

**МОДЕЛЬ ОПТИЧЕСКОГО ДАТЧИКА КОНТРОЛЯ УРОВНЯ ЗАГРЯЗНЕНИЙ
НЕФТЕПРОДУКТАМИ****Г. Т. Пипия*, С. В. Хлопин, Л. В. Черненькая***Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Россия*** gogpipiy@ya.ru*

Аннотация. Объектом исследования является анализ фото- и видеоизображений, полученных с использованием специального оптического датчика в целях определения наличия нефтепродуктов в жидкостях. Используются такие методы исследования, как машинное (компьютерное) зрение, обнаружение объектов, цветокоррекция, поиск контуров и цифровая обработка изображений. При разработке программного комплекса использованы язык программирования Java 8 и фреймворки Java Swing, OpenCV 3.4.10, Processing.

Ключевые слова: компьютерное зрение, детектирование объектов, анализ изображения, глобальная оптимизация, целевые функции.

Благодарности: исследование выполнено при поддержке Российского научного фонда, грант № 24-21-20116 (<https://rscf.ru/project/24-21-20116/>), и Санкт-Петербургского научного фонда, грант № 24-21-20116.

Ссылка для цитирования: Пипия Г. Т., Хлопин С. В., Черненькая Л. В. Модель оптического датчика контроля уровня загрязнений нефтепродуктами // Изв. вузов. Приборостроение. 2025. Т. 68, № 3. С. 267–274. DOI: 10.17586/0021-3454-2025-68-3-267-274.

MODEL OF OPTICAL SENSOR FOR MONITORING THE LEVEL OF OIL POLLUTION**G. T. Pipiy*, S. V. Khlopin, L. V. Chernenkaya***Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russia*** gogpipiy@ya.ru*

Annotation. The object of the study is the analysis of photo and video images obtained using a special optical sensor to determine the presence of petroleum products in liquids. The following research methods were used: machine (computer) vision, object detection, color correction, contour search, and digital image processing. The Java 8 programming language and the Java Swing, OpenCV 3.4.10, Processing frameworks were used in the development of the software package.

Key words: computer vision, object detection, image analysis, global optimization, target functions

Acknowledgements: The study was supported by grant No. 24-21-20116 from the Russian Science Foundation (<https://rscf.ru/project/24-21-20116/>) and grant No. 24-21-20116 from the St. Petersburg Science Foundation.

For citation: Pipiy G. T., Khlopin S. V., Chernenkaya L. V. Model of optical sensor for monitoring the level of oil pollution. *Journal of Instrument Engineering*. 2025. Vol. 68, N 3. P. 267–274 (in Russian). DOI: 10.17586/0021-3454-2025-68-3-267-274.

Введение. Загрязнение окружающей среды, в частности нефтью и нефтепродуктами, считается одной из самых больших проблем, от которых страдает человечество. Экологическая проблема, вызванная нефтезагрязнениями, приобретает особую остроту из-за сложности их оперативного обнаружения и удаления, а также дороговизны применяемых в настоящее время методов. Возникает вопрос, как без лабораторных методов обнаружить утечку масла в воде, например в потоке жидкости (река, канал сброса теплоносителя, трубопровод).

Одно из возможных решений — использование цифровых технологий, основанных на применении оптического датчика и компьютерного зрения, с помощью которых можно определять концентрацию примеси в режиме реального времени.

В настоящей статье представлена модель оптического датчика контроля уровня загрязнений нефтепродуктами, позволяющая определять объем примесей нефтепродуктов в воде.

Лабораторный стенд для оптического датчика контроля уровня загрязнений. Схема лабораторного стенда датчика представлена на рис. 1. Состав стенда:

- оптический датчик;
- емкость (бак), содержащая рабочий объем жидкости;
- труба перемещения жидкости;
- дозатор нефтепродуктов, через который можно поместить определенный объем нефтепродуктов в трубопровод для последующего обнаружения;
- фотокамера, используемая для получения серии цифровых изображений потока жидкости, содержащей примеси;
- персональный компьютер (ПК), предназначенный для анализа и обработки изображения;
- фотокамера датчика, позволяющая облучать поток жидкости УФ-излучением;
- УФ-светодиод, производящий ультрафиолетовое освещение с заданной частотой импульсов;
- датчик измерения скорости потока, передающий на ПК значение скорости потока воды, проходящей по трубе;
- плата управления, предназначенная для решения следующих задач:
 - определение скорости потока жидкости путем анализа показаний датчика измерения скорости потока;
 - установка частоты зажигания светодиода;
 - передача импульса синхронизации в компьютер обработки данных;
 - регулировка частоты оборотов мотора насоса;
 - лазерный модуль с лучом „Линия“, применяемый для обнаружения примесей (используется эффект преломления).

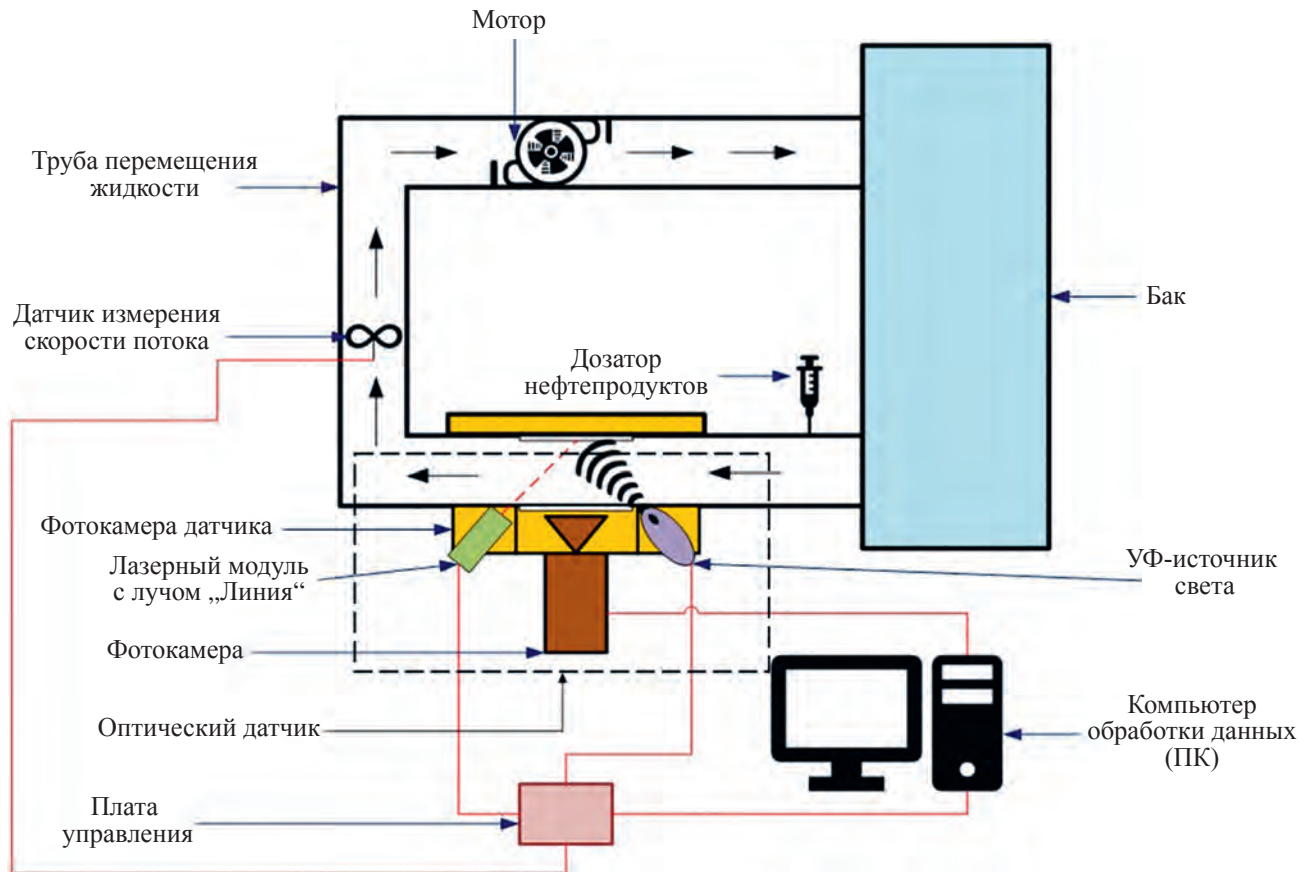


Рис. 1

Представленный стенд позволит протестировать датчик контроля в условиях, приближенных к реальным условиям эксплуатации, что обеспечит своевременное выявление недоработок в датчике, аппаратной или программной части.

Алгоритм оптимальной фильтрации и извлечения контуров бинарных изображений. Рассмотрим алгоритм анализа изображений, полученных с помощью фотокамеры датчика в процессе протекания потока воды. Алгоритм позволяет идентифицировать загрязнение окружающей среды нефтепродуктами путем фильтрации изображений и последовательного выделения контуров обработанных изображений.

Используем алгоритм вычисления содержания примесей и алгоритм выделения объектов в определенном цветовом диапазоне. Проводя детектирование вторичного флуоресцентного излучения, получаем бинарные изображения [1], представленные на рис. 2: *а* — бинарное изображение воды с примесью тяжелых нефтепродуктов при естественном освещении; *б* — бинарное изображение частиц нефтепродуктов при облучении коротковолновым ультрафиолетовым излучением в дистиллированной воде, обработанное с помощью алгоритмов эрозии и дилатации последовательно (методы *Imgproc* и библиотеки *OpenCV*).

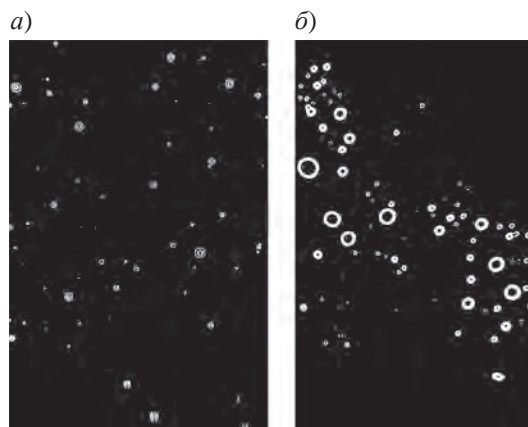


Рис. 2

Для обнаружения контуров на полученных бинарных изображениях используется алгоритм *suzuki85* [2]. Сначала необходимо заполнить границы края бинарного I^{Binary} изображения нулевыми пикселями. Исходное бинарное изображение I^{Binary} размером $M + 2 \times N + 2$ содержит значения „0“ и „1“ (ненулевые пиксели обрабатываются как „1“, а нулевые остаются равными нулю, поэтому изображение рассматривается как бинарное) для обозначения фона (background) и переднего плана (foreground) соответственно [3]. Для любого другого значения используется нумерация (маркировка) областей (region label), т. е. значения пикселей:

$$I^{\text{Regions}}(x, y) = \begin{cases} 0 & \text{— background;} \\ 1 & \text{— foreground;} \\ 2, 3, \dots & \text{— region label.} \end{cases} \quad (1)$$

Шаг 1 алгоритма *suzuki85* — сканирование входного бинарного изображения слева направо, пока не будет найден пиксел переднего плана (foreground) $I^{\text{Regions}}(x, y) \neq 0$.

Шаг 2. Определяем, является ли бинарный пиксел переднего плана внешней границей (outer border) или „дырой“ (hole), используя выражение:

$$I^{\text{Border}}(x, y) = \begin{cases} \text{ОВ, } [I^{\text{Regions}}(x, y - 1) = 0] \wedge [I^{\text{Regions}}(x, y) = 1]; \\ \text{НВ, } [I^{\text{Regions}}(x, y + 1) = 0] \wedge [I^{\text{Regions}}(x, y) \geq 1], \end{cases} \quad (2)$$

где ОВ — внешняя граница бинарного изображения (матрицы), НВ — граница „дыры“ бинарного изображения (матрицы), $I^{\text{Border}}(x, y)$ — граница пиксела после определения его типа с координатами (x, y) .

Присваиваем каждой новой найденной границе уникальный номер, обозначая ее NBD. Принимаем NBD-рамки бинарного изображения как $NBD = 1$. Остальные границы нумеруются последовательно: $NBD = NBD + 1$. Сохраняем информацию о родительском пикселе любой границы в LNBD (Last NBD). Каждый раз, когда начинаем сканировать новую строку бинарного изображения (матрицы), сбрасываем $LNBD = 1$. Далее выполняем следующие условия:

$$I^{Neighbor}(x, y) = \begin{cases} \left. \begin{array}{l} NBD = NBD + 1 \\ (x_2 = x, y_2 = y - 1) \end{array} \right\}, I^{Border}(x, y) = OB; \\ \left. \begin{array}{l} NBD = NBD + 1 \\ (x_2 = x, y_2 = y + 1) \end{array} \right\}, I^{Border}(x, y) = HB; \\ \left. \begin{array}{l} NBD = NBD + 1 \\ (x_2 = x, y_2 = y + 1) \\ LNBD = I^{Border}(x, y) \end{array} \right\}, [I^{Border}(x, y) = HB] \wedge [I^{Border}(x, y) > 1], \end{cases} \quad (3)$$

где $I^{Neighbor}(x, y)$ — соседний пиксел относительно $I^{Border}(x, y)$ с координатами (x_2, y_2) .

Если условия в формуле (3) не выполняются, то переходим к шагу 8 алгоритма. В зависимости от типа В новых найденных границ В' и границы с порядковым номером LNBD (т.е. последней границы, встречающейся в текущей строке), определяем родителя текущей границы, как показано в таблице.

В/В'	ОВ	НВ
ОВ	Родительская граница границы В'	Граница В'
НВ	Граница В'	Родительская граница границы В'

Далее выполняем следующие шаги:

Шаг 3. Начиная с $I^{Neighbor}(x, y)$, рассмотрим соседние пикселы $I^{Border}(x, y)$ по часовой стрелке и найдем ненулевой пиксел. Пусть $I^{NonZero}(x, y)$ с координатами $(x_1 = x, y_1 = y)$ — первый найденный ненулевой пиксел. Если не найдено ни одного ненулевого пикселя, то назначим NBD для $I^{Border}(x, y)$ и переходим к шагу 8.

Шаг 4. Назначим пикселу $I^{Neighbor}(x, y)$ координаты $I^{NonZero}(x, y)$ ($x_2 = x_1, y_2 = y_1$) и пикселу $I^{Border}(x, y)$ — координаты ($x_3 = x, y_3 = y$).

Шаг 5. Начиная со следующего элемента пикселя $I^{Neighbor}(x, y)$, в порядке против часовой стрелки, рассмотрим соседние пикселы текущего пикселя $I^{Border}(x, y)$ для того, чтобы найти ненулевой пиксел, и пусть первым будет $I^{NonZero}(x, y)$ с координатами (x_4, y_4) .

Шаг 6. Меняем значение пикселя $I^{Border}(x, y)$ следующим образом:

$$I^{Border}(x, y) = \begin{cases} -NBD, I(x_3, y_3 + 1) = 0, \\ NBD, [I(x_3, y_3 + 1) \neq 0] \wedge [I^{Border}(x, y) = 1]. \end{cases} \quad (4)$$

Если условия в формуле (4) не выполняются, то значение $I^{Border}(x, y)$ не меняется.

Шаг 7. Если $x_4 = x, y_4 = y$ и $x_3 = x_1, y_3 = y_1$, переходим к шагу 8. В противном случае присваиваем пикселу с координатами (x_2, y_2) значения координат ($x_2 = x_3, y_2 = y_3$) и пикселу с координатами (x_3, y_3) значения координат ($x_3 = x_4, y_3 = y_4$), затем возвращаемся к шагу 1.

Шаг 8. Если $I^{Binary}(x, y) \neq 1$, то $LNBD = |I^{Binary}(x, y)|$ и продолжаем сканирование со следующего пикселя $I^{Binary}(x, y + 1)$. Алгоритм завершается, когда сканирование достигает правого нижнего угла изображения [4].

Преимущество алгоритма suzuki85 заключается в том, что он позволяет построить иерархию контуров в бинарном изображении: см. рис. 3, а, где показаны значения пикселей, а на рис. 3, б — выделенные структуры иерархии границ (контуров).

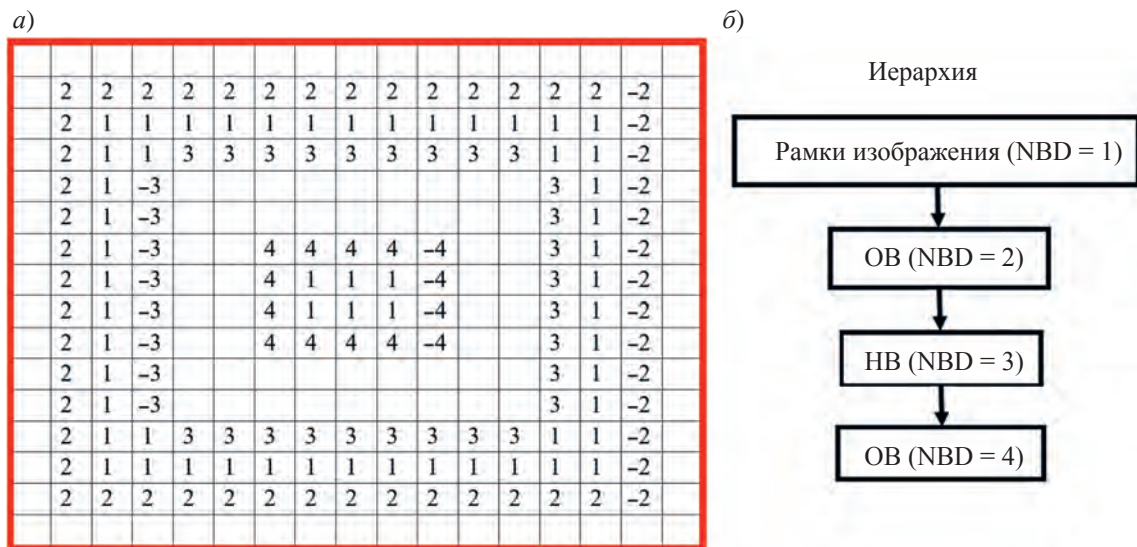


Рис. 3

Далее сохраняем каждый контур ОБ в виде последовательности точек V в ячейке массива. Для описания контуров ОБ представим их как последовательность (вектор) дискретных двумерных координат точек $V = (v_0, v_1, \dots, v_{m-1})$ по следующей формуле [2, с. 665–666]:

$$C = (v_0, v_1, \dots, v_{m-1}),$$

где $v_k = (x_k, y_k)$ — точка (пиксел) двумерного контура с координатами (x, y) и $0 \leq k \leq m$ (двумерная вещественная плоскость) [5].

На рис. 4 показан непрерывный двумерный контур, представленный в виде последовательности m , где равномерно расположены векторы $V = (v_0, v_1, \dots, v_{m-1})$ в двумерной вещественной плоскости [6].

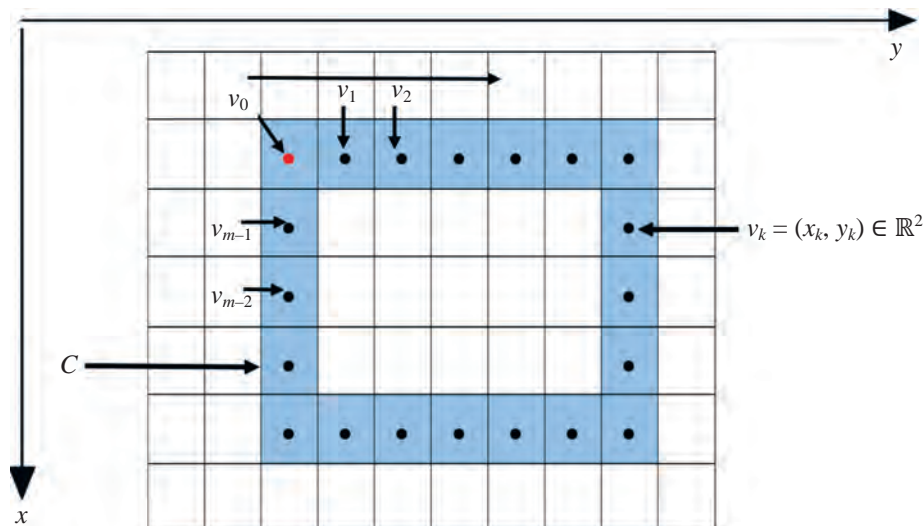


Рис. 4

Математическая модель определения объема примесей методом глобальной оптимизации. При вычислении объема примесей, образованных попаданием нефтепродуктов, полагаем, что фигура примесей близка к эллипсу или шару [7] (рис. 5).

При определении контура (выделен красным цветом на рис. 5) находим радиус вращения, равный минимальному радиусу образующегося эллипса — половине малой полуоси, большая полуось равна максимальному диаметру эллипса [8, 9]. Однако при наличии нескольких

максимумов и минимумов известные алгоритмы нахождения оптимумов возвращают только минимальное по шагу оптимальное значение, при этом данное значение может и не делить сферу или эллипс на две равные части. Пример такой неправильной фигуры представлен на рис. 6.



Рис. 5

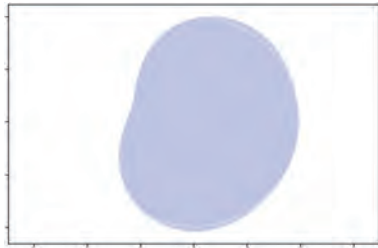


Рис. 6

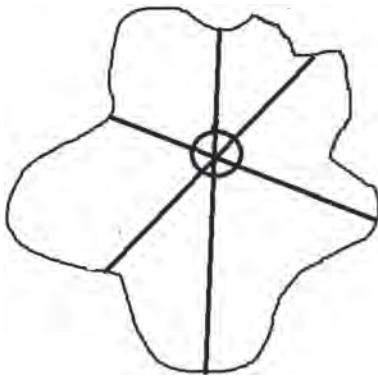


Рис. 7

На основании изложенного приходим к выводу, что проблема вычисления объема примесей заключается в разнообразии фигур с несколькими окрестностями максимальных и минимальных радиусов, что не поддается описанию классическими методами вычисления геометрических параметров эллипсов и (или) сфер [10].

Рассмотрим следующую задачу. Пусть дана область определения окружности или сферы в двумерном пространстве S^2 , в которой существует внутреннее множество $\text{int}T \subset S^2$, такое что $\forall \bar{x} \in \bar{X} = [\bar{x}: \bar{x} \geq x \in X]$, где $|\bar{x}|$ — нормированная метрика, определяющая максимальный радиус эллипса [11]. Соответственно, выделим подмножество \bar{X} , где $\bar{X} \subset X \subset \text{int}T \subset S^2$. Для решения задачи необходимо найти область на множестве $\text{int}T \subset S^2$, удовлетворяющую решениям $|\bar{x}| \geq |x|$, затем на множестве \bar{X} необходимо найти центр тяжести x_c , удовлетворяющий условию $\bar{x}_1 = \bar{x}_2 = \dots = \bar{x}_n$, где n — количество найденных точек максимума на множестве \bar{X} . По аналогии с поиском максимума необходимо найти точки минимума $|\bar{x}| \leq |x|$, удовлетворяющие условию $\bar{x}_1 = \bar{x}_2 = \dots = \bar{x}_n$. После выполнения вычислений должны быть образованы пересекающиеся векторы на неправильной фигуре, как показано на рис. 7.

Для решения приведенной задачи воспользуемся методом двухуровневой нелинейной оптимизации.

Задачу двухуровневой нелинейной оптимизации запишем следующим образом [12, 13]:

$$\begin{aligned} \min & F(x, y(x)); \\ G(x, y(x)) & \leq 0; \\ \min_{y \in Y} & f(x, y); \\ g((x, y)) & \leq 0, \end{aligned}$$

где $x \in X$, $X \subset R^{n_1}$ и $Y \subset R^{n_2}$ — закрытые множества; $G(\dots): X \times Y \rightarrow R^l$ и $g(\dots): X \times Y \rightarrow R^m$ образуют множество функционалов, а $F(\dots), f(\dots)$ — выходные функции, полученные на пересечении двух множеств.

Перечислим условия, обеспечивающие решение задачи двухуровневой нелинейной оптимизации [14, 15]:

— область допустимых решений:

$$S = \{(x, y) | x \in X, y \in Y, G(x, y) \leq 0, g(x, y) \leq 0\};$$

— допустимая область $y(x) \in Y$ при фиксированных $x \in X$:

$$Y(x) = \{y(x) \in Y | g(x, y) \leq 0\};$$

— рациональная реакция или условие оптимума $y(x) \in Y$:

$$O(x) = \{y(x) \in Y | y \in \arg \min_{y \in Y(x)} f(x, y)\};$$

— условие существования оптимальных точек $y(x) \in Y, x \in X$:

$$\bar{S} = \{(x, y) | (x, y) \in S, y \in O(x)\}.$$

Для вычисления объема нефтепродуктов в капле воды необходимо решить три подзадачи.

1. Построить множество $O_1(x) = \{y(x)_1 \in Y_1 | y_1 \in \arg \min_{y_1 \in Y(x)_1} f(x, y)\}$.
2. Построить множество $O_2(x) = \{y(x)_2 \in Y_2 | y_2 \in \arg \min_{y_2 \in Y(x)_2} f(x, y)\}$.
3. Решить задачу

$$\begin{cases} \frac{\max F(x, y(x)) - \min F(x, y(x))}{2}; \\ y_1 \in \arg \min_{y_1 \in Y(x)_1} f(x, y), \\ y_2 \in \arg \min_{y_2 \in Y(x)_2} f(x, y). \end{cases}$$

Рассмотренная задача определения объема примесей с помощью решения задачи двухуровневой оптимизации позволит найти максимальный и минимальный радиусы в неправильной фигуре и точку пересечения найденных радиусов, что обеспечит восстановление объема неправильной фигуры.

Заключение. Предложен алгоритм, позволяющий фильтровать контуры изображений таким образом, чтобы они содержали только границы, не нарушая при этом обнаружение объектов на бинарных изображениях, и аппроксимировать размер кадров цифрового изображения в соответствии с предполагаемым физическим объемом нефтесодержащей примеси.

Последующая обработка серии фотоизображений (видеорядов) позволит создать системы динамического контроля уровня загрязненности жидкости. По результатам представленного исследования возможно создание рабочего прототипа датчика для определения количественного состава примесей.

СПИСОК ЛИТЕРАТУРЫ

1. Салех Л. О. А., Хлопин С. В., Черненькая Л. В., Тарасевский Ф. Г., Царев М. М. Алгоритм определения концентрации примесей в жидкости по оптическим данным // Изв. ТулГУ. Технические науки. 2023. № 1. С. 247–256.
2. Suzuki S. Topological Structural Analysis of Digitized Binary Images by Border Following // Computer Vision, Graphics, and Image Processing. 1985. N 30. P. 32–46.
3. Shaban A., Eddaif L., Telegdi J. Sensors for water and wastewater monitoring // Advanced Sensor Technology Biomedical, Environmental, and Construction Applications. 2023. Ch. 14. P. 517–563.
4. Coupling membrane processes with wet air oxidation for the remediation of industrial effluents / E. Gout, F. Toure Lo, M. Monnot, O. Boutin, P. Vanloot, M. Claeys-Bruno, P. Moulin // Chemical Engineering Journal. 2023. Vol. 472 (15). P. 144937.
5. Белов А. В. Микроконтроллеры AVR: от азов программирования до создания практических устройств. СПб: Наука и техника, 2016. 544 с.
6. Хлопин С. В., Салех Л. О. А. Программная реализация оптического метода динамического вычисления объема содержания примесей в жидкости // Системный анализ в проектировании и управлении: Сб. науч. тр. XXV Междунар. науч. конф. СПб, 2021. С. 229–235.
7. Садыков С. С., Терехин А. В. Классификация геометрических фигур, вписанных в квадрат, по их форме с использованием диагональных признаков // Прикаспийский журнал: управление и высокие технологии. 2014. № 1 (25). С. 081–092.
8. Маклашов В. А., Пиганов М. Н. Алгоритм для определения смещения оценок координат // Тр. МАИ. 2022. № 122. С. 10.
9. Ушаков В. Н., Лебедев П. Д., Лавров Н. Г. Алгоритмы построения оптимальных упаковок в эллипсы // Вестн. Южно-Уральского гос. ун-та. Сер. Математическое моделирование и программирование. 2017. Т. 10, №. 3. С. 67–79.
10. Сюй А. В., Кравцова Н. А., Строганов В. И., Криштон В. В. Параметрический метод построения эллипса поляризации излучения // Изв. вузов. Приборостроение. 2008. Т. 51, № 10. С. 65–69.

11. Клочков Ю. В., Николаев А. П., Киселева Т. А. К вопросу о непрерывной параметризации пространственных фигур, имеющих в сечении эллипс // Изв. вузов. Математика. 2017. № 9. С. 30–35.
12. Chen Y., Florian M. The nonlinear bilevel programming problem: Formulations, regularity and optimality conditions // Optimization. 1995. Vol. 32, N 3. P. 193–209.
13. Edmunds T. A., Bard J. F. Algorithms for nonlinear bilevel mathematical programs // IEEE Trans. on Systems, Man, and Cybernetics. 1991. Vol. 21, N 1. P. 83–89.
14. Sinha A., Malo P., Deb K. A review on bilevel optimization: From classical to evolutionary approaches and applications // IEEE Trans. on Evolutionary Computation. 2017. Vol. 22, N 2. C. 276–295.
15. Colson B., Marcotte P., Savard G. An overview of bilevel optimization // Annals of Operations Research. 2007. Vol. 153. P. 235–256.

СВЕДЕНИЯ ОБ АВТОРАХ

Георгий Тенгизович Пипия

— канд. техн. наук; Санкт-Петербургский политехнический университет Петра Великого, Высшая школа компьютерных технологий и информационных систем; инженер-исследователь, E-mail: gogpipiy@ya.ru

Сергей Владимирович Хлопин

— канд. техн. наук, доцент; Санкт-Петербургский политехнический университет Петра Великого, Высшая школа компьютерных технологий и информационных систем; E-mail: x@spbstu.ru

Людмила Васильевна Черненькая

— д-р техн. наук, профессор; Санкт-Петербургский политехнический университет Петра Великого, Высшая школа компьютерных технологий и информационных систем; E-mail: ludmila@qmd.spbstu.ru

Поступила в редакцию 26.06.24; одобрена после рецензирования 01.07.24; принята к публикации 28.01.25.

REFERENCES

1. Saleh L.O.A., Khlopin S.V., Chernenkaya L.V., Tarasevskii Ph.G., Tsarev M.M. *News of the Tula state university. Technical sciences*, 2023, no. 1, pp. 247–256. (in Russ.)
2. Suzuki S., Abe K. *Computer Vision, Graphics, and Image Processing*, 1985, no. 1(30), pp. 32–46, DOI:10.1016/0734-189x(85)90016-7.
3. Shaban A., Eddaif L., Telegdi J. *Advanced Sensor Technology Biomedical, Environmental, and Construction Applications*, Elsevier, 2023, pp. 517–563.
4. Gout E., Toure Lo F., Monnot M., Boutin O., Vanloot P., Claeys-Bruno M., Moulin Ph. *Chemical Engineering Journal*, 2023, vol. 472, pp. 144937.
5. Belov A.V. *Mikrokontrollery AVR: ot azov programmirovaniya do sozdaniya prakticheskikh ustroystv* (AVR Microcontrollers: from the Basics of Programming to Creating Practical Devices), St. Petersburg, 2016, 544 p. (in Russ.)
6. Khlopin S.V., Saleh L.O.A. *Sistemnyy analiz v proyektirovani i upravlenii* (Systems Analysis in Design and Management), Collection of Scientific Papers of the XXV International Scientific and Educational-Practical Conference, St. Petersburg, 2021, pp. 229–235. (in Russ.)
7. Sadykov S.S., Terekhin A.V. *Caspian Journal: Management and High Technologies*, 2014, no. 1(25), pp. 081–092. (in Russ.)
8. Maklashov V.A., Piganov M.N. *Trudy MAI*, 2022, no. 122, pp. 10. (in Russ.)
9. Ushakov V.N., Lebedev P.D., Lavrov N.G. *Bulletin of the South Ural State University. Series: Mathematical Modeling and Programming*, 2017, no. 3(10), pp. 67–79. (in Russ.)
10. Syuy A.V. et al. *Journal of Instrument Engineering*, 2008, no. 10(51), pp. 65–69. (in Russ.)
11. Klochkov Y.V., Nikolaev A.P., Kiselyova T.A. *Russian Mathematics*, 2017, no. 9, pp. 27–31.
12. Chen Y., Florian M. *Optimization*, 1995, no. 3(32), pp. 193–209.
13. Edmunds T.A., Bard J.F. *IEEE Transactions on Systems, Man, and Cybernetics*, 1991, no. 1(21), pp. 83–89.
14. Sinha A., Malo P., Deb K. *IEEE Transactions on Evolutionary Computation*, 2017, no. 2(22), pp. 276–295.
15. Colson B., Marcotte P., Savard G. *Annals of Operations Research*, 2007, vol. 153, pp. 235–256.

DATA ON AUTHORS

Georgy T. Pipiya

— PhD; Peter the Great St. Petersburg Polytechnic University, Higher School of Computer Technologies and Information Systems; Engineer-Researcher, E-mail: gogpipiy@ya.ru

Sergey V. Khlopin

— PhD, Associate Professor; Peter the Great St. Petersburg Polytechnic University, Higher School of Computer Technologies and Information Systems; E-mail: x@spbstu.ru

Ludmila V. Chernenkaya

— Dr. Sci., Professor; Peter the Great St. Petersburg Polytechnic University; Higher School of Computer Technologies and Information Systems; E-mail: ludmila@qmd.spbstu.ru

Received 26.06.24; approved after reviewing 01.07.24; accepted for publication 28.01.25.