ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

УДК 004.93’12

В. В. Девятков, А. Н. Алфимцев

РАСПОЗНАВАНИЕ МАНИПУЛЯТИВНЫХ ЖЕСТОВ

Предложен оригинальный метод распознавания манипулятивных жестов, основанный на использовании модели нечетких конечных автоматов, описываются его программная реализация и результаты экспериментов.

В последние годы все больший интерес проявляется к интерфейсу человека с компьютером [1–8] на основе использования им жестов. Цели такого интерфейса могут быть различными. В настоящей статье рассматривается случай, когда целью является управление компьютером или другим оборудованием. Для того чтобы такое управление стало возможным, необходимо достаточно надежно распознавать жесты человека. В общем случае человек использует жесты различных типов: эпистемологические, коммуникативные, манипулятивные. Эпистемологические жесты выполняются в целях исследования окружающего пространства путем тактильных ощущений. Коммуникативные жесты сопутствуют речи и применяются в целях помощи в выражении определенных мыслей. Манипулятивные жесты используются для передачи информации другому человеку или устройству, в частности для управления, поэтому в настоящей статье используются именно манипулятивные жесты. Распознавание жестов может быть определено как процесс установления сходства вновь выполненных жестов, которые будем называть распознаваемыми, со структурой (моделями) известных жестов, называемых обычно эталонными. Каждый эталонный жест отражает какие-то характерные черты класса схожих жестов.

Эталонные модели жестов. *Жесты* — это определенные перемещения частей тела человека (объектов). Жесты служат одним из важнейших инструментов человеческого общения [9]. В нашем случае таким объектом является рука. Эталонной моделью жеста будем называть определенную структуру представления этого жеста в двух-, трех- или n-мерном пространствах. Позднее понятие модели жеста будет дано более формально. Изложим принципы формирования эталонных моделей жестов.

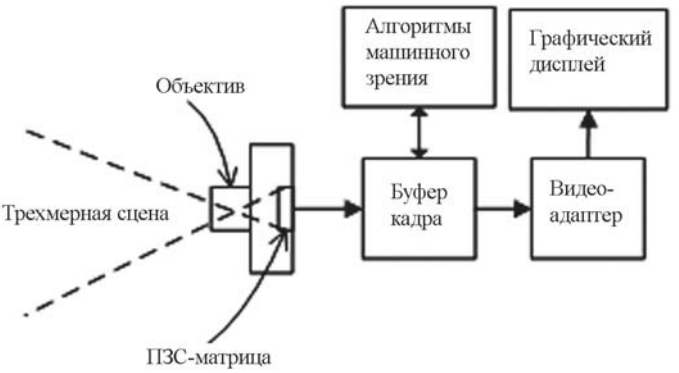

Рис. 1. Схема получения кадра изображения с помощью видеокамеры

Эталонная модель каждого жеста формируется в несколько этапов:

а) захват камерой кадров изображений, содержащих двигающийся объект, выполняющий жест;

б) нахождение двигающегося объекта в последовательности кадров изображений;

в) выделение кисти руки в изображении двигающегося объекта;

г) построение траекторий перемещения изображения кисти;

д) формирование эталонной модели жеста.

Рассмотрим эти этапы подробнее.

Захват камерой кадров изображений. Для этого использовали Web-камеру с разрешением 640×480, 8 бит, 30 кадр./с. Большинство современных видеокамер построено с использованием технологии приборов с зарядовой связью (ПЗС) [10]. На рис. 1 показана схема получения кадра изображения с помощью Web-камеры.

Кадр изображения представляет собой изображение в формате bitmap (сокращенно BMP) — это формат растровой графики для операционной системы Windows (пример кадра изображения приведен на рис. 2). В файлах BMP информация о цвете каждого пикселя кодируется тремя параметрами R, G, B (обычно о таком кодировании говорят, как о формате RGB: Red (красный), Green (зеленый), Blue (синий). Файл BMP разбит на разделы: заголовок файла растровой графики, информационный заголовок растрового массива, и таблица цветов или значений параметров R, G, B. Заголовок файла растровой графики содержит информацию о файле, в том числе адрес, с которого начинается

Рис. 2. Кадр изображения, полученный от Web-камеры

область данных растрового массива. В информационном заголовке содержатся сведения об изображении, хранящемся в файле, например, о высоте и ширине (в пикселях). В таблице цветов представлены собственно значения параметров R, G, B [11].

Комбинация трех параметров (основных цветов) R, G, B позволяет представить произвольный цвет видимого спектра. Но для распознавания объектов цветовое пространство, задаваемое в формате RGB, имеет ряд недостатков: большую корреляцию между компонентами, смешивание яркостной и цветовой составляющих, существенную неоднородность по восприятию. Поэтому, чтобы явно получить значения яркости и цвета пикселя, обычно переходят к другим форматам (другим цветовым пространствам), например к полутоновому формату или к формату HSV [10], использующему параметры H, S, V (здесь параметры H, S, V соответственно обозначают Hue (тон), Saturation (насыщенность), Volume (яркость). Параметр тона H характеризует преобладающий основной цвет (длину волны, преобладающую в излучении) и изменяется от 0 до 360, параметр насыщенности S характеризует близость к тоновой волне и изменяется от 0 до 1 (например, у белого цвета — насыщенность равна 0, так как невозможно выделить его цветовой тон), параметр V характеризует яркость пикселя (у черного цвета $V = 0$, у белого $V = 1$). Одинаково насыщенные оттенки могут иметь различные яркости.

Переход в полутоновое пространство из цветового пространства в формате RGB осуществляется по следующей формуле [10]:

$$I(x, y) = 0{,}2125R_(x, y) + 0{,}7154G(x, y) + 0{,}0721B(x, y),$$

где $(x,\ y)$ — координаты пикселя кадра изображения, $I(x, y)$ — значение яркости пикселя, $R(x, y)$, $G(x, y)$, $B(x, y)$ — значения параметров пикселя с координатами $(x,\ y)$ в цветовом пространстве RGB. Минимальное и максимальное значения яркости, получаемые по этой формуле, равны соответственно 0 и 255.

Переход в цветовое пространство HSV из цветового пространства в формате RGB осуществляется по следующим формулам [10]:

$$H = \frac{\pi}{3}\begin{cases} Cb - Cg, & \text{если } R = V; \\ 2 + Cr - Cb, & \text{если } G = V; \\ 4 + Cg - Cr, & \text{если } B = V, \end{cases}$$

где $Cr = \dfrac{(V - R)}{(V - v)}$; $Cg = \dfrac{(V - G)}{(V - v)}$; $Cb = \dfrac{(V - B)}{(V - v)}$;

$$S = \begin{cases} 0, & \text{если } V = 0; \\ \dfrac{(V - v)}{V}; \end{cases}$$

$$V = \max(R, G, B); \quad v = \min(R, G, B).$$

Нахождение двигающегося объекта в последовательности кадров изображений. На этом этапе, используя последовательность кадров, получаемую в процессе захвата изображения, находят двигающийся объект, для чего используется фактор изменения яркости пикселей, относящихся к движущемуся объекту, в последовательности двух смежных i-го и $(i + 1)$-го кадров изображений. Яркость каждого пикселя i-го кадра сравнивается с яркостью соответствующего пикселя $(i+1)$-го кадра. Если разность яркостей $D(x, y) = |I_i(x, y) - I_{i+1}(x, y)|$ пикселей превышает заданный порог, то этот пиксель $(i + 1)$-го кадра считается принадлежащим двигающемуся объекту. Порог может быть различным в зависимости от чувствительности используемой аппаратуры, освещенности и других факторов. В нашем случае порог выбран экспериментально для нормальной освещенности в помещении и равен 20.

В результате на данном этапе выделяется двигающийся объект, например рука человека (рис. 3).

Выделение кисти руки в изображении двигающегося объекта. Чтобы выделить в кадре изображения только кисть, ищем пиксели, цвет которых совпадает с цветом кожи человека (рис. 4). В цветовом пространстве HSV для этого цвета значения параметра H находятся в промежутке от 110 до 160 [10].

Построение траекторий перемещения изображения кисти. Траекторией перемещения (движения) кисти руки является последовательность координат ее центра тяжести (аналогично для других объектов). Координаты (x_R, y_R) центра тяжести вычисляются по формулам:

Рис. 3. Выделенный двигающийся объект в кадре изображения

Рис. 4. Центр тяжести кисти

$$x_R = \frac{\sum_{i=1}^{N} x_i}{N}, \quad y_R = \frac{\sum_{i=1}^{N} y_i}{N},$$

где x_i, y_i — координаты i-го пикселя, принадлежащего кисти руки; N — число таких пикселей. На рис. 4 центр тяжести показан крестиком.

При построении эталонной траектории движения кисти или другого объекта один и тот же жест повторятся многократно. Траектории движения каждого повторяемого жеста при этом не совпадают. Например, если жесты имеют вид буквы Z, то вместо одной траектории можем иметь множество траекторий, показанных на рис. 5. По осям x и y здесь отложены координаты пикселей.

Формирование моделей эталонных жестов. Каждое множество траекторий манипулятивных жестов имеет свои характерные особенности. Так, если жесты имеют вид буквы Z, то имеется участок траекторий, параллельный оси x и получаемый во времени слева направо, затем участок траекторий, получаемый во времени сверху вниз примерно под 45° к оси, и затем опять участок траекторий, параллельный оси x и получаемый во времени справа налево. Обобщенно траекторию движения всех жестов, имеющих вид буквы Z, можно представить в виде графа, показанного на рис. 6.

Вершина *1* этого графа объединяет множество точек, принадлежащих началу траекторий, вершины *2* и *3* соответствуют множествам точек перегиба траекторий, вершина *4* объединяет множество точек концов траекторий, дуги графа указывают на направление движения центра тяжести объекта по траекториям. Этот граф может служить основой для построения модели эталонного жеста.

Спрашивается, как представлять подобные графы — как модели эталонных жестов? как их алгоритмически формировать для различных жестов, чтобы можно было в дальнейшем использовать их для распознавания?

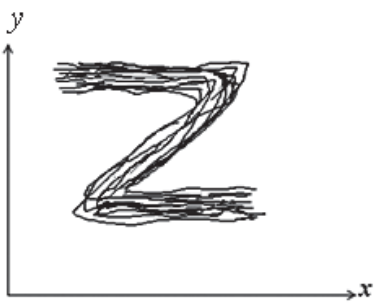

Рис. 5. Траектории жеста, повторенного несколько раз

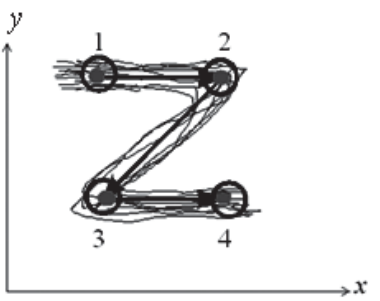

Рис. 6. Граф модели эталонного жеста

Поскольку каждая вершина графа объединяет характерные точки с определенным сходством, то, прежде всего, необходимо определить, какие точки относятся к одной и той же вершине. Множество точек, относящихся к одной вершине, назовем кластером. Число точек кластера обозначим N. Каждый кластер включает множество наборов значений характерных признаков $y_{k1}, y_{k2}, \ldots, y_{km}$. Каждый набор характерных признаков $y_{k1}, y_{k2}, \ldots, y_{km}$ образует точку y_k $(k = 1, \ldots, n)$ в m-мерном пространстве. В нашем случае набор значений характерных признаков помимо собственно геометрических координат точек может содержать дополнительно, например, бинарные признаки, определяющие, является ли данная точка началом траектории или нет, является ли она точкой перегиба или нет, является ли данная точка концом траектории или нет и т.п

Для нахождения кластеров воспользуемся одним из известных методов четкой кластеризации, чаще всего называемым методом кластеризации c-средних [12]. Напомним, в чем заключается суть этого метода.

В основе метода кластеризации c-средних лежит метод целевой функции. Целевая функция (критерий) создается таким образом, чтобы

• минимизировать расстояние между точкой в кластере и центром кластера;

• максимизировать расстояние между центрами кластеров.

Один из таких критериев известен как сумма квадратичных ошибок внутри класса, использующая евклидову норму для характеристики расстояния.

Этот критерий обозначается как $J(U, \boldsymbol{v})$, где U — разбиение всех точек на кластеры (непересекающиеся подмножества точек, объединение которых совпадает с исходным множеством точек, разбиваемым на кластеры). Параметр $\boldsymbol{v}$ — это вектор кластерных центров (множество кластерных центров, соответствующих разбиению U).

Формула критерия (целевой функции) будет следующей:

$$J(U, \boldsymbol{v}) = \sum_{k=1}^{n} \sum_{i=1}^{c} \chi_{ik} (d_{ik})^2.$$

Здесь d_{ik} — мера евклидова или какого-либо другого расстояния (в m-мерном пространстве характеристических параметров R^m) между k-м m-мерным вектором $\boldsymbol{y}_k$ и i-м кластерным центром $\boldsymbol{v}_i$, вычисляемая по формуле

$$d_{ik} = d(\boldsymbol{y}_k - \boldsymbol{v}_i) = \left[\sum_{j=1}^{m} (y_{kj} - v_{ij})^2 \right]^{1/2}.$$

Координаты кластерных центров $\boldsymbol{v}_i = \{v_{i1}, v_{i2}, \ldots, v_{im}\}$ вычисляются по формуле

$$v_{ij} = \frac{\sum\limits_{k=1}^{n} \chi_{ik} y_{kj}}{\sum\limits_{k=1}^{n} \chi_{ik}},$$

где χ_{ik} — характеристическая функция,

$$\chi_{ik} = \begin{cases} 1, \text{ если } y_{kj} \in A_i \\ 0, \text{ если } y_{kj} \notin A_i \end{cases},$$

а A_i — кластер.

Требуется найти оптимальное разбиение U^* на кластеры, для которого значение функции цели минимально, т.е.

$$J(U^*, \boldsymbol{v}^*) = \min_{U \in M_c} J(U, \boldsymbol{v}),$$

где M_c — множество всех различных разбиений на c кластеров.

Одна из стратегий метода кластеризации c-средних известна как итеративная оптимизация.

По концепции эта стратегия подобна другим итеративным стратегиям и включает в себя следующие шаги.

1. Зафиксировать число c кластеров ($2 \leqslant c < n$) и выбрать начальное разбиение $U^{(0)}$ множества точек траекторий на кластеры A_i, затем выполнить следующие шаги для $r = 0, 1, 2, \ldots$.

2. Вычислить центры $v_i^{(r)}$ всех кластеров, определяемых разбиением $U^{(r)}$.

3. Вычислить новые характеристические функции для всех i, k:

$$\chi_{ik}^{(r+1)} = \begin{cases} d_{ik}^{(r)} = \min\{d_{ik}^{(r)}\} \text{ для всех } j \in c; \\ 0 \text{ в противном случае.} \end{cases}$$

4. Построить новое разбиение $U^{(r+1)}$.

5. Если $U^{(r+1)} = U^{(r)}$, то остановить процесс и считать разбиение $U^{(r+1)}$ оптимальным. В противном случае принять $r = r + 1$ и перейти к шагу *2*.

Используя эту стратегию, можем получить модели всех эталонных жестов, которые можно представить графами, вершинам которых соответствуют кластеры со своими центрами, а дугам — направления траекторий движения. Пример жеста в виде буквы Z показан на рис. 7. Здесь вершинам соответствуют кластеры A_1, A_2, A_3, A_4. Координаты центров кластеров указаны на осях.

Модель эталонного жеста (см. рис. 7) не содержит информации о времени перемещения центров кластеров. Для того чтобы можно было

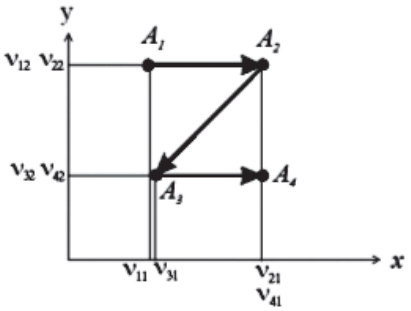

Рис. 7. Модель эталонного жеста в виде буквы Z

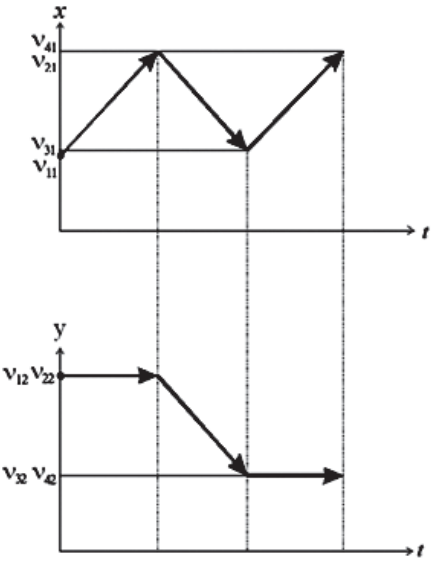

Рис. 8. Проекции модели эталонного жеста

учесть время, вместо модели, показанной на рис. 7 для случая двумерного пространства, будем использовать также модели, показанные на рис. 8, полученные в результате проекции траекторий движения центров тяжести соответственно на оси абсцисс и ординат. В общем случае m-мерного пространства число таких моделей $y_i(t)$ $(i = 1, \ldots, m)$ равно m.

Постановка задачи распознавания жестов. Рассмотрим одну из моделей $y_i(t)$ какого-либо жеста. Значение $y_i(t)$ в некоторый момент времени будем называть **отсчетом** $y_i(t)$. Число отсчетов совпадает с числом кластеров в алгоритме кластеризации c-средних.

Последовательность $n+1$ отсчетов $Y_i[t_0, t_n] = \{y_i(t_0), y_i(t_1), y_i(t_2), \ldots, y_i(t_n)\}$ i-й модели одного и того же жеста в течение нескольких последовательных моментов времени $t_0, t_1, t_2, \ldots, t_n$ (в течение временного интервала $[t_0, t_n]$) назовем **сигналом**. Множество отсчетов $K(t) = \{y_1(t), y_2(t), \ldots, y_m(t)\}$ m различных моделей одного и того же жеста в момент времени t назовем **кадром**. Последовательность кадров $K(t_0), K(t_1), \ldots, K(t_n)$, получаемых по m моделям одного и того же жеста в течение нескольких моментов времени $t_0, t_1, t_2, \ldots, t_n$ (в течение временного интервала $[t_0, t_n]$), назовем **потоком кадров**. Совокупность $n+1$ сигналов $Y_1[t_0, t_n], Y_2[t_0, t_n], \ldots, Y_n[t_0, t_n]$, относящихся к одному временному интервалу $[t_0, t_n]$, назовем **потоком сигналов**.

Сопоставим каждому отсчету $y_j(t_i)$ одного и того же сигнала состояние $b_j(t_i)$ конечного автомата M_j. Введем функцию выходов ко-

нечного автомата M_j

$$\varphi(b_j(t_i)) = y_j(t_i)$$

и функцию переходов автомата M_j

$$f(b_j(t_i), t_{i+1}) = b_j(t_{i+1}).$$

Таким образом, каждый отсчет — это значение функции выхода $y_j(t) = \varphi(b_j(t))$ автомата M_j; каждый сигнал — последовательность значений функций выхода $\boldsymbol{y}_j(t) = (y_j(t_0), y_j(t_1), \ldots, y_j(t_n))$ одного и того же автомата M_j; каждый кадр — это набор $\boldsymbol{y}(t) = (y_1(t), \ldots, y_m(t))$ значений функций выхода различных автоматов $M_1, M_2, \ldots, M_m$; поток сигналов представляется набором последовательностей значений функций выхода $\boldsymbol{y}_1(t), \boldsymbol{y}_2(t), \ldots, \boldsymbol{y}_m(t)$ соответственно конечных автоматов $M_1, M_2, \ldots, M_m$; поток кадров — это последовательность кадров $\boldsymbol{y}(t_0), \boldsymbol{y}(t_1), \ldots, \boldsymbol{y}(t_n)$. Поскольку значение функции выхода $y_j(t)$ однозначно определяется функцией выхода $y_j(t) = \varphi(b_j(t))$, то наряду с введенными обозначениями также используем:

последовательность состояний $\boldsymbol{b}_j(t) = (b_j(t_0), b_j(t_1), \ldots, b_j(t_n))$ автомата M_j, соответствующую сигналу;

макросостояние $\boldsymbol{b}(t) = (b_1(t), \ldots, b_m(t))$ автоматов $M_1, M_2, \ldots, M_m$, соответствующее кадру;

множество последовательностей состояний $\boldsymbol{b}_1(t), \boldsymbol{b}_2(t), \ldots, \boldsymbol{b}_m(t)$ автоматов $M_1, M_2, \ldots, M_m$, соответствующих потоку сигналов;

последовательность макросостояний $\boldsymbol{b}(t_0), \boldsymbol{b}(t_1), \ldots, \boldsymbol{b}(t_n)$ автоматов $M_1, M_2, \ldots, M_m$, соответствующую потоку кадров.

В теоретическом плане представляет интерес решение следующих задач распознавания жестов, представленных потоками.

1. Формирование эталонных потоков путем специальной обработки каждого манипулятивного жеста, вводимого в компьютер с помощью камеры.

2. Распознавание жестов путем сравнения по определенным критериям вновь вводимых потоков с эталонными потоками.

3. Выявление характерных свойств жестов путем формального вывода (доказательства) наличия определенных отношений на потоках.

4. Эквивалентные преобразования потоков, состоящие в минимизации, композиции и кодировании состояний автоматов $M_1, M_2, \ldots, M_m$, представляющих потоки.

Задачи 3 и 4 в настоящей статье не рассматриваются.

Распознавание жестов. Представим автомат M, соответствующий какой-либо модели некоторого жеста, его графом переходов (рис. 9). Каждая вершина графа помечена символом b_i, $i = 0, 1, \ldots, 12$ (вершины обозначены кружками). Каждая пара соседних вершин b_i, b_{i+1}

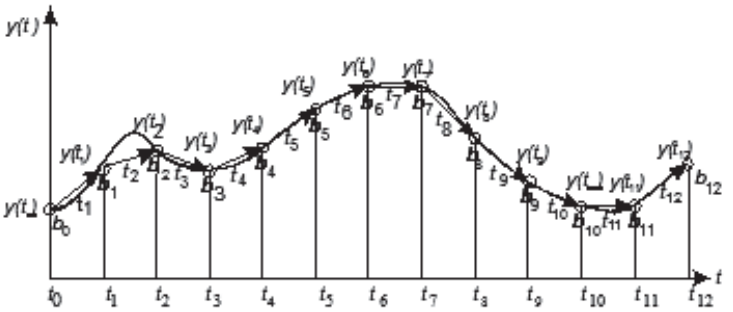

Рис. 9. Граф переходов сигнала

$(i = 0, 1, 2, \ldots, 11)$ соединена дугой, направленной от вершины i к вершине $i+1$. Дуги, направленные от вершины i к вершине $i+1$, помечены символом t_i в алфавите $T = \{t_0, t_1, t_2, t_3, \ldots, t_{m-1}\}$. Если выписать обозначения всех дуг слева направо, то получим последовательность символов $t_1t_2t_3t_4t_5t_6t_7t_8t_9t_{10}t_{11}t_{12}\Lambda$ (здесь Λ — пустой символ, который можно опустить). Эту последовательность можно рассматривать как слово или предложение некоторого языка $L = L(G)$, порождаемого автоматной грамматикой $G = \{V, T, P, S = b_0\}$,
$V = \{b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}, b_{11}\}$,
$T = \{t_0, t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}, \Lambda\}$,
$P = \{b_0 \to t_1b_1, b_1 \to t_2b_2, b_2 \to t_3b_3, b_3 \to t_4b_4, b_4 \to t_5b_5, b_5 \to t_6b_6, b_6 \to t_7b_7, b_7 \to t_8b_8, b_8 \to t_9b_9, b_9 \to t_{10}b_{10}, b_{10} \to t_{11}b_{11}, b_{11} \to \to t_{12}b_{12}b_{12} \to \Lambda\}$.

Если бы все было идеально, то для распознаваемого жеста можно построить автомат, которому бы в точности соответствовала одна из эталонных грамматик. Тогда язык, соответствующий этому автомату, мог бы быть однозначно распознан с помощью этой эталонной грамматики, а значит, был бы однозначно распознан и жест, соответствующий автомату. Однако в реальности такая идеальная ситуация недостижима.

Построим для каждой модели эталонного жеста по его четкой грамматике G нечеткую грамматику, базируясь на следующих принципах.

Каждой дуге графа соответствуют две инцидентные вершины b_i и $b_{i+1} = b_j$ (рис. 10). Координатой вершины b_i на оси абсцисс является t_i и $\varphi(b_i(t_i)) = y_i(t_i)$, а координата вершины b_j на оси абсцисс есть t_j и $\varphi(b_j(t_j)) = y_j(t_j)$. Допустим, что отсчеты обеих вершин по оси ординат для распознаваемого жеста могут изменяться в пределах среднеквадратического отклонения точек кластера от центра кластера:

$$s_i = \sqrt{\frac{\sum\limits_{i=1}^{N}(y_i - \nu_i)^2}{N}},$$

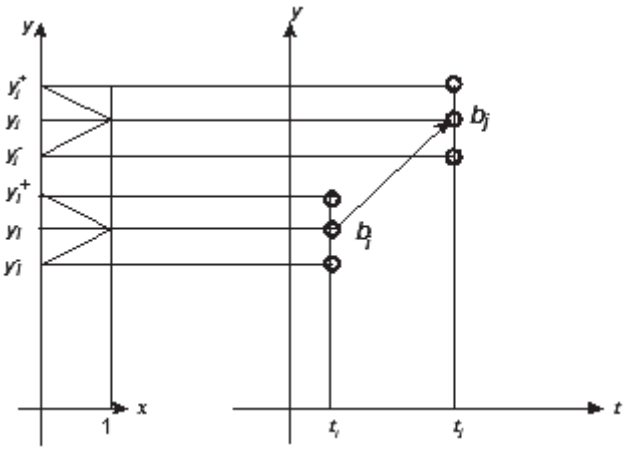

Рис. 10. Функции принадлежности вершин синтаксического графа

где N — число точек, принадлежащих кластеру; ν_i — координата центра кластера; y_i — координаты точек кластера. Для простоты будем считать, что s_i одно и то же для всех i. Обозначим это значение как s.

Таким образом, вместо одной вершины b_i с координатами t_i, y_i будем иметь множество вершин $b_{ri} \in B(b_i)$ с координатами, изменяющимися в пределах области $y_i^- = y_i - s$, $y_i^+ = y_i + s$, причем мощность множества $B(b_i)$ гипотетически будет равна $2s$, если по оси абсцисс откладываются координаты в пикселях. Таким образом, вместо одной вершины b_j с координатами t_j, y_j гипотетически будем иметь множество вершин $b_{sj} \in B(b_j)$ с координатами, изменяющимися в пределах области $y_j^- - y_j^+$, а вместо одной дуги t_j, ведущей из вершины b_i в вершину b_j, будем иметь множество всех дуг $t_{lj} \in T(t_j) = \{(b_{ri}, b_{sj}) | b_{ri} \in B(b_i), b_{sj} \in B(b_j)\}$, соединяющих каждую вершину множества $B(b_i)$ с каждой вершиной множества $B(b_j)$.

Зададим две треугольные функции принадлежностей $\mu_i(y)$, $\mu_j(y)$ на универсуме Y тройками значений соответственно в точках $\{y_i^-, y_i, y_i^+\}$ и $\{y_j^-, y_j, y_j^+\}$ (см. рис. 10). Каждая из этих функций определяется следующим выражением:

$$\mu(y) = \begin{cases} \dfrac{y}{y_k - y_k^-} - \dfrac{y_k^-}{y_k - y_k^-}, & \text{если } y_k^- \leqslant y \leqslant y_k; \\ \dfrac{-y}{y_k^+ - y_k} + \dfrac{y_k^+}{y_k^+ - y_k}, & \text{если } y_k \leqslant y \leqslant y_k^+; \end{cases}$$

здесь $k = i, j$. Эти функции определяют меру близости координат вершин графа к “наилучшим координатам”, которым соответствует значение функций принадлежности, равное 1.

Положим, что функция принадлежности каждой дуги $t_{lj} \in T(t_j)$, инцидентной вершинам $b_{ri} \in B(b_i)$ и $b_{sj} \in B(b_j)$, определяется следу-

ющим образом:

$$\mu_{T(t_j)}(t_{lj}) = \min\{\mu_i(y_{ri}), \mu_j(y_{sj})\}.$$

Нечеткая грамматика $G_F = \{V, T_F, P_F, S_F\}$ получается из четкой грамматики $G = \{V, T, P, S\}$ следующим образом. Алфавит нечеткой грамматики

$$T_F = \bigcup_j T(t_j).$$

Единственный начальный нетерминальный символ четкой грамматики заменяется множеством начальных нетерминальных символов

$$S_F = B(b_0).$$

Множество правил P_F нечеткой грамматики G_F будет следующим:

$$P_F = \{b_{ri} \to t_{lj} b_{sj}, \mu(b_{ri} \to t_{lj} b_{sj}) = \mu_{T(t_j)}(t_{lj}), t_{lj} \in T(t_j),$$
$$j = 1, \ldots, n, \quad l \leqslant 2s\}.$$

Нечеткая автоматная грамматика G_F порождает нечеткий язык $\{L(G_F), R_{L(G_F)}\}$:

$$L(G_F) = \{l^* | l^* = t_{l1} t_{l2} \ldots t_{ln}, \ \ t_{lj} \in T(t_j), \ \ l \leqslant 2s\};$$

$$R_{L(G_F)} = \{\mu_{L(G_F)}(l^*)/l^* | l^* =$$
$$= t_{l1} t_{l2} \ldots t_{ln}, \ t_{lj} \in T(t_j), \ \ \mu_{L(G_F)}(l^*) = \min_{t_{lj} \in \{t_{l1}, \ldots, t_{ln}\}} \{\mu_{T(t_j)}(t_{lj})\}.$$

С учетом введенных понятий распознавание жестов по нечетким эталонным грамматикам может осуществляться следующим образом.

1. Распознаваемый жест обрабатывается с теми же шагами дискретизации по временной оси, что и эталонные жесты.

2. Для распознаваемого жеста строится автомат, а затем язык, ему соответствующий.

3. Осуществляется синтаксический разбор языка, соответствующего распознаваемому жесту, с помощью всех эталонных нечетких грамматик.

4. Если синтаксический разбор для какой-либо нечеткой эталонной грамматики оказался успешным, то на этом работа алгоритма может быть закончена и вычислено результирующее значение функции принадлежности, характеризующее близость распознаваемого жеста к эталонному.

5. Если еще не все эталонные нечеткие грамматики использованы для разбора, то разбор может быть продолжен для них.

6. Если нерассмотренных эталонных грамматик не осталось и не было ни одного успешного синтаксического разбора, то распознавание жеста заканчивается неудачей (жест не был распознан).

7. Если было несколько удачных синтаксических разборов, то распознавание закончилось успешно, и распознаваемый жест относится к тому эталонному жесту, при разборе с помощью грамматики которого функция принадлежности оказалась максимальной.

Структура программной системы распознавания жестов показана на рис. 11. Она состоит из следующих блоков.

Видеокамера. Изображение с видеокамеры в формате bitmap поступает на вход блока формирования образов жестов через равные промежутки времени (30 кадров в секунду).

В **блоке формирования моделей жестов** (ФМЖ) на основе рассмотренных алгоритмов определения и отслеживания руки человека создается модель выполненного жеста.

На этапе обучения системы полученная модель жеста поступает на вход **блока обучения**, который формирует эталонные нечеткие конеч-

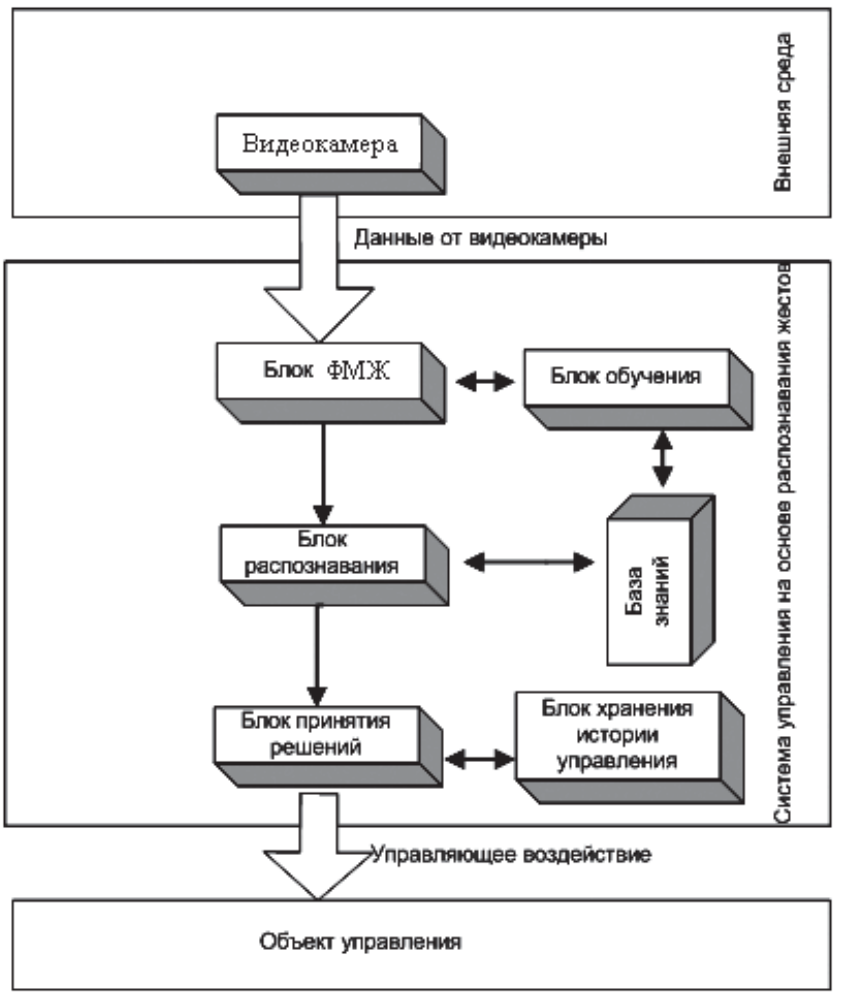

Рис. 11. Структура системы

ные автоматы в соответствии с рассмотренной методологией. Эталонные автоматы и ряд настроечных параметров системы сохраняются в **базе знаний**.

На этапе распознавания модель жеста, сформированная блоком ФМЖ, обрабатывается в **блоке распознавания**. В этом блоке осуществляется распознавание жестов на основе синтаксического разбора с помощью нечетких эталонных грамматик из базы знаний.

Если распознавание закончилось успешно, то **блок принятия решений** выдает управляющее воздействие в зависимости от типа распознанного жеста.

В **блоке хранения истории управления** сохраняется последовательность распознанных жестов и соответствующих им управляющих воздействий за определенное время, в частности, в целях интерпретации принятых решений по управлению.

Экспериментальные результаты работы системы. Целью экспериментов было определение точности и устойчивости работы системы распознавания манипулятивных жестов. Были проведены следующие четыре эксперимента.

Нахождение порога чувствительности алгоритма обнаружения движущегося объекта.

Определение точности распознавания жестов, выполняемых одной рукой одним человеком; двумя руками по очереди одним человеком и одной рукой различными людьми.

В трех последних экспериментах система должна была распознать каждый жест в реальном времени, т.е. без заметной для человека задержки.

Порог чувствительности алгоритма обнаружения движения определяли следующим образом. Пользователь выполнял движение, в нашем случае — это взмах рукой, в условиях нормальной освещенности (коэффициент естественной освещенности составлял 0,6; мощность освещения 300 лк [13]). Система фиксировала число пикселей, принадлежащих двигающемуся объекту, с использованием рассмотренного алгоритма обнаружения движения. Число пикселей, действительно принадлежащих двигающемуся объекту, определялось вручную.

Точность алгоритма обнаружения движения рассчитывалась по следующей формуле:

$$Точность\_алгоритма = \frac{Количество\_пикселей\_действительно\_принадлежащих\_двигающемуся\_объекту}{Количество\_найденных\_пикселей} \cdot 100\% .$$

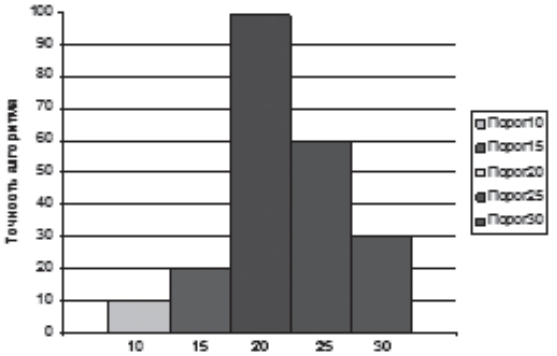

Рис. 12. Выбор порога в алгоритме обнаружения двигающегося объекта

На рис. 12 приведены гистограммы порогов чувствительности для данного алгоритма. По вертикальной оси расположена точность алгоритма в процентах, по горизонтальной — столбцы разных значений порогов. Из гистограмм видно, что при пороге, равном 20, 99 % зафиксированных пикселей принадлежит двигающемуся объекту.

Точность распознавания жеста, выполняемого одной рукой одним человеком, определяли следующем образом. Пользователь выполнял жесты в нормальных условиях освещения, находясь перед камерой. Жесты выполнялись одной рукой. Рука могла быть как левой, так и правой. Распознавали жесты в виде букв Z, M, P, N, W и фигур — окружность, волна, треугольник, квадрат, крест. Во время обучения каждый жест был повторен 10 раз. Жест выполнялся со средней скоростью ($\sim$1 жест/с). Число тестовых жестов равно 50.

Точность распознавания рассчитывалась по формуле

$$Точность\_распознавания = \\ = \frac{Количество\_распознанных\_жестов}{Количество\_тестовых\_жестов} \cdot 100\%.$$

Результаты распознавания представлены в табл. 1.

Нахождение точности распознавания жестов одного человека, выполняющего их двумя руками по очереди, проводилось аналогично предыдущему распознаванию. Жесты выполнялись правой и левой рукой поочередно. Во время обучения каждый жест был повторен 10 раз левой и правой рукой соответственно. Число тестовых жестов равнялось 100.

Метод расчета точности распознавания был повторен 3 раза, чтобы для каждого жеста можно было найти лучшую, худшую и среднюю точности. Результаты для тестовых данных приведены на рис. 13, вертикальная ось — точность распознавания в процентах, горизонтальная — распознаваемые жесты.

Таблица 1

Результаты распознавания жестов, выполненных одной рукой

Номер пользователя	Число повторений жеста / Число правильных ответов	Точность распознавания, %
1	5/5	100
2	5/4	80
3	5/4	80
4	5/5	100
5	5/4	80
6	5/4	80
7	5/5	100
8	5/5	100
9	5/4	80
10	5/5	100

Основное отличие следующего эксперимента нахождения точности распознавания состояло в том, что система обучалась одним пользователем, а тестировалась группой других пользователей.

Статистическое обоснование достоверности и работоспособности системы с реальными пользователями проводилась на доверительной выборке, сформированной по алгоритму стратификации [14]. Использование этого алгоритма позволяет с одинаковой вероятностью выбрать любой элемент из выборки, что является основным условием правильного формирования тестовой выборки. В данном случае выборка состоит из жестов, выполненных студентами, отобранных следующим образом. Из группы студентов был составлен общий список. После проверки на ошибки и отсутствие повторяемости, был выбран каждый пятый элемент списка.

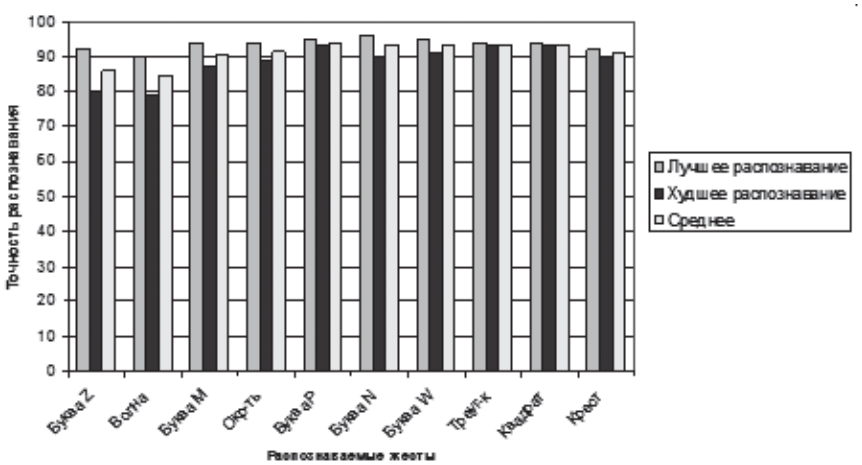

Рис. 13. Среднее результатов распознавания двух рук

Результаты распознавания жеста “окружность” для десяти пользователей представлены в табл. 2.

Таблица 2

Результаты распознавания для различных пользователей

Данные	Число выполненных жестов	Число распознанных жестов	Точность распознавания, %
Тестирующие	50	47	94

Точность распознавания второго, третьего и четвертого экспериментов превышает 90 %, что позволяет рассчитывать на применение данной системы для построения робастного интерфейса человек-компьютер. В этом интерфейсе данные о распознанных манипулятивных жестах могут быть использованы как команды управления программным обеспечением компьютера и интерфейс управления с помощью клавиатуры и мыши становится ненужным.

Заключение. Разработана методология распознавания манипулятивных жестов, базирующаяся на использовании модели нечетких конечных автоматов. Методология была апробирована в программной системе на персональном компьютере с использованием видеокамеры. Для захвата изображения использовались видеокамеры низкого качества (web-камеры) разрешением 640×480, 8 бит, 30 кадр./с. Система распознает десять манипулятивных жестов с точностью около 90 %, при этом не требуется больших вычислительных ресурсов, может совместно работать с несколькими приложениям в операционной системе Windows XP. Выбранные манипулятивные жесты исключали двусмысленность передаваемой информации. Это обеспечивалось выбором жестов, неиспользующихся в обычном общении и состоящих из базовых жестов языка глухонемых, интуитивно понятных пользователю.

Популярными моделями для распознавания жестов являются скрытые марковские модели, байесовы сети, нейронные сети. Главные недостатки этих моделей — это необходимость жестко предопределенной внутренней структуры, хорошо сегментированный набор обучающей выборки, частое переобучение.

Главными преимуществами использованной модели нечеткого конечного автомата является возможность строить распознаватель, имея всего несколько примеров в обучающей выборке, строить автоматы разной длины, распознавать жесты с траекторией, включающей циклы и пересечения.

Кроме того, разработан алгоритм захвата и отслеживания области интересов на сложном фоне. Алгоритм не требует дополнительных

маркеров на теле человека, выполняющего жест, распознает жесты в комнате с электрическим и дневным освещением, различным фоном.

В дальнейшем развитие системы распознавания предполагается связать с распознаванием групп жестов как последовательностных процессов, выявить и описать логические отношения между жестами; учесть достижения в области кинесики, физиолингвистики, дактилологии.

СПИСОК ЛИТЕРАТУРЫ

1. Alsabti K., Ranka S., and Singh V. An efficient K-means clustering algorithm. http://www.cise.ufl.edu/ *ranka/*, 1997.
2. Akyildiz F., Su W., Sankarasubramaniam Y., Cayirci E. A survey on sensor networks / IEEE Communication Magazine, August 2002. – P. 102–114.
3. Bonnet P., Gehrke J. E., Seshadri P. Towards sensor database systems. ASM Mobile Data Management. 2001.
4. Burschka D., Ye G., Corso J., and Hager G. A practical approach for integrating vision-based methods into interactive 2d/3d applications // CIRL Lab Technical Report CIRL-TR-05-01, The Johns Hopkins University, 2005.
5. Carbini S., Viallet J. E., Bernier O., Bascle B. Tracking body parts of multiple people for multi-person multimodal interface // Computer Vision in Human–Computer Interaction, in: ICCV 2005 Workshop, Beijing, China, 2005. P. 16–25.
6. Estrin D., Govindan R., Heidemann J., Kumar S. Next century challenges: Scalable coordination in sensor networks. ASM MOBICOM, 1999.
7. Maesschalck R., Jouan-Rimbaud D., Massart D. L. Tutorial. The Mahalanobis distance, Chemom. Intell. Lab. Syst., 50, 1, 2000.
8. Визильтер Ю. В., Желтов С. Ю., Ососков М. В. Системы распознавания и визуализация характерных черт человеческого лица в реальном времени на персональной ЭВМ с использованием web-камеры // Труды конф. Графикон 2002. – Нижний Новгород, Россия: 2002. – С. 251–254.
9. Григорьева Е. В. Обучение невербальным компонентам иноязычного общения (жестовый комплекс) // Университетские чтения. Симп. 1. Сек. № 1-20. Актуальные проблемы языкознания и литературы, 2006. – С. 1–3.
10. Шапиро Л., Стокман Д. Компьютерное зрение / Пер. с англ. – М.:БИНОМ, Лаборатория знаний, 2006. – 752 с.
11. Просис Д. Файлы растровой графики: взгляд внутрь. – PC Magazine, December 3, 1996. – P. 321.
12. Штовба С. Д. Введение в теорию нечетких множеств и нечеткую логику. http://rrc.dgu.ru/res/matlab/fuzzylogic/book1/12.html
13. Физические факторы производственной среды, оценка освещения рабочих мест; МУ 2.2.4.706-98/МУ ОТ РМ 01-98. http://allru.org/BPravo/DocumShow.asp?DocumID=80276
14. Мешкова Т. А., Малых С. Б., Куравский Л. С. Стандартизация психологических тестов: проблема формирования репрезентативной выборки // Учеб.-метод. пособ. – М.: МГППУ, 2003. – 72 с.

Статья поступила в редакцию 9.02.2007

Владимир Валентинович Девятков родился в 1939 г., окончил Ленинградский институт точной механики и оптики в 1963 г. Д-р техн. наук, профессор, заведующий кафедрой "Информационные системы и телекоммуникации" МГТУ им. Н.Э. Баумана. Академик международной академии информатизации. Автор более 80 научных трудов в области логического управления, компьютерных систем и комплексов технической кибернетики.

V.V. Devyatkov (b. 1939) graduated from the Leningrad Institute for Precise Mechanics and Optics in 1963. D. Sc. (Eng.), professor, head of "Information Systems and Telecommunications" department of the Bauman State Technical University. Academician of International Academy of Informatization. Author of over 80 publications in the field of logical control, computer systems and complexes, technical cybernetics.

Александр Николаевич Алфимцев родился в 1983 г. В 2005 г. окончил МГТУ им. Н.Э. Баумана. Аспирант кафедры "Информационные системы и телекоммуникации" МГТУ им. Н.Э. Баумана. Автор 8 научных работ и 2 патентов на изобретение в области методов искусственного интеллекта, распознавания образов, компьютерного зрения.

A.N. Alfimtsev (b. 1983) graduated from the Bauman Moscow State Technical University in 2005. Post-graduate of "Information Systems and Telecommunications" department of the Bauman Moscow State Technical University. Author of 8 publications and 2 invention patents in the field of methods of artificial intelligence, image identification, computer vision.

Адрес редакции журнала: 105005, Москва, 2-я Бауманская ул., д.5.

Тел.: (495) 263-62-60; 263-60-45. Факс: (495) 261-45-97.

E-mail: press@bmstu.ru