

К.В. СМЕРНОВ<sup>1</sup>, Б.А. ФАУСТОВ<sup>1</sup>, И.Л. ФАУСТОВА<sup>2</sup>

## СИСТЕМА РАСПРЕДЕЛЕННОГО ВЫЧИСЛЕНИЯ ПОЛЯ СКОРОСТЕЙ

Рассмотрена реализация метода определения поля скоростей, который можно применять для нахождения оптического потока радионуклидных изображений. Поле скоростей – одна из форм представления движения объектов на изображении, при котором каждой точке изображения ставится в соответствие вектор скорости, отвечающий соответствующей точке на объекте. Приведено математическое описание используемого метода и его реализация с применением облачных технологий.

**Ключевые слова:** поле скоростей, оптический поток, обработка изображений.

### Введение

Информатизация технологий и объектов атомной отрасли в настоящее время является актуальной задачей. Методы обработки и анализа изображений применяются в самых различных информационных и технических системах. В данной работе рассмотрена реализация метода определения поля скоростей, который можно применять для нахождения оптического потока радионуклидных изображений. Поле скоростей – одна из форм представления движения объектов на изображении, при котором каждой точке изображения ставится в соответствие вектор скорости, отвечающий соответствующей точке на объекте. Построение поля скоростей было проведено многими авторами. Наиболее известная постановка задачи с использованием понятия оптического потока предполагает постоянство функции плотности распределения вдоль траекторий изучаемой системы [1]. В работе приведено математическое описание используемого метода [2] и его реализация с применением облачных технологий.

### Описание метода

Возьмем последовательность дискретных изображений какого-либо предмета. Обозначим ширину и высоту изображения из последовательности  $w$  и  $h$  соответственно. Количество изображений обозначим  $d$ . При переходе от одного изображения к другому каждая точка с координатами  $(x_1, x_2) \in \Omega$ ,  $\Omega = \{x_1 \in [0, w + 1], x_2 \in [0, h + 1]\}$  меняет цвет.

Для упрощения описания будем считать, что изображения полутонные и цвет каждой точки кодируется одним числом. Далее эту числовую характеристику цвета будем называть яркостью и обозначать как  $I_i(x) = I_i(x_1, x_2)$ , где  $i$  – номер изображения в последовательности. В случае цветного изображения каждый цвет рассматривается отдельно. Поле скоростей для пары изображений с номерами  $i$  и  $i + 1$  будем обозначать как

$$u(x_1, x_2) = \begin{pmatrix} u_1(x_1, x_2) \\ u_2(x_1, x_2) \end{pmatrix}, (x_1, x_2) \in \Omega. \quad (1)$$

Метод оптического потока базируется на задаче минимизации интеграла. В случае, если рассматривается неизменность яркости объектов при переходе от одного изображения к другому, это можно выразить следующим образом:

$$\int_{\Omega} \left( [I_{i+1}(x + u) - I_i(x)]^2 \right) dx + \alpha \int_{\Omega} \left( \|\nabla u_1\|^2 + \|\nabla u_2\|^2 \right) dx \xrightarrow{u} 0, x \in \Omega. \quad (2)$$

Первый интеграл отвечает за постоянство яркости, второй – за гладкость потока,  $\alpha$  – параметр регуляризации. Переходя к уравнениям Эйлера – Лагранжа, получаем систему относительно  $u$

$$\begin{cases} \frac{\partial I_i(x)}{\partial x} [I_{i+1}(x + u) - I_i(x)] + \alpha \nabla^2 u_1 = 0, \\ \frac{\partial I_i(x)}{\partial y} [I_{i+1}(x + u) - I_i(x)] + \alpha \nabla^2 u_2 = 0, \end{cases} \quad (3)$$

где  $\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$  – лапласиан.

При переходе к дискретной сетке его можно раскрыть с помощью формул приближенного вычисления производных [3] и получить систему линейных уравнений для расчета поля скоростей  $u(x_1, x_2), (x_1, x_2) \in \Omega$ .

Например, можно использовать следующую итерационную схему:

$$\begin{aligned} (I_{i,z}^k + I_{i,x}^k u_1^k + I_{i,y}^k u_1^k) - \alpha \nabla^2 u_1^{k+1} &= 0, \\ (I_{i,z}^k + I_{i,x}^k u_1^k + I_{i,y}^k u_2^k) - \alpha \nabla^2 u_2^{k+1} &= 0, \end{aligned} \quad (4)$$

где  $I_{i,x} = \frac{\partial I_i(x)}{\partial x}$ ,  $I_{i,y} = \frac{\partial I_i(x)}{\partial y}$ ,  $I_{i,z} = \frac{\partial I_i(x)}{\partial z}$  – частные производные яркости по координатам и времени,  $k$  – номер итерации в процессе вычисления  $u_1, u_2$ . Частные производные будем оценивать с помощью формул конечных разностей.

### Общая схема

Для построения схемы вычислительного процесса за основу возьмем итерационный процесс (4), для ускорения сходимости этот алгоритм будем применять не к исходному изображению, а к пирамиде изображений, построенной на его основе. Основанием пирамиды является пара исходных изображений, каждый следующий уровень представляет собой вдвое меньшие по размерам усредненные изображения.

Исходное изображение можно разделить на несколько изображений меньшего размера делением пополам или на четыре части. Таким образом, оценка поля скоростей по формулам (4) может выполняться параллельно на нескольких узлах.

Схема вычислительного процесса показана на рис. 1.

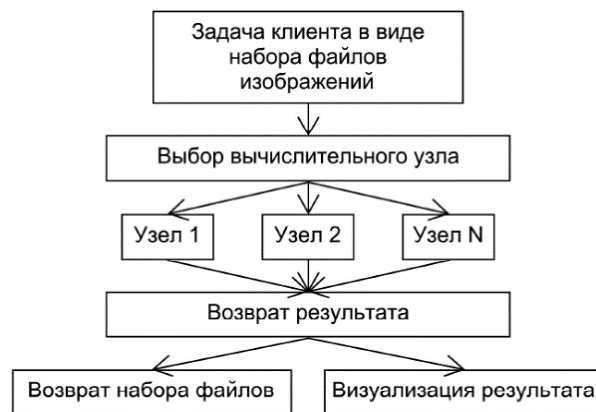


Рис. 1. Общая схема вычислительного процесса

Пользователь на вход подает несколько изображений, для каждой пары необходимо вычислить поле скоростей. Процесс вычисления состоит из следующих этапов:

1. Выбор свободных вычислительных узлов для новой задачи в требуемом количестве.
2. Загрузка данных на вычислительные узлы.
3. Вычисление поля скоростей для задачи.
4. Предоставление результата пользователю в виде набора файлов.

### Средства реализации

Для реализации web-сервиса были выбраны следующие средства.

1) Негоку. Облачный сервис Негоку предоставляет вычислительные ресурсы для размещения проектов, созданных на базе различных языков программирования: PHP, Java, Ruby, Python и некоторые другие. Каждый проект запускается в отдельном контейнере, что обеспечивает его защищенность и легкую масштабируемость.

В описываемой реализации используется вариант однопоточных проектов для организации вычислительных сервисов, каждый из которых может вычислять поле скоростей для набора изображений, и сервиса-контроллера, который будет предоставлять информацию о состоянии вычислительных сервисов. Все они реализованы с использованием языка программирования JAVA.

Доступ к вычислительным сервисам (вычислительные узлы) организуется через POST-запрос.

2) Heroku DB. Кроме вычислительных ресурсов облачный сервис Heroku предоставляет возможность размещения базы данных под СУБД PostgreSQL, эта возможность используется для хранения состояний вычислительных узлов. Каждый из них в начале вычисления задачи отмечает в базе, что он занят. При завершении вычисления задачи – что он освобожден.

3) Файловый сервер и web-сервис. Основной сайт, через который предоставляется доступ к вычислительной системе, представляет собой отдельный сервис, реализованный на PHP, также развернутый на Heroku. Временное хранилище изображений, полученных от пользователей, и результаты вычислений размещаются на ftp-сервисе byethost.

### Описание жизненного цикла задачи

Схема процесса обработки отдельной задачи представлена на рис. 2. Пользователь выбирает на основном сайте [4] набор изображений и запускает процесс обработки задачи.

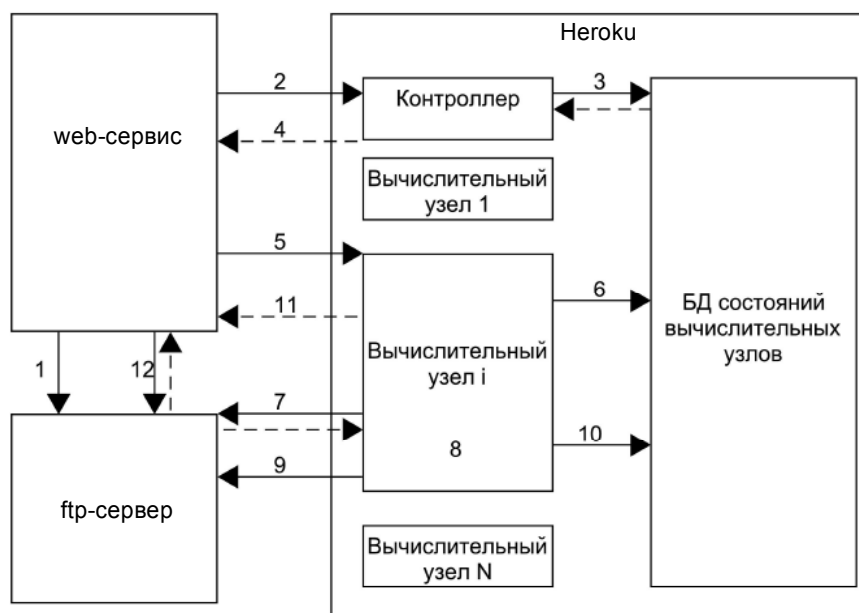


Рис. 2. Схема базы данных состояний вычислительных узлов

Далее приведено описание шагов обработки задачи:

1. Web-сервис загружает изображения на ftp-сервер.
2. Web-сервис обращается к узлу-контроллеру (Heroku) для получения информации о состоянии вычислительных узлов.
3. Узел-контроллер соединяется с базой данных состояний вычислительных узлов (Heroku PostgreSQL) и загружает информацию о состоянии системы.
4. Узел-контроллер возвращает web-сервису информацию о состоянии вычислительных узлов.
5. Web-сервис выбирает узел, на который подается запрос на обработку с параметрами количества изображений в последовательности и массив имен файлов.
6. Вычислительный узел указывает в базе данных состояний узлов Heroku PostgreSQL информацию о том, что он занят.
7. По этим ссылкам вычислительный узел загружает изображения из ftp-сервера.
8. Вычислительный узел производит решение задачи: для каждой пары изображений вычисляет поле скоростей в виде пары матриц в одном файле данных.
9. Вычислительный узел сохраняет файлы результата на ftp-сервер.

10. Вычислительный узел указывает в базе данных состояний узлов Heroku PostgreSQL информацию о том, что он свободен.
11. Web-сервис получает ответ на POST запрос от вычислительного узла, содержащий информацию о завершении обработки задачи. Пользователю предоставляется результат в виде набора ссылок на файлы результата, хранящиеся на ftp-сервере
12. Пользователь может скачать файлы результата с ftp-сервера или просмотреть визуализацию потока на сайте.

### Пример: сдвиг карты

Метод оптического потока можно применять для анализа карт местности, получаемых в результате аэрофотосъемки. Рассмотрим сдвиг карты на вектор 1, 1 вниз-вправо (рис. 3).

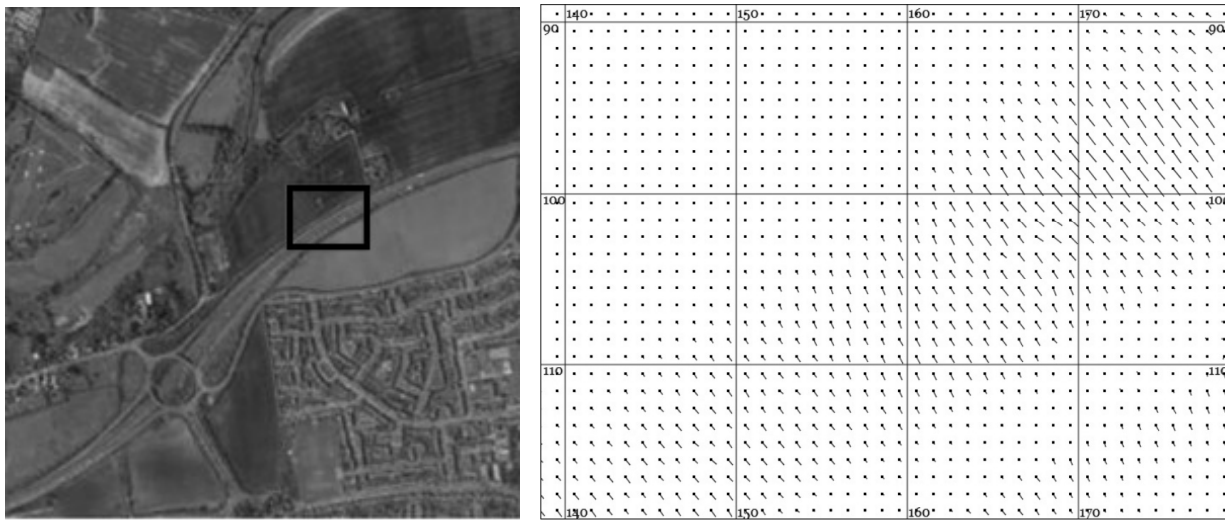


Рис. 3. Исходное изображение и результат работы

### Анализ

Для проведения анализа работы алгоритмов необходимо выбрать величины, по которым будем производить сравнение. В качестве таковых выберем величины, которые будут отражать средние отклонения вычисленных величин от реальных

$$E_{\phi} = \frac{1}{wh} \sum_{i,j=i_0,j_0}^{i_w,j_h} \phi_{ij} - \phi_0,$$

$$E_r = \frac{1}{wh} \sum_{i,j=i_0,j_0}^{i_w,j_h} (u_{xij} - u_x)^2 - (u_{yij} - u_y)^2. \quad (5)$$

Здесь область  $i = i_0 \dots i_w$ ,  $j = j_0 \dots j_h$  – фрагмент изображения размером  $wh$ ;  $\phi_0$ ,  $u_x, u_y$  – соответственно фазовый угол и компоненты известного вектора сдвига изображения;  $\phi_{ij}$ ,  $u_{xij}$ ,  $u_{yij}$  – соответственно фазовый угол и компоненты вектора, вычисленного для точки  $i, j$ .  $E_{\phi}$  измеряется в радианах, так как это нормированная сумма ошибок при определении фазового угла по всем точкам изображения.  $E_r$  измеряется в квадратных пикселях, так как мы определяем компоненты вектора скорости в пикселях, деленных на кадр.

Приведем сравнение различных вариантов линеаризации схемы (3), (4):

1. Оригинального алгоритма Horn – Schunck.
2. Модифицированного смешанного алгоритма Horn – Schunck и Лукаса – Канаде, где точке с координатами  $i_0, j_0$  соответствует множество точек

$$\{(i_0 + i, j_0 + j) \mid i \in \{-1, 0, 1\}, j \in \{-1, 0, 1\}\}.$$

3. Модифицированного смешанного алгоритма Horn – Schunck и Лукаса – Канаде с подчеркиванием границ.

Результаты сравнения методов для примера сдвига карты представлены в таблице.

**Сравнение методов для сдвига карты**

| $E_\phi$ – средняя ошибка фазового угла вектора поля скоростей     |       |       |       |       |
|--|-------|-------|-------|-------|
| Число итераций   | 5     | 20    | 40    | 60    |
| Оригинальный Horn – Schunck  | 0.516 | 0.295 | 0.236 | 0.207 |
| Смешанный Horn – Schunck и Лукаса – Канаде                         | 0.619 | 0.627 | 0.668 | 0.739 |
| Смешанный Horn – Schunck и Лукаса – Канаде с подчеркиванием границ | 0.619 | 0.627 | 0.668 | 0.739 |
| Оригинальный Horn – Schunck с подчеркиванием границ                | 0.516 | 0.295 | 0.236 | 0.207 |
| $E_r$ – средняя ошибка длины вектора поля скоростей                |       |       |       |       |
| Число итераций   | 5     | 20    | 40    | 60    |
| Оригинальный Horn – Schunck  | 2.637 | 2.498 | 2.347 | 2.246 |
| Смешанный Horn – Schunck и Лукаса – Канаде                         | 2.268 | 1.877 | 1.795 | 1.578 |
| Смешанный Horn – Schunck и Лукаса – Канаде с подчеркиванием границ | 2.268 | 1.877 | 1.795 | 1.578 |
| Оригинальный Horn – Schunck с подчеркиванием границ                | 2.637 | 2.498 | 2.347 | 2.246 |

Анализируя результаты, можно сделать вывод, что смешанный алгоритм дает более точные оценки поля скоростей.

### Заключение

Рассмотрена реализация метода определения поля скоростей, которую можно применять для анализа движения объектов на изображениях. Проведено сравнение различных вычислительных схем. Данную систему можно улучшить несколькими способами, а именно, добавить визуализацию работы описанного распределенного приложения [5], помимо обработки последовательности изображений прибавить обработку файла видео, ускорить вычислительный процесс, добавив распределение одной задачи на несколько узлов, и усовершенствовать отказоустойчивость системы при выходе из строя вычислительных узлов.

### СПИСОК ЛИТЕРАТУРЫ

1. Horn B.K.P. and Schunck B.G. // Artificial Intelligence. – 1981. – No. 17. – P. 185–203.
2. Смирнов К.В. // Процессы управления и устойчивость. – 2014. – Т. 1. – № 1. – С. 272–276.
3. Вержбицкий В.М. Основы численных методов: учебник для вузов. – М.: Высшая школа, 2002. – 840 с.
4. Velocity Flow [Электронный ресурс]: URL: <http://velocityflow.herokuapp.com> (дата обращения: 15.03.16).
5. Якушкин О.О., Гришкин В.М. // Компьютерные исследования и моделирование. – 2015. – Т. 7. – № 3. – С. 529–532.

Поступила в редакцию 20.10.2020.

<sup>1</sup> Санкт-Петербургский государственный университет, г. Санкт-Петербург, Россия

<sup>2</sup> Северский технологический институт Национального исследовательского ядерного университета «МИФИ», г. Северск, Россия

*K.V. SMIRNOV<sup>1</sup>, B.A. FAUSTOV<sup>1</sup>, I.L. FAUSTOVA<sup>2</sup>*

## **DISTRIBUTED SYSTEM FOR CALCULATING THE VELOCITY FIELD**

This paper considers the implementation of the method for determining the velocity field, which can be used to find the optical flux of radionuclide images. The velocity field is one of the forms of representing the movement of objects in the image, in which each point of the image is associated with a velocity vector that corresponds to the point on the object. The article provides a mathematical description of the method used and its implementation using cloud technologies.

**Keywords:** *velocity field, optical flow, image processing.*

<sup>1</sup> Saint-Petersburg State University, Saint-Petersburg, Russia

<sup>2</sup> Seversk Technological Institute National Research Nuclear University «MEPHI», Seversk, Russia

---

**Smirnov** Konstantin Valer'yevich, Postgraduate Student SPbU, e-mail: constcoh@gmail.com;

**Faustov** Bogdan Andreyevich, Postgraduate Student SPbU, e-mail: bogfaust@gmail.com;

**Faustova** Inna Leontievna, Candidate of Physical and Mathematical Sciences, Docent STI NRNU «MEPHI», e-mail: infaust@rambler.ru.