

NEWS

OF THE NATIONAL ACADEMY OF SCIENCES OF THE REPUBLIC OF KAZAKHSTAN

PHYSICO-MATHEMATICAL SERIES

ISSN 1991-346X

Volume 3, Number 307 (2016), 77 – 82

QUESTIONNAIRES RECOGNITION DURING SOCIAL RESEARCH**N. Litvinenko**

Institute of mathematics and mathematical modeling CS MES RK, Almaty, Kazakhstan

E-mail: n.litvinenko@inbox.ru**Keywords:** Bitmap, gdi+, respondent, questionnaire, graphic image, pixel.

Abstract. In social and psychological research there is the topical problem of input and processing the survey data. Manual data input is not optimal due to large time and labor costs. Moreover, manual input is highly unreliable as directly linked to the human factor that can cause a lot of errors when typing. The goal of this paper is solving of some technical problems that appear in the development of the automated system for input of filled questionnaires of social or psychological survey. The first task is solving problems related to various physical distortions of filled questionnaires that appear in generation of graphic images of these questionnaires (photography, scanning, etc.). The second task is the recognition of the marked field on the graphical image of the questionnaire. The methodology of this work consists in the development of technical methods of recognition of graphic images of filled questionnaires in the most common formats – BMP, GIF, JPG in Windows operating system with Visual Studio 2013, C#, and using of gdi+. The result of this paper is a software product that allows to recognize graphic images of some types of sociological or psychological questionnaires in automatic or manual mode. The scope results can be different scientific and practical research in the field of sociology or psychology if you have a large number of respondents. The results can be used in different scientific and practical studies in the field of sociology or psychology involving the large number of respondents.

УДК 004.43; 004.93; 519.254

**РАСПОЗНАВАНИЕ БЛАНКОВ АНКЕТИРОВАНИЯ ПРИ
ПРОВЕДЕНИИ СОЦИОЛОГИЧЕСКИХ ИССЛЕДОВАНИЙ****Н. Литвиненко**

Институт математики и математического моделирования КН МОН РК, Алматы, Казахстан

Ключевые слова: Bitmap, gdi+, респондент, анкета, графический образ, пиксель.

Абстракт. В социологических и психологических исследованиях актуальна проблема ввода и обработки данных анкетирования. Ручной ввод данных не является оптимальным в связи с большими временными и трудовыми затратами. Кроме того, ручной ввод весьма ненадежен, так как напрямую связан с человеческим фактором, что может повлечь за собой большое количество ошибок при вводе. Целью данной статьи является решение некоторых технических проблем, возникающих при организации автоматизированной системы ввода заполненных анкет социологического или психологического опроса. Первой задачей является решение вопросов, связанных с различными физическими искажениями заполненных анкет, возникающих при получении графических образов данных анкет (фотографирование, сканирование и прочее). Второй задачей является распознавание отмеченных полей на графическом образе анкеты. Методология данной работы заключается в разработке технических способов распознавания графических образов заполненных анкет в наиболее популярных форматах – BMP, GIF, JPG в среде Windows, VisualStudio 2013, C#, с использованием средств gdi+. Результатом работы является программный продукт, позволяющий распознавать графические образы некоторых типов социологических или психологических анкет в автоматическом или ручном режиме. Областью применения результатов могут быть различные научные и

практические исследования в области социологии или психологии при наличии большого количества респондентов.

Описание проблемы. При проведении социологических и психологических исследований существует проблема ввода и обработки данных заполненных анкет. При объемных исследованиях ввод данных анкетирования оператором требует достаточно много времени. Обычно появляется большое количество ошибок и поэтому представляется весьма перспективным организовать ввод заполненных анкет в виде их графических изображений в наиболее распространенных форматах – BMP, GIF, JPG. Данные форматы поддерживаются большинством современных камер и фотографических аппаратов, а также практически всеми сканерами. При решении данной проблемы возникает несколько интересных задач:

- При фотографировании возникают искажения в размерах заполненной анкеты. Прямоугольник заполненной анкеты превращается в некоторый произвольный четырехугольник с углами, отличными от прямых, с искаженными длинами сторон, анкета может быть сфотографирована с любой стороны. Каким образом привести данное искаженное изображение к некоторому стандартному виду с минимальными искажениями изображения.

- При фотографировании и сканировании возникают искажения в гамме цветов изображения анкеты. Каким образом преобразовать данную гамму к исходной цветовой гамме анкеты.

- Каким образом распознать отмеченные поля заполненной анкеты.

В данной работе освещаются 1 и 3 пункты. Предполагается, что фотографирование или сканирование производится качественно, с минимальным искажением исходной гаммы цветов заполненной анкеты.

Для разработки программного продукта был выбран язык C#, поскольку он удачно сориентирован на платформу .NET. В нем хорошо сочетаются проверенные средства программирования и последние системные разработки. Базовые понятия и принципы работы на платформе (.NET+C#) доступно описаны в книгах [1-2].

Решение задачи физического искажения изображения анкеты. Возможны 2 подхода к решению сложных графических задач - воспользоваться низкоуровневыми средствами OpenGL или использовать принципы работы gdi+ в среде .NET. Возможности OpenGL хорошо описаны в литературе [3-5]. OpenGL позволяет писать быстро работающий программный код. Однако написание кода достаточно трудоемко и требует больших временных затрат. Программы, написанные в среде gdi+ , работают медленнее, но библиотеки gdi+ хорошо сориентированы на работу в .NET, в особенности для языка C#. Основные приемы работы с графикой и описание базовых классов .NET, предназначенных для отправки управляющих инструкций драйверам графических устройств для обеспечения корректного вывода графики на экран, хорошо представлено в книге [6]. Для решения задачи определения истинного положения искаженного изображения анкеты было принято решение разместить на бланке анкеты 5 специальных маркеров, обозначающих контуры анкеты. Маркеры представляют собой квадраты черного цвета размером 24*24 пикселя. 4 маркера расположены в углах анкеты, 5 маркер расположен на нижней стороне анкеты и позволяет определить положение анкеты на плоскости. Маркеры пронумерованы по часовой стрелке. 1 маркер – левый верхний угол, 2 маркер – правый верхний угол. 3 маркер – правый нижний угол. 5 маркер – левый нижний угол. 4 маркер лежит между 3 и 5 маркерами.

Идея определения истинных размеров и положения анкеты заключается в следующем. Подготовим панель с размерами, соответствующими размерам исходного бланка анкетирования. Трансформируем графическое изображение анкеты таким образом, чтобы 1, 2, 3 и 5 маркеры оказались в углах данной панели, а 4 маркер оказался на нижней стороне панели между 3 и 5 маркерами.

Массив изображений анкет должен находиться в отдельной директории. В каждой директории должны находиться анкеты только одного вида. Оператор выбирает директорию с нужными анкетами. Директория с анкетами помещается в специальное окно, где можно выбрать интересующие анкеты, пометив их галочками. В автоматическом режиме (кнопка ==>> или кнопка

«Прочитать выбранное») все выбранные анкеты будут распознаны и информация с анкет помещена в базу, смотреть Рисунок 1.

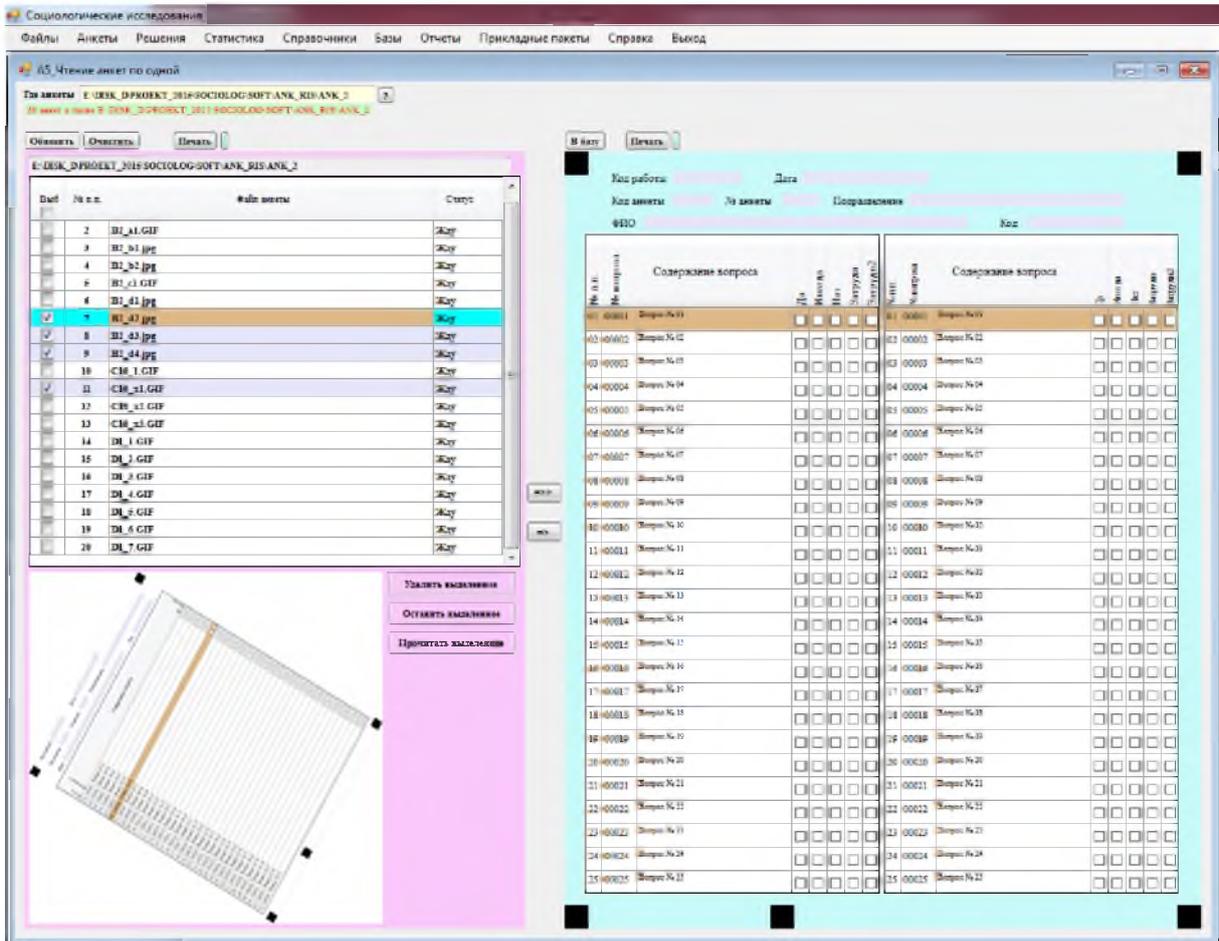


Рисунок 1 – Окно распознавания изображений анкет в потоке

В одиночном режиме (кнопка =>) будет распознаваться текущая анкета и только ее информация будет помещаться в базу. В случае, если анкет для обработки в выбранной директории много, можно воспользоваться кнопками «Удалить выделенное» - из окна будут удалены все выделенные анкеты и «Оставить выделенное» - в окне останутся только выделенные анкеты. Работа будет продолжаться только с анкетами, оставшимися в окне. В случае ошибочных действий с анкетами можно воспользоваться кнопкой «Обновить», которая возвращает в окно исходный набор анкет из выбранной директории. Очистить окно можно при помощи кнопки «Очистить».

Распознавание данных анкет происходит в следующем порядке:

1. Очередная анкета выбирается из верхнего окна. Ее изображение помещается в нижнее окно.
2. Программа находит пять маркеров анкеты.
3. Программа преобразует изображение анкеты таким образом, чтобы четыре маркера находились в углах окна, а пятый маркер находился на нижней стороне окна. После преобразования анкета будет занимать полностью все окно.
4. Теперь данные на анкете можно определять по известным координатам каждого поля анкеты.

Алгоритм поиска маркеров анкеты. Алгоритм поиска маркеров также работает в среде gdi+.

Цвет маркеров может быть не абсолютно черным, но должен находиться в пределах – (RED< 30, GREEN< 30, BLUE< 30). Это вызвано тем, что при фотографировании или сканировании цвета могут искажаться. Проблемы цветовосприятия, различные цветовые модели, растровые алгоритмы и преобразования на плоскостях хорошо описаны в [7]. Размеры маркеров 24*24, однако при фотографировании изображение может быть уменьшено.

Последовательность поиска маркеров следующая:

1. Строим Bitmap(bitmap) панели с анкетой.

panel1.DrawToBitmap(bitmap,rc1);

2. Рассматриваем bitmap порциями по квадратам 30*30. Определяем в очередной рассматриваемой порции наличие черных точек. Если черных точек достаточно много (например, больше 50), требуется выяснить, не маркер ли это. Если это маркер, вычисляем его центр и запоминаем. Очевидно, черных точек в порции должно быть достаточно много, чтобы гарантировать, что это не текст, не линии и т.д. Данный способ позволяет найти именно маркеры.

3. Таким образом, будут найдены все пять маркеров. Теперь необходимо определить номера каждого маркера. По нашему предположению, 1 маркер находится в левом верхнем углу, 2 маркер в правом верхнем углу, 3 маркер в правом нижнем углу, 5 маркер в левом нижнем углу, 4 маркер находится между 3 и 5 маркерами. 3, 4 и 5 маркеры определяются из условия, что их центры лежат на одной прямой. Условием принадлежности трех точек одной прямой является равенство нулю определителя:

$$\begin{vmatrix} x_i & y_i & 1 \\ x_j & y_j & 1 \\ x_k & y_k & 1 \end{vmatrix} = 0$$

Где $i, j, k = 1, 2, \dots, 5, i \neq j \neq k$,

$(x_i, y_i), i = 1, 2, \dots, 5$ – координаты центров маркеров

Таких определителей будет $C_5^3 = \frac{5 \cdot 4 \cdot 3}{1 \cdot 2 \cdot 3} = 10$

В силу приближенности наших вычислений может оказаться, что ни один из определителей не равен 0. Поэтому вместо равенства нулю, мы ищем определитель с наименьшим значением. Вычисляем все 10 определителей и сравниваем их значения. Определитель с наименьшим значением позволяет нам выбрать 3, 4 и 5 маркеры. Однако какой из них 3, какой 4, какой 5 - пока неизвестно.

3 и 5 маркеры определяются из условия, что расстояние между ними больше, чем расстояние между 3 и 4 маркерами и 4 и 5 маркерами. 3 маркер определяется из условия, что расстояние между 3 и 4 маркерами больше, чем расстояние между 4 и 5 маркерами.

Определив 3, 4 и 5 маркеры, приступаем к определению - какой из оставшихся маркеров является 1 маркером, а какой 2 маркером.

1 маркер определяется из условия, что 2 и 3 маркеры лежат по одну сторону от прямой, проходящей через центры 1 и 5 маркеров.

Условием, что 2 точки (x_2, y_2) и (x_3, y_3) лежат по одну сторону от прямой, является то, что значения определителей

$$\begin{vmatrix} x_2 & y_2 & 1 \\ x_1 & y_1 & 1 \\ x_5 & y_5 & 1 \end{vmatrix} = a_1$$

$$\begin{vmatrix} x_3 & y_3 & 1 \\ x_1 & y_1 & 1 \\ x_5 & y_5 & 1 \end{vmatrix} = a_2$$

будут одинаковых знаков.

Где $(x_n, y_n), n = 1, 2, \dots, 5$ - координаты центра соответствующего маркера.

Если получим результаты одного знака, т.е. $a_1 \cdot a_2 > 0$ – это 1 маркер, если разных знаков – это 2 маркер.

Алгоритм восстановления изображения. Для восстановления исходного изображения анкеты строим 2 Bitmap пустой панели и панели с изображением анкеты:

```

bmp2 = new Bitmap(panel2.Width, panel2.Height);
panel1.DrawToBitmap(bmp1,rc1);

```

Проведем в bmp1 две прямые. Первая прямая (a1) проходит через центры 1 и 5 маркеров, вторая прямая (b1) проходит через центры 2 и 3 маркеров. В bmp2 также проведем две прямые. Первая прямая (a2) проходит через центры 1 и 5 маркеров. Вторая прямая (b2) проходит через центры 2 и 3 маркеров. Разобьем эти 4 прямые на достаточно большое количество отрезков, больше, чем диагональ панели. В нашем случае возьмем 1000. Обозначим точки разбиения прямых через $A1_j$, $B1_j$, $A2_j$, $B2_j$ ($0 \leq j \leq 1000$). Проведем одну прямую через точки $A1_j$ и $B1_j$ (прямая $a1_j$), а другую через точки $A2_j$ и $B2_j$ (прямая $a2_j$). Каждую из прямых $a1_j$ и $a2_j$ разобьем на 1000 отрезков точками $AB1_j$ и $AB2_j$ соответственно. Для каждой точки разбиения $AB1_j$ определим цвет ближайшего пиксела и ближайшему к точке $AB2_j$ пикселу присвоим данный цвет.

Ниже приведен соответствующий блок программы.

```

floatA1X, A1Y, A2X, A2Y, B1X, B1Y, B2X, B2Y;
floatdA1X, dA1Y, dA2X, dA2Y, dB1X, dB1Y, dB2X, dB2Y;
float XA1, YA1, XB1, YB1, XA2, YA2, XB2, YB2;
floatdXA, dYA, dXB, dYB;
int XA, YA, XB, YB;
A1X = (float)(p_A[1].X - p_A[0].X); A1Y = (float)(p_A[1].Y - p_A[0].Y);
A2X = (float)(p_A[2].X - p_A[3].X); A2Y = (float)(p_A[2].Y - p_A[3].Y);
B1X = (float)(p_B[1].X - p_B[0].X); B1Y = (float)(p_B[1].Y - p_B[0].Y);
B2X = (float)(p_B[2].X - p_B[3].X); B2Y = (float)(p_B[2].Y - p_B[3].Y);
dA1X = A1X / 1000; dA1Y = A1Y / 1000; dA2X = A2X / 1000; dA2Y = A2Y / 1000;
dB1X = B1X / 1000; dB1Y = B1Y / 1000; dB2X = B2X / 1000; dB2Y = B2Y / 1000;
bmp2 = new Bitmap(panel2.Width, panel2.Height);
for (int j1 = 0; j1 < 1000; j1++)
{
    XA1 = p_A[0].X + dA1X * j1; XB1 = p_B[0].X + dB1X * j1;
    YA1 = p_A[0].Y + dA1Y * j1; YB1 = p_B[0].Y + dB1Y * j1;
    XA2 = p_A[3].X + dA2X * j1; XB2 = p_B[3].X + dB2X * j1;
    YA2 = p_A[3].Y + dA2Y * j1; YB2 = p_B[3].Y + dB2Y * j1;
    dXA = (XA2 - XA1) / 1000; dXB = (XB2 - XB1) / 1000;
    dYA = (YA2 - YA1) / 1000; dYB = (YB2 - YB1) / 1000;
    for (int j2 = 0; j2 < 1000; j2++)
    {
        XA = (int)(XA1 + dXA * j2); XB = (int)(XB1 + dXB * j2);
        YA = (int)(YA1 + dYA * j2); YB = (int)(YB1 + dYB * j2);
        cw = bmp1.GetPixel(XA, YA); bmp2.SetPixel(XB, YB, cw);
    }
}
gr2.DrawImage(bmp2, rc2);

```

Алгоритм чтения данных анкетирования с изображения анкеты. Требуется определить отмеченные поля на изображении анкеты. Неотмеченные поля будут иметь цвет, близкий к белому. Будем считать (RED > 230, GREEN > 230, BLUE > 230). В каждом поле будем искать пиксели с цветом, где хотя бы один из RED, GREEN, BLUE меньше 230. Если таких точек в поле больше 10, считаем, что поле отмечено.

Для определения местоположения на изображении очередного поля используется экранная форма, с которой был распечатан чистый бланк анкетирования. На первом этапе изображение заполненного бланка было подогнано к размерам формы. Координаты полей на форме легко определить, и они будут соответствовать координатам поля на изображении. Программно просматривая все нужные поля на форме, находим соответствующие поля на изображении.

Заключение. Рассмотренный метод определения данных анкетирования на изображении бланка анкеты ориентирован на социологические или психологические типы анкет. Однако

данный метод может быть без труда расширен на любые типы анкет. Такой способ работы с данными анкетирования является очень перспективным ввиду высокой скорости и точности обработки данных.

ЛИТЕРАТУРА

- [1] Шилдт Г. C# 4.0: Полное руководство. – ООО «И.Д. Вильямс». – 2011. – 1056 с.
- [2] Рихтер Д. CLR via C#. Программирование на платформе Microsoft .NET FRAMEWORK 4.0 на языке C#. 4-е издание. – СПб.: Питер. – 2016, 896 с.
- [3] The Khronos OpenGL ARB Working Group. OpenGL Programming Guide. Eight Edition. The Official Guide to learning OpenGL, Version 4.3. – 2013, 986 p.
- [4] Wright R., Lipchak B., Haemel N. OpenGL Superbible: Comprehensive Tutorial and Reference. – Addison-Wesley. – 2007, 1205 p.
- [5] Хилл Ф. OpenGL Программирование компьютерной графики. – СПб.: ДиасофтЮП. – 2002, 1088 с.
- [6] Нейгел К., Ивэн Б., Глинн Д., Уотсон К. C# и платформа .NET 4 для профессионалов. – ООО «И.Д. Вильямс». – 2011, 1440 с.
- [7] Шикин Е.В., Боресков А.В. Компьютерная графика. Полигональные модели. М.: Диалог-МИФИ. – 2005, 464 с.

REFERENCES

- [1] Shildt G. C# 4.0: Polnoe rukovodstvo. – ООО «I.D. Vil'jams». – 2011. – 1056 s. (in Rus.)
- [2] Rihter D. CLR via C#. Programmirovaniye na platforme Microsoft .NET FRAMEWORK 4.0 najazyke C#. 4-e izdanie. – SPb.: Piter. – 2016, 896 s. (in Rus.)
- [3] The Khronos OpenGL ARB Working Group. OpenGL Programming Guide. Eight Edition. The Official Guide to learning OpenGL, Version 4.3. – 2013, 986 p. (in Eng.)
- [4] Wright R., Lipchak B., Haemel N. OpenGL Superbible: Comprehensive Tutorial and Reference. – Addison-Wesley. – 2007, 1205 p. (in Eng.)
- [5] Hill F. OpenGL Programmirovaniye komp'yuternoj grafiki. – SPb.: DiaSoftJuP. – 2002, 1088 s. (in Rus.)
- [6] Neygel K., Iven B., Glinn D., Uotson K. C# I platforma .NET 4 dlya professionalov. – ООО «I.D. Vilyams». – 2011, 1440 s. (in Rus.)
- [7] Shikin E. V., Boreskov A. V. Komp'yuternaja grafika. Poligonal'nye modeli. – M.: Dialog-MIFI. – 2005, 464 s. (in Rus.)

СОЦИОЛОГИЯЛЫҚ ЗЕРТТЕУЛЕР ЖҮРГІЗГЕН КЕЗДЕ САУАЛНАМА ЖҮРГІЗУ БЛАНКТЕРІН АЙЫРЫП ТАНУ

Н. Литвиненко

ҚР БҒМҒК Математика және математикалық модельдеу институты, Алматы, Қазақстан

Түйін сөздер: Bitmap, gdi+, респондент, сауалнама, графикалық бейне, пиксель.

Аннотация. Әлеуметтік және психологиялық зерттеулерде сауалнама жүргізу мәліметтерін енгізу мен өңдеу мәселесі көкейтесті болып отыр. Мәліметтерді қолмен енгізу, көп уақыт пен еңбектің шығындалуына орай, оңтайлы болып табылмайды. Онымен қоса, адам факторымен тікелей байланысты болғандықтан, бұл енгізген кезде қателердің көп санының жіберілуіне алып келуі мүмкін болғандықтан, қолмен енгізу аса сенімсіз. Бұл мақаланың мақсаты әлеуметтік және психологиялық сұрау салудың толтырылған сауалнамасын енгізудің автоматтандырылған жүйесін ұйымдастырған кезде туындайтын кейбір техникалық мәселелерді шешу болып табылады. Бірінші міндет сауалнама мәліметтерінің графикалық бейнелерін алған (фотоға түсіру, сканерлеу және т.б.) кезде туындайтын, толтырылған сауалнамалардың әртүрлі физикалық бұрмалануларымен байланысты мәселелерді шешу болып табылады. Екінші міндет сауалнамалардың графикалық бейнесіндегі белгіленген алаңдарды айырып тану болып табылады. Осы жұмыстың әдіснамасы толтырылған сауалнамалардың графикалық бейнелерін gdi+ құралдарын қолдана отырып, ең танымал форматтарда - Windows, Visual Studio 2013, C# ортасында BMP, GIF, JPG форматтарында айырып танудың техникалық амал-тәсілдерін, құралдарын жасаудан тұрады. Жұмыстың нәтижесі әлеуметтік немесе психологиялық сауалнаманың кейбір түрлерінің графикалық бейнелерін автоматтандырылған режимде немесе қол режимінде айырып тануға мүмкіндік беретін бағдарламалық өнім болып табылады. Нәтижелерді қолдану саласы, респонденттердің көп саны болған кезде, әлеуметтану немесе психология саласындағы әртүрлі ғылыми және практикалық зерттеулер болуы мүмкін.

Поступила 04.04.2016 г.