

УДК 004.93

**ФИЛЬТРАЦИЯ ДАННЫХ, ПОЛУЧЕННЫХ  
ТРЕХМЕРНОЙ РЕКОНСТРУКЦИЕЙ  
ПО РЯДУ ИЗОБРАЖЕНИЙ**

***В. Л. Никитенков, А. Е. Ануфриев***

*В данной статье описаны методы фильтрации данных, полученных трёхмерной реконструкцией по ряду изображений. При получении трёхмерных точек по ряду изображений часто вместе с точками интересующего нас объекта попадают точки фона и точки, которые были ошибочно распознаны как похожие (например, точки неба за объектом), поэтому необходимо делать фильтрацию точек фона и объекта до самой трёхмерной реконструкции. Кроме удаления лишней данных для обработки, удаление точек фона до этапа реконструкции приводит к тому, что в алгоритм вычисления трёхмерных точек и параметров камеры не попадают точки с большим соотношением  $st/rix$ , что приводит к более быстрой сходимости и лучшему решению систем уравнений, описывающих положения камер.*

**Ключевые слова:** *фильтрация трёхмерных точек, вычисление точек фона.*

## **1. Введение**

Трёхмерная реконструкция по ряду фотографий применяется во многих областях человеческой деятельности, таких как вычисление мерок для пошива одежды, размеров и объема тел, сохранение данных о памятниках культуры в виде фотографий и облака точек. В основе трёхмерной реконструкции по одной камере лежит следующая последовательность шагов:

1. Поиск особых точек на изображениях [7, с. 209-222].
2. Поиск соответствующих точек на соседних изображениях [7, с. 222-234].

3. Вычисление внутренних и внешних параметров камеры [6, с. 155-157].
4. Вычисление точек в трёхмерном пространстве с точностью до масштаба.
5. Нахождение масштаба по шаблонному объекту на изображениях.

Но в силу искажений на объективах, изменения освещённости в различные моменты съёмки и влияния фона на нахождение точек соответствия на парах изображений полученный результат будет содержать много ложных точек. Также результат реконструкции будет зависеть от того, насколько далеко находятся найденные соответствующие точки изображений от центра камеры в мировых координатах.

Кроме самой реконструкции актуальным является вопрос извлечения нужной информации из облака точек, но, в отличие от задач выделения данных из облака точек при лазерном сканировании, в случае реконструкции по ряду изображений имеется дополнительная информация по цвету каждой точки и есть возможность найти связные области сразу при выделении одинаковых по цвету областей на изображении.

В данной статье будут описаны алгоритмы фильтрации на примере трёхмерной реконструкции по фотографиям торцов штабелей круглых лесоматериалов при допущении некоторых ограничений по выбору точек съёмки.

## 2. Удаления точек фона на изображении

Будем считать, что объект, который интересует нас на изображении, находится в центре изображения и занимает большую его часть. Выделим из центра изображения его часть  $IC$  размером  $N \times N$ , где  $N$  подбирается эмпирически и в рассматриваемом нами случае окно должно вмещать в себя несколько торцов. Для множества пикселей  $X = \{x_1, x_2, \dots, x_n\}$ ,  $x_i = (r_i, g_i, b_i)$ , где  $n = N \times N$ , принадлежащих  $IC$ , рассчитаем центры цветовых кластеров и пороги, по которым будем определять, принадлежит ли пиксель цветовому кластеру, методами *k-means* [3, с. 424-430] или *Mixtures of Gaussian* [3, 3, с. 430-439]. Обозначим через  $k$  — число кластеров, на которое разделим множество  $X$ . После применения метода *Mixture of Gaussian* получим множество центров кластеров  $M = \{\mu_1, \mu_2, \dots, \mu_k\}$  и ковариационные матрицы для каждого центра  $\Sigma = \{\Sigma_1, \Sigma_2, \dots, \Sigma_k\}$ , по которым можно оценить достоверность принадлежности пикселя к каждому кластеру. Так как в алгоритме *k-means* рассчитываются только центры кластеров, то пороги будем рассчитывать по следующей формуле: пусть  $X_l = \{x_{1_l}, x_{2_l}, \dots, x_{n_l}\}$

подмножество пикселей из  $X$ , для которых минимум евклидова расстояния до центров кластеров достигается на центре  $\mu_l, l \leq l \leq k$ , тогда

$$\Sigma_l = \begin{bmatrix} \sigma_l(1) & 0 & 0 \\ 0 & \sigma_l(2) & 0 \\ 0 & 0 & \sigma_l(3) \end{bmatrix}, \quad \text{где } \sigma_l = \frac{1}{n_l} \cdot \sum_{x \in X_l} \begin{bmatrix} (\mu_l(1) - x(1))^2 \\ (\mu_l(2) - x(2))^2 \\ (\mu_l(3) - x(3))^2 \end{bmatrix}.$$

Таким образом подобран критерий, по которому можно отнести каждый пиксель изображения либо к фону, либо к объекту. Экспериментально для задачи реконструкции торцов оптимальным числом кластеров было  $k$ , равное четырём.

Удаление точек фона при поиске соответствий на изображениях ускоряет алгоритм вычисления параметров камер и удаляет из обработки точки, цена деления  $\text{cm}/\text{pix}$  в которых больше, чем на интересующем нас объекте, что приводит к более точным оценкам параметров камер и координат реконструированных точек (см. рис. 1).

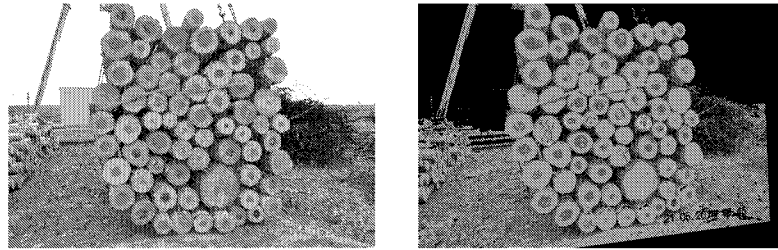


Рис. 1. Слева: исходное изображение; справа: отфильтрованное изображение с  $k$ , равным 3

### 3. Фильтрация точек при построении карты глубины

Допустим, что имея ряд изображений  $\{I_1, I_2, \dots, I_n\}$  и соответствующие точки между несколькими соседними изображениями, мы оценили внутренние и внешние параметры камер алгоритмами Incremental SFM или Global SFM и с точностью до масштаба знаем координаты центров камер и их повороты в пространстве  $\{[R_1|t_1], [R_2|t_2], \dots, [R_n|t_n]\}$ , где  $R_i \in R^{3 \times 3}$  и  $t_i \in R^{3 \times 1}$ .

Зная внешние параметры камер изображений, можно оценить фундаментальную матрицу [1, 241-247] между соседними изображениями и ректифицировать пары изображений, то есть, используя эппиполярные ограничения, преобразовать исходные изображения (рис. 2) так, чтобы соответствующие точки лежали на одинаковых строках изображения (рис. 3).



Рис. 2. Исходные изображения



Рис. 3. Ректифицированные изображения

После выполненного преобразования для каждой точки левого изображения необходимо найти соответствующую точку на правом изображении, но из-за погрешности вычисления фундаментальной матрицы лучше искать соответствия не только на одной строке, но и несколькими строками выше и ниже. В программе, написанной для вычисления плотного облака точек круглых лесоматериалов по карте глубины, был выбран порог для поиска на одну строчку выше и ниже, что дало вполне приемлемый результат. Пусть для некоторых точек из левого изображения мы нашли соответствующие точки на правом изображении, обозначим это множество точек символом  $L = \{(x_1^l, y_1^l), (x_2^l, y_2^l), \dots, (x_n^l, y_n^l)\}$ , а соответствующие им точки правого изображения — символом  $R = \{(x_1^r, y_1^r), (x_2^r, y_2^r), \dots, (x_n^r, y_n^r)\}$ , тогда, если представить карту глубины как двумерный массив из целых чисел размером  $h \times w$ , где  $h$  и  $w$  — высота и ширина исходных изображений, значения в данном массиве можно представить как

$$DepthMap(y, x) = \begin{cases} -2 \cdot w, & (x, y) \notin G \\ x_i^r - x_i^l, & (x, y) = (x_i^l, y_i^r) \in G \end{cases} .$$

Сама задача построения карты глубины является весьма сложной и ресурсоёмкой, так как требует сравнения большого числа дескрипторов пикселей на соседних изображениях. Простейший способ построения карты глубины — метод *normalized cross correlation* (NCC) [2], а более точным методом является применение дескрипторов DAISY [1], но применение DAISY требует больших расходов оперативной памяти. Стоит также заметить, что хорошим улучшением в построении карты глубины является предварительное вычисление границ объектов на изображении с помощью краевого фильтра Canny [4] и выделение этих точек в множестве  $L$  как устойчивых, так как наиболее вероятно, что в окрестности краевых точек карта глубины вычислена с хорошей точностью.

Допустим, что одним из указанных выше методом мы получили карту глубины, выделили на ней устойчивые краевые точки и применили к карте глубины фильтр Гаусса, или медианный фильтр (рис. 4). На карте глубины, приведенной в примере видно, что поверхность торцов состоит из крупных однородных частей, а фоновые объекты состоят из множества разнородных по глубине частей, чем и можно воспользоваться для дальнейшей фильтрации. Выделим на карте глубины области связности по критерию близости значения глубины для соседних пикселей и отсечём области, меньшие по площади, чем заранее заданный или вычисленный порог. После предыдущего шага добавим к карте глубины ранее выделенные краевые точки (рис. 5), если они были удалены, а затем восстановим соответствующие точки и применим к ним операцию, обратную операции ректификации.

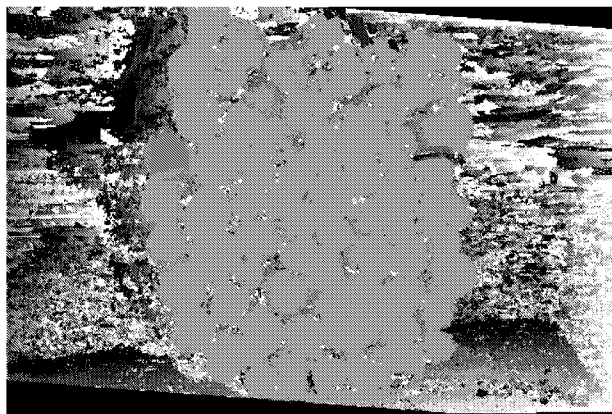


Рис. 4. Исходная карта глубины

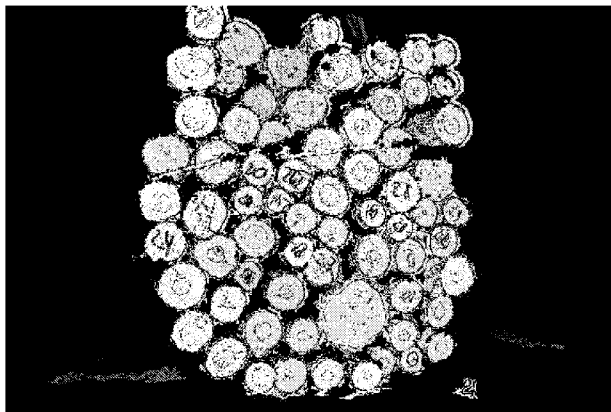


Рис. 5. Отфильтрованная карта глубины

#### 4. Фильтрация данных на этапе обработки трёхмерных точек

После вычисления соответствующих точек на соседних парах изображений внутренних и внешних параметров камер можно применить метод триангуляции и вычислить трёхмерное облако точек  $P = (x_1, y_1, z_1), (x_2, y_2, z_2), \dots, (x_n, y_n, z_n)$  (рис. 6). Даже после фильтрации на этапах ректификации и построения карты глубины остаются точки фона, которые при дальнейшей кластеризации будут увеличивать время работы алгоритмов, поэтому полезно сделать удаление фоновых объектов до этапа кластеризации. Рассмотрим эту задачу на примере облака точек торцевой стороны круглых лесоматериалов. Основным инструментом на данном этапе будет построение гистограмм плотности точек по осям координат, но данные гистограммы будут зависеть от того, какие повороты были применены к облаку точек в пространстве, поэтому нужно применить к облаку точек повороты так, чтобы ось  $Y$  была примерно перпендикулярна плоскости, на которой стоит штабель с лесоматериалами, ось  $X$  проходила через плоскость, образованную торцами, и была перпендикулярна оси  $Y$ , а ось  $Z$  была перпендикулярна на плоскости  $OXY$ .

Первым шагом данного метода будет оценка плоскости, проходящей через поверхности торцов. Для того чтобы устойчиво оценить параметры, воспользуемся методом RANSAC (Random Sample Consensus). Уравнение искомой плоскости выглядит следующим образом:  $A \cdot x + B \cdot y + C \cdot z + D = 0$ , тогда оценку параметров  $A, B, C, D$  можно описать следующими шагами:

1. Ввести переменные:  $bestQuality = 0$  — для оценки плоскости,  $nIterations$  — число итераций для алгоритма,  $nPoints$  — число точек для оценки параметров плоскости.
2. Пока  $nIterations > 0$ , выполнить следующие действия:
  - (a) Сделать случайную выборку  $P' \subset P$  размером  $nPoints$ .
  - (b) Оценить параметры плоскости  $A', B', C', D'$  по выборке  $P'$ .
  - (c) Вычислить количество точек, лежащих близко к плоскости, и записать результат в переменную  $tmpQuality$ .
  - (d) Если  $tmpQuality > bestQuality$ :  $bestQuality = tmpQuality$ ;  $A = A'; B = B'; C = C'; D = D'$ .
  - (e)  $nIterations = nIterations - 1$ .

Оценка числа итераций и сам метод RANSAC описаны в статье [5]. После применения данного алгоритма получим оценку плоскости, проходящей через торцы. Обозначим полученную плоскость через  $L_{oxy}$ . Далее для оценки необходимых поворотов воспользуемся координатами центров камер  $t_i, 1 \leq i \leq n$ . Будем считать, что несколько снимков из ряда были сделаны приблизительно на одном уровне от земли, что вполне реализуемо в современных цифровых фотоаппаратах с акселерометром. Тогда из полученных координат камер можно сделать выборку так, чтобы они лежали на одной прямой, методом RANSAC. Обозначим эту выборку символом  $T$ . Построим плоскость через выбранные точки, перпендикулярную плоскости  $L_{oxy}$ , и обозначим эту плоскость через  $L_{oxz}$ . За центр координат возьмём точку проекции  $\frac{1}{|T|} \sum_{t \in T} t$  на плоскость  $L_{oxy}$ . За ось  $Y$  примем перпендикуляр к плоскости  $L_{oxz}$ , так чтобы проекция точек, лежащих в положительной части оси  $Y$ , лежали в верхней части изображения, а точки в отрицательной части лежали в нижней части изображения (имея все параметры камеры, легко найти проекцию трёхмерных точек на плоскость изображений). За ось  $Z$  возьмём перпендикуляр к плоскости  $L_{oxy}$ , а ось  $X$  вычислим как векторное произведение направляющих векторов осей  $Y$  и  $Z$ . Теперь применим к облаку точек смещение и поворот так, чтобы новая система координат совместилась со старой, и получим множество точек, по которому можно строить гистограмму распределения точек и отсекал точки, которые лежат в хвостах распределений (рис. 6, 7).

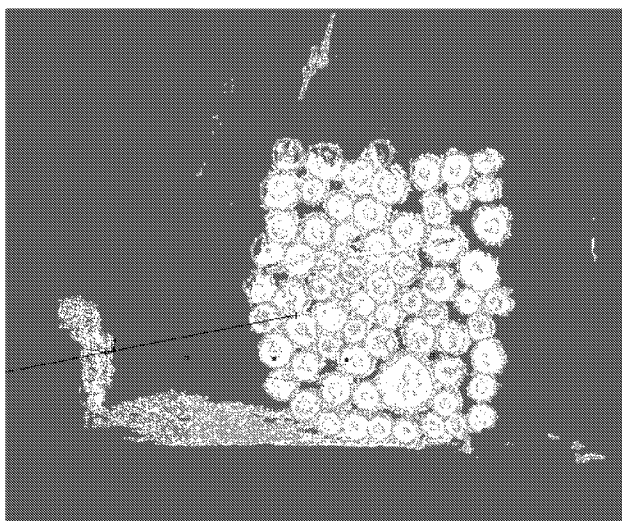


Рис. 6. Исходное облако точек

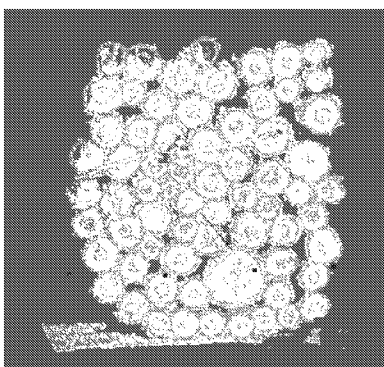


Рис. 7. Облако точек с применённым смещением, поворотом и фильтрацией

## 5. Вывод

При выполнении трёхмерной реконструкции по ряду изображений важным является фильтрация результатов на каждой стадии алгоритма, от этапа поиска соответствующих точек до этапа получения облака точек, и каждый из описанных алгоритмов даёт улучшенный результат реконструкции, при котором на экспериментальных данных удаляется от 80 % фоновых объектов и неправильно реконструированных точек.



## Список литературы

1. **Engin Tola, Vincent Lepetit, Pascal Fua.** A Fast Local Descriptor for Dense Matching // Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference, 23-28 June 2008. Pp 1–8. DOI: 10.1109/CVPR.2008.4587673.
2. **Charles Loop, Zhengyou Zhang.** Computing Rectifying Homographies for Stereo Vision. // Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Vol.1, pages 125–131, June 23–25, 1999. Fort Collins, Colorado, USA.
3. **Christopher M. Bishop.** Pattern Recognition and Machine Learning. Springer, 2006. 738 p.
4. **John Canny,** A Computational Approach to Edge Detection. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. PAMI-8, NO. 6, NOVEMBER 1986. Pp. 679–698.
5. **Martin A. Fischler, Robert C. Bolles.** Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography // Comm. Of the ACM 24: 381–395. DOI:10.1145/358669.358692
6. **Richard Hartley, Andrew Zisserman.** Multiple View Geometry in Computer Vision. Cambridge: University Press, 2003. 655 p.
7. **Richard Szeliski.** Computer Vision: Algorithms and Applications. Springer, 2011. 812 p.

### Summary

**Nikitenkov V. L., Anufriev A. E.** Filtering data obtained by 3D reconstruction from multiple images

In the given article the problem of filtering data obtained by 3D reconstruction from multiple images is considered and methods for solving this problem are represented. For the problem of data clustering from points cloud obtained by 3D reconstruction the noise removal in all steps of reconstruction algorithm is very important.

*Keywords: 3D points filtering, filtering background.*

*СыктГУ, ООО "Си Тех"*

*Поступила 01.11.2014*