными компонентами в приведенном изображении.

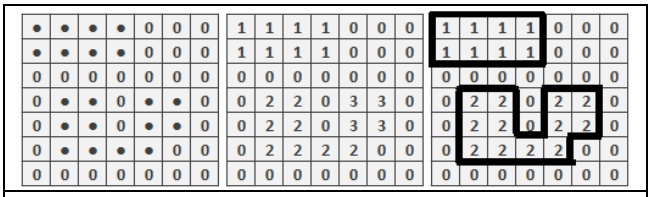

Рисунок 1 − Пример маркировки точек связанных компонент в дальностном изображении. Слева направо показаны соответственно дальностное изображение, где точками обозначены пиксели, значения которых больше нуля; маркировка после первого прохода и маркировка после второго прохода.

Обозначим через $K(x, y)$ метку точки (x, y) в изображении S. Следующим шагом алгоритма является вычисление центров найденных связанных компонентов. Центры вычисляются отдельно для каждого компонента посредством моментов. Пусть C - связанная компонента в изображении S, точки которого были маркированы меткой *c*. Определим моменты первого порядка $M_{0,0}$, $M_{0,1}$, $M_{1,0}$ компонента *C* следующим образом:

$$M_{0,0} = \sum_x \sum_y I(x, y), \qquad M_{0,1} = \sum_x \sum_y y \cdot I(x, y),$$

$$M_{1,0} = \sum_x \sum_y x \cdot I(x, y),$$ где

$$I(x, y) = \begin{cases} 1, \textit{ если } K(x, y) = c \\ 0, \textit{ в противном случае} \end{cases}.$$

Вычислить центр компонента *C* можно следующим образом:

$$(x_c, y_c, z_c) = \left(\frac{M_{1,0}}{M_{0,0}}, \frac{M_{0,1}}{M_{0,0}}, S(x_c, y_c)\right).$$ Ориентация ладони может быть вычислена посредством моментов второго порядка.

Следующим шагом алгоритма является фильтрация найденных связанных компонентов. Из всех компонентов удаляются:

1) компоненты, размер которых слишком мал, чтобы быть изображением ладони человека;
2) компоненты, центры которых не меняют расположения в течение времени. Примечание: для отслеживания местоположения центров рассматривается фиксированное число предыдущих кадров видеоряда.

На основе предложенного алгоритма было разработано программное приложение, позволяющее создавать рисунки на персональном компьютере посредством динамических жестов руки. На рисунке 2 показан интерфейс программы. Объекты разного цвета на рисунке 2(a) - это распознанные связанные компоненты в кадрах видеоряда. Для каждого компонента цвет выбирается случайным образом, но так, чтобы в одном видеокадре два разных компонента не имели одинакового цвета. Распознавание позиции руки в каждом кадре видеоряда осуществляется в течение нескольких миллисекунд, что позволяет рисовать на экране компьютера в реальном времени, без ощущения задержки отклика. Заметим, что количество людей и рук не ограничено. Видео-демонстрация программы доступна в [7].

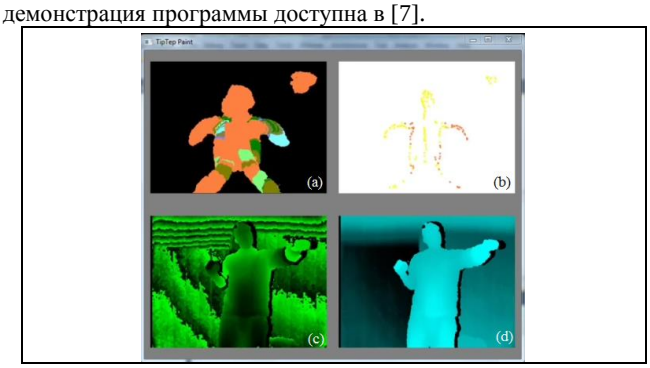

Рисунок 2 – Интерфейс программы для рисования с помощью жестов руки.
(a) Множество всех распознанных компонентов во всех кадрах видеоряда. (b) Множество центров всех распознанных компонентов во всех кадрах видеоряда. (c) Дальностное изображение. (d) Сглаженное дальностное изображение.

4. ИДЕНТИФИКАЦИЯ ДИНАМИЧЕСКИХ ЖЕСТОВ РУКИ

Задачей идентификации жеста является поиск эталонного образца, который имеет минимальное расстояние до показанного жеста. В случае динамических жестов задача идентификации усложняется неопределенностью начала и конца показанного жеста. Обойти эту трудность можно посредством наложения ограничений на жестикулирующего человека – началом жеста определить время, когда расстояние руки от сенсора становится меньше чем заданное значение d и концом жестикуляции - время, когда расстояние становится больше чем d. В этом случае вычислить координаты центров ладоней можно посредством алгоритма приведенной в разделе 3.

Идентификация динамического жеста осуществляется в два этапа:

1. Создание эталонных жестов.
2. Сопоставление показанного жеста с эталонными образцами.

Представим динамический жест в виде временного ряда (рис. 3).

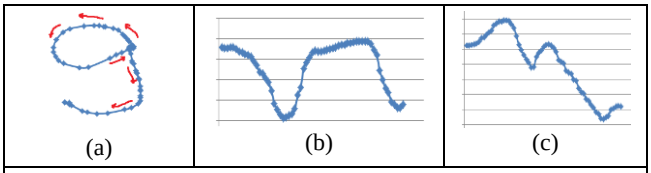

Рисунок 3 – (a) Траектория жеста. (b) – Изменение координат центра руки по оси абсцисс (c) - Изменение координат центра руки по оси ординат.

Обозначим буквой P временной ряд $\{p_1, p_2, ..., p_m\}$, где $p_1 = (x_1, y_1)$ представляет собой координаты центра ладони в первом кадре видеоряда и $p_m = (x_m, y_m)$ - в последнем кадре. Заметим, что цифра m, представляющая собой количество кадров в видеоряде при показе одного жеста, может меняться во время разных показов одного и того же жеста.

Сопоставление двух динамических жестов осуществляется путем нормализации и вычисления расстояния между соответствующими временными рядами с помощью алгоритма динамической трансформации шкалы времени.

Введем обозначения: $S_x = \min\{x_1, x_2, ..., x_m\}$, $E_x = \max\{x_1, x_2, ..., x_m\}$, $S_y = \min\{y_1, y_2, ..., y_m\}$, $E_y = \max\{y_1, y_2, ..., y_m\}$, $D_x = 1/(E_x - S_x)$, $D_y = 1/(E_y - S_y)$, $C_x = (E_x + S_x)/2$, $C_y = (E_y + S_y)/2$. Нормализованный временной ряд записывается в виде $P' = \{p'_1, p'_2, ..., p'_m\}$, где $p'_i = ((x_i - C_x) \cdot D_x, (y_i - C_y) \cdot D_y)$, для всех значений $i \in \{1,2,...,m\}$. Сравнение двух нормализованных временных рядов $P' = \{p'_1, p'_2, ..., p'_m\}$ и $Q' = \{q'_1, q'_2, ..., q'_n\}$ осуществляется посредством применения алгоритма DTW [8]. Для заданных временных рядов строится матрица расстояний $A \in R^{mxn}$ на метрике Манхеттена:

$$a_{i,j} = d(p'_i, q'_j) = | p'_i(x) - q'_j(x) | + | p'_i(y) - q'_j(y) |$$

Следующим шагом является поиск пути в матрице $A \in R^{mxn}$, начинающегося с элемента $a_{1,1}$ и достигающего элемента $a_{m,n}$, для которого сумма значений элементов данного пути минимальна. Найти такой путь за полиномиальное время можно посредством алгоритма динамического программирования. Для матрицы $A \in R^{mxn}$ создается новая матрица $B \in R^{(m+1)x(n+1)}$. Элементу $b_{1,1}$ присваивается значение 0, а прочим элементам первой строки и первого столбца матрицы B - значение ∞. Остальные элементы матрицы вычисляются следующим образом: $b_{i,j} = a_{i,j} + \min\{b_{i-,j}, b_{i,j-1}, b_{i-1,j-1}\}$. В качестве коэффициента схожести двух временных рядов выбирается значение элемента $b_{m+1,n+1}$.

На основе предложенного алгоритма была разработана система распознавания динамических жестов руки человека. Система распознает 12 жестов – 10 цифр и две геометрические фигуры. Человек с помощью движения руки рисует цифры и геометрические фигуры. Каждый жест сопоставляется со всеми эталонными жестами. В качестве распознанного жеста выбирается эталонный жест, который имеет наибольший коэффициент схожести с показанным жестом. Интерфейс реализованной системы показан на рисунке 4.

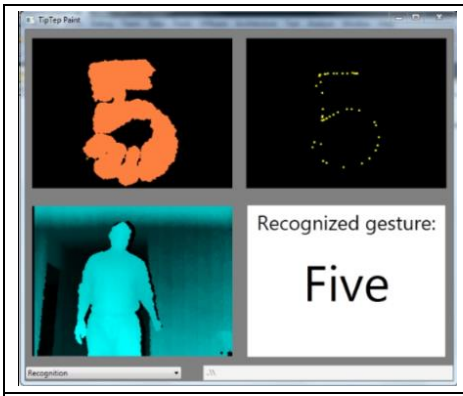

Рисунок 4 – Интерфейс программы распознавания динамических жестов руки.

Заметим, что направление показа жеста влияет на результаты распознавания. Например, если показать цифру ноль по

часовой стрелке и против часовой, то программа примет эти жесты за два разных. Распознать жесты независимо от направления движения руки можно посредством хранения для каждого эталонного жеста временных рядов, выполненных по разным направлениям.

5. РЕЗУЛЬТАТЫ

Предложенный алгоритм распознавания динамических жестов руки был протестирован на базе жестов двух разных людей. Тестовая база включала в себе 2400 жестов двух человек, показывающие цифры от нуля до девяти и две геометрические фигуры – квадрат и треугольник. В качестве эталонных жестов из тестовой базы произвольным образом были выбраны 12 образцов, по одному для каждого класса. В таблице 1 приведены характеристики качества распознавания алгоритма, где точность распознавания определяется как доля жестов действительно принадлежащих данному классу относительно всех жестов, которые система отнесла к этому классу. Полнота распознавания определяется как доля найденных классификатором жестов принадлежащих классу относительно всех жестов этого класса в тестовой выборке.

Таблица 1 – Характеристики качества распознавания

Характеристики качества распознавания	**Тестовая выборка**											
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	Δ	□
Точность	.75	.83	1	1	.79	.82	.81	1	1	.73	.88	.95
Полнота	.88	.98	.86	.9	1	.76	1	1	.7	.64	1	.7

Из таблицы 1 видно, что средняя точность распознавания составляет 88 процентов, а средняя полнота – 87, что является приемлемым результатом для использования предложенного алгоритма во многих системах взаимодействия человека с машиной. Ошибки классификатора можно объяснить как ошибками оператора во время жестикуляции, так и схожестью временных рядов некоторых жестов. Количество шагов, требуемых для сравнения двух жестов с числом кадров m и n соответственно, оценивается как $O(m \times n)$, что позволяет распознать жест сразу же после завершения его показа.

6. ЗАКЛЮЧЕНИЕ

Выполненные эксперименты показывают, что предложенные алгоритмы и методы могут быть использованы для создания новых типов человеко-машинного интерфейса. Они могут быть расширены и использованы для рисования с помощью жестов рук как альтернатива сенсорным экраном и как способ перевода жестового языка глухонемых на естественный язык. В дальнейшем планируется создать прототип программы, которая позволит распознавать статические и динамические жесты языка глухонемых.

7. БЛАГОДАРНОСТИ

Работа выполнена при поддержке проекта РФФИ №13-07-00025 А.

8. ССЫЛКИ

[1] Chu S., Tanaka J. Hand Gesture for Taking Self Portrait // Proceedings of the 14th international conference on Human-computer interaction: interaction techniques and environments - Part II. Springer-Verlag: 2011. — P.238-247

[2] Chen F., Fu C., Huang C. Hand gesture recognition using a real-time tracking method and hidden Markov models // Journal Image and Vision Computing. 2003. — P. 745-758

[3] Kim J., Thang N., Kim T. 3-D hand motion tracking and gesture recognition using a data glove // Industrial Electronics, 2009. ISIE 2009. IEEE International Symposium on. 2009. — P.1013-1018

[4] Shotton J., Fitzgibbon A., Cook M., Sharp T., Finocchio M., Moore R. Real-time human pose recognition in parts from single depth images // Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society: 2011. — P. 1297-1304

[5] Asus Xtion Pro Live. URL: http://www.asus.com/Multimedia/Xtion_PRO_LIVE/

[6] Shapiro L. Computer Vision. New Jersey: Prentice Hall, 2001. — p.608

[7] TipTep — Humanizing of digital environment. URL: http://tiptep.com/index.php/research

[8] Theodoridis S., Pikrakis A., Koutroumbas A., Cavouras D. Introduction to Pattern Recognition: A Matlab Approach. Academic Press, 2010 — p.231

Об авторах

Ваагн Нагапетян – аспирант кафедры информационных технологий факультета физико-математических и естественных наук Российского университета дружбы народов. Его адрес: vahagnahapetyan@gmail.com