

# ПРЕДВАРИТЕЛЬНАЯ СЕГМЕНТАЦИЯ ИЗОБРАЖЕНИЯ ДЛЯ УСКОРЕНИЯ АЛГОРИТМОВ СТЕРЕОЗРЕНИЯ

О. А. Лавягина<sup>1</sup>, Е. В. Водолазский<sup>2</sup>

<sup>1</sup> *Национальный технический университет Украины  
«Киевский политехнический институт имени Игоря Сикорского»,  
Физико-технический институт*

<sup>2</sup> *Международный научно-учебный центр информационных технологий и систем  
НАН Украины и МОН Украины*

## Аннотация

Задача бинокулярного стереозрения является одной из актуальных проблем компьютерного зрения. В данной работе приводится постановка задачи стереозрения по двум выровненным изображениям одного объекта или поверхности, алгоритм диффузии для решения этой задачи, а также предлагается метод ускорения этого алгоритма с помощью предварительной сегментации изображения.

**Ключевые слова:** бинокулярное стереозрение, алгоритм диффузии, сегментация, карта глубин, трёхмерная модель

## Введение

В основе получения трёхмерной модели объекта или поверхности лежит построение карты глубин по паре стереоизображений, известное как задача стереозрения. Алгоритм диффузии – один из популярных методов для решения задачи стереозрения, который часто показывает хорошие практические результаты, однако работает достаточно медленно. В данной работе предложен способ ускорения алгоритма диффузии для решения задачи стереозрения с помощью предварительной сегментации изображения.

## 1. Постановка задачи

Дано левое полутоновое изображение  $L : T \rightarrow C$  и правое полутоновое изображение  $R : T \rightarrow C$ , где  $T = \{(x, y) \mid 1 \leq x \leq w, 1 \leq y \leq h\}$  – множество координат пикселей изображения,  $w$  – ширина изображения,  $h$  – высота изображения, а  $C = \{0, \dots, 255\}$  – множество интенсивностей пикселей.

Предполагается, что  $L$  и  $R$  – пара изображений неподвижной сцены, снятая одной и той же камерой. При этом, правое изображение  $R$  было получено при том же направлении съёмки, что и левое  $L$ , но при перемещении камеры строго горизонтально вправо. Это значит, что проекции объектов на изображение тоже перемещаются исключительно по горизонтали. Строим такую модель, согласно которой объекты не перекрываются друг другом. Ниже в экспериментах будет показано, что это не сильно ограничивает область применения алгоритма – на практике его можно использовать и для перекрывающихся объектов.

Таким образом, строка с вертикальной координатой  $y$  на левом изображении соответствует стро-

ке с вертикальной координатой  $y$  на правом изображении. Горизонтальная координата  $x$  каждого пикселя левого изображения  $(x, y) \in T$  сдвигается влево на величину  $d(x, y)$  для получения координаты соответствующего пикселя на правом изображении  $(x - d(x, y), y)$ . Множество всех возможных сдвигов горизонтальной координаты пикселя равно  $D = \{0, \dots, M\}$ , где  $M$  – фиксированный максимальный сдвиг.

Строится  $|T|$ -дольный граф-решётка, в котором каждой доле соответствует один пиксель (рис. 1). Будем называть доли графа объектами. Каждый объект  $(x, y) \in T$  имеет  $|D|$  вершин  $(x, y, d)$ , которые соответствуют всем возможным сдвигам  $d \in D$  пикселя  $(x, y)$ . Значения сдвигов  $d$  будем называть метками.

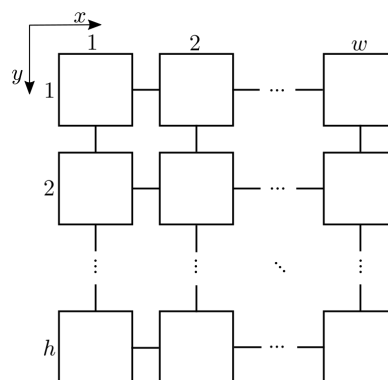


Рис. 1.  $|T|$ -дольный граф-решётка

На каждую вершину  $(x, y, d)$ ,  $d \in D$  в каждой доле графа  $(x, y) \in T$  накладывается штраф за отклонение между интенсивностями соответствующих пикселей  $f_{(x,y)}(d) = f(L(x, y), R(x - d, y))$ .

Каждый объект графа имеет не более четырёх соседних объектов: верхний, правый, нижний и левый. Объекты, соответствующие пикселям на краях изображения, имеют по три соседа, а объекты, соответствующие угловым пикселям, – по два. Пусть  $\mathcal{N}(x, y)$  – множество всех соседних объектов для объекта  $(x, y)$ . Все вершины каждого объекта соединены со всеми вершинами в соседних объектах. На дужку между вершиной с меткой  $d$  в объекте  $(x, y)$  и вершиной с меткой  $d'$  в объекте  $(x', y') \in \mathcal{N}(x, y)$  накладывается штраф за несоответствие выбранных сдвигов в соседних объектах  $g_{(x,y),(x',y')}(d, d')$ ,  $d, d' \in D$ .

Отображение  $\mathbf{d} : T \rightarrow D$  назовём разметкой. Каждому пикселю изображения (каждому объекту графа) оно ставит в соответствие метку, то есть выбирает одну и только одну вершину в каждом объекте. Задача состоит в выборе такой разметки  $\mathbf{d} \in D^T$ , которая минимизирует штрафную функцию

$$G(\mathbf{d}) = \sum_{x=1}^w \sum_{y=1}^h f_{(x,y)}(\mathbf{d}(x, y)) + \sum_{x=1}^w \sum_{y=1}^h \sum_{(x', y') \in \mathcal{N}(x, y)} g_{(x,y),(x',y')}(\mathbf{d}(x, y), \mathbf{d}(x', y')). \quad (1)$$

## 2. Алгоритм диффузии

Алгоритм диффузии (min-sum diffusion algorithm) является блочно-координатным спуском [1]. Каждой вершине  $(x, y, d)$ ,  $(x, y) \in T$ ,  $d \in D$  ставится в соответствие блок переменных  $(\varphi_{(x,y),(x',y')}(d) \in \mathbb{R} \mid (x', y') \in \mathcal{N}(x, y))$ .

Вводятся репараметризованные штрафы вершин

$$f_{(x,y)}^{\varphi}(d) = f_{(x,y)}(d) - \sum_{(x', y') \in \mathcal{N}(x, y)} \varphi_{(x,y),(x',y')}(d), \quad (x, y) \in T, \quad d \in D$$

и репараметризованные штрафы дуг

$$g_{(x,y),(x',y')}^{\varphi}(d, d') = g_{(x,y),(x',y')}(d, d') + \varphi_{(x,y),(x',y')}(d) + \varphi_{(x',y'),(x,y)}(d').$$

Элементарный шаг алгоритма диффузии состоит из двух операций для каждого объекта  $(x, y) \in T$ :

$$\begin{aligned} \forall (x', y') \in \mathcal{N}(x, y) \quad \forall d \in D \\ \varphi_{(x,y),(x',y')}^{t+1}(d) = \varphi_{(x,y),(x',y')}^t(d) - \min_{d' \in D} g_{(x,y),(x',y')}^{\varphi^t}(d, d') \end{aligned} \quad (2)$$

и

$$\begin{aligned} \forall (x', y') \in \mathcal{N}(x, y) \quad \forall d \in D \\ \varphi_{(x,y),(x',y')}^{t+2}(d) = \varphi_{(x,y),(x',y')}^{t+1}(d) + \frac{f_{(x,y)}^{\varphi^{t+1}}(d)}{|\mathcal{N}(x, y)|}, \end{aligned} \quad (3)$$

где  $t$  обозначает номер итерации. Операции (2) и (3) можно выполнять параллельно для разных объектов, так как  $(t+1)$ -й шаг требует данные только от шага  $t$ , которые зафиксированы.

На первой итерации  $\varphi_{(x,y),(x',y')}(d) = 0$  для всех объектов  $(x, y) \in T$ , всех меток  $d \in D$  и всех соседних объектов  $(x', y') \in \mathcal{N}(x, y)$ .

Известно [1], что последовательное применение операций (2) и (3) максимизирует двойственную функцию Лагранжа

$$E(\Phi) = \sum_{x=1}^w \sum_{y=1}^h \min_{d \in D} f_{(x,y)}^{\varphi}(d) + \sum_{x=1}^w \sum_{y=1}^h \sum_{(x', y') \in \mathcal{N}(x, y)} \min_{d, d' \in D} g_{(x,y),(x',y')}^{\varphi}(d, d')$$

по переменным

$$\Phi = \{(\varphi_{(x,y),(x',y')}(d) \mid (x', y') \in \mathcal{N}(x, y)) \mid (x, y) \in T, \quad d \in D\},$$

а поэтому минимизирует штрафную функцию (1).

## 2.1. Поиск карты глубин

После минимизации штрафной функции нужно найти одну из тех разметок, штраф которых равен минимальному. Для этого используется алгоритм вычёркивания второго порядка (relaxation labeling algorithm) [1]. Для гарантии существования разметки после применения алгоритма диффузии штрафная функция  $g$  должна иметь свойство супермодулярности [2].

## 3. Предварительная сегментация изображения

Исходное изображение разбивается на прямоугольную решётку с одинаковым заданным размером ячеек. Все пиксели, принадлежащие одной ячейке, делятся на две группы по средней интенсивности пикселей ячейки (рис. 2). Назовём эти группы суперпикселями. Таким образом, в процессе сегментации изображение разбивается на ячейки, в которых находится два суперпикселя: более светлый и более тёмный.



(а) Ячейка изображения



(б) Ячейка, разбитая на два суперпикселя

Рис. 2. Визуализация суперпикселей

После сегментации строится  $|T_s|$ -дольный граф с множеством координат суперпикселей

$T_s = \{(x_s, y_s, i) \mid 1 \leq x_s \leq m, 1 \leq y_s \leq n, i \in \{0, 1\}\}$ , где  $m$  – количество ячеек по горизонтали,  $n$  – количество ячеек по вертикали,  $i$  – индекс суперпикселя

в ячейке. Каждой доле соответствует суперпиксель – объект графа. Каждый объект  $(x_s, y_s, i) \in T_s$  содержит  $|D|$  меток, которые соответствуют возможным сдвигам горизонтальной координаты пикселей, принадлежащих объекту. Таким образом, все пиксели, принадлежащие одному суперпикселю, будут иметь одинаковую глубину на результирующей карте глубин.

В объекте  $(x_s, y_s, i) \in T_s$  на вершину с меткой  $d \in D$  накладывается штраф, равный сумме штрафов за выбор соответствующих вершин во всех пикселях, принадлежащих данному суперпикселю

$$f_{(x_s, y_s, i)}^s(d) = \sum_{(x, y) \in (x_s, y_s, i)} f_{(x, y)}(d).$$

Каждый объект может иметь не более девяти соседних объектов: по два объекта в верхней, правой, нижней и левой ячейках, а также второй объект, принадлежащий той же ячейке (рис. 3). Объекты, соответствующие пикселям на краях изображения, имеют по семь соседей, а объекты, соответствующие угловым пикселям, – по пять. Штраф, накладываемый на дуги между вершинами разных объектов, такой же, как и при постановке задачи без суперпикселей.

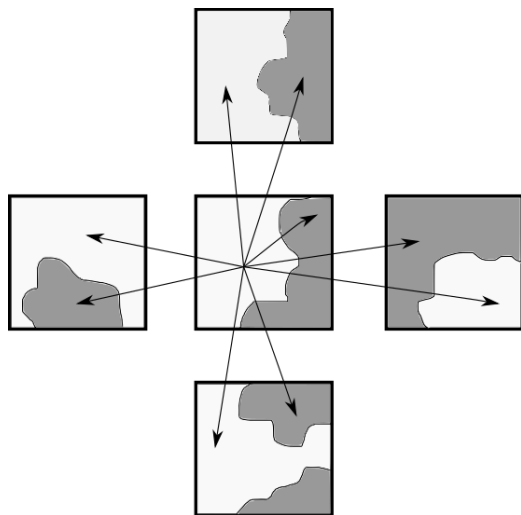


Рис. 3. Структура соседства при использовании суперпикселей. Квадратами обозначены ячейки, содержащие два суперпикселя (объекта): светлый и тёмный. Стрелками обозначены дуги, выходящие из светлого суперпикселя центральной ячейки в соседствующие с ним объекты

Аналогично штрафной функции (1) исходной задачи получаем штрафную функцию модифицированной задачи

$$G_s(\mathbf{d}) = \sum_{t_s \in T_s} \sum_{i \in \{0,1\}} \left[ f_{(t_s, i)}^s(d(t_s, i)) + \sum_{(t'_s, i') \in \mathcal{N}(t_s, i)} g_{(t_s, i), (t'_s, i')} (d(t_s, i), d(t'_s, i')) \right],$$

где  $t_s$  обозначает ячейку с координатами  $(x_s, y_s)$ .

## 4. Практические результаты

Исходные изображения были взяты из набора стереопар, которые были сделаны в Миддлберийском колледже в 2001, 2003 и 2006 годах [3] [4] [5] [6]. На рисунке 4 приведены левые изображения из стереопар, для которых строились карты глубин в данном разделе статьи.

На рисунке 5 изображены карты глубин, полученные с помощью алгоритма диффузии, где каждой доле графа соответствует один пиксель. Для каждого изображения указано количество итераций диффузии, время выполнения всех итераций, а также количество меток  $|D|$ .

На рисунке 6 изображены карты глубин, полученные с помощью алгоритма диффузии с применением предварительной сегментации изображения при размере ячеек 5 на 5 пикселей. Для каждого изображения указано количество итераций диффузии, время выполнения всех итераций, а также количество меток  $|D|$ . Получены достаточно гладкие карты глубин, однако видны неточности на краях объектов.

В качестве штрафной функции для вершин  $f$  был выбран модуль разности интенсивностей соответствующих пикселей на двух изображениях

$$f_{(x, y)}(d) = |L(x, y) - R(x - d, y)|,$$

а в качестве штрафной функции для дуг  $g$  – модуль разности выбранных сдвигов в соседних объектах

$$g_{(x, y), (x', y')} (d, d') = \alpha \cdot |d - d'|,$$

где  $\alpha = 1.4$  – коэффициент сглаживания.

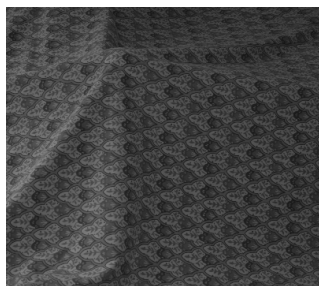
Дополнительно были введены ограничения на возможные метки в каждом объекте: для объекта  $(x, y) \in T$  с горизонтальной координатой  $x$  не может быть выбран сдвиг  $d > x$ , который перевёл бы координату пикселя в отрицательное число. Также наложены ограничения на сдвиги в соседних объектах по горизонтали:  $d' \leq d + 1$ , где  $d$  – метка в объекте  $(x, y) \in T$ ,  $d'$  – метка в объекте  $(x', y') \in \mathcal{N}(x, y)$ , таком что  $x' = x + 1$ .

Алгоритм был реализован на языке программирования Rust. В ходе работы был использован компьютер с процессором Intel(R) Core(TM) i5-7400 и ОЗУ DDR4 2133MHz.

## Выводы

Построение карты глубин – сложная задача и на сегодняшний день не решена точно. Вычислительная сложность некоторых алгоритмов решения задачи стереозрения достаточно большая, что приводит к большой длительности их работы.

Данная работа содержит постановку задачи стереозрения, а также её решение алгоритмом диффузии. Описан и испытан новый способ ускорения алгоритма с помощью предварительной сегментации изображения, при котором не теряется много информации о глубине объектов. При незначительных потерях качества удалось ускорить работу алгоритма в несколько раз.



(а) Ткань (Cloth1),  $400 \times 355$  пикселей

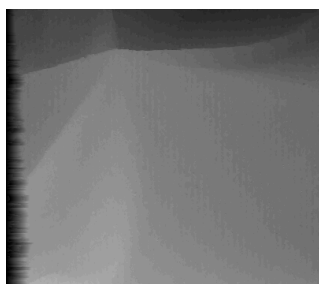


(б) Цветочные горшки (Flowerpots),  $400 \times 338$  пикселей



(в) Плакат (Poster),  $400 \times 352$  пикселей

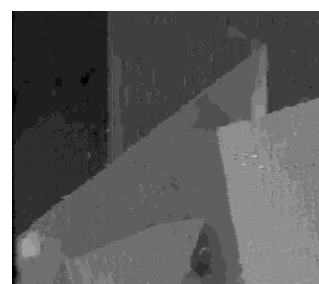
Рис. 4. Левые изображения стереопар



(а) 2'400 итераций, 5 часов 40 минут,  $|D| = 40$



(б) 2'800 итераций, 1 час 20 минут,  $|D| = 16$



(в) 1'600 итераций, 46 минут,  $|D| = 16$

Рис. 5. Карты глубин, полученные алгоритмом диффузии без применения сегментации изображения



(а) 100 итераций, 19 минут,  $|D| = 40$



(б) 450 итераций, 15 минут,  $|D| = 16$



(в) 400 итераций, 14 минут,  $|D| = 16$

Рис. 6. Карты глубин, полученные алгоритмом диффузии после применения сегментации изображения

## Перечень использованных источников

1. Savchynskyy Bogdan. Discrete Graphical Models — An Optimization Perspective // Foundations and Trends® in Computer Graphics and Vision. — 2019. — Vol. 11, no. 3-4. — P. 160-429. — URL: <http://dx.doi.org/10.1561/06000000084>.
2. Шлезингер М. И., Антонюк К. В. Анализ алгоритмов диффузии для решения оптимизационных задач структурного распознавания // Кибернетика и системный анализ. — 2011.
3. Scharstein D., Szeliski R., Zabih R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms // Proceedings IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV 2001). — 2001. — P. 131-140.
4. Scharstein D., Szeliski R. High-accuracy stereo depth maps using structured light // 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings. — Vol. 1. — 2003. — P. 195-202.
5. Scharstein Daniel, Pal Chris. Learning Conditional Random Fields for Stereo. — 2007. — 06.
6. Hirschmüller Heiko, Scharstein Daniel. Evaluation of Cost Functions for Stereo Matching. — 2007. — 06.