

AAM+SVM (признаки, выделенные вручную)	RBM+SVM (автоматически полученные признаки)
<ol style="list-style-type: none"> 1. Ручное выделение ключевых точек. 2. Обучение AAM по выделенным точкам. 3. Использование обученной модели для определения координат точек на новых изображениях. 4. Обучение и применением SVM на полученных координатах и метках эмоций. 	<ol style="list-style-type: none"> 1. Выделение случайных регионов изображений. 2. Обучение RBM на этих регионах и получение высокоуровневых признаков (фильтров). 3. Использование свётки начальных изображений полученными фильтрами. 4. Обучение и применение SVM на новых изображениях и метках эмоций.

Для обучения обоих методов был использован расширенный набор данных Cohn-Kanade ([7]), включающий 10708 изображений с 66-ю отмеченными ключевыми точками, 327 из которых также имеют метку одной из 7 базовых эмоций (вместе с нейтральным выражением лица). Размер вырезаемого региона был установлен в 12 пикселей (при размере полного изображения в 256 пикселей по X и 200 пикселей по Y), а количество компонент для RBM - 72. Предобучение RBM проводилось на всех изображениях, в то время как SVM в обоих подходах использовал лишь те изображения, для которых существует метка эмоции.

По результатам перекрёстной проверки (cross-validation) связка AAM+SVM показала в среднем 92% корректно классифицированных результатов, в то время как RBM+SVM - 64%. Это, однако, значительно выше чистого SVM, чья средняя точность всего чуть выше 20%.

Таким образом, несмотря на относительно низкий результат классификации с автоматически выделенными признаками, данный эксперимент явно показывает перспективность поиска альтернативных способов выделения признаков. Вероятно, наилучшим подходом будет некий гибрид RBM и AAM, на что и будет направлена будущая работа.

Список использованных источников:

1. Cootes T. F., Edwards G. J., Taylor C. J. // In Proc. European Conf. on Computer Vision. 1998. Vol. 2, P. 484–498.
2. Matthews I., Baker S. // International Journal of Computer Vision. 2004, Vol. 60 (2) P. 135 - 164.
3. Ratliff, M. S. Active appearance models for affect recognition using facial expressions: thesis by MD in CS. Wilmington, 2010.
4. Hinton, G. E., Salakhutdinov, R. R. Reducing the Dimensionality of Data with Neural Networks // Science, 2006, Vol. 313 (5786), P. 504–507.
5. Metropolis, N., Rosenbluth, A.W. Equations of State Calculations by Fast Computing Machines. Journal of Chemical Physics. 1953, Vol. 21 (6), P. 1087–1092.
6. Hinton, G. E. A Practical Guide to Training Restricted Boltzmann Machines. University of Toronto, 2010.
7. Lucy, P., Cohn, J. F., Prkachin, K. M., Solomon, P., & Matthews, I. Painful data: The UNBC-McMaster Shoulder Pain Expression Archive Database // IEEE International Conference on Automatic Face and Gesture Recognition, 2011.

ГЕНЕРАЦИЯ СУДОКУ И ОЦЕНКА КОЛИЧЕСТВА РЕШАЕМЫХ СУДОКУ

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Митьковец А.А., Савчик П.А.

Борисенко О. Ф. – канд. физ-мат. наук, доцент

Судоку – математическая головоломка с числами. В нашей работе рассматривается классическая версия судоку. Игровое поле представляет собой квадрат размером 9x9 клеток, разделённый на меньшие квадраты со стороной в 3 клетки. Цель игры – заполнить пустые ячейки так, чтобы каждый ряд, столбик и квадрат 3x3 содержали цифры от 1 до 9 по одному разу.

Постановка задачи:

Создать алгоритм (а также программу на его основе), генерирующий незаполненное поле судоку с заданным количеством изначально указанных чисел. Реализовать создание заполненного поля на основании венгерского алгоритма, создание маски для удаления при помощи «банковской последовательности».

Оценить количество всех незаполненных полей судоку, имеющих решение.

Работа состоит из нескольких частей. Краткое описание каждой из них:

1) Генерация заполненного поля судоку.

Для создания судоку по нашему алгоритму необходимо иметь полностью заполненное (по правилам судоку) поле (рис.1, а). Генерация основывается на применении венгерского метода. Поле судоку заполняется построчно. Задача о заполнении строки рассматривается как задача о назначениях, матрица стоимостей изменяется с учётом уже заполненных строк.

2) Генерация масок.

Маска представляет собой матрицу 9x9, состоящую из 0 и 1 (рис.1, б). Служит для удаления чисел из заполненного поля sudoku. Алгоритм составления маски основывается на использовании «банковской последовательности» [1]. Каждая маска имеет свой номер, что позволяет последовательно перебрать все возможные маски.

Составление масок случайным образом имеет недостаток – при генерации выборки, мы не можем быть уверены, что не получим множество похожих друг на друга масок. «Банковская последовательность» дает нам возможность сделать выборку из n-го количества действительно различных масок, что необходимо при вычислении количества всех возможных незаполненных полей sudoku (п. 4).

3) Получение незаполненного sudoku.

Реализуется путём совмещения маски и сгенерированного заполненного поля. (рис.1, в)

0 в маске означает, что соответствующая клетка в заполненном поле становится пустой, 1-- что число в соответствующей клетке не убирается.

4) Вычисление количества всех возможных незаполненных полей sudoku.

Мы применяли 1000 масок с одинаковым количеством нулей к различным заполненным полям. Затем повторяли действие, постепенно увеличивая количество нулей в масках до 60. При этом было замечено, что доля sudoku с одним решением убывает схожим образом для разных полей (рис. 2).

Имея такой график для одного заполненного поля, а также зная общее количество масок, можем получить среднее количество всех возможных sudoku, приводящих к данному заполненному полю.

А зная общее количество всех заполненных полей [2], можем получить количество всех возможных незаполненных sudoku вообще.

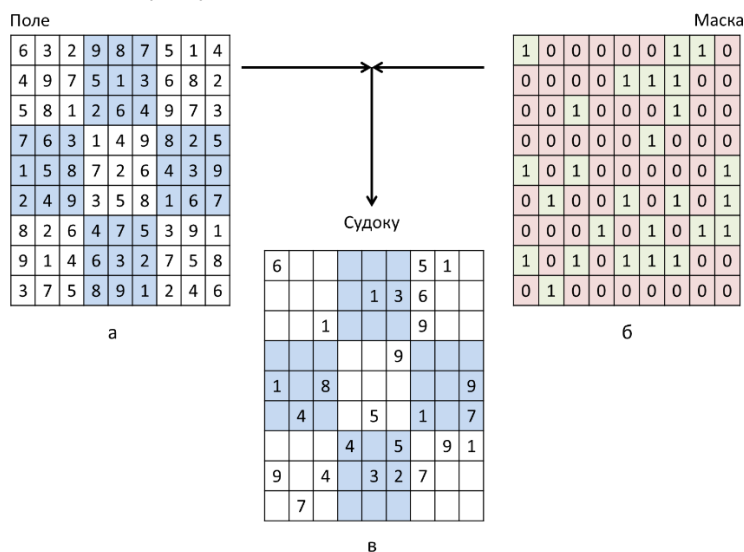
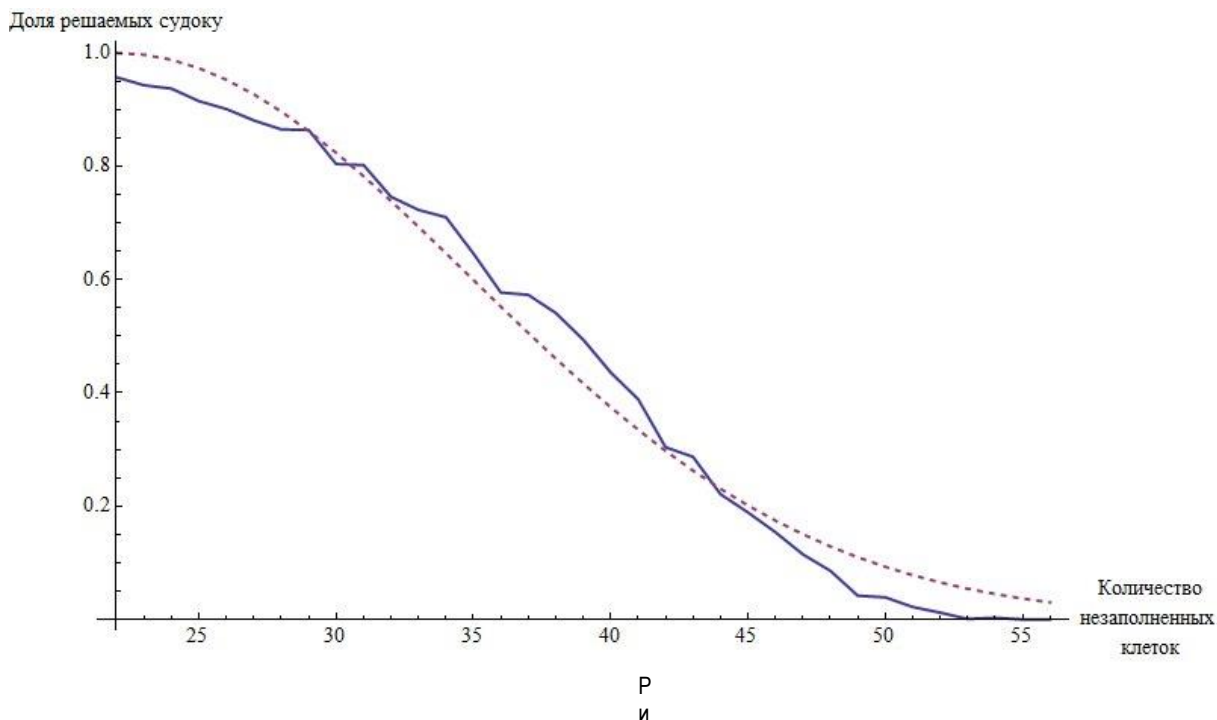


Рис. 2. Сплошной график: уменьшение отношения решаемых sudoku к нерешаемым с уменьшением начальных чисел с 60 до 25 чисел (60 – 35) для одного заполненного поля. Штриховой график: функция $y = e^{-x^2/220}$.



В результате работы получено:

- 1) Алгоритм генерации любого заполненного поля судоку из множества всех возможных полей.
- 2) Алгоритм генерации всех возможных масок (с нумерацией).
- 3) Алгоритм генерации судоку с заданным количеством изначально указанных чисел.
- 4) Среднее количество всех возможных незаполненных судоку.

Список использованных источников:

1. J. Loughry, J.I. van Hemert, L. Schoofs. Efficiently Enumerating the Subsets of a Set. - (<http://www.applied-math.org/subset.pdf>)
2. E. Russell, F. Jarvis. Sudoku enumeration: the symmetry group. (<http://www.afjarvis.staff.shef.ac.uk/sudoku/sudgroup.html>)

ИМИТАТОР ПРОСТРАНСТВЕННЫХ ПЕРЕМЕЩЕНИЙ НА ОСНОВЕ МЕХАНИЗМА ПАРАЛЛЕЛЬНОЙ КИНЕМАТИКИ С ТРЕМЯ СТЕПЕНЯМИ СВОБОДЫ

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Кекиш Н. И.

Карпович С. Е. – д-р техн. наук, профессор

Рассмотрен разработанный имитатор пространственных перемещений, который представляет собой механо-аппаратно-программный комплекс на трех независимо управляемых параллельных кинематических группах подвижных звеньев с компьютерным управлением.

Имитаторы пространственных перемещений, как правило, базируются на механизмах параллельной кинематики, приводящих в движение исполнительную платформу, на которой реализуются программируемые движения в трехмерном пространстве. Наряду с имитаторами, обеспечивающими сложные движения с шестью степенями свободы [1], которые требуют шести управляемых параллельных кинематических цепей с шестью сервоприводами, в спецтехнологическом оборудовании производства изделий электронной техники во многих случаях вполне достаточно программируемых перемещений с тремя степенями свободы: одним поступательным перемещением вдоль фиксированной оси и двух ограниченных поворотов во взаимно перпендикулярных плоскостях.

На основе анализа возможных схем реализации трехстепенных имитаторов движения была предложена кинематическая схема, по которой разработан механизм параллельной кинематики, выбраны и обособлены двигатели и система их управления, создано программное обеспечение. На механо-аппаратно-программном уровне разработанный имитатор пространственных перемещений структурно состоит из механизма параллельной кинематики, трех исполнительных двигателей, контроллера системы управления.

Контроллер системы управления перемещениями платформы представляет собой функциональный блок на базе контроллера FX3U-48 фирмы Mitsubishi с установленным программным обеспечением. Он взаимодействует через последовательный интерфейс RS-232C с управляющим компьютером, выполняет полученные команды управления движением и сообщает об ошибках, происходящих во время их выполнения. Контроллер управляет трехфазными асинхронными двигателями через инвертор FR-A700 фирмы Mitsubishi. Инвертор имеет специальные настройки для задания режимов функционирования систем, содержащих трехфазные асинхронные двигатели в составе приводов позиционирования. Преимущества такой схемы «контроллер – инвертор – двигатель» заключается в функциональной и информационной согласованности на всех уровнях: электрическом уровне сигналов; уровне управления на основе специализированных команд контроллера; уровне внешних интерфейсов и среды разработки управляющего программного обеспечения. В случае использования платформы в качестве манипулятора в технологическом процессе программа управления является именно той частью системы, с которой оператор работает наибольшее количество времени. В управляющем программном обеспечении также предусмотрены процедуры отслеживания возможных критических ситуаций и реагирования на них.

Для реализации программных движений платформы имитатора пространственных перемещений в системе управления исполнительными двигателями в управляющее программное обеспечение заложен алгоритм решения обратной задачи кинематики. Для рассматриваемого параллельного механизма обратная задача кинематики формулируется в постановке нахождения угловых координат подвижных элементов (валов) трех управляющих двигателей, которые обеспечивают заданные позицию и ориентацию подвижной платформы в пространстве [1]. Непосредственное решение может быть получено как частный случай решения обратной задачи кинематики для подобного параллельного механизма с шестью степенями свободы, рассмотренного в работе [2]. Кроме того, на основе алгоритма решения обратной задачи кинематики рассчитана граничная рабочая область параллельного механизма, которая представляет собой диапазон всевозможных положений исполнительной платформы имитатора для заданных входных значений поворотных координат трех двигателей. Граничная рабочая область в виде файлов, содержащих упорядоченный набор рассчитанных значений, была использована в аппаратно-программном комплексе системы управления для проверки допустимых положений параллельного