

Метод автокалибровки параметров управления учебным роботом с помощью библиотеки машинного зрения OpenCV

Н.О. БЕСШАПОШНИКОВ^I, А.Г. КУШНИРЕНКО^I, А.А. ЛЕВИН^{I,II}

^I Федеральное государственное учреждение «Федеральный научный центр Научно-исследовательский институт системных исследований» Российской академии наук, Москва, Россия

^{II} Федеральное государственное образовательное учреждение высшего образования «Московский государственный университет имени М.В. Ломоносова», Москва, Россия

Аннотация. В статье рассказано об опыте практического применения методов машинного зрения библиотеки OpenCV для автоматической настройки параметров управления роботом «Ползун», который используется в учебных занятиях по алгоритмике в системе программирования «ПиктоМир». Обсуждаются технологические причины погрешностей движения робота, анализируются способы решения. Предлагается алгоритм калибровки соответствующих параметров движения.

Ключевые слова: ПиктоМир, робототехника, алгоритмика, машинное зрение, OpenCV.

DOI: 10.14357/20790279180511

Еще 20-30 лет назад, использование методов машинного зрения при решении инженерных задач требовало привлечения высококвалифицированных специалистов, использования дорогостоящего оборудования и непростого в освоении программного обеспечения (рис. 1) [1].



Рис. 1. Выставочный стенд компании Automatix (сегодня подразделение Omron), демонстрирующий систему машинного зрения Autovision II в 1983 году

В наши дни применение методов машинного зрения стало доступно инженерам для решения широкого круга повседневных задач. Во многих та-

ких задачах результат может быть получен с минимальными затратами – ценой написания пары сотен строк кода с использованием свободно распространяемого программного обеспечения, исполняемого на бытовой компьютерной технике. В статье рассказано об одном опыте такого применения методов машинного зрения в инженерной практике.



Рис. 2. Радиоуправляемый робот «Ползун»

Учебный робот «Ползун»[2] (рис. 2) предназначен для иллюстрации принципа программного управления во вводных курсах алгоритмики и программирования для дошкольников и младших школьников. Для повышения наглядности при из-

ложении принципов программного управления и методов программирования в подобных вводных курсах используются как виртуальные роботы, изображаемые на экране, так и реальные роботы, перемещающиеся по полу игровой комнаты. «Ползун» – пример одного из таких реальных учебных роботов, используемый под управлением системы программирования «ПиктоМир» в курсе «Алгоритмика»[12]. Целесообразность и методика, использованная при обучении алгоритмике и программированию для дошкольников, описана в статьях [7–11, 13–19].

На первом вводном занятии «Ползун» может быть использован в режиме пульта управления, а после того, как дети овладеют методикой составления программ по управлению виртуальным роботом «Ползуном» в системе «ПиктоМир», воспитатель организует выполнение составленных детьми программ реальным «Ползуном» – двойником виртуального. Программы, составляемые на этом этапе обучения, – очень простые. Прост и сам учебный робот «Ползун». Он предназначен для движения по учебному полю (из квадратных ковриков) на полу игровой комнаты или класса и умеет выполнять только команды движения: «вперед», «повернуться налево», «повернуться направо». Число команд, получаемых роботом при выполнении одной программы, как правило, не превосходит двух трех десятков. Например, при выполнении программы на рис. 3 роботу выдается 12 команд.

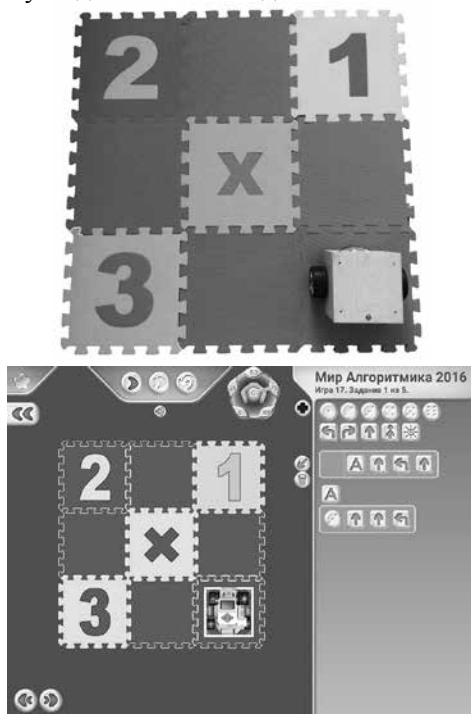


Рис. 3. Программа для реального «Ползуна» и его виртуального двойника

1. Сценарии использования робота

Внешние команды могут поступать «Ползуну» [2] по одному из двух каналов связи. Инфракрасный канал используется для режима ручного пульта управления и bluetooth – канал для режима программной генерации потока команд. Методика использования робота такова, что в обоих режимах в течение одного сеанса работы с роботом ему последовательно выдается не более двух – трех десятков команд. Это сделало возможным – в целях упрощения конструкции робота – применить систему навигации робота на игровом поле без обратной связи (dead reckoning). Робот передвигается по полю, составленному из квадратных клеток размером примерно 30x30 см. После выполнения каждой команды робот должен оказаться в центре очередной клетки, будучи ориентирован строго по одному из четырех направлений, параллельных сторонам клеток. Команда «вперед» предписывает роботу переместиться на одну клетку, команды «повернуться налево» и «повернуться направо» предписывают роботу повернуться на 90 градусов против часовой или по часовой стрелке. При выполнении команд возникают ошибки положения и ошибки ориентации: центр робота может сместиться относительно центра текущей клетки и может появиться угол между ориентацией робота и ориентацией сторон текущей клетки. При последовательном выполнении команд ошибки накапливаются.

Робот должен быть сконструирован и изготовлен так, чтобы после выполнения последовательности из двух – трех десятков команд суммарная ошибка положения не превосходила нескольких сантиметров, а ошибка угла не превосходила нескольких градусов. После изготовления опытного образца робота выяснилось, что команда «вперед» выполняется достаточно точно, а при выполнении команд поворота ошибки велики и для их уменьшения во встроенном программном обеспечении должны быть предусмотрены изменяемые извне параметры. Эти параметры необходимо настраивать индивидуально для каждого экземпляра робота по завершению его изготовления в процессе выходного контроля. В установочной партии роботов настройка параметров проводилась вручную. Оператор выдавал с инфракрасного пульта команду поворота, измерял угол, на который повернулся робот, и менял параметр во встроенном программном обеспечении. Для серийного производства подобная процедура слишком продолжительна и затратна. Поэтому был разработан стенд выходного контроля, автоматизирующий описанную выше процедуру. Кинематическая схема робота показана на рис. 4, общий вид стенда выходного контроля – на рис. 5.

Для автоматизации процедуры настройки параметров робота, нужно было научиться определять положение и ориентацию робота без участия человека. Этого оказалось возможным достичь путем написания собственного мини-кода объемом менее 300 строк с использованием библиотеки машинного зрения OpenCV [4]. Выполнение кода возможно на любом смартфоне с ОС iOS или Android.

Повышение точности выполнения команд может оказаться полезным и для роботов, снабженных системой навигации с обратной связью, поскольку позволит успешно корректировать положение робота и в случае, когда помехи сделали невозможным проведение одного или нескольких сеансов обратной связи.

2. Техническое описание робота

Опишем конструкцию робота, обращая особое внимание на те компоненты, которые могут повлиять на возникновение ошибки ориентации.

Корпус робота выполнен методом лазерной резки из фанеры с последующей механической обработкой и лакировкой поверхностей. Робот оснащен двумя двухфазными шаговыми двигателями (stepper motor) STH-36c1018 Япония Sinano, управляемыми драйверами шаговых двигателей EasyDriver A3967 [6]. Один шаг привода заставляет поворачиваться вал мотора на угол. То есть полный оборот вала двигателя состоит из 400 шагов.

На плате управления роботом установлены bluetooth модуль CC2541 с поддержкой технологии BLE [5] и микроконтроллер ATmega168, снабженный перепрограммируемым ПЗУ (EEPROM). В процессе выходного контроля индивидуально, для каждого изготавливаемого экземпляра робота подбираются оптимальные параметры управления моторами робота. Эти параметры заносятся в ПЗУ.

3. Кинематическая схема робота

Как уже говорилось, робот «Ползун» умеет выполнять следующие команды: «вперед», «повернуться направо» и «повернуться налево». Команды могут запускаться с кнопочного пульта дистанционного управления, имеющего инфракрасный интерфейс связи, а также посредством интерфейса bluetooth посылкой запросов, соответствующих каждой из команд. Перед непосредственным исполнением команд для каждого мотора устанавливается направление вращения. При выполнении команды «вперед» эти направления одинаковы, а при выполнении команд «повернуться направо» и «повернуться налево» – противоположны. Выполнение каждой из команд

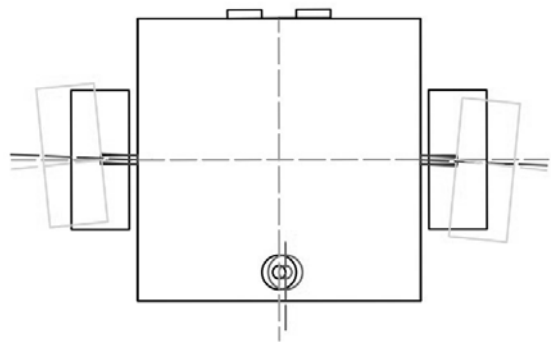


Рис. 4. Кинематическая схема робота

происходит путем посылки нужного числа шаговых импульсов, каждый из которых одновременно поступает двум моторам. Обозначим через `steps_forward`, `steps_right` и `steps_left` количество посылок, необходимых для выполнения команд «вперед», «повернуться направо» и «повернуться налево», соответственно. Эти три натуральных числа и являются параметрами настройки робота.

Конструкция робота такова, что параметр `steps_forward`, соответствующий команде движения робота «вперед», можно установить одинаковым для всех роботов данной партии (его приходится менять только при изменении размерных параметров комплектующих, используемых в данной партии). Он находится единожды для всей партии ручным подбором.

Как уже говорилось, параметры `steps_right` и `steps_left` установить едиными для всей партии невозможно. Для выполнения поворотов на 90 градусов с требуемой точностью эти параметры приходится индивидуально подбирать для каждого робота. При одном и том же значении параметра `steps_right` различные роботы могут поворачиваться на заметно отличающиеся углы по нескольким причинам:

- Отклонение центра опорного шарнирного колеса от плоскости симметрии корпуса робота приводит к перемещению точки приложения силы трения опорного колеса.
- Моторы могут быть установлены не перпендикулярно боковой поверхности корпуса робота, на разном расстоянии от двух боковых поверхностей и могут быть установлены несоосно.
- Колесо может быть установлено не симметрично относительно оси вала мотора.

Иными словами, избранная технология изготовления и сборки робота не позволяют гарантировать достаточную точность закрепления моторов ведущих колес и третьего опорного колеса на

корпусе робота. Кроме того, дополнительные погрешности в геометрию робота вносятся технологическими процессами лазерной резки корпуса робота, усыханием фанерного корпуса, процессами шлифовки и лакового покрытия корпуса, а также несовершенной геометрией резиновых покрышек колес.



Рис. 5. Общий вид стенда выходного контроля

4. Стенд выходного контроля и автоматизированной калибровки

Калибровка робота проводится на поле из стандартных пенных ковриков, которые включены в образовательный комплект, прилагаемый к роботу «Ползун» для занятий по алгоритмике [2].

Разработанная программа автоматизированной калибровки устанавливается на смартфоне с камерой высокого разрешения, снабженном модулем bluetooth с поддержкой стандарта BLE [5].

При сборке стенда смартфон закрепляется в лапке штатива робота на высоте 40 – 70 сантиметров от поверхности поля. Далее робота нужно расположить на поле под камерой смартфона и включить на смартфоне разработанное приложение для калибровки. На экране должен появиться видеопоток с камеры смартфона, на изображении

размер робота должен быть примерно в три раза меньше размера кадра. Если это не так, то необходимо подрегулировать высоту штатива.

При используемой ранее ручной настройке подача команд выполнялась с пульта по инфракрасному каналу связи, процедура настройки одного робота занимала 15-20 минут. В процессе автоматизированной настройки оператор а) включает питание робота, б) устанавливает робота на стенде в некоторое, достаточно произвольное исходное положение, в) запускает калибровочную программу на смартфоне, г) по звуковому сигналу со смартфона снимает со стенда полностью настроенного робота и выключает на нем питание. При калибровке одного робота программа выдает ему несколько десятков команд. Калибровка занимает 5–10 минут.

Тем самым процесс калибровки полностью автоматизирован, он не требует вмешательства человека после размещения на стенде и включения робота и по окончании процесса калибровки мы получаем полностью настроенного робота.

5. Алгоритм машинного распознавания положения робота

Для автоматического определения фактически выполненного роботом угла поворота при выполнении внешних команд используется компьютерный анализ изображений. А именно, с камеры смартфона снимается видеопоток, и стоп-кадры анализируются с частотой примерно 1 раз в секунду. Для анализа используется открытая библиотека машинного зрения OpenCV [4]. Исходное изображение сначала подвергается предобработке с помощью следующих методов коррекции:

```
cv::cvtColor(currentSample, gray, CV_BGR2GRAY),
cv::GaussianBlur(gray, gray, cv::Size(5,5), 0),
threshold(gray, threshold, 0, 255, CV_THRESH_BINARY | CV_THRESH_OTSU).
```

На предобработанном изображении производится поиск контуров всех прямоугольников методом

```
cv::findContours(threshold, rectContours,
CV_RETR_TREE, CV_CHAIN_APPROX_SIMPLE).
```

Это делается с целью найти вертикальные проекции двух колес и корпуса робота, которые имеют вид прямоугольников. Кроме трех искомым прямоугольных контуров – корпуса робота и двух колес – метод, «как правило», выдает лишние контуры. Для того чтобы отобрать нужные контуры и отсеять лишние, найдем прямым перебором такие пары прямоугольников, у которых отношение площадей находятся в некотором промежутке

значений $[R_1 - \varepsilon_1, R_1 + \varepsilon_1]$, где $R_1 = 10.4$ – известное, вычисляемое по документации робота, отношение площадей горизонтальных проекций корпуса робота и колеса, ε_1 – некоторая константа, определяющая разброс выбираемых прямоугольников. При изменениях высоты камеры над полем и малых изменениях угла наклона камеры изображение в кадре меняется «почти конформно», поэтому отношение площадей исследуемых объектов меняется незначительно. Выполняемый перебор пар должен найти две пары контуров: (контур корпуса, контур правого колеса) и (контур корпуса и контур левого колеса), удовлетворяющих заданному выше условию «правильности отношения площадей». Первым элементом каждой из этих пар должен быть прямоугольный контур, близкий к квадрату, поэтому в алгоритме для контроля проверяется условие «квадратности».

Получив необходимые прямоугольные контуры, найдем центр квадрата проекции корпуса робота. Эту точку возьмем за центр новой полярной системы координат. Теперь найдем вектор направления для отсчета угла поворота робота. Для этого нужно найти переднюю сторону проекции корпуса робота. Это делается с помощью определения положения инфракрасного датчика на верхней крышке корпуса робота. Этот датчик выглядит, как ограниченный окружностью темный кружок на светлой квадратной крышке корпуса робота. Поэтому на исходном изображении найдем контуры окружностей, который находится в границах квадрата корпуса робота. Затем выберем единственную окружность (вокруг изображения модуля инфракрасного датчика), как окружность, площадь которой имеет заданное отношение к площади корпуса робота (это отношение известно из документации). Соответствующее отношение должно находиться в промежутке $[R_2 - \varepsilon_2, R_2 + \varepsilon_2]$, где R_2 – отношение площади корпуса робота и места установки инфракрасного датчика, ε_2 – константа, определяющая разброс выбираемых окружностей. Параметры ε_1 и ε_2 выбраны опытным путем. «Передняя» сторона робота определяется как сторона, наиболее удаленная от центра окружности.

6. Алгоритм калибровки

При выполнении каждой команды движения устанавливается направление движения каждого из моторов, и затем на оба мотора подается одинаковое количество импульсов. Это количество индивидуально для каждой команды и записано в памяти робота. Как уже говорилось ранее, количество импульсов, используемых в команде «вперед» подбирается единожды для всей партии роботов. Алгоритм калибровки подбирает индивидуально

для каждого робота два целых числа – количество тактов работы каждого из моторов, необходимое для выполнения команд «повернуться влево» и «повернуться вправо».

Выполнение процедуры автоматической калибровки обеспечивается многократным вызовом одной единственной функцией.

Аргументом этой процедуры является число импульсов. При выполнении процедура по «выхватыванию из видеопотока» стоп-кадру описанным выше путем определяется ориентация робота относительно поля. Затем на шаговые двигатели подается s импульсов и снова определяется ориентация робота. По этим двум ориентациям и определяется угол, на который повернулся робот после подачи заданного числа импульсов на двигатели.

При помощи этой функции числа и находятся итеративной процедурой, близкой к методу половинного деления.

Наиболее точно угол, на который повернулся робот, определяется в случае почти полного оборота. Поэтому итеративным путем находится число импульсов, необходимое для совершения полного оборота, которое затем делится на 4.

Авторы благодарны Леонову А.Г. за всестороннюю поддержку и обсуждение проблематики данной статьи.

Литература

1. *Фото выставочного стенда компании Automatrix*. Автор: ArnoldReinhold – собственная работа, CC BY-SA 3.0, // [Электронный ресурс]. URL: <https://commons.wikimedia.org/w/index.php?curid=7048891> (дата обращения: 10.08.2018).
2. *ИнфоМир* // [Электронный ресурс]. URL: <http://infomir.ru/> (дата обращения: 10.08.2018).
3. *Свободно распространяемая программная система «ПиктоМир» для изучения азов программирования дошкольниками и младшими школьниками* // [Электронный ресурс]. URL: <https://piktomir.ru/> (дата обращения: 10.08.2018).
4. *Open Source Computer Vision Library*. // [Электронный ресурс]. URL: <https://opencv.org/about.html> (дата обращения: 10.08.2018).
5. *Bluetooth Smart or Version 4.0+ of the Bluetooth specification*. // [Электронный ресурс]. URL: <https://www.bluetooth.com/specifications/> (дата обращения: 10.08.2018).
6. *Easy Driver Stepper Motor Driver An Open Source Hardware Stepper Motor Drive Project*. // [Электронный ресурс]. URL: <http://www.schmalzhaus.com/EasyDriver/> (дата обращения: 10.08.2018)

7. Кушниренко А.Г., Леонов А.Г. Архитектура смешанной пиктограммно-текстовой системы программирования для дошкольников и младших школьников // Вестник кибернетики. – 2017. – Т. 28, № 4.
8. Nikita Beshaposhnikov, Anatoli Kushnirenko, Alexander Leonov // Pictomir: how and why do we teach textless programming for preschoolers, first graders and students of pedagogical universities // Proceedings of the 13th Central & Eastern European Software Engineering Conference in Russia. – CEE-SECR '17. – New York, N.Y., United States: New York, N.Y., United States, 2017.
9. Кушниренко А.Г., Леонов А.Г. «Кумир» – учебное программное обеспечение базового курса информатики // Информатика в школе. – 2017. – № 8 – С.5-10.
10. Леонов А.Г., Первин Ю.А. Программные средства представления графической и музыкальной информации в пропедевтическом курсе информатики начальной школы // Информатика в школе. – 2017. – № 7. – С. 71–75.
11. Бешапошников Н.О., Кушниренко А.Г., Леонов А.Г. Простейшие вычисления в курсе «Алгоритмика для дошкольников и младшекласников» // Сборник научных статей по итогам международной научно-практической конференции». Россия, Санкт-Петербург. – 2015. – С. 101–110.
12. Кушниренко А.Г., Леонов А.Г., Ройтберг М.А. Знакомим дошкольников младших школьников с азами алгоритмики с помощью систем ПиктоМир и Кумир // Труды НИИСИ РАН. – 2015. – Т. 5, № 1. – С. 134–137.
13. Леонов А.Г., Первин Ю.А. Качественные оценки эффективности методики обучения элементов информатики в пропедевтическом курсе // Ярославский педагогический вестник. – 2015. – № 5. – С. 92–96.
14. Леонов А.Г., Первин Ю.А. Учебные и тестовые логические задачи в пропедевтическом курсе информатики // Информатика и образование. – 2015. – № 9. – С. 32–36.
15. ПиктоМир для планшетных компьютеров как инструмент пропедевтического курса информатики / А.Г. Леонов, Н.О. Бешапошников, Д.Б. Еремин, А.Н. Дедков // Труды Большого Московского семинара по методике раннего обучения информатике. – 2014. – Т. 1, № 4.
16. Кушниренко А.Г., Леонов А.Г. Элективный курс для младшекласников «Управляем настоящим Роботом» // Новое слово в науке и практике: гипотезы и апробация результатов исследований. – 2014. – Санкт-Петербург – Изд. КультИнформПресс. - С. 105–109
17. Практикумы и исполнители. Средства интенсификации обучения / Д. В. Хачко, В. В. Яковлев, Н. М. Субоч и др. // Труды седьмой конференции Свободное программное обеспечение в высшей школе. – 2012. – С. 36–37.
18. Леонов А.Г., Кушниренко А.Г. Программирование для дошкольников младших школьников // Информатика – Первое сентября. – 2011. – № 15.
19. Леонов А.Г., Кушниренко А.Г. Структура программного обеспечения для проведения ЕГЭ по информатике и ИКТ в компьютеризированной форме // Шестая конференция Свободное программное обеспечение в высшей школе. – М.: AltLinux, 2011. – С. 21–23.

Бешапошников Никита Олегович. Федеральное государственное учреждение «Федеральный научный центр Научно-исследовательский институт системных исследований» Российской академии наук, г. Москва. Аспирант. Количество печатных работ: 5. Область научных интересов: вычислительная математика, информационные технологии. E-mail: nbeshaposhnikov@vip.niisi.ru

Кушниренко Анатолий Георгиевич. Федеральное государственное учреждение «Федеральный научный центр Научно-исследовательский институт системных исследований» Российской академии наук, г. Москва. Заведующий отделом, кандидат физико-математических наук, доцент. Количество печатных работ 80. Область научных интересов: математика, вычислительная математика, информационные технологии в образовании. E-mail: agk_@mail.ru

Левин Андрей Александрович. Федеральное государственное учреждение «Федеральный научный центр Научно-исследовательский институт системных исследований» Российской академии наук, г. Москва. Сотрудник. Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный университет имени М.В.Ломоносова». Аспирант. Количество печатных работ: 1. Область научных интересов: вычислительная математика, информационные технологии, бесконечномерный анализ. E-mail: letlevin@gmail.com

A method for auto-calibration of the educational robot control parameters using computer vision library OpenCV

N.O. Beshaposhnikov¹, A.G. Kushnirenko¹, A.A. Levin^{1,II}

¹ Federal State Institution "Scientific Research Institute for System Analysis of the Russian Academy of Sciences"

^{II} Federal State Budget Educational Institution of Higher Education M.V.Lomonosov Moscow State University

Abstract. At present, methods of machine vision are available for a wide range of tasks. This article is devoted to one experience of practical application of the OpenCV computer vision library for automatic adjustment of some control parameters of the robot "Polzun", which is used in the learning process in the programming environment "PiktoMir". We discuss the technological causes of errors in the movement of the robot and explore various solutions. A calibration algorithm for the corresponding robot motion parameters is proposed.

Keywords: *piktoMir, robotics, algorithmics, computer vision, OpenCV.*

DOI: 10.14357/20790279180511

References

1. *Photo of the Automatix machine vision system.* Author: ArnoldReinhold – own work [Rules for the citing of sources]. Available at: <https://commons.wikimedia.org/w/index.php?curid=7048891> (accessed: 10.08.2018).
2. *InfoMir* [Rules for the citing of sources]. Available at: <http://infomir.ru> (accessed: 10.08.2018).
3. *Freely distributed program system "PictoMir" for studying programming basics by preschool children and junior schoolchildren* [Rules for the citing of sources]. Available at: <https://piktomir.ru> (accessed: 10.08.2018).
4. *Open Source Computer Vision Library* [Rules for the citing of sources]. Available at: <https://opencv.org/about.html> (accessed: 10.08.2018).
5. *Bluetooth Smart or Version 4.0+ of the Bluetooth specification* [Rules for the citing of sources]. Available at: <https://www.bluetooth.com/specifications> (accessed: 10.08.2018).
6. *Easy Driver Stepper Motor Driver An Open Source Hardware Stepper Motor Drive Project* [Rules for the citing of sources]. Available at: <http://www.schmalzhaus.com/EasyDriver/> (accessed: 10.08.2018).
7. *Anatoli Kushnirenko, Alexander Leonov* 2017. The architecture of a mixed pictogram-text programming system for preschool and primary school children. 2017. Proceedings in Cybernetics. 2017. Volume 28, №4.
8. *Nikita Beshaposhnikov, Anatoli Kushnirenko, Alexander Leonov.* Pictomir: how and why do we teach textless programming for preschoolers, first graders and students of pedagogical universities. Proceedings of the 13th Central & Eastern European Software Engineering Conference in Russia. – CEE-SECR '17. – New York, N.Y., United States: New York, N.Y., United States, 2017.
9. *Anatoli Kushnirenko, Alexander Leonov* 2017. "KuMir" as a learning environment for the basic school course of computer science. Informatics in school. 2017, №8, 5 p.
10. *Alexander Leonov, Pervin Yu.A.* 2017. Software for presentation of graphic and music information in the elementary course of informatics of elementary school. Informatics in school. 2017, № 7, 71 p.
11. *Nikita Beshaposhnikov, Anatoli Kushnirenko, Alexander Leonov* 2015. The simplest calculations in the course "Algorithmics" for preschool and primary school children. Collection of scientific articles on the results of the international scientific-practical conference". Russia, Saint-Petersburg. 2015. 101 p.
12. *Anatoli Kushnirenko, Alexander Leonov, Roytberg M.A.* 2015. We propose to introduce preschool children and junior schoolchildren to the basics of "Algorithmics" using the Pictomir and Kumir systems. The journal Proceedings of NIISI RAS – 2015. – V. 5, № 1.134 p.
13. *Alexander Leonov, Pervin Yu.A.* 2015. Qualitative assessments of the effectiveness of the method of teaching the elements of computer science in the propaedeutic course. Yaroslavl Pedagogical Bulletin. – 2015. – № 5. – 92 p.
14. *Alexander Leonov, Pervin Yu.A.* 2015. Educational and test logic tasks in the propaedeutic course of computer science. Education and Informatics. – 2015. – № 9. – 32 p.
15. *Pictomir for tablet computers as a tool for the propaedeutic course of computer science.* Proceedings of the Great Moscow Seminar on the method of early learning computer science. Alexander Leonov, Nikita Beshaposhnikov, D.B. Eremin, A.N. Dedkov.. 2014.- V. 1, No. 4.
16. *Anatoli Kushnirenko, Alexander Leonov* 2014. Elective course for primary school children

- “Manage real Robot”. KultInformPress, Saint Petersburg, 2014, p 105.
17. *D.V. Chacko, V.V. Yakovlev, N.M. Suboch et al.* 2012. Virtual robots as a means of intensifying learning programming. Materials of the Seventh Conference Free software in high school. 2012. 36 pp.
18. *Alexander Leonov, Anatoli Kushnirenko* 2011. Programming for preschool children and junior schoolchildren. Informatics – «first of September». 2011. – № 15.
19. *Alexander Leonov, Anatoli Kushnirenko* 2011. The structure of software for the federal test on computer science in computerized form. Materials of the sixth conference Free software in high school. M., AltLinux, 2011. 21 p.

N.O. Beshaposhnikov. PhD student, Federal State Institution “Scientific Research Institute for System Analysis of the Russian Academy of Sciences”, Russia, 117218, Moscow, Nakhimovsky prospect, d. 36, k. 1.

E-mail: nbesshaposhnikov@vip.niisi.ru

A. G. Kushnirenko. PH.D. Senior Research Fellow, Federal State Institution “Scientific Research Institute for System Analysis of the Russian Academy of Sciences”, Russia, 117218, Moscow, Nakhimovsky prospect, d. 36, k. 1. E-mail: agk_@mail.ru

A.A. Levin. Staff, Federal State Institution “Scientific Research Institute for System Analysis of the Russian Academy of Sciences”, Russia, 117218, Moscow, Nakhimovsky prospect, d. 36, k. 1; PhD student, Federal State Budget Educational Institution of Higher Education M.V. Lomonosov Moscow State University, Russia, 119991, Moscow, GSP-1, 1 Leninskiye Gory. E-mail: letlevin@gmail.com