УДК 64.011.56
DOI: 10.25206/1813-8225-2023-186-60-66

К. В. АВЕРКОВ[1]
Д. С. МАКАШИН[1, 2]

[1]Омский государственный университет путей сообщения, г. Омск

[2]Омский государственный технический университет, г. Омск

# ИСПОЛЬЗОВАНИЕ ТЕХНИЧЕСКОГО ЗРЕНИЯ В ГИБКИХ ПРОИЗВОДСТВЕННЫХ СИСТЕМАХ ДЛЯ ОПРЕДЕЛЕНИЯ КООРДИНАТ БЕСПОРЯДОЧНО РАСПОЛОЖЕННЫХ ОБЪЕКТОВ

**Статья посвящена разработке алгоритма определения координат и ориентации объектов с помощью технического зрения с использованием языка программирования Python и библиотеки компьютерного зрения OpenCV. В статье представлена программа, позволяющая установить координаты объекта, произвольно расположенного в области видимости камеры, а также определить его ориентацию. Эти данные позволят выполнить эффективный захват объекта схватом манипулятора. В современном машиностроении задачи такого рода являются достаточно актуальными, они позволяют повысить автономность гибких производственных систем и сделать производство более безопасным.**

**Ключевые слова: гибкая производственная система, промышленный робот, техническое зрение, распознавание объектов, координаты объекта, ориентация схвата.**

**Введение.** Гибкие производственные системы (ГПС) являются высшей ступенью автоматизации. Они обеспечивают высокую производительность, точность обработки и возможность быстрой переналадки на выпуск новой продукции. Ключевыми факторами развития ГПС являются использование станков с числовым программным управлением (ЧПУ) и достижения современной промышленной робототехники. Промышленный робот, обслуживающий станок с ЧПУ, позволяет полностью исключить человека из технологического процесса, устранить любые возможные простои, связанные с человеческим фактором, и обеспечить безопасность [1, 2].

Целью данной работы является попытка разработать программу, позволяющую промышленному роботу находить объекты манипулирования и определять их координаты с помощью технического зрения. Результаты могут быть применены в реальных промышленных условиях на роботизированных участках, работа на которых связана с обнаружением объектов в рабочей зоне промышленного робота.

Промышленные роботы, оснащенные системами технического зрения, способны решать самые разнообразные задачи [3–7], в том числе с использованием искусственного интеллекта [8]. Однако, несмотря на кажущуюся простоту, загрузка станка деталями является достаточно сложной задачей даже для современного промышленного робота. Одна из наиболее сложных проблем — правильная ориентация заготовки при установке в приспособление, а также ориентация готовой детали при извлечении ее.

Захваты современных промышленных роботов специализированы под детали определенных типов. Чаще всего применяются клещевые захваты (рис. 1).

Ширина раскрытия захвата лимитирована и должна соответствовать определенным размерам объекта манипулирования. Рабочие поверхности схватов, как правило, адаптированы под детали определенной формы. Например, для деталей цилиндрической формы могут использоваться призматические или полукруглые рабочие поверхности. Для надежного захвата объекта манипулирования схват должен находиться в определенном положении, то есть быть ориентированным относительно объекта манипулирования.

К примеру, для надежного захвата детали типа вал рабочие плоскости схвата должны быть параллельны образующим вала, иначе захват либо не произойдет вообще, либо положение вала в схвате будет неопределенным, так как во время срабатывания схвата вал повернется на некоторый угол. Поэтому требуется развернуть схват таким образом, чтобы плоскость движения рабочих элементов была перпендикулярна оси вала, а сам захват

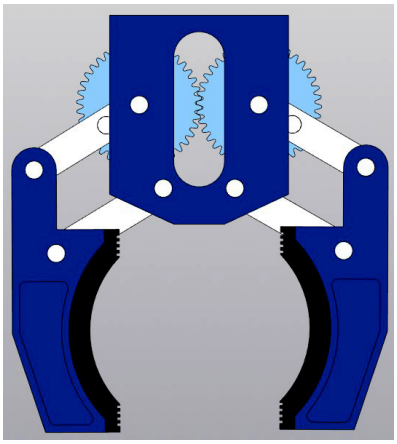


**Рис 1. Конструкция типичного клещевого захватного устройства промышленного робота**

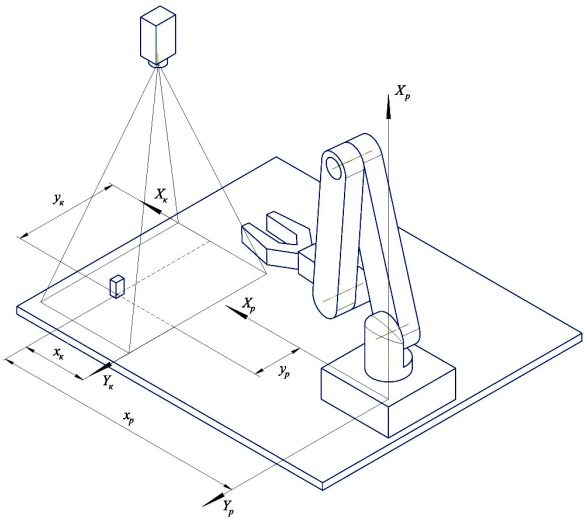



**Рис. 2. Системы координат камеры и робота**

должен происходить в строго определенном месте детали, характеризуемом определенным диаметром.

**Реализация технического зрения в гибких производственных системах для определения координат беспорядочно расположенных объектов.** Проблема точного ориентирования детали может быть решена одним из трех способов.

Во-первых, определенное положение детали может быть придано вручную, то есть перед началом цикла обработки рабочий вручную выкладывает детали в определенном положении на определенной позиции в пределах рабочей области промышленного робота. Для этого могут использоваться специальные палеты или иные приспособления, исключающие случайное смещение детали. Данный способ является самым простым, но наименее приемлемым. Наличие ручных операций, связанных с перекладыванием детали, снижает степень автоматизации и производительность, добавляет человеческий фактор.

Другим способом является применение специальных вибрационных бункеров, обеспечивающих быструю и надежную ориентацию деталей. Детали и заготовки могут быть загружены в бункер навалом, правильная ориентация происходит автоматически. Данный способ является надежным и высокотехнологичным, однако он применим только к деталям и заготовками небольшого размера. Кроме того, вибрационные бункеры занимают существенное пространство в рабочей зоне, а взаимное соударение деталей при вибрации может привести к снижению качества поверхностей.

Наиболее прогрессивным на сегодняшний день способом точной ориентации деталей и заготовок, обрабатываемых на станках с ЧПУ, является использование компьютерного зрения.

Использование компьютерного зрения делает работу промышленного робота и станка с ЧПУ полностью автономной. Оно полностью согласуется с парадигмами «Индустрии 4.0».

Самым распространенным на сегодняшний день средством поддержки компьютерного зрения является OpenCV — библиотека компьютерного зрения с открытым исходным кодом. Данная библиотека содержит в себе несколько тысяч алгоритмов, позволяющих установить не только ориентацию деталей и определение ее координат, но и распознавание, контроль и обеспечение безопасности [9]. Использование данной библиотеки приближает возможности промышленного робота к возможностям искусственного интеллекта. При этом в наилучшей степени возможности библиотеки раскрываются в сочетании с применением языка программирования Python.

Для успешного использования промышленного робота в составе ГПС необходима стыковка систем координат промышленного робота и станка ЧПУ. При использовании компьютерного зрения добавляется система координат камеры [10]. Минимальные погрешности при перемещении достигаются при условии параллельности осей координат всех трех систем.

Начало системы координат промышленного робота связано с его основанием. Начало системы координат камеры связано с одним из углов кадра (рис. 2).

В техническом зрении объекты наиболее часто идентифицируют с помощью цвета и формы, причем автоматизировать обнаружение по цвету проще. При этом используют цветовые маски, накладываемые на изображение. Одной из проблем является фильтрация изображения, так как оно содержит множество объектов такого же или близкого оттенка, что и объект манипулирования. В предложенной программе выполнена фильтрация контуров по размерам. Определив контуры границ объекта, легко могут быть вычислены координаты его центра.

Таким образом, алгоритм работы программы должен включать следующие шаги:

1) получение изображения;
2) наложение цветовой маски;
3) определение контуров;
4) фильтрация «лишних» контуров;
5) определение координат объектов в системе координат камеры и их ориентации в пространстве (это необходимо, чтобы установить захват робота в требуемое положение;
6) пересчет координат в систему координат робота;
7) передача полученных координат в систему управления робота и выполнение операций.

Ниже представлен код программы, позволяющий обнаруживать, вычислять координаты и необходимые углы разворота захвата для манипулирования объектами. Код программы имеет необходимые

пояснения, кроме того, для удобства программа разбита на отдельные блоки и дано пояснение к каждому блоку.

Сначала нужно выполнить импорт библиотек. Работа с компьютерным зрением требует библиотек: OpenCV (собственно библиотека компьютерного зрения), NumPY (линейная алгебра). Также понадобится библиотека Treading (обработка потоков данных, например, видеопотоков) и DobotDllType (управление роботом).

```
# Импорт необходимых библиотек
import threading
import DobotDllType as dType
import cv2
import numpy as np
```

Ниже представлен код подключения робота к ПК и программирование операции «Выход в 0».

```
# Подключение робота
CON_STR = {dType.DobotConnect.
DobotConnect_NoError:
"DobotConnect_NoError",
dType.DobotConnect.DobotConnect_NotFound:
"DobotConnect_NotFound",
dType.DobotConnect.DobotConnect_Occupied:
    "DobotConnect_Occupied"}

api = dType.load()

state = dType.ConnectDobot(api, "", 115200)[0]
print("Connect status:",CON_STR[state])

# Вывод робота в "0"
dType.SetHOMECmdEx(api, 0, 1)
```

Все движения робота будут программироваться одной функцией, содержащей в себе команды, связанные с перемещением манипулятора робота к объекту манипулирования, его захватом и перемещением в нужную точку. После выполнения каждого физического действия с помощью метода dType.dSleep(500) обеспечивается пауза длительностью 0,5 с.

```
# Функция, программирующая манипуляции
объекта роботом
def robot_moove(x, y, angle, y_end):
# Выбор схвата в качестве рабочего инструмента
dType.SetEndEffectorParamsEx(api, 59.7, 0, 0, 1)
# Безопасная высота 30 мм
dType.SetPTPJumpParamsEx(api,30,100,1)
# Открыть захват
dType.SetEndEffectorGripperEx(api, 1, 0)
dType.dSleep(500)
# Переместиться к объекту
dType.SetPTPCmdEx(api, 0, x, y, -21.8, angle, 1)
dType.dSleep(500)
# Захватить объект
dType.SetEndEffectorGripperEx(api, 1, 1)
dType.dSleep(500)
# Переместить объект
dType.SetPTPCmdEx(api, 0, 145.6, y_end, -24.2,
0, 1)
dType.dSleep(500)
# Отпустить объект
dType.SetEndEffectorGripperEx(api, 1, 0)
dType.dSleep(500)
```

Представленный ниже код обеспечивает нахождение объекта манипулирования на столе, определение его координат и угла поворота. Поиск объекта выполняется по цвету (в данном случае объект синего цвета), для этого на полученное с камеры изображение «накладывается» цветовая маска, заданная в цветовом пространстве HSV. Маска выделяет нужный объект. Далее найденный объект обводится в прямоугольный контур.

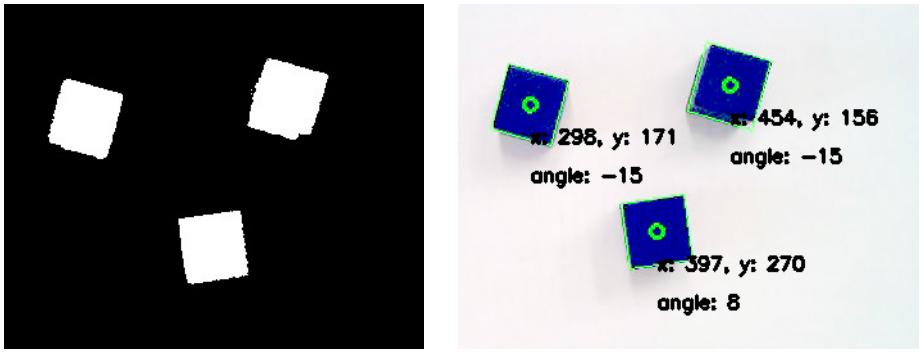
Поскольку в кадре могут быть другие объекты синего цвета, необходимо выполнить фильтрацию найденных контуров. Отсеивание контуров выполняется по размеру, то есть отсеиваются контуры размером меньше заданного.

Переменная cl_pr обеспечивает прерывание видеопотока после нахождения объектов. Прерывание необходимо, так как, перемещаясь, схват манипулятора попадает в поле зрения камеры и нарушает видимость.

Также в данном блоке определяются координаты центра объекта манипулирования.

```
# Создание пустого списка для хранения координат объекта
Obj = []
# Связывание видеопотока с камеры с переменной capImg
capImg = cv2.VideoCapture(1)

while (capImg.isOpened()):
# Считывание кадров в переменную frame
ret, frame = capImg.read()
# Переменная для прерывания показа текущего кадра
cl_pr = False
# Запуск цикла со «стоп-кадром»
while True:
# Перевод изображения в цветовое пространство HSV
frame_hsv = cv2.cvtColor(frame,
cv2.COLOR_BGR2HSV)
# Установка границ цветового диапазона
low_blue = np.array([100, 80, 30], dtype = "uint8")
high_blue = np.array([140, 255, 255], dtype =
"uint8")
# Накладывание цветовой маски
blue_mask = cv2.inRange(frame_hsv, low_blue,
high_blue)
# Вывод на экран изображения с наложенной маской
cv2.imshow("Objects", blue_mask)
# Поиск контуров и запись их в переменную contours
contours, hierarchy = cv2.findContours(blue_
mask,
cv2.RETR_LIST,
cv2.CHAIN_APPROX_SIMPLE)
# Перебор найденных контуров
for icontour in contours:
# Поиск прямоугольных контуров
rect = cv2.minAreaRect(icontour)
# Вычисление площади контура для выполнения
# фильтрации
area = int(rect[1][0]*rect[1][1])
# Фильтрация контуров
if area > 1500:
# Поиск вершин прямоугольника
box = cv2.boxPoints(rect)
# Округление координат вершин
```




а) б)

**Рис. 3. Представление, полученное в программе:**
**а) изображение, полученное при наложении цветовой маски;**
**б) отображение координат и угла ориентации детали**

```
box = np.int0(box)
# Отрисовка прямоугольного контура
cv2.drawContours(frame, [box], -1, (0, 255, 0), 1)
# Определение координат центров контуров в СК
# камеры
x_cam = rect[0][1]
y_cam = rect[0][0]
```

Следует учесть, что координаты определены в системе координат камеры, начало которой находится в левом верхнем углу кадра. Сами координаты будут получены в пикселях.

Кроме определения координат, необходимо также определить угол поворота объекта. Перед выполнением захвата схват манипулятора разворачивается под тем же углом.

```
# Определение угла поворота схвата
if rect[2] <= 45:
angle = int(-rect[2])
else:
angle = int(90-rect[2])
# Отрисовка точки в центре конутра
cv2.circle(frame, (int(rect[0][0]), int(rect[0][1])),
5, (0, 255, 0), 2)
# Указание координат и угла на экране
cv2.putText(frame, "x: %d, y: %d" % (rect[0][0], rect[0][1]),
(int(rect[0][0]), int(rect[0][1])+30),
cv2.FONT_HERSHEY_SIMPLEX,
0.5, (0, 255, 0), 2)
cv2.putText(frame, "angle: %d" % (angle),
(int(rect[0][0]), int(rect[0][1])+60),
cv2.FONT_HERSHEY_SIMPLEX,
0.5, (0, 255, 0), 2)

# Запись координат и угла в список
Obj_tuple = (x_cam, y_cam, angle)
Obj.append(Obj_tuple)

# Вывод найденных контуров
cv2.imshow("Camera", frame)

# Ожидание закрытия видеопотока нажатием клавиши # «q»
key_press = cv2.waitKey(0)
if key_press == ord('n'):
break
elif key_press == ord('q'):
cl_pr = True
break
if cl_pr:
break
```

Далее необходимо перевести найденные координаты в систему координат робота (рис 2). Для этого необходимо учесть разницу в положении начал координат и тот факт, что полученные программой координаты выражены в пикселях. Таким образом, нужно пересчитать координаты из СК камеры в СК робота и перевести координаты в миллиметры.

```
# Определение координат объектов
# Перебор элементов списка Obj
for i in range(len(Obj)):
x_rob = int(Obj[i][0]*0.472 + 182.2)
y_rob = int(Obj[i][1]*0.472 - 165.2)
angle = int(Obj[i][2])
y_end = -150 + 25*i
```

Далее в программе выполняется перемещение объекта с помощью функции robot_moove, отвод робота на безопасное расстояние и выключение робота. На экран выводятся координаты объекта манипулирования и угол поворота схвата (рис. 3).

```
# Отработка движений роботом
robot_moove(x_rob+0.2*i, y_rob, angle, y_end)

# Поднять захват на 30 мм
dType.SetPTPCmdEx(api, 7, 0, 0, 30, 0, 1)
dType.dSleep(500)
# Выключить захват
dType.SetEndEffectorGripperEx(api, 0, 0)

capImg.release()
cv2.destroyAllWindows()

print("x =", x_cam, "y =", y_cam, "angle =", angle)
# Вывод списка с координатами
print(Obj)
```

Из приведенных рисунков (рис. 3а и б) видно, что определение координат объекта с помощью технического зрения является достаточно надежным методом, вероятность возникновения ошибки здесь низкая. Кроме того, несомненно, данный способ позволит не только облегчить труд рабочего, но и упростит подготовку управляющей программы для робота. Отпадет необходимость в «ручном» вычислении координат объектов.

**Заключение.** Полная автоматизация технологического процесса является одной из главных целей Индустрии 4.0. На сегодняшний день автоматизация процессов обработки достигла очень высокого уровня за счет повсеместного использования систем числового программного управления. Основной ресурс дальнейшего повышения уровня автоматизации — вспомогательные процессы, связанные с перемещением деталей и заготовок, нанесением меток, заменой изношенного инструмента и пр.

Преимуществами предложенного алгоритма является:

1. Эффективное применение фильтрации контуров.
2. Надежное определение координат.
3. Простота реализации.

Использование компьютерного зрения позволяет обеспечить высокие показатели производительности и качества, убрать человеческий фактор, оградить человека от опасных и вредных производственных факторов, повысить культуру производства.

**Библиографический список**

1. Бахрамов Н. М., Храмов В. В. Проблемы использования технического зрения роботов-ассистентов в контексте цифровой трансформации // Интеллектуальные ресурсы — региональному развитию. 2022. № 2. С. 163—167.

2. Андросов А. Ю., Горшков А. А., Луцков Ю. И. Размещение системы технического зрения на манипуляторе мобильного робота // Известия ТулГУ. Технические науки. 2014. № 11-1. С. 418—426.

3. Гуров В. С., Колодько Г. Н., Костяшкин Л. Н. [и др.]. Обработка изображений в авиационных системах технического зрения: моногр. / под ред. Л. Н. Костяшкина, М. Б. Никифорова. Москва: Физматлит, 2016. 240 с.

4. Котляр Д. И., Ломанов А. Н., Корнейчук В. С. Применение технического зрения для сканирования кромки лопатки в процессе ремонта // Вестник Череповецкого государственного университета. 2022. № 1 (106). С. 42—54. DOI: 10.23859/1994-0637-2022-1-106-4.

5. Юдин Д. А., Магергут В. З. Программный комплекс системы технического зрения для оценки состояния процесса обжига // Программные продукты и системы. 2013. № 2. С. 257—262.

6. Бублик Д. А., Кононенко Р. В. К вопросу контроля геометрических параметров объектов двойной кривизны больших размеров при помощи системы технического зрения // Вестник Иркутского государственного технического университета. 2019. № 4. С. 670—677. DOI: 10.21285/1814-3520-2019-4-670-677.

7. Блохин К. О., Матлахов В. П., Хандожко В. А. Наукоемкая технология контроля качества керамической плитки с использованием технического зрения // Наукоемкие технологии в машиностроении. 2018. № 2. С. 27—34.

8. Прукс В. Э. Пример системы технического зрения на основе нейросетевого классификатора // Вестник Балтийского федерального университета им. И. Канта. 2012. № 10. С. 99—103.

9. Пелевин Е. Е., Балясный С. В. Оптимальные алгоритмы выделения контуров изображения в системе технического зрения // Juvenis scientia. 2016. № 6. С. 6—8. DOI: 10.15643/jscientia.2016.6.195.

10. Сокол Е. Е. Концепция системы технического зрения для контроля качества и сортировки готовых деталей // Молодежный вестник ИрГТУ. 2021. № 3. С. 20—25.

**АВЕРКОВ Константин Васильевич,** кандидат технических наук, доцент (Россия), доцент кафедры «Технологии транспортного машиностроения и ремонта подвижного состава» Омского государственного университета путей сообщения (ОмГУПС), г. Омск.
SPIN-код: 8407-0488
AuthorID (РИНЦ): 674150
AuthorID (SCOPUS):55257667700
Адрес для переписки: averok@yandex.ru

**МАКАШИН Дмитрий Сергеевич,** кандидат технических наук, доцент кафедры «Металлорежущие станки и инструменты» Машиностроительного института Омского государственного технического университета, г. Омск; доцент кафедры «Технологии транспортного машиностроения и ремонта подвижного состава» ОмГУПС, г. Омск.
SPIN-код: 1763-1883
AuthorID (РИНЦ): 926848
AuthorID (SCOPUS): 57203642272
Адрес для переписки: dima.makashin@gmail.com

**Для цитирования**

Аверков К. В., Макашин Д. С. Использование технического зрения в гибких производственных системах для определения координат беспорядочно расположенных объектов // Омский научный вестник. 2023. № 2 (186). С. 60—66. DOI: 10.25206/1813-8225-2023-186-60-66.

**Статья поступила в редакцию 23.01.2023 г.**

UDC 64.011.56
DOI: 10.25206/1813-8225-2023-186-60-66

K. V. AVERKOV[1]
D. S. MAKASHIN[1, 2]

[1]Omsk State Transport University, Omsk, Russia
[2]Omsk State Technical University, Omsk, Russia



# THE USE OF TECHNICAL VISION IN FLEXIBLE PRODUCTION SYSTEMS TO DETERMINE COORDINATES OF RANDOMLY LOCATED OBJECTS

**The article is devoted to the development of an object recognition algorithm using technical vision using Python and the OpenCV computer vision library. The article presents a program that allows you to set the coordinates of an object arbitrarily located in the field of view of the camera, as well as determine its orientation. This data will allow you to perform an effective capture of the object by the grip of the manipulator. In modern mechanical engineering, tasks of this kind are quite relevant, they make it possible to increase the autonomy of flexible production systems and make production safer.**

**Keywords: flexible production system, industrial robot, technical vision, object recognition, object coordinates, orientation of the grip.**

**References**

1. Bakhramov N. M., Khramov V. V. Problemy ispol'zovaniya tekhnicheskogo zreniya robotov-assistentov v kontekste tsifrovoy transformatsii [Directions for improving and integrating the robot vision system for integrated data mining] // Intellektual'nyye resursy — regional'nomu razvitiyu. *Intellectual Resources for Regional Development.* 2022. No. 2. P. 163–167. (In Russ.).

2. Androsov A. Yu., Gorshkov A. A., Lutskov Yu. I. Razmeshcheniye sistemy tekhnicheskogo zreniya na manipulyatore mobil'nogo robota [A placement of the vision system on the mobile robot manipulator] // Izvestiya TulGU. Tekhnicheskiye nauki. *Izvestiya TulGU. Technical Sciences.* 2014. No. 11-1. P. 418–426. (In Russ.).

3. Gurov V. S., Kolod'ko G. N., Kostyashkin L. N. [et al.]. Obrabotka izobrazheniy v aviatsionnykh sistemakh tekhnicheskogo zreniya [Image processing in aviation vision systems] / ed. by L. N. Kostyashkina, M. B. Nikiforova. Moscow, 2016. 240 p. (In Russ.).

4. Kotlyar D. I., Lomanov A. N., Korneychuk V. S. Primeneniye tekhnicheskogo zreniya dlya skanirovaniya kromki lopatki v protsesse remonta [The application of computer vision for scanning the blade edge during repair procedures // Vestnik Cherepovetskogo gosudarstvennogo universiteta. *Cherepovets State University Bulletin.* 2022. No. 1 (106). P. 42–54. DOI: 10.23859/1994-0637-2022-1-106-4 (In Russ.).

5. Yudin D. A., Magergut V. Z. Programmnyy kompleks sistemy tekhnicheskogo zreniya dlya otsenki sostoyaniya protsessa obzhiga [Computer vision system software for assessment of firing process] // Programmnyye produkty i sistemy. *Software Products and Systems.* 2013. No. 2. P. 257–262. (In Russ.).

6. Bublik D. A., Kononenko R. V. K voprosu kontrolya geometricheskikh parametrov ob"yektov dvoynoy krivizny bol'shikh razmerov pri pomoshchi sistemy tekhnicheskogo zreniya [To control of geometrical parameters of large size double curvature objects using computer vision system] // Vestnik Irkutskogo gosudarstvennogo tekhnicheskogo universiteta. *Proceedings of Irkutsk State Technical University.* 2019. No. 4. P. 670–677. DOI: 10.21285/1814-3520-2019-4-670-677. (In Russ.).

7. Blokhin K. O., Matlakhov V. P., Khandozhko V. A. Naukoyemkaya tekhnologiya kontrolya kachestva keramicheskoy plitki s ispol'zovaniyem tekhnicheskogo zreniya [Science intensive technology for ceramic tile quality control using computer vision] // Naukoyemkiye tekhnologii v mashinostroyenii. *Science-based Technology in Mechanical Engineering.* 2018. No. 2. P. 27–34. (In Russ.).

8. Pruks V. E. Primer sistemy tekhnicheskogo zreniya na osnove neyrosetevogo klassifikatora [Example of a vision system based on a neural network] // Vestnik Baltiyskogo federal'nogo universiteta im. I. Kanta. *Bulletin of I. Kant Baltic Federal University.* 2012. No. 10. P. 99–103. (In Russ.).

9. Pelevin E. E., Balyasnyy S. V. Optimal'nyye algoritmy vydeleniya konturov izobrazheniya v sisteme tekhnicheskogo zreniya [Optimal algorithm of edge detection within the system of computer vision] // Juvenis scientia. *Juvenis Scientia.* 2016. No. 6. P. 6–8. DOI: 10.15643/jscientia.2016.6.195. (In Russ.).

10. Sokol E. E. Kontseptsiya sistemy tekhnicheskogo zreniya dlya kontrolya kachestva i sortirovki gotovykh detaley [Vision system concept for quality control and sorting of finished parts] // Molodezhnyy vestnik IrGTU. *Young Researchers Journal of ISTU.* 2021. No. 3. P. 20–25. (In Russ.).

**AVERKOV Konstantin Vasilyevich,** Candidate of Technical Sciences, Associate Professor, Associate Professor of Technologies of Transport Engineering and Rolling Stock Repair Department, Omsk State Transport University (OSTU), Omsk.

SPIN-code: 8407-0488
AuthorID (RSCI): 674150
AuthorID (SCOPUS):55257667700
Correspondence address: averok@yandex.ru
**MAKASHIN Dmitriy Sergeyevich,** Candidate of Technical Sciences, Associate Professor of Metal-cutting Machines and Tools Department, Mechanical Engineering Institute, Omsk State Technical University, Omsk; Associate Professor of Technologies of Transport Engineering and Rolling Stock Repair Department, OSTU, Omsk.
SPIN-code: 1763-1883
AuthorID (RSCI): 926848
AuthorID (SCOPUS): 57203642272
Correspondence address: dima.makashin@gmail.com

**For citations**

Averkov K. V., Makashin D. S. The use of technical vision in flexible production systems to determine coordinates of randomly located objects // Omsk Scientific Bulletin. 2023. No. 2 (186). P. 60–66. DOI: 10.25206/1813-8225-2023-186-60-66.

**Received January 23, 2023.**