

Усенков Дмитрий Юрьевич

3D-СКАНЕР, РАБОТАЮЩИЙ... НА МОЛОКЕ

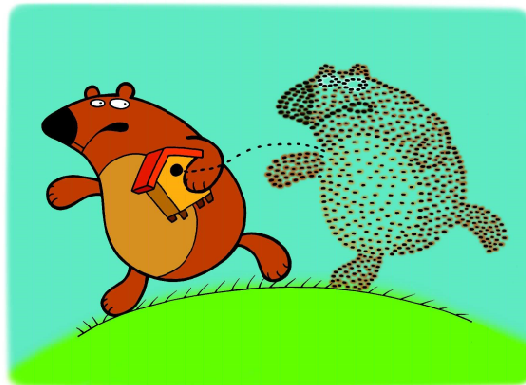
3D-сканер (устройство, позволяющее оцифровать прежде всего форму некоторого материального объекта, и иногда – раскраску его поверхности, для последующего использования полученной 3D-модели объекта при конструировании или для получения его копии при помощи 3D-печати) многие считают сложным и чрезвычайно дорогостоящим техническим устройством, которое невозможно изготовить в домашних условиях. На самом же деле сделать самому 3D-сканер вполне реально. Можно сделать 3D-сканер из приставки Kinect для Xbox 360 или из обычной лазерной указки, используемой совместно со специальной программой DAVID. Наконец, можно выполнить 3D-сканирование вообще без 3D-сканера – при помощи обычного фотоаппарата и специальных программ для распознавания формы объекта по его фотографиям, сделанным с разных ракурсов. Однако еще одна идея, предложенная пользователем *yenfre* на портале <http://www.instructables.com>, заслуживает особого внимания благодаря ее исключительной оригинальности. И действительно: кто бы мог подумать, что для 3D-сканирования будет достаточно пары пластиковых контейнеров (в качестве ёмкостей), смартфона (в качестве фотоаппарата) и... любой белой непрозрачной жидкости, например, обычного молока?!

Чтобы разобраться, как этот набор, казалось бы, разнородных предметов превратить в 3D-сканер, вспомним, как вообще обычно выполняется 3D-сканирование.

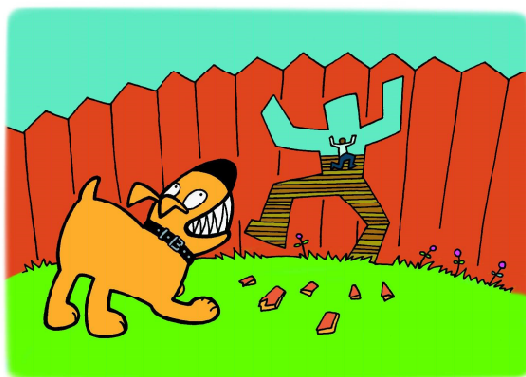
По крайней мере, половина существующих моделей 3D-сканеров использует «контурный» принцип:

- исходный объект фотографируется с разных ракурсов (обычно на контрастном белом фоне или на специальном фоне с разметкой в виде черных кружочков, требуемой компьютерным программам для распознавания расположения сфотографированного объекта в пространстве), после чего из полученных фотоизображений выделяются внешние контуры – абрис объекта;

- на объект проецируется лазерный луч, которому при помощи специальной насадки придана форма тонкой световой полоски, и программа при помощи видеокамеры считывает изменения (искажения) этой световой полоски на поверхности объекта, извлекая из этой информации сведения о его контурах, и т. д.



...строит «облако точек»...



...как получить набор контуров исходного объекта...

Далее же обрабатывающая программа по набору полученных контуров, соответствующим образом ориентированных в пространстве, строит «облако точек» – набор опорных точек, для которых вычислены пространственные координаты и которые соответствуют характерным элементам формы объекта (например, расположены на его выступах, перегибах контуров и пр.). Затем полученные опорные точки соединяются отрезками для построения сетки, определяющей форму поверхности (она может оставаться многогранной или может быть сглажена путем замены прямолинейных отрезков кривыми линиями – сплайнами). И, на-

конец, при необходимости полученная пространственная векторная фигура – собственно 3D-модель объекта – может быть «обтянута» растровой картинкой (скином, или «шкуркой»), полученной из тех же фотографий объекта и дающей информацию о его внешней раскраске.

Таким образом, от нас требуется решить задачу – как получить набор контуров исходного объекта, – а всё остальное сделают уже существующие программы распознавания и формирования 3D-модели. И вот как предлагает получать такой набор контуров в виде «последних срезов» автор описываемой идеи – *yenfre*.

Собственно, суть идеи понятна уже из первой же фотографии «сканирующей установки», помещенной в начале публикации на <http://www.instructables.com> (рис. 1).

Два пластиковых контейнера ставятся один на другой. В дне верхнего контейнера и, соответственно, в крышке нижнего сделаны маленькие отверстия, позволяющие жидкости (автор разработки использовал молоко) медленно перетекать из верхней ёмкости в нижнюю. Можно, впрочем, и изменить конструкцию, предусмотрев для стока жидкости тонкую трубочку – главное, чтобы жидкость могла вытечь из верхнего



Рис. 1

контейнера целиком, обнажив его дно, а процесс вытекания имел малую скорость.

На крышке верхнего контейнера располагается смартфон (или фотоаппарат); всё, что при этом нужно сделать, – это обеспечить ровное, достаточно яркое освещение с отсутствием бликов.

На дно верхнего контейнера кладется сканируемый объект и заливается молоком, чтобы оно покрыло его полностью. «Включается» сток молока в нижний контейнер, и... по мере вытекания молока его белая поверхность будет очерчивать перед фотокамерой те самые нужные нам контуры объекта в виде его «срезов» на разной высоте от дна контейнера. А когда молоко вытечет вниз целиком, 3D-сканирование можно считать оконченным и приступить к обработке полученных фотографий.

На рис. 2 показан еще один вариант предложенной автором конструкции. Смартфон или фотоаппарат может быть любым, главное, чтобы имелась возможность делать серии фотографий через равные промежутки времени («Timelapse-съемка»). Для эксперимента *yenfref* использовал смартфон на базе ОС Android с установленным приложением *Auto Camera* (<https://play.google.com/store/apps/details?id=com.daybrush.autocamera&hl=ru>), позволяющее снимать серии фотографий (требование: Android версии 4.2 или более поздней). В качестве «держателя» для смарт-



На дно верхнего контейнера кладется сканируемый объект и заливается молоком...

фона использовался кусок пластика с вырезанной под объектив фотокамеры дыркой (рис. 3). Для удобства управления 3D-сканером для перетекания молока из отверстия в дне верхнего контейнера в отверстие в крышке нижнего наружу была выведена трубочка, которую можно пережимать зажимом, а для исключения подтеканий потребовался клеевой пистолет, при помощи которого трубочка была приклеена к контейнерам (рис. 4).

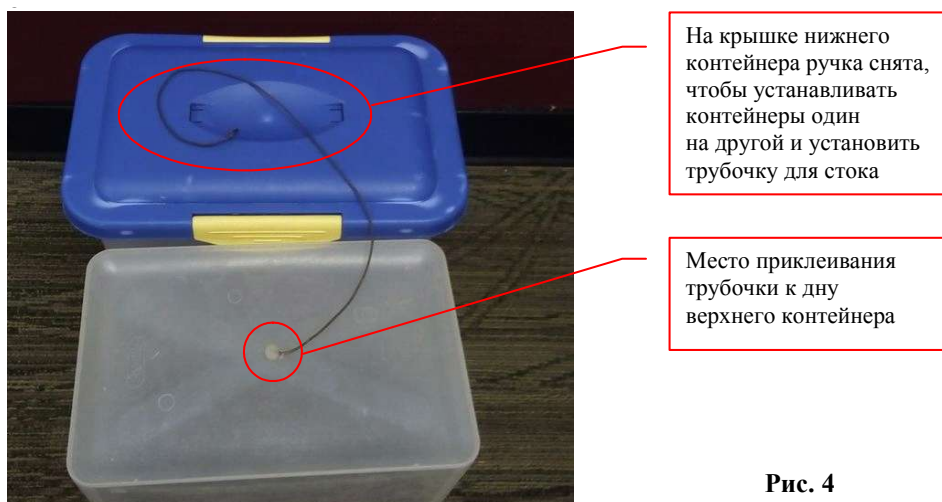
Чтобы затем обрабатывать на компьютере полученные фотографии контуров объекта, автор разработки (*yenfref*) написал программу (скрипт) на языке Python с использованием открытой библиотеки программ компьютерного зрения *OpenCV* (<http://opencv.org>). Данные программы распространяются под лицензией BSD и доступны для свободного использования как в образова-



Рис. 2



Рис. 3



На крышке нижнего контейнера ручка снята, чтобы устанавливать контейнеры один на другой и установить трубочку для стока

Место приклеивания трубочки к дну верхнего контейнера

Рис. 4

тельных, так и в коммерческих целях, предназначены для создания программ на языках C++, C, Python и Java и могут работать в среде Windows, Linux, Mac OS, iOS или Android (для чего на указанном сайте предлагаются для скачивания соответствующие версии библиотек). Соответствующий скрипт на языке Python можно скачать на сайте-репозитории <https://github.com/gcb5083/gotmesh>.

В «комплекте поставки» имеются как сами Python-скрипты, так и набор «молочных» фотографий, полученных с помощью столь необычного 3D-сканера, а также примеры созданных по ним 3D-моделей камешка в различных форматах.

Чтобы запустить распознавание контуров, нужно прежде всего установить на своем компьютере библиотеку *OpenCV*. Затем

потребуется запустить на исполнение Python-скрипт.

Автор разработки предлагает два варианта этого скрипта.

В первой, более старой его версии (файл *point_conversion.py* в папке *HackPSU 2016*), в командной строке для запуска скрипта нужно указать три параметра: имя папки, в которой расположены фотографии «полосных срезов», а также два числовых параметра, которые обозначают пороговые яркости и используются при обработке фотографий для выделения контуров. Возможно, эти значения потребуется подобрать; автор же рекомендует использовать значения 2000 и 2500.

Пример командной строки:

```
$ python point_conversion.py
    '/rock/' '2000' '2500'
```

Вторая, более новая версия скрипта (*pointcloud.py*) находится в основной папке проекта. Для нее достаточно указать в командной строке только один параметр – имя папки с фотографиями. Далее на экране появится диалоговое окно (рис. 5), в котором опять-таки нужно выбрать значения числовых параметров пороговых яркостей (их можно сохранить для последующего использования при помощи соответствующей «ползунковой» кнопки «Save Settings»). Подбор пороговых значений производится путем визуального контроля, при этом нужно стараться добиться четкой прорисовки контуров самого объекта при минимизации помех



...нужно стараться добиться четкой прорисовки контуров самого объекта при минимизации помех...

(таких, например, как блики внизу справа на рис. 5). Обработав первое фото, можно перейти к следующему (кнопка «Start») и проверить выделение контуров на нем, и так – до завершения просмотра всех фотографий.

Другая «ползунковая» кнопка («Start») запускает собственно процесс распознавания.

Пример командной строки:

```
$ python pointcloud.py '/input/'
```

Программный код скрипта нетрудно посмотреть, поскольку файл *.py* имеет обычный текстовый формат. В листинге 1 показано, как выглядит скрипт *point_conversion* (скрипт *pointcloud.py* несколько сложнее за счет реализации обслуживания диалогового окна).

Полученное «облако точек» программа сохраняет в формате *.xyz*, то есть, попросту говоря, в виде списка значений трех координат (X, Y и Z) каждой опорной точки, записанных в отдельных строках. Чтобы преобразовать такой формат в другой типовой формат представления 3D-моделей, исполь-

зуемый в большинстве существующих программ конструирования и 3D-печати (например, в формат STL), предлагается использовать популярную программу 3D-визуализации *MeshLab* (<http://meshlab.sourceforge.net>):

1. Меню **File** → **Import Mesh...**, выбрать файл *.xyz*.

2. Убрать нежелательные артефакты: меню **Edit** → **Select Vertexes**, выбрать ненужные опорные точки и удалить их, используя кнопку **Delete the current set of vertices** в панели инструментов.

3. Меню **Filters** → **Normals, Curvatures, Orientation a Compute normals** – подготовка «облака точек» для его преобразования.

4. Меню **Filters** → **Point Set** → **Surface Reconstruction: Poisson for surface reconstruction** – собственно преобразование «облака точек» в поверхность по методу Пуассона (можно поэкспериментировать, выбирая различные параметры этого преобразования).

5. Экспорт результата в *.stl*-файл: меню **File** → **Export Mesh As...**

Для примера на рис. 6 показаны некоторые из набора фотографий, полученных *yenfre* с помощью его «3D-сканера». На рис. 7 показан вид «облака точек», импортированного в программу *MeshLab* и обработанного для удаления нежелательных артефактов (лишних опорных точек). А на рис. 8 можно увидеть полученную по этому «облаку точек» поверхность (для повышения скорости работы включен режим упрощенного отображения геометрии).

Конечно, представленная разработка – это в каком-то смысле курьёз или даже шутка. Далеко не всякий объект можно для сканирования окунуть в молоко, да и молоко нынче – вещь недешевая, чтобы

использовать его в качестве одноразового «расходного материала», хотя автор разработки утверждает, что по его опыту молоко дает наилучшие результаты по сравнению с другими опробованными им жидкостями вроде рас-



Блики

Рис. 5

Листинг 1

```
#!/usr/bin/env python

import cv2
import numpy as np
import sys
import os
import string
import math
import time

path = sys.argv[1]
t1 = float(sys.argv[2])
t2 = float(sys.argv[3])

heightcount = 1
valuecount = 1

pointcount = 0
linecount = 0
layercount = 0

pointmesh = open(path[0:string.rfind(path, '\\')]
                  + 'xyzmesh.xyz', 'w')

for imageslice in os.listdir(path):

    if math.sqrt(heightcount) >= valuecount:

        RGBslice = cv2.imread(path + imageslice)
        grayslice = cv2.cvtColor(RGBslice, cv2.COLOR_BGR2GRAY)
        edgeslice = cv2.Canny(grayslice, t1, t2, apertureSize=5)

        baseslice = RGBslice.copy()
        baseslice = np.uint8(baseslice/2.)
        baseslice[edgeslice != 0] = (0, 255, 0)

        for line in edgeslice:
            for point in line:

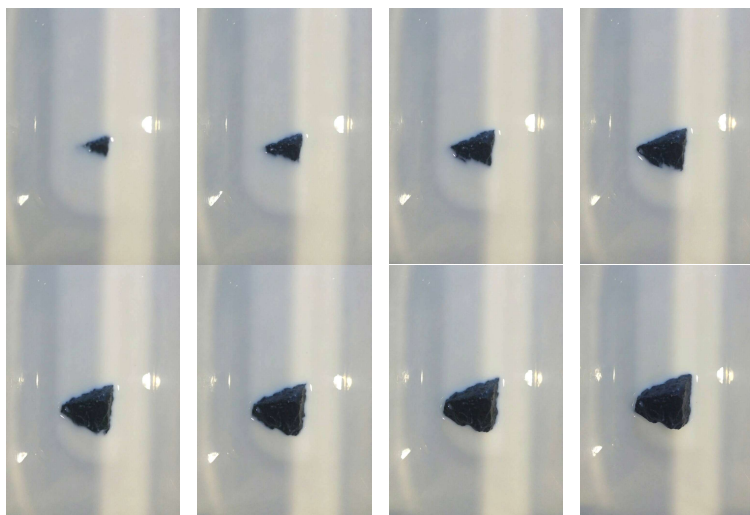
                if point > 0:
                    pointmesh.write('%s\t%s\t%s\n'
                                     % (pointcount, linecount, layercount))
                    pointcount += 1

                pointcount = 0
                linecount += 1

            linecount = 0
            layercount += 1

            valuecount += 1
            heightcount += 1

pointmesh.close()
```



твора пищевых красителей или красок. Но она вместе с тем позволяет получить наглядное представление о самом процессе 3D-сканирования и, например, может с успехом быть использована для демонстрации этого процесса школьникам.

Рис. 6

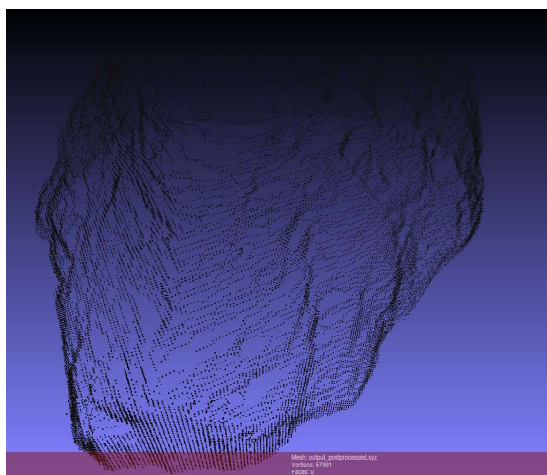


Рис. 7

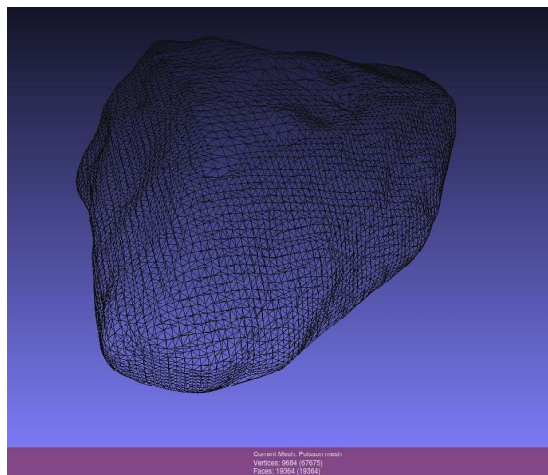


Рис. 8

Источник:

<http://www.instructables.com/id/GotMesh-the-Most-Cheap-and-Simplistic-3D-Scanner>

*Усенков Дмитрий Юрьевич,
Московский государственный
институт индустрии туризма
имени Ю.А. Сенкевича, г. Москва.*

