

**В. А. КОЛБАСИН**, канд. техн. наук;  
**Е. В. УЛЬЯНКО**, магистрант НТУ «ХПИ»

## **РЕАЛИЗАЦИЯ СЕНСОРНОГО ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ С ИСПОЛЬЗОВАНИЕМ ОБЫЧНОГО МОНИТОРА И ВЕБ- КАМЕРЫ**

Запропоновано метод реалізації функцій сенсорного екрану за допомогою веб-камери та звичайного монітору. Розглядаються методи пошуку та аналізу об'єктів по серії зображення.

Предложен метод реализации функций сенсорного экрана с помощью веб-камеры и обычного монитора. Рассматриваются методы поиска и анализа объектов по серии изображений.

The method for simulating functions of the touch screen using the web camera and typical monitor. The methods of search and analysis objects in a series of images.

**Введение.** Еще недавно для диалога пользователя с компьютером могли использоваться только манипулятор «мышь» и клавиатура. Однако активное развитие альтернативных технологий человеко-машинного интерфейса привело к появлению и внедрению в повседневную жизнь новых, более удобных способов взаимодействия пользователя с машиной. Одним из них является сенсорный интерфейс пользователя, обычно реализуемый с использованием сенсорных экранов.

Сенсорный экран позволяет пользователю управлять процессами, происходящими в компьютере при помощи прикосновений к экрану. При этом появляется возможность обращаться с виртуальными экранными объектами практически так же, как с объектами реального физического мира. Собственно, эта возможность и сделала сенсорный экран таким популярным. В настоящее время сенсорные экраны успешно используются в таких прикладных устройствах, как мобильные телефоны, планшетные компьютеры, платежные терминалы, информационные киоски, панели управления в промышленности и в медицине. Однако стандартные сенсорные экраны могут быть использованы не везде.

Технически сенсорный экран состоит из двух частей: собственно дисплей, который отображает информацию, и сенсорный элемент – дополнительное устройство, которое обычно устанавливается поверх дисплея и служит для определения координат прикосновения к экрану. В настоящее время используются сенсорные элементы, построенные на основе различных принципов функционирования: резистивные, эмиссионные, проекционно-эмиссионные. Сенсорный элемент должен быть с одной стороны чувствительным к нажатию, а с другой – устойчивым к механическим повреждениям и влиянию окружающей среды. Естественная противо-

речивость этих двух требований усложняет конструкцию и удорожает изготовление сенсорных экранов, особенно в случае экранов большой диагонали [1].

Так как суть функционирования сенсорного экрана сводится к определению позиции прикосновения к экрану, сенсорный интерфейс пользователя можно реализовать при помощи видеокамеры, наблюдающей за дисплеем. В этом случае программное обеспечение системы должно в реальном режиме времени определять изменения в кадре, определять среди них движения указателя, нажатие указателем на экран и генерировать понятные для прикладного программного обеспечения (ПО) сигналы, содержащие координаты нажатия. Данная работа посвящена исследованию возможности созданию программной системы, реализующей сенсорный интерфейс пользователя при помощи стандартного дисплея и простой видеокамеры (веб-камеры). В качестве указателя в данной работе используется цветовой маркер, прикрепленный к пальцу. Это упрощает распознавание позиции указателя, но вводит следующее ограничение – цвет маркера не должен использоваться элементами графического интерфейса пользователя.

**Поиск позиции указателя.** Для определения позиции указателя необходимо найти движущиеся объекты в последовательности кадров с видеокамеры и выбрать среди них объект, соответствующий параметрам указателя. Движущиеся объекты в данной работе определяются на основе анализа разницы кадров.

Разница  $P$  двух последовательных кадров  $S1$  и  $S2$  определяется как

$$P_{i,j} = |S1_{i,j} - S2_{i,j}|, \quad i = \overline{1, W}, j = \overline{1, H}, \quad (1)$$

где  $S1$ ,  $S2$  – предыдущий и последующий кадры;

$W$ ,  $H$  – ширина и высота кадра.

Значения элементов матрицы  $P$ , соответствующие неподвижным объектам изображения, будут иметь значения, близкие к нулю. Если же в какой-то области кадра выполняется движение, то пиксели объекта будут замещаться пикселями фона и наоборот. За счет этого значения точек межкадровой разницы на границах движущихся объектов будут существенно отличаться от нуля.

Также значения элементов матрицы межкадровой разницы могут быть отличны от нуля вследствие влияния шума сенсора видеокамеры, изменения освещения и колебаний камеры и экрана. Чтобы уменьшить влияние этих факторов в работе выполняется двухэтапная фильтрация матрицы  $P$ .

На первом этапе выполняется медианная фильтрация матрицы межкадровой разницы фильтром со скользящим окном со стороной 5 пикселей. За счет медианной фильтрации подавляется шум сенсора видеокамеры и шум, получающийся в результате дрожания видеокамеры.

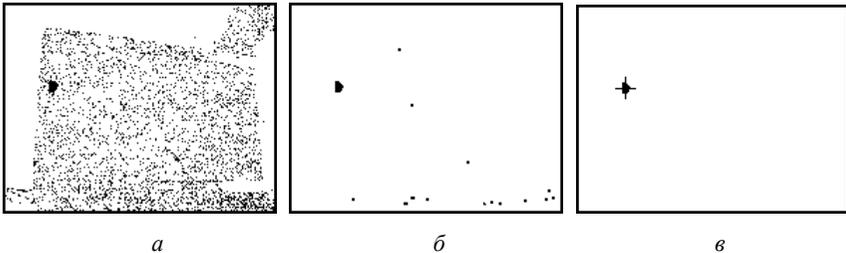
На втором этапе выполняется пороговая бинаризация матрицы межкадровой разницы  $P$  по формуле

$$F_{i,j} = \begin{cases} 0, & |P_{i,j}| \leq \theta, \\ 1, & |P_{i,j}| > \theta, \end{cases} \quad (2)$$

где  $\theta$  – значение порогового уровня, выбираемое экспериментально в процессе калибровки.

К полученному в результате фильтрации и бинаризации результату применяется метод оценки геометрических характеристик объектов на изображении, предложенный в работе [2]. Применяя данный метод, находятся оценки площади, центра тяжести и линейных размеров объектов на бинаризованном изображении. Далее в качестве искомого объекта выбирается объект с наибольшей площадью, расположенный в границах монитора, в цветном изображении которого преобладает выделенный для указателя цвет. В качестве координат объекта используются координаты его центра тяжести.

Процесс получения межкадровой разницы, фильтрации и выделения объекта представлен на приводимом ниже рисунке.



Поэтапная обработка разницы кадров для обнаружения указателя ( $a$  – межкадровая разница,  $b$  – результат фильтрации и бинаризации,  $в$  – выделенный объект)

**Определение экранных координат указателя.** Полученные координаты указателя определяют его позицию в кадре, но для использования данной системы в качестве сенсорного экрана эти координаты необходимо перевести в экранные координаты. Для этого можно выполнить обратное перспективное преобразование [3], но это требует большого объема калибровочных данных (в частности знания точного взаимного расположения камеры и экрана) и значительных вычислительных ресурсов. Поэтому для определения экранных координат в работе предлагается метод на основе использования калибровочной сетки.

Предлагается в процессе первичной настройки системы (ее калибровки) разделить экран на условные прямоугольники с некоторым заданным шагом. Далее, в каждом из узлов получившейся калибровочной сетки выводить

мигающий указатель, для которого в соответствии с описанным выше алгоритмом будут определяться координаты указателя в кадре. Экранные координаты и кадровые координаты всех узлов в кадре сохраняются в качестве калибровочных данных.

Чтобы преобразовать кадровые координаты указателя  $(x_k, y_k)$  в экранные  $(x_s, y_s)$  необходимо найти ближайший узел калибровочной сетки и выполнить билинейную интерполяцию по формулам

$$x_s = \frac{x_k - xk_{i,j}}{xk_{i+1,j} - xk_{i,j}} \cdot \Delta x \cdot i; \quad y_s = \frac{y_k - yk_{i,j}}{yk_{i+1,j} - yk_{i,j}} \cdot \Delta y \cdot j, \quad (3)$$

где  $i, j$  – позиция ближайшего к указателю узла калибровочной сетки;

$xk_{i,j}, yk_{i,j}$  – его координаты.

**Результаты.** Предложенная методика была реализована программно, и был выполнен ряд экспериментов по определению точности определения позиции указателя. Для эксперимента было выбрано два варианта расположения камеры относительно монитора с разным углом наклона к нему и три варианта размеров калибровочной сети. Во всех тестах в наилучшем случае позиция указателя определялась с нулевой ошибкой, средние и максимальные значения ошибки определения позиции указателя представлены в таблице.

Ошибка определения позиции указателя

Угол	150 °			120 °			
	Шаг сетки, пикс.	315×225	265×175	215×165	315×225	265×175	215×165
Ошибка средняя / максимальная, пиксели	x	14 / 41	13 / 38	12 / 34	9 / 24	9 / 22	8 / 19
	y	19 / 36	18 / 31	17 / 30	17 / 28	16 / 29	15 / 26

Таким образом, точность определения позиции указателя является достаточной для многих практических применений. Скорость работы программной реализации позволяет использовать ее для определения позиции указателя в режиме реального времени и данный метод может найти применение при создании сенсорного интерфейса пользователя.

**Список литературы:** 1. *Вихрев Л.* Легким движением руки: сенсорные панели / *Л. Вихрев* // Компоненты и технологии. – 2005. – № 5. 2. *Соифер В. А.* Методы компьютерной обработки изображений / *В. А. Соифер.* – М. : Физматлит, 2003. – 784 с. 3. *Иванов В. П., Батраков А. С.* Трехмерная компьютерная графика / *Г. С. Полищук.* – М. : Радио и связь, 1995. – 224 с.