Министерство образования и науки РФ
ФГБОУ ВО «Тверской государственный университет»
Факультет прикладной математики и кибернетики
Направление «Прикладная математика и информатика»
Профиль «Системный анализ, исследование операций и управление»

ВЫПУСКНАЯ РАБОТА БАКАЛАВРА

Оценка положения и ориентации детали с помощью вебкамеры

Автор:
Наливайко Ярослава Олеговна

Научный руководитель:
к.ф.-м.н.
Бобышев Владимир Николаевич

Допущена к защите:

Руководитель ООП:

_____/А.В. Язенин/

(подпись, дата)

Заведующий кафедрой: математической статистики и системного анализа

(наименование)

_____/В.Н. Михно/

(подпись, дата)

Тверь 2018

Содержание

Список основных сокращений

СТЗ система технического зрения

ЧПУ численное программное управление

Введение

Прогресс науки и техники затрагивает каждую область деятельности человека, привнося с собой совершенно новые методы и усовершенствования старых. Особенно это заметно в области производства, в которой компании стараются сделать свой товар как можно более привлекательным для покупателя, улучшая его качество и понижая его цену путем повышения эффективности процесса производства. В настоящее время ученые повсеместно формализуют операции физического и интеллектуального труда для их последующей механизации и автоматизации.

На современных предприятиях с широкой номенклатурой продукции зачастую используются управляемые компьютером станки, незаменимые в тех случаях, когда предприятие нацелено на точность изготовления товаров и их повторяемость. Подобные станки выполняют именно те команды, которые были в них загружены, например, переместиться в заданное место на рабочей платформе с заданной скоростью. При этом предполагается, что деталь находится в конкретном месте платформы и имеет определенный наклон относительно осей станка. Даже малейшее отклонение от ожидаемого может привести в результате к большим ошибкам. Поэтому очень важно для каждой детали определить эти отклонения и перерасчитать под них программу. К сожалению, эта задача невыполнима для неквалифицированного работника и требует специального дорогостоящего оборудования.

На производстве для решения этой задачи зачастую используется контактный метод, основанный на том, что чувствительный элемент прибора приводится в контакт с объектом измерения. Однако при применении этого метода объект из легко деформируемого материала будет поврежден, поэтому требуется подход, получающий информацию бесконтактным способом. Это приводит нас к системам технического зрения (СТЗ), разраба-

тываемым специально для выполнения подобных задач.

В связи с развитием цифровых камер и вычислительной техники, являющихся важнейшими компонентами СТЗ, до уровня доступности широкому кругу пользователей стали возможны фиксация изображений нужного качества и их обработка для получения необходимой информации в режиме реального времени. Автоматизация технологического процесса, основной целью применения которой является повышение эффективности производства и качества продукции, широко используется для проверки на наличие дефектов и оценки положения и ориентации детали.

Для оценки координат и ориентации детали проще всего использовать изображения высокого разрешения, на которых исследуемая деталь помещется целиком, и применять алгоритмы, основанные на выделении контуров. Подобные изображения можно получить с камеры, расположенной достаточно высоко над рабочей поверхностью и обладающей высоким разрешением. Однако эти условия не всегда выполнимы, в частности, если рассматриваемая деталь чересчур большая и/или бюджет не допускает приобретения столь дорогостоящего оборудования.

В данной работе рассматривается подход, в котором используется недорогая камера (например, вебкамера), предназначенная скорее для фиксирования изображений в реальном времени, а не высоком качестве и расположенная сравнительно близко к исследуемому объекту. Этот подход основан на обработке не одного статичного изображения с высоким разрешением, а последовательности изображений отдельных частей объекта, которая должна быть построена таким образом, чтобы извлечь как можно больше необходимой информации и достичь лучшей точности.

Целью данной работы является разработка алгоритма, обеспечивающего оценку положения и ориентации детали, а также его программная реализация на языке программирования Java.

Данная выпускная квалификационная работа состоит из введения, 4

глав, заключения и списка литературы.

В первой главе перечисляются составляющие используемых СТЗ и описываются изучаемые детали, а также приводятся примеры подходящих и неподходящих систем.

Во второй главе описывается алгоритм нахождения на изображении границы между деталью и фоном и дается определение оптическому потоку и преобразованию Хафа, широко используемым в данной работе. Рассматриваются некоторые условия, при которых результат этого алгоритма может оказаться неверным, и метод определения достоверности результата.

В третьей главе перечисляются необходимые для работы данные о СТЗ, а также описываются алгоритмы получения тех из них, которые вычисляются, а не предоставляются оператором.

В четвертой главе доказывается, что для определения ориентации детали необходимо исследовать ее стороны по всей длине, и излагается алгоритм этого исследования, в котором учитывается возможность недостоверных данных от СТЗ. Описываются результаты статистических наблюдений. Приводится и разъясняется пример результата реальной работы программы. Описывается метод определения положения детали.

1 Требования к СТЗ и изучаемым деталям

Система технического зрения - это специальная система устройств, предназначенная для фиксации изображений, их последующей обработки и извлечения из них информации. Есть множество видов СТЗ, тем не менее каждая из них содержит три следующих основных элемента: источник энергии, камеру и процессор.

Алгоритм, описанный в данной работе, предназначен для работы на технологическом оборудованиии с числовым программным управлением (ЧПУ), то есть на оборудовании, действиями которого руководит компьютеризированная система. В зависимости от загруженной в них программы они выполняют те или иные действия. Примерами оборудования с ЧПУ являются станки (к примеру, фрезерные) и 3D принтеры.

Отказ от камер с высоким разрешением, позволяющих получить изображения, на которых объекты помещаются полностью, смягчает требования, выставляемые по отношению к устройствам захвата изображений, но в то же время приводит к другому вопросу: какими компонентами должна обладать СТЗ, чтобы отвечать условиям поставленной задачи?

1.1 Состав СТЗ

После анализа оборудования, подходящего описанному в данной работе алгоритму, были выделены следующие элементы, которые должна включать в себя СТЗ:

1. Камера
2. Система освещения
3. Технология позиционирования в трехмерном пространстве (например, оборудование с ЧПУ)

4. Источник энергии

5. Компьютер

6. Кабели связи или механизм беспроводной связи

7. Программное обеспечение

1.1.1 Пример удовлетворяющей требованиям СТЗ

На рисунке 1 представлен пример устройства подходящей СТЗ на основе 3D принтера. Энергию она получает от бытовой электрической сети.

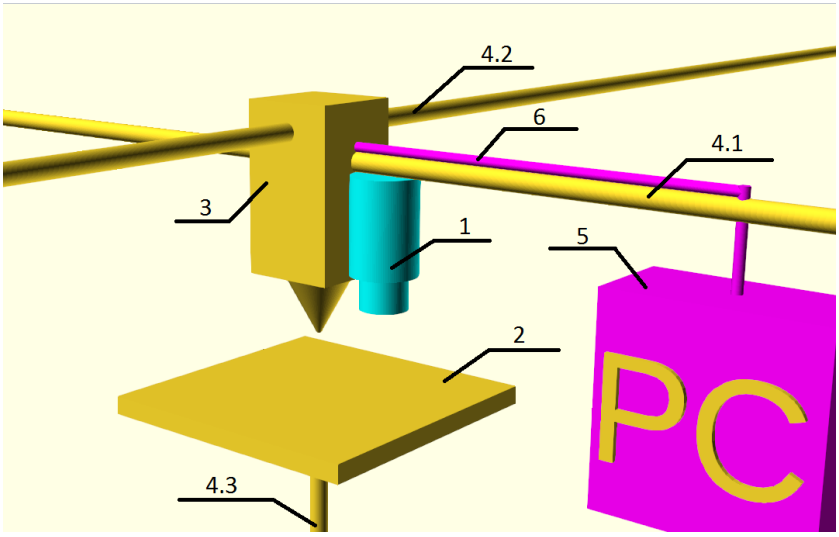

Рис. 1: Удовлетворяющая требованиям СТЗ

В состав данной системы входят следующие части:

1 — камера, включающая в себя систему освещения;

2 — рабочая платформа, на которой располагается объект;

3 — монтажная пластина, на которой закреплены камера и рабочая головка, с помощью которой объект будет в дальнейшем обработан;

4.1, 4.2, 4.3 — технология позиционирования по осям X, Y, Z, по которым монтажная пластина может перемещаться в трехмерном пространстве;

5 — компьютер, управляющий СТЗ;

6 — кабели связи, по которым передаются данные в компьютер с камеры и остальной системе от компьютера.

1.1.2 Пример не удовлетворяющей требованиям СТЗ

Представленная на рисунке 2 СТЗ не удовлетворяет указанным выше условиям, так как не обладает технологией позиционирования в трехмерном пространстве - деталь перемещается относительно камеры только по ленте конвейера, то есть по оси Y.

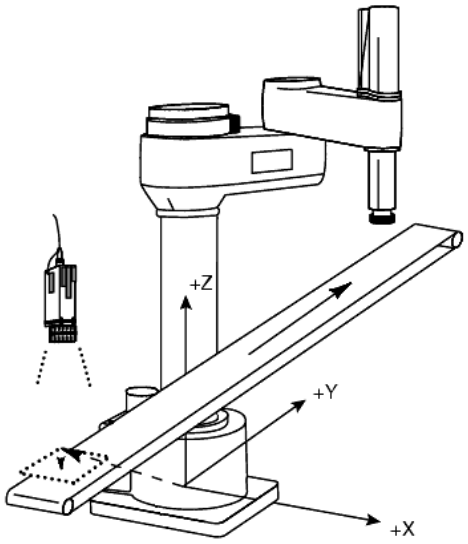

Рис. 2: Не удовлетворяющая требованиям СТЗ

1.2 Особенности СТЗ

Кроме того немаловажно упомянуть следующие условия и характеристики подходящих СТЗ.

Достаточный размер рабочей платформы

Рассматриваемый объект должен полностью помещаться на рабочей платформе, чтобы камера могла свободно передвигаться вдоль его края.

Технология позиционирования не оказывает влияния на результат работы программы

Представленная на рисунке 1 технология позиционирования не является единственной верной — к примеру, существуют 3д принтеры, в которых монтажная пластина перемещается лишь по одной оси, а рабочая платформа по двум, а также дельта-роботы, принцип работы которых основан не на трех взаимно-перпендикулярных направляющих, а на трех параллелограммах. Программа каждой отдельной модели оборудования реализует технологию позиционирования по-своему.

Камера должна быть жестко закреплена

Во время работы программы камера должна оставаться в одном и том же положении относительно монтажной пластины.

1.3 Требования к изучаемым деталям

Детали, положение и ориентацию которых должна определить система, имеют прямоугольную форму с прямыми или закругленными углами и могут иметь вырезы радиальной формы вдоль границ. Длина и ширина этих деталей соизмеримы, так что в поле зрения камеры будет попадать только одна из параллельных границ. Также детали должны иметь высоту по крайней мере в 1 см.

Требование о том, что края детали обязаны быть абсолютно ровными, было бы избыточно, так как алгоритм может справиться с небольшими неровностями кромки. Однако на рассматриваемых границах не должно быть стыков и значительных повреждений. Кроме того от детали не должно исходить никакого вредного для находящейся рядом с ней камеры излучения - теплового или электромагнитного.

2 Нахождение границы между деталью и фоном

Для СТЗ камера является источником информации о текущем положении и ориентации заготовки. В этой главе описываются задействованные в алгоритме методы определения детали на изображении и нахождения границы между ней и фоном, а также даются определения основным понятиям, использующимся в этих методах.

Программная реализация алгоритма, описанного в данной главе, представлена в **Приложении**.

2.1 Определение взаимного расположения детали и фона на изображении

Детали, изучаемые СТЗ, могут отличаться друг от друга размером, цветом, формой и материалом как, например, детали на рисунке 3. Кроме того цвет фона может быть любым, а кадры одной и той же сцены, сделанные с различным освещением, получаются разными. Все эти замечания ведут к постановке следующей задачи: нужно разработать такой алгоритм для СТЗ, который мог бы определить взаимное расположение детали и фона на изображении вне зависимости от перечисленных выше характеристик.

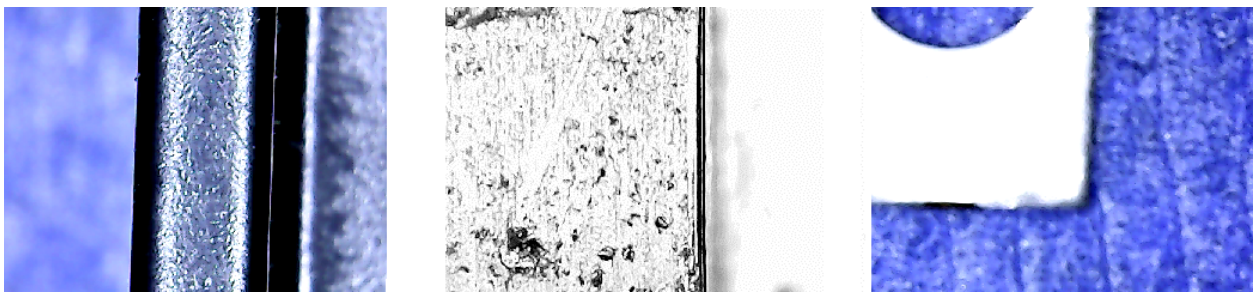

Рис. 3: Примеры кадров с камеры

В отличии от изображения, на котором деталь помещается полностью,

на кадре, снятом рядом с гранью, невозможно однозначно определить, что является границей, а что - лишь дефекты детали. Человек использует свое бинокулярное зрение для восприятия глубины сцены, а СТЗ, обладающая лишь одним зрительным анализатором, достигает его за счет движения относительно этой сцены, определяя относительные скорости перемещения объектов или так называемый оптический поток.

2.1.1 Оптический поток

Оптический поток [7] — это изображение видимого движения сцены (объектов, поверхностей и краев), получаемое в результате перемещения наблюдателя относительно нее. Он предоставляет информацию о сдвиге сцены между двумя изображениями, не затрагивая само их содержание.

Оптический поток, приписывающий каждому пикселю изображения некоторую скорость или, что равнозначно, некоторое смещение, называется плотным (dense) оптическим потоком. Также существуют алгоритмы вычисления выборочных (sparse) оптических потоков, определяющих скорость лишь для заданного множества точек, найденных, к примеру, некоторым детектором контуров.

Очевидно, что выборочный поток вычисляется быстрее, но при сравнении различных алгоритмов разница оказывается незначительной. Также сущесвуют алгоритмы, требующие вычисления скорости всех пикселей изображения, к ним относится и алгоритм, описанный в данной работе.

2.1.2 Вычисление оптического потока

Определение оптического потока состоит в нахождении сдвигов Δx и Δy для заданного массива точек первого изображения, полученного в момент времени t, относительно второго изображения, полученного в момент

времени t+Δt. Данный подход можно записать следующим уравнением

$$I(x, y, t) \approx I(x + \Delta x, y + \Delta y, t + \Delta t), \tag{1}$$

где I(x,y,t) – это яркость пикселя (x,y) в момент времени t.

Яркость пикселя не является единственно возможной функцией для нахождения оптического потока и может давать сбои из-за изменения освещенности между кадрами, но при рассмотрении видеопотока с малым промежутком времени между кадрами можно предполагать, что освещенность изменится лишь незначительно.

В предположении, что перемещение мало, можно использовать ряд Тейлора и получить следующее равенство

$$I(x + \Delta x, y + \Delta y, t + \Delta t) \approx I(x, y, t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t. \tag{2}$$

Из уравнений 1 и 2, пренебрегая погрешностью, можно вывести следующее

$$\frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t = 0,$$

что равнозначно

$$\frac{\partial I}{\partial x}\frac{\Delta x}{\Delta t} + \frac{\partial I}{\partial y}\frac{\Delta y}{\Delta t} + \frac{\partial I}{\partial t}\frac{\Delta t}{\Delta t} = 0,$$

отсюда следует

$$\frac{\partial I}{\partial x}V_x + \frac{\partial I}{\partial y}V_y + \frac{\partial I}{\partial t} = 0,$$

где V_x, V_y – компоненты скорости оптического потока, а $\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}, \frac{\partial I}{\partial t}$ – частные производные яркости пикселя (x,y) изображения в момент времени t в соответствующих направлениях.

В результате имеем

$$I_x V_x + I_y V_y + = -I_t.$$

Найденное уравнение имеет две неизвестные и не может быть однозначно разрешено. Есть множество различных подходов к решению этой

проблемы, в том числе широко известный алгоритм Лукаса-Канаде [3], обходящий неоднозначность за счет использования значений соседних пикселей в каждой точке, и алгоритм Horn-Schunck [8], опирающийся на предположение о гладкости изображения, то есть отсутствии резкого изменения скорости. В данной работе используется алгоритм Farnebäck [6], аппроксимирующий яркость в окрестности с помощью квадратичной формы, реализованный в пакете OpenCV.

2.1.3 Использование оптического потока

Основными областями применения алгоритмов, основанных на оптическом потоке, являются анализ движения и сжатие видео. Кроме того подобные алгоритмы зачастую используются в технологиях компьютерного зрения и робототехники для анализа трехмерных сцен [10].

Одним из самых известных устройств, использующих для своей работы оптические потоки, является оптическая компьютерная мышь. Сенсор современной мыши фиксирует от тысячи кадров в секунду, каждый из которых будет являться смещением предыдущего на один или несколько пикселей. Вычислив это смещение, можно определить, в какой направлении и насколько далеко была подвинута мышь.

В описанном в данной работе алгоритме определение детали на фоне осуществляется на основе параллакса движения — оптического эффекта, заключающегося в том, что при смещении зрительного анализатора проекция объектов, расположенных дальше от наблюдателя, смещается медленнее, чем проекция ближе расположенных объектов. Вследствие этого эффекта значения оптического потока в точках, соответствующих фону, будут заметно меньше, чем в точках, соответствующих детали, что позволяет определить взаимное расположение фона и детали на изображении.

2.2 Определение границ детали

Видимая на получаемом с камеры изображении часть детали будет иметь форму четырехугольника (выделено зеленым на рисунке 4), стороны которого нужно найти для определения параметров детали на рабочей платформе.

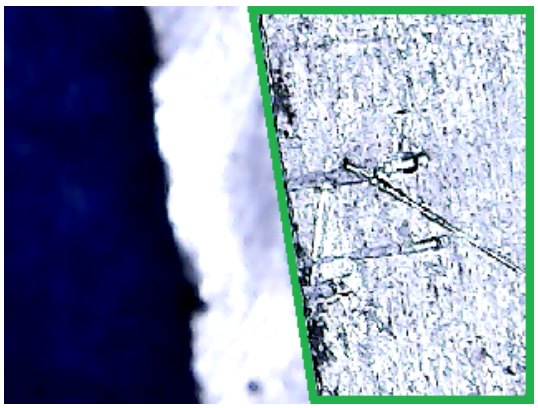

Рис. 4: Изображение с камеры

2.2.1 Определение границ детали по оптическому потоку

Для простоты восприятия оптического потока человеком вычисленные скорости переводятся из вещественного типа данных в целочисленный. Каждый пиксель на изображении окрашивается в один из четырнадцати цветов согласно соответствующему значению оптического потока:

- В ходе работы алгоритма Farnebäck значения некоторых точек не может быть вычислено корректно (например, из-за появления новой части детали на изображении или освещения). При нормальной работе СТЗ количество пикселей с каждым таким значением не превышает одного процента от числа всех пикселей на изображении. Они будут отмечены красным.

- Пикселям с двенадцатью самыми часто встречающимися значениями, количество которых превышает один процент от числа всех пик-

селей на изображении, ставятся в соответствие различные цвета. Черный цвет при этом обозначает самое маленькое смещение.

- Остальные пиксели отмечаются белым.

Слишком большое число обозначенных белым или красным значений указывает на то, что между кадрами был совершен шаг не оптимальной длины. Результат такой обработки оптического потока представлен на рисунке 5.

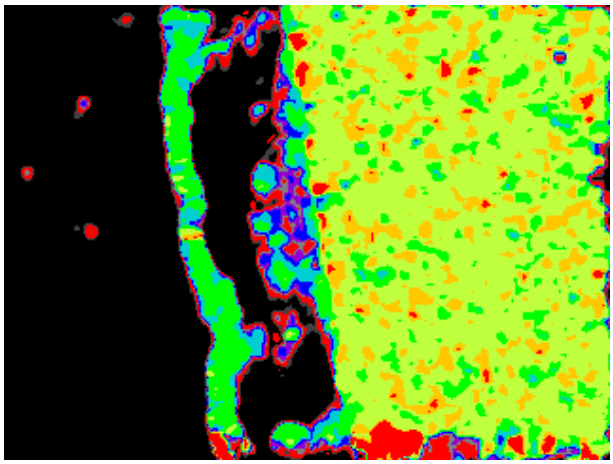

Рис. 5: Оптический поток

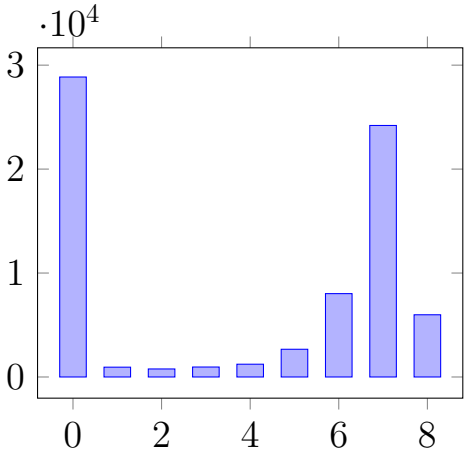

Рис. 6: Распределение скоростей

На рисунке 6 представлена гистограмма распределения скоростей оптического потока, на которой явно видны два пика, соответствующие фону с нулевым перемещением и детали со скоростью 7. Кроме того к детали относятся и соседние значения, выделяющиеся среди остальных, как, например, скорости 6 и 8. После обработки рисунка 5 получается множество точек детали, для наглядности обозначенных на рисунке 7 черным цветом.

Для дальнейшей работы нужно найти контур детали на изображении. Используя алгоритм Square Tracing, описанный в работе [9], система обходит один за другим все замкнутые контуры обоих цветов в предположении, что искомыми являются самые длинные контуры черного цвета. После обработки рисунка 7 получается следующий массив точек контура, представленный на рисунке 8.

Рис. 7: Деталь из оптического потока

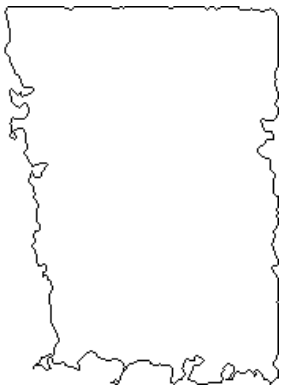

Рис. 8: Граница детали

Для нахождения уравнений прямых, соответствующих границам, программа находит точки, принадлежащие каждой из них. Для этого объявляется четыре массива: два размером W для верхней и нижней границ и два размером H для левой и правой границ, где W - это ширина изображения с камеры, а H - это его высота. Каждому элементу этих массивов присваивается значение W+H. Так как точки границ являются минимумами или максимумами в соответствующих столбцах или строках изображения, программа проходит по массиву точек контура, сравнивая их со значениями в четырех массивах. Ход вычислений для левой границы показан в таблице 3 на примере изображения 9, на котором синим отмечен контур. Жирным выделено измененное на данном шаге значение (для наглядности за один шаг в таблице может быть взято несколько точек контура). Длина и ширина этого изображения равны 10.

Таблица 1: Данные об ориентации детали

Y	0	1	2	3	4	5	6	7	8	9
	20	20	20	20	20	20	20	20	20	20
(6,0)	**6**	20	20	20	20	20	20	20	20	20
(7,0)-(8,0)	6	20	20	20	20	20	20	20	20	20
(9,0) - (9,9)	6	**9**	**9**	**9**	**9**	**9**	**9**	**9**	**9**	**9**

(8,9) - (1,9)	6	9	9	9	9	9	9	9	9	**1**
(2,8)	6	9	9	9	9	9	9	9	**2**	1
(2,7)	6	9	9	9	9	9	9	**2**	2	1
(3,6)	6	9	9	9	9	9	**3**	2	2	1
(4,5)	6	9	9	9	9	**4**	3	2	2	1
(4,4)	6	9	9	9	**4**	4	3	2	2	1
(4,3)	6	9	9	**4**	4	4	3	2	2	1
(4,2)	6	9	**4**	4	4	4	3	2	2	1
(5,1)	6	**5**	4	4	4	4	3	2	2	1

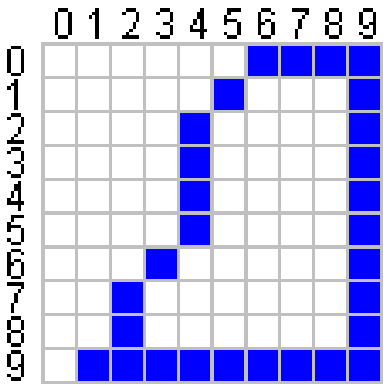

Рис. 9: Пример замкнутого контура

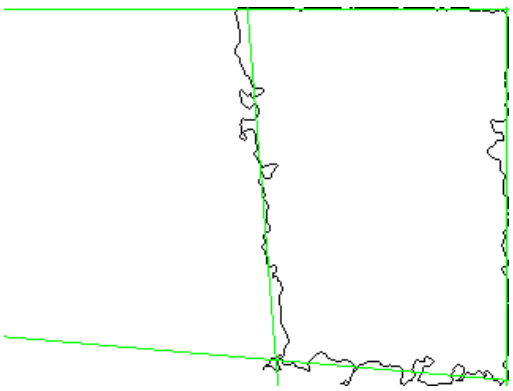

Рис. 10: Аппроксимированные границы

Для сглаживания неровностей оптического потока каждая из сторон разделяется на три части, в каждой из которых высчитывается среднее значение. При этом не учитываются те значения, отклонение которых от нормы слишком велико. Затем находятся параметры прямых, проходящих через эти соседние средние значения. После этого параметры для каждой стороны усредняются и таким образом получаются искомые уравнения границ. Аппроксимированные границы для рисунка 8 представлены на рисунке 10 зеленым цветом.

2.2.2 Преобразование Хафа для поиска прямых

Найденные из оптического потока границы подходят для определения взаимного расположения детали и фона, однако они недостаточно точны для последующей оценки положения и ориентации детали. Поэтому для повышения точности необходимо найти прямые на оригинальном изображении с камеры.

Перед поиском прямых изображение проходит через три последовательные стадии обработки: приведение цветного изображения к изображению в оттенках серого, сегментация изображения с помощью порогового метода и выделение границ на изображении посредством оператора Кэнни (эти методы обработки изображений описаны в работе [4]) — результаты каждой из этих обработок представлены на рисунках 11, 12 и 13 соответственно. Рисунок 13 будет использован для поиска прямых, проходящих через точки белого цвета, соответствующие выделенным оператором Кэнни границам.

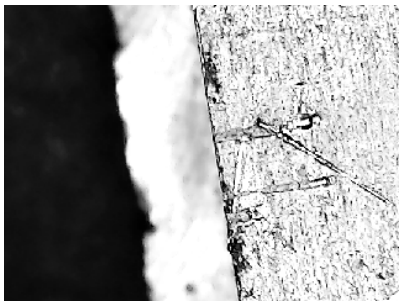

Рис. 11: Изображение в оттенках серого

Рис. 12: Изображение после сегмантации

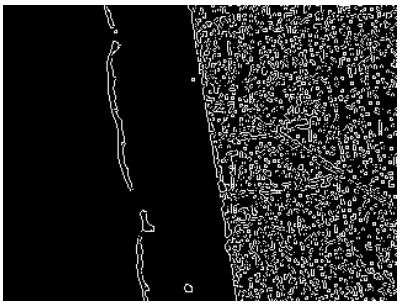

Рис. 13: Выделенные границы

Одним из методов нахождения на изображении простых фигур, таких как прямые, круги и эллипсы, является преобразование Хафа [5]. Самый простой вариант этого алгоритма предназначен для поиска прямых. Прямая при этом представляется набором точек (x, y), удовлетворяющих уравнению

$$xcos\theta + ysin\theta + r = 0. \quad (3)$$

Параметр r - это длина перпендикуляра, опущенного на прямую из начала координат, а θ - это угол между положительным направлением оси абсцисс и этим перпендикуляром. Каждая пара параметров (θ, r) соответствует уникальной прямой при условии, что $r \geq 0$ и $0 \leq \theta < 2\pi$. Кроме того параметр r нужно ограничить сверху, чтобы не рассматривать те прямые, которые не видны на изображении.

Для обнаружения прямых преобразованием Хафа координаты (x, y) каждого пикселя белого цвета на прошедшем обработку изображении и каждая пара параметров прямой (θ, r) подставляются в уравнение 3. Если оно выполняется с некоторой заранее заданной точностью, то прямая получает голос. Эта процедура называется процедурой голосования. Чем больше голосов в итоге получает прямая, тем больше на изображении есть пикселей, через которых она проходит, и тем яснее она видна.

На рисунке 14 представлен результат работы преобразования Хафа на изображении размером 800 x 515 пикселей. Зеленым нарисованы семнадцать линий, прошедшие заданный порог в 250 голосов.

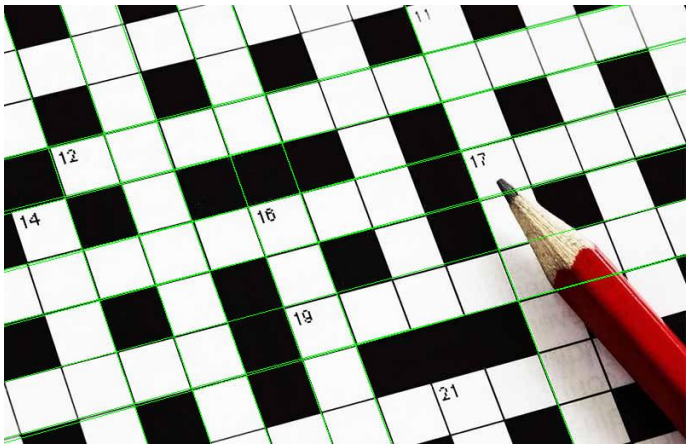

Рис. 14: Пример применения преобразования Хафа

При применении преобразования Хафа задается минимальное количество голосов или точек, через которые должна проходить прямая, чтобы метод включил ее в результирующий массив. Чем больше это число, тем меньше прямых будет найдено. Используя алгоритм двоичного поиска,

система находит такое минимальное число голосов, при котором количество найденных прямых не будет превышать десяти. Одной из них будет искомая граница между деталью и фоном, являющаяся одной из самых длинных на изображении. Большее количество прямых может привести к противоречиям. На рисунке 15 красным цветом представлены прямые, полученные с помощью преобразования Хафа.

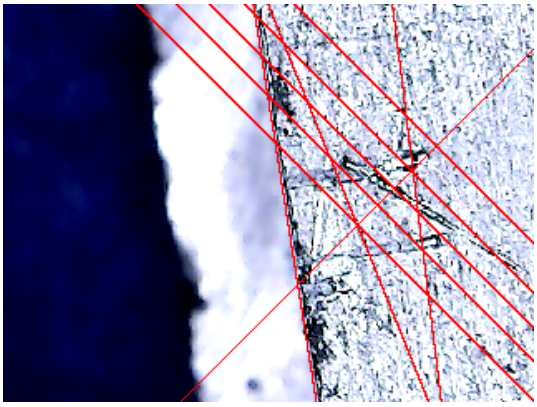

Рис. 15: Полученные прямые

2.2.3 Определение результирующих прямых

Полученные в пункте 2.2.1 прямые могут быть использованы для определения взаимного расположения фона и детали, но они недостаточно точны для последующего вычисления положения и ориентации детали. Прямые, полученные в пункте 2.2.2 с помощью преобразования Хафа обладают достаточной точностью, но без дополнительной информации не определить, какая именно из них является искомой. Однако из этих двух массивов уже можно получить необходимые данные.

Каждой из прямых, полученных из оптического потока (группа 1), ставится в соответствие или одна из прямых, вычисленных с помощью преобразования Хафа (группа 2.1), или одна из прямых рамки изображении (группа 2.2). Для этого сравнивается каждая пара прямых из групп 1 и $2.1 \cup 2.2$. Для вертикальных линий вычисляются значения абсциссы в точ-

ках $y = 0$ и $y = 100$, для горизонтальных – значения ординаты в точках $x = 0$ и $x = 100$. Самой похожей объявляется та пара прямых, абсолютная разница значений которых в двух точках является минимальной.

Так как доподлинно неизвестно, насколько сильно прямые группы 1 отличаются от искомых, вполне возможна ситуация, в которой границей между деталью и фоном на самом деле являются не самые на них похожые, а близкие к ним. Поэтому кроме самых похожих для каждой прямой группы 1 система определяет массив достаточно похожих на нее прямых группы $2.1 \cup 2.2$, который может включать и только ее саму. Так как реализованный в пакете OpenCV метод преобразования Хафа располагает найденные прямые в порядке убывания количества голосов, можно предположить, что прямая из этого массива, обладающая самым маленьким номером, соответствует искомой границе между деталью и фоном. В том случае, если эта прямая совпадает с прямой, являющейся самой похожей на соответствующую границу, можно с уверенностью утверждать, что граница между деталью и фоном найдена.

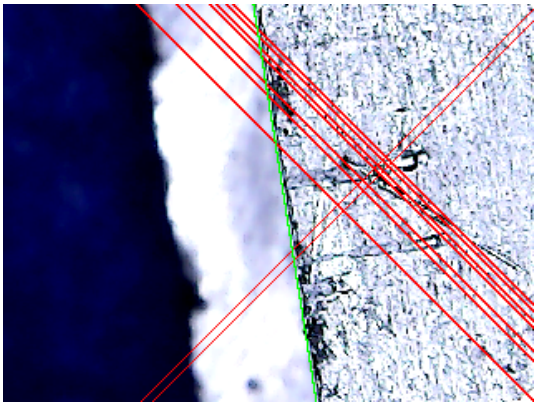

Рис. 16: Удачная попытка

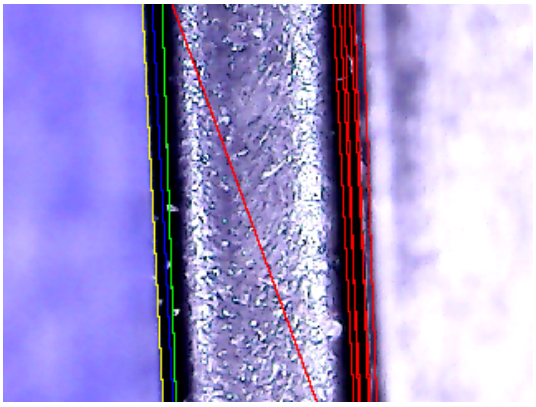

Рис. 17: Неудачная попытка

На рисунке 16 представлен результат сопоставления прямых с рисунков 10 и 15, зеленым на нем выделена граница между деталью и фоном, а красным - все остальные прямые группы 2.1. На рисунке 17 представлен провалившийся из-за слишком сильного освещения результат выпол-

нения алгоритма – так как зеленая прямая, обозначающая самую близкую к группе 1, и желтая прямая, предполагаемая искомой границей, не совпали, полученные данные могут оказаться ошибочными.

3 Стадия подготовки

Для успешной работы программа должна обладать некоторыми знаниями о СТЗ, в том числе теми, которые не меняются от запуска к запуску. Для этого оператору нужно занести известную ему информацию в файл о СТЗ и расположить камеру, чтобы система могла определить и сохранить оставшуюся необходимую информацию. Этот процесс называется стадией подготовки и он проводится один раз перед началом работы.

3.1 Файл с информацией о СТЗ

На данный момент существует огромное множество разных станков для всевозможных целей - шлифования, фрезерования, сверления - и для обработки разнообразных материалов - металла, дерева, пластмассы. В связи с этим станки могут значительно отличаться друг от друга. Эти различия необходимо учитывать для успешной работы алгоритма и безопасной эксплуатации оборудования. Для этого оператору нужно создать файл, информация об осях в котором записывается в массивы в порядке X, Y, Z, со следующей информацией о СТЗ:

1. **Максимальные перемещения по осям X, Y, Z**

   Нужно следить за тем, чтобы станку не была отдана команда передвинуться в отрицательные или превышающие максимальные координаты, так как это может привести к его поломке. Эти значения для осей записываются в массив "borders".

2. **Скорость передачи данных**

   Для передачи команд станку и получения от него ответов нужно знать правильную скорость. Измеряется в битах в секунду.

3. **Обратный Z**

Этот параметр может принимать значения *true/false*. *False* означает, что при увеличении параметра Z расстояние от камеры до стола будет увеличиваться, а *true* - что оно будет уменьшаться.

4. **Минимальный шаг**

   Минимальное расстояние, на которое СТЗ может передвинуть монтажную пластину по каждой из осей. Эти значения записываются в массив "min_shifts".

Данный файл должен иметь расширение .json. Ниже представлен пример файла с информацией о 3D принтере с размерами рабочей платформы 350 x 250 x 300, скоростью передачи данных 250 000 бит/сек., без обратного Z, с минимальным шагом 0.1 мм по каждой из осей.

```
{
    "borders": [350, 250, 300],
    "speed": 250000,
    "inverse_z": false,
    "min_shifts": [0.1, 0.1, 0.1]
}
```

Кроме этих данных программе нужна информация о положении камеры относительно осей движения СТЗ. Эта информация определяется при помощи оператора один раз после закрепления камеры, записывается в тот же файл и состоит из следующих частей:

1. **Соответствие перемещения камеры перемещению детали на избражении**

   Если перемещению камеры по оси X соответствует перемещение детали на изображении по горизонтали, то параметр "accordance" принимает значение *true*. В ином случае - *false*.

2. **Ориентация камеры по осям X, Y**

   Если ориентация камеры неправильна, то ее положительному перемещению по оси будет соответствовать положительное перемещение детали на изображении. В этом случае изображение с камеры нужно отражать по горизонтали и/или вертикали. Если ориентация по оси правильная, то соответствующий элемент массива "orientation" будет иметь значение *true*, в ином случае - *false*.

3.2 Определение ориентации камеры

Вычислив оптический поток, описанный в пункте 2.1.1, становится возможным получение информации о положении камеры относительно осей движения СТЗ.

Шаг 1. Первым делом нужно выяснить, соответствует ли перемещение камеры по оси X перемещению детали по горизонтали (аналогично по оси Y и по вертикали). Для этого делается первый снимок, совершается шаг оптимальной длины по одной из осей и делается второй снимок, затем программа высчитывает оптический поток этих двух снимков. Пример результата визуализации этой операции представлен на рисунках 18 и 19.

Рис. 18: Оптический поток движения по горизонтали

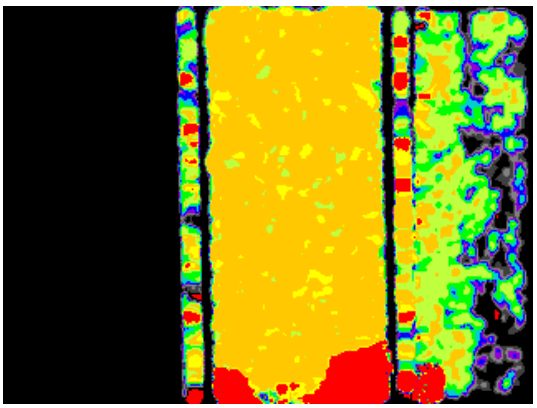

Рис. 19: Оптический поток движения по вертикали

Как видно из этих рисунков, по одному из направлений, в данном случае, по горизонтали, будут преобладать нулевые значения скоростей, отмеченные черным цветом. На другом рисунке присутствует много ненулевых перемещений, указывающих на то, что деталь переместилась именно по этому направлению, в данном случае, по вертикали. Если сложить значения оптического потока отдельно для каждого из направлений, то получается две суммы: 4649 (рисунок 18) и 386892 (рисунок 19). Так как одно из чисел значительно больше другого, можно с легкостью определить, в какую именно сторону передвинулась деталь на изображении.

Если направление перемещения детали не совпадает с направлением движения камеры, то перед обработкой изображение нужно будет поворачивать на 90 градусов.

Шаг 2. Для правильной работы алгоритма также необходимо, чтобы перемещению камеры по какой-либо из осей соответствовало перемещение детали на изображении в противоположную сторону, в ином случае положение и ориентация заготовки будут вычислены неправильно. Для определения этой информации по алгоритму, использованному в **шаге 1**, с камеры получаем две пары снимков - по осям X и Y соответственно. Для каждой из этих пар вычисляется свой оптический поток.

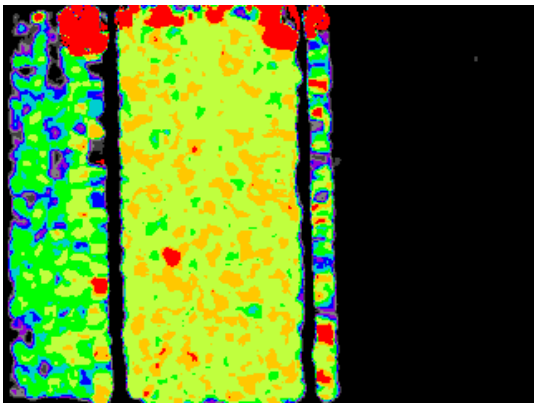

Рис. 20: Правильный порядок

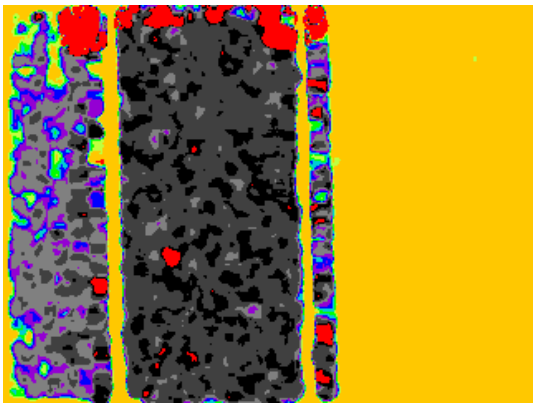

Рис. 21: Неправильный порядок

На рис. 20 и 21 представлены результаты визуализации оптических по-

токов двух изображений. На левом рисунке эти изображения заданы в одном порядке, а на правом - в обратном. Если сложить значения оптического потока для каждого из этих порядков, то получается две суммы: 323289 и -375822. Из визуализаций можно заключить, что если сумма оптического потока по оси положительна, то ориентация камеры по этой оси правильна. В ином случае получаемое с нее изображении нужно будет отразить по вертикали или горизонтали.

3.3 Определение положения оптической оси объектива

Оптическая ось - это линия, относительно которой в оптической системе, например, в объективе или микроскопе, присутствует вращательная симметрия. Из-за конструкции камеры, технологической погрешности изготовления или неточности установки оптическая ось объектива может не совпадать с геометрическим центром изображения. Камера находится над гранью детали, если эта грань на изображении проходит через оптическую ось. Поэтому для последующего правильного расположения камеры необходимо вычислить эту точку. Ее координаты по осям X, Y сохраняются в файл в массив "optaxis".

Для нахождения точки оптической оси на изображении используется следующий факт: если камера находится над гранью, то при изменении расстояния от камеры до рабочей платформы грань на изображении остается на том же месте.

Для решения этой задачи был разработан следующий алгоритм:

Шаг 1. Разместить камеру так, чтобы вертикальная грань на изображении проходила через его геометрический центр. Объявить массив для 6 прямых.

Шаг 2. Сохранить текущий кадр, сделать шаг оптимальной длины

по любой оси, сохранить следующий кадр, вычислить оптический поток и определить параметры прямой, соответствующей границе между фоном и изображением. Записать прямую в массив под нулевым номером. Установить счетчик k на единицу.

Шаг 3. Увеличить расстояние до рабочей платформы на 1 см. Используя преобразование Хафа, найти линии на изображении, сравнить их с k-1 в массиве, сохранить под номером k самую похожую. Прибавить к k единицу.

Шаг 4. Если счетчик k меньше 5, перейти к **шагу 3**. В ином случае вычислить разницу между прямыми под номерами 0 и 5. Если разница отсутствует, то точка оптической оси была успешно найдена и вычисления останавливаются. В ином случае вернуться к исходному расстоянию до рабочей платформы, передвинуть камеру по горизонтальной оси на минимальный шаг в положительном направлении, если разница положительна, или в отрицательном, если она отрицательна, перейти к **шагу 2**.

3.4 Определение расстояния между центром камеры и рабочей головкой

Последними необходимыми для работы параметрами является массив "head_shift", характеризующий расстояние между камерой и головкой станка по осям X, Y. Эти значения должны быть учтены при перерасчете программы обработки детали.

Для определения этих расстояний станок выполняет заранее заданную программу, оставляя на рабочей платформе или на расположенной на ней детали след. 3D принтер, к примеру, оставляет на поверхности слои застывшего термопластика. Затем СТЗ проводит исследование данного следа и сравнивает полученные значения его позиции со значениями в заданной программе, вычисляя искомое расстояние.

4 Определение положения и ориентации детали

После прохождения стадии подготовки можно перейти к стадии исследования детали. Используя алгоритм определения границы между деталью и фоном на изображении с камеры из главы 2, можно приступить к обработке последовательности изображений определенной грани детали, откуда уже можно получить информацию о ее положении и ориентации.

Так как при разработке СТЗ мы не требуем, чтобы ориентация камеры совпадала с осями координат рабочей платформы, неизбежны некоторые отклонения. Эти отклонения кроме того могут меняться от запуска к запуску программы вследствие, например, снятия камеры и ее повторного закрепления, поэтому нет смысла сохранять их в файл с информацией о СТЗ, в котором хранятся данные только о постоянных параметрах системы.

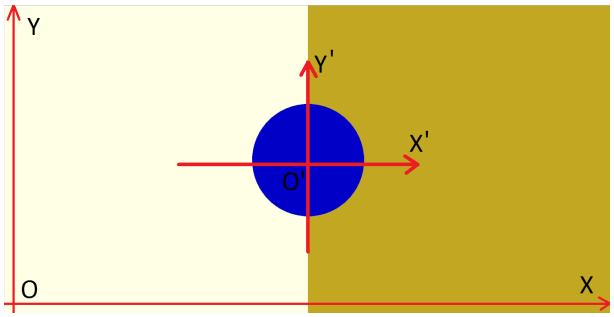

Рис. 22: Идеальная ориентация камеры

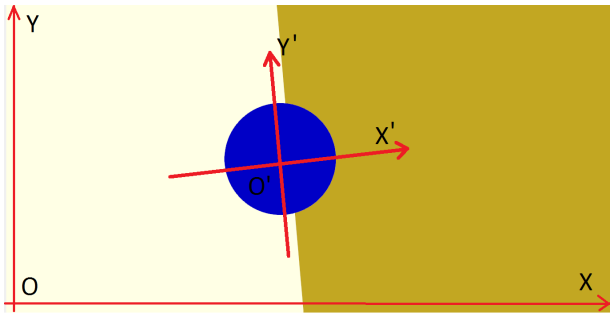

Рис. 23: Неидеальная ориентация камеры

На представленных на рисунках 22 и 23 ситуациях ориентация камеры $(O'X'Y')$ в обоих случаях совпадает с наклоном детали, вследствие чего границы между фоном и деталью на получаемых с камер изображениях будут иметь один и тот наклон. Левый рисунок показывает идеальную

ориентацию камеры, совпадающую с осями координат рабочей платформы (OXY). Если бы можно было с уверенностью сказать, что камеры всегда будет расположена именно так, то можно было бы определить ориентацию детали не по последовательности изображений, а всего по двум кадрам. Однако скорее всего будет присутствовать некоторое неизвестное отклонение от идеальной ориентации, как на правом рисунке.

Если в ситуации, представленной на рисунке 23, перемещать камеру исключительно по оси Y рабочей платформы, не делая движений по направлению X, то деталь на изображении будет постепенно смещаться. Замеряя это смещение при различных положениях камеры, можно определить ориентацию детали.

4.1 Алгоритм определения ориентации детали

После размещения оператором камеры над границей детали СТЗ делает шаг оптимальной длины в любом направлении, чтобы вычислить оптический поток и определить местоположение детали. Для решения задачи определения ориентации детали был разработан следующий алгоритм. Он применяется для тех случаев, когда граница между деталью и фоном проходит по вертикали кадра. В ином случае в алгоритме нужно лишь заменить движение по оси X на движение по оси Y и наоборот.

Первым делом СТЗ совершает шаги оптимальной длины по положительному направлению оси X, вычисляя оптический поток каждой последовательной пары и сохраняя информацию о найденной на изображении границе. Этот процесс продолжается до тех пор, пока деталь не будет больше видна на изображении, затем камера возвращается на исходную позицию.

Такие же действия производятся и по оси Y с той лишь разницей, что после фиксации каждой пары кадров для получения оптического потока

делается большой шаг, чтобы уменьшить затрачиваемое время, так как настолько подробное исследование стороны детали не требуется. Мы будем предполагать, что деталь расположена на рабочей платформе достаточно ровно, так что СТЗ сумеет собрать достаточное количество данных до того, как деталь выйдет за пределы изображения с камеры.

Необходимо обследовать сторону заготовки по всей ее длине, так как из-за обработки материала на ее сторонах могут присутствовать технологические неровности, такие как выпуклость, вогнутость или шероховатость.

В таблице 2 представлен пример получаемых в результате работы этого алгоритма данных. В левой ее части представлены результаты движения по оси X, в правой - по оси Y. Столбцы X и Y показывают изменение положения камеры по соответствующей оси, положение по второй оси во время исследования при этом фиксировано. В столбцах *Граница* отражено положение на изображении границы между фоном и деталью в пикселях. Столбцы *Наклон* представляют угол между этой границей и осью координат.

Таблица 2: Данные об ориентации детали

X	Граница	Наклон	Линии	Y	Граница	Наклон	Линии
0.0	83	0.035	+	0.0	109	0.017	+
0.1	87	0.035	+	2.5	60	-0.017	+
0.2	96	0.035	+	5.0	110	0.017	+
0.3	101	0.035	+	7.5	115	0.017	-
0.4	110	0.035	+	10.0	72	-0.017	+
0.5	115	0.035	+	12.5	113	0.017	+
0.6	121	0.035	-	15.0	70	-0.1	+
0.7	129	0.035	-	17.5	113	0.017	-
0.8	134	0.035	+	20.0	115	0.017	-

Таблица 2: Данные об ориентации детали

X	Граница	Наклон	Линии	Y	Граница	Наклон	Линии
0.9	142	0.035	+	22.5	113	0.035	+
1.0	147	0.035	+	25.0	116	0.017	+
1.1	239	0.0	+	27.5	113	0.017	-
1.2	163	0.0	+	30.0	114	0.017	+
1.3	169	0.0	+	32.5	111	0.035	+
				35.0	113	0.017	+
				37.5	77	-0.1	-
				40.0	114	0.017	+
				42.5	114	0.017	+
				45.0	114	0.017	-
				47.5	71	0.0	-
				50.0	110	0.035	-
				52.5	114	0.017	-
				55.0	112	0.035	+
				57.5	120	0.017	-
				60.0	120	0.017	+

4.2 Выборка достоверных границ между деталью и фоном

Описанный в главе 2 алгоритм может совершать ошибки из-за, например, резкого изменения освещенности, вследствие чего полученные из последовательности изображений данные нужно проверять на однородность, отбрасывать все неподходящие и в дальнейшем работать только с досто-

верными.

На диаграмме 24 представлен результат описываемого далее алгоритма, примененного к данным о передвижении по оси Y из таблицы 2. По горизонтали указано положение камеры по этой оси, по вертикали - положение границы между фоном и деталью.

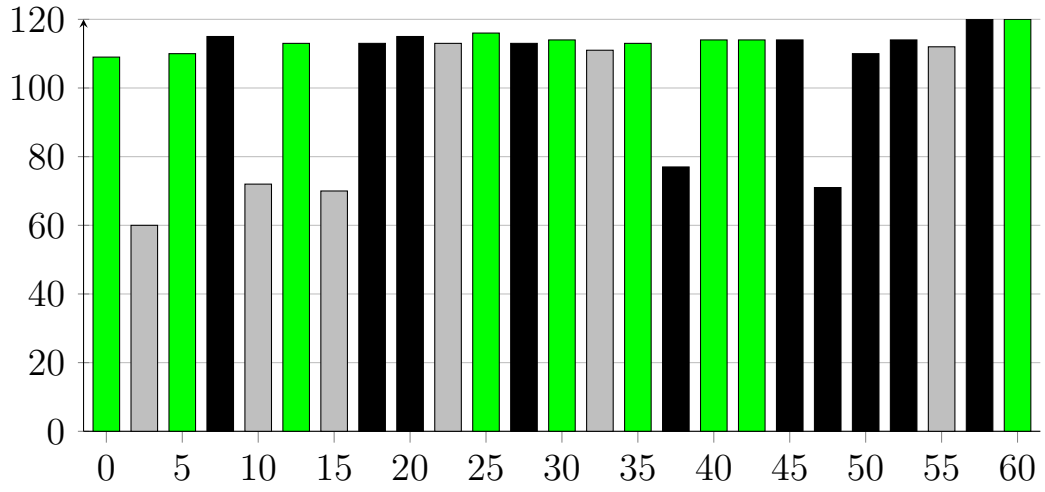

Рис. 24: Достоверность данных из таблицы 2 по оси Y

Первым шагом к нахождению ошибочных данных является сравнение двух прямых, полученных по алгоритму из главы 2. В таблице 2 эта информация представлена в столбцах *Линии*. Знаком "+" отмечены те данные, в которых эти прямые совпадают и полученной информации можно доверять, знаком "-" отмечены те, в которых прямые не совпадают. На диаграмме 24 в черный цвет окрашены те столбцы, которые не прошли эту проверку.

Второй шаг проверки данных связан с наклоном границы между фоном и деталью. Самое часто встречающееся значение наклона предполагается правильным. В таблице 2 для оси Y таким значением является 0.017. Те данные, которые прошли первый шаг проверки и наклон которых отличается от 0.017, окрашены на диаграмме 24 в серый цвет.

Прошедшие эти два шага проверки данные представлены столбцами, окрашенными в зеленый цвет. По ним видно, что край детали не идеально

ровный, а немного вогнутый на значениях Y 30-50. Этот подход проверки данных на достоверность позволяет делать выводы об ориентации даже тех деталей, работа с которыми оказывается весьма трудной из-за сложностей нахождения оптического потока.

В ходе экспериментов были проведены статистические наблюдения по количеству достоверных данных при изучении границы. Их доля среди всех данных не опускается ниже 25%. Этого достаточно для успешной оценки положения и ориентации детали.

4.3 Определение относительного отклонения

Ориентация детали будет определяться по относительным смещениям камеры по осям. К примеру, соотношение x/y означает, что при смещении камеры на x мм по оси X и на -y мм по оси Y линия между фоном и деталью на изображении останется примерно на том же месте. Одна из задач данной работы заключается в нахождении этого соотношения по данным, отобранным в предыдущем пункте. Для ее решенния был разработан следующий агоритм.

Для каждой из осей вычисляются параметры прямой $y = ax + b$, где аргумент x означает перемещение по соответствующей оси, а y - положение границы между фоном и деталью на изображении. Эти данные аппроксимируются с помощью метода наименьших квадратов (МНК).

Таблица 3: Отобранные из таблицы 2 данные

X	0.0	0.1	0.2	0.3	0.4	0.5	0.8	0.9	1.0
Граница	83	87	96	101	110	115	134	142	147
Y	0.0	5.0	12.5	25.0	30.0	35.0	40.0	42.5	60.0
Граница	109	110	113	116	114	113	114	114	120

Отобранные из таблицы 2 данные представлены в таблице 3. На их основе мы для примера найдем ориентацию детали. По алгоритму МНК мы вычисляем следующие параметры:

$$a_x = 65.416, b_x = 82.25; \quad a_y = 0.142, b_y = 109.715.$$

Угловые коэффициенты означают, что при смещении камеры по оси X на 1 мм граница между деталью и фоном на изображении переместится на 65.416 пикселей, а при смещении по оси Y - на 0.142 пикселя. Из этих двух соотношений - x/пиксель и y/пиксель - можно определить искомое соотношение x/y. В данном примере оно будет равно $\frac{1}{65.416}/\frac{1}{0.142} = 0.0153/7.0423$ или 0.153/70.423.

4.4 Определение положения детали

В областях, занимающихся оборудованием с ЧПУ, было разработано множество различных методов определения положения заготовки. Одними из самых широко используемых среди них являются метод определения места пересечения двух перпендикулярных сторон детали и метод нахождения центров двух отверстий. Разработанный алгоритм определения положения детали опирается на идею, представленную в первом методе.

Используя алгоритм, описанный в пункте 4.3, система вычисляет функции отношений для двух перпендикулярных сторон. Алгоритм определения положения детали будет описан в общем виде.

Функции отношения для вертикальной прямой, которая была исследована при положении по вертикальной оси y_0 и по горизонтальной оси x_0 соответственно:

$$f_x(x) = a_x x + b_x, \quad f_y(y) = a_y y + b_y,$$

для горизонтальной прямой, которая была исследована при положении по

вертикальной оси y_1 и по горизонтальной оси x_1 соответственно:

$$g_x(x) = c_x x + d_x, \quad g_y(y) = c_y y + d_y.$$

Затем для каждой прямой находится уравнение от двух переменных x и y, имеющее следующий вид. Для вертикальной прямой:

$$f(x, y) = ax + by + e, \tag{4}$$

где $a := a_x$, $b := a_y$, $e := \frac{b_x - a_y y_0 + b_y - a_x x_0}{2}$. Для горизонтальной прямой:

$$g(x, y) = cx + dy + f, \tag{5}$$

где $c := c_x$, $d := c_y$, $f := \frac{d_x - c_y y_1 + d_y - c_x x_1}{2}$.

После этого для прямых вычисляется функция границы на рабочей платформе. Считается, что она проходит через точку оптической оси системы. Подставляя из файла с информацией о СТЗ в формулы 4 и 5 значения "optaxis_x"вместо f(x,y) и "optaxis_y"вместо g(x,y), можно найти точки, через которые эти функции проходят, и вычислить их параметры.

Найдя точку пересечения этих функций и зная, где она предположительно должна находиться, система высчитывает разницу между этими значениями, тем самым определяя положение детали.

Заключение

Целью данной выпускной квалификационной работе была разработка алгоритма автоматизации процесса оценки положения и ориентации детали с помощью вебкамеры. Изначально этот процесс выполнялся вручную, что увеличивало время, затрачиваемое на подготовку детали, и требовало от работников большого опыта, высокой квалификации и аккуратности. При этом не достигались достаточная точность и повторяемость продукции.

Для достижения цели работы был поставлен ряд задач: определение границы между деталью и фоном вне зависимости от их характеристик, нахождение информации о СТЗ, оценка положения и ориентации детали. В ходе работы для каждой из этих задач был разработан метод решения. Эти методы были программно реализованы на языке Java. Кроме того были рассмотрены и решены обнаруженные проблемы, возникшие при применении алгоритма в практических целях, а также выставлены требования к СТЗ и исследуемым деталям.

Были проведены эксперименты на станке Cutmaster CM-1500, оборудованном дозатором клеев и герметиков для вклейки стекол, в ООО "Элк", занимающемся разработкой и производством электронного оборудования для железнодорожного транспорта. Результаты данной работы были внедрены на этом производстве, вследствие чего требования к квалификации оператора станка были серьезно смягчены, точность и повторяемость продукции улучшены, а время подготовки детали к обработке укорочено.

Таким образом, задачи решены в полном объеме, алгоритм успешно внедрен в производственный процесс, вследствие чего поставленная цель повышения эффективности изготовления деталей и улучшения их качества достигнута.

Список литературы

[1] Б.М. Базров. Основы технологии машиностроения. Машиностроение, 2005. С. 736.

[2] Дэвид А. Форсайт, Жан Понс. Компьютерное зрение: Современный подход. Pearson, 2004. С. 928.

[3] Jean-Yves Bouguet. Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm. Intel Corporation, 2000. С. 9.

[4] Gary Bradski, Adrian Kaehler. Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Media, 2008. С. 556.

[5] Duda, R. O., P. E. Hart. Use of the Hough Transformation to Detect Lines and Curves in Pictures. Stanford Research Institute, 1972. С. 5.

[6] Gunnar Farnebäck. Two-Frame Motion Estimation Based on Polynomial Expansion. Linköping University, 2003. С. 8.

[7] Fleet D., Weiss Y. Optical Flow Estimation. In: Paragios N., Chen Y., Faugeras O. (eds) Handbook of Mathematical Models in Computer Vision. Springer, 2006. С. 237-257.

[8] Berthold K.P. Horn, Brian G. Shunck. Determining Optical Flow. MIT, 1980. С. 28.

[9] P. Rajashekar Reddy, V. Amarnadh, Mekala Bhaskar. Evaluation of Stopping Criterion in Contour Tracing Algorithms. IJCSIT, 2012. С. 7.

[10] Jörg Rett, Jorge Dias. Autonomous Robot Navigation-A study using Optical Flow and log-polar image representation. University of Coimbra. С. 10.