



KAUNAS UNIVERSITY OF TECHNOLOGY
FACULTY OF MECHANICAL ENGINEERING AND DESIGN
DEPARTMENT OF PRODUCTION ENGINEERING

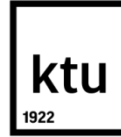
Eglė Dubauskė

**DEVELOPMENT OF 3D SCANNER WITH
LOW HARDWARE REQUIREMENTS AND EASY
USABILITY**

Final project for Master degree

Academic supervisor:
Assoc. Prof. M. Rimašauskas

Kaunas, 2015



KAUNAS UNIVERSITY OF TECHNOLOGY
FACULTY OF MECHANICAL ENGINEERING AND DESIGN
DEPARTMENT OF PRODUCTION ENGINEERING

(Faculty)

Eglė Dubauskė

(Student's name, surname)

621H77003 Industrial Engineering and Management

(Title and code of study programme)

Development of 3D scanner with low hardware requirements and easy usability

DECLARATION OF ACADEMIC HONESTY

29 May 2015
Kaunas

I confirm that a final project by me, **Eglė Dubauskė**, on the subject “Development of 3D scanner with low hardware requirements and easy usability” is written completely by myself; all provided data and research results are correct and obtained honestly. None of the parts of this thesis have been plagiarized from any printed or Internet sources, all direct and indirect quotations from other resources are indicated in literature references. No monetary amounts not provided for by law have been paid to anyone for this thesis.

I understand that in case of a resurfaced fact of dishonesty penalties will be applied to me according to the procedure effective at Kaunas University of Technology.

(name and surname filled in by hand)

(signature)

KAUNAS UNIVERSITY OF TECHNOLOGY
FACULTY OF MECHANICAL ENGINEERING AND DESIGN
DEPARTMENT OF PRODUCTION ENGINEERING

**DEVELOPMENT OF 3D SCANNER WITH LOW
HARDWARE REQUIREMENTS AND EASY USABILITY**

Final project for Master degree

Prepared by Stud. E.Dubauskė, MEM-3/9

(Signature)

(Date)

Supervisor Assoc. Prof. M. Rimašauskas

(Signature)

(Date)

Reviewer Assoc. Prof. S. Kilikevičius

(Signature)

(Date)

KAUNAS, 2015

Dubauskė, E. Nesudėtingos konstrukcijos ir lengvai naudojamo 3D skaitytuvo sukūrimas. Magistrantūros studijų baigiamojo darbo vadovas Prof. Dr. Marius Rimašauskas; Kauno Technologijos Universitetas, mechanikos inžinerijos ir dizaino fakultetas, gamybos inžinerijos katedra.

Kaunas, 2015. 43 p.

SANTRAUKA

Įvadiniamame magistro darbo skyriuje pateiktas temos aktualumas, darbo tikslas (lengvai naudojamo 3D skaitytuvo sukūrimas ir testavimas), uždaviniai, produkto naujumas, naudoti tyrimo metodai bei priemonės.

Pirmajame skyriuje išnagrinėta 3D spausdintuvų ir 3D skaitytuvų rinka, taip pat buvo apžvelgta duomenų rinkimas, skaitmeninis modeliavimas bei apgražos inžinerija. Atlikta glaudi trianguliacijos metodo lazeriniu nuskaitymu apžvalga.

Antrajame skyriuje pateikti: skaitytuvo aparato sandara ir prietaiso darbo principas: struktūra, prietaiso nustatymai, filmavimas, aukščio kalibravimas ir naudotos programos. Taip pat skyrių apžvelgta foto aparato parametrai, motoro specifikacijos, lazerio techninės dalies apžvalga ir pateiktas išsamus skaitytuvo kodo pristatymas.

Trečiajame skyriuje pateikti skaitymo kokybės tyrimai, rezultatai ir matavimai, taip pat buvo pristatyti siūlomi pakeitimai ir patobulinimai. Be to pateiktas skaitytuvo prototipas.

Išvadose apibendrinami pasiekti rezultatai ir gauta informacija apie 3D skaitytuvą.

Dubauskė, E. Development of 3D scanner with low hardware requirements and easy usability. Master degree final project supervisor Assoc. Prof. Marius Rimašauskas; Kaunas University of Technology, faculty of Mechanical Engineering and Design, Department of Production Engineering.

Kaunas, 2015. 43 p.

SUMMARY

In the introductory chapter of the master work theme actuality is presented, the aim (development of 3D scanner with low hardware requirements and easy usability), objectives, and novelty of a 3D scanner, methods and tools used.

The first chapter the market of 3D printers and 3D scanner has been overviewed and analyzed as well as data collection, digital modeling and a reverse engineering. Performed brief overview of a triangulation method laser scanning.

In the second chapter presented: physical setup and working principle of the device: structure and a working principle: setting the device, taking a video, calibration, programs. Also the overview of a photo camera specifications, motor specifications, overview of a laser mechanics, and a detailed presentation of a scanner code.

In the third chapter test of the scanning quality, results and measurements are provided, also potential modifications and improvements were represented. Moreover the prototype of a scanner are submitted.

The conclusions summarize results of the tests and received information about the 3D scanner.

KAUNO TECHNOLOGIJOS UNIVERSITETAS
MECHANIKOS INŽINERIJOS IR DIZAINO FAKULTETAS

Tvirtinu:

Gamybos inžinerijos
katedros vedėjas

(parašas, data)

(vardas, pavardė)

MAGISTRANTŪROS STUDIJŲ BAIGIAMOJO DARBO UŽDUOTIS
Studijų programa PRAMONĖS INŽINERIJA IR VADYBA

Magistrantūros studijų, kurias baigus įgyjamas magistro kvalifikacinis laipsnis, baigiamasis darbas yra mokslinio tiriamojo ar taikomojo pobūdžio darbas (projektas), kuriam atlikti ir apginti skiriama 30 kreditų. Šiuo darbu studentas turi parodyti, kad yra pagilinęs ir papildęs pagrindinėse studijose įgytas žinias, yra įgijęs pakankamai gebėjimų formuluoti ir spręsti aktualią problemą, turėdamas ribotą ir (arba) prieštaringą informaciją, savarankiškai atlikti mokslinius ar taikomuosius tyrimus ir tinkamai interpretuoti duomenis. Baigiamuoju darbu bei jo gynimu studentas turi parodyti savo kūrybingumą, gebėjimą taikyti fundamentines mokslo žinias, socialinės bei komercinės aplinkos, teisės aktų ir finansinių galimybių išmanymą, informacijos šaltinių paieškos ir kvalifikuotos jų analizės įgūdžius, skaičiuojamųjų metodų ir specializuotos programinės įrangos bei bendrosios paskirties informacinių technologijų naudojimo įgūdžius, taisyklingos kalbos vartosenos įgūdžius, gebėjimą tinkamai formuluoti išvadas.

1. Darbo tema:

Nesudėtingos konstrukcijos ir lengvai naudojamo 3D skaitytuvo sukūrimas.

Developement of 3D scanner with low hardware requirements and easy usability

Patvirtinta 2015 m. gegužės mėn. 11d. dekanų įsakymu Nr. ST17-F-11-2

2. Darbo tikslas Nesudėtingos konstrukcijos ir lengvai naudojamo 3D skaitytuvo sukūrimas ir testavimas

3. Darbo struktūra **Apžvalginė dalis:** 3D spausdintuvų ir 3D skaitytuvų standartų ir testavimo tyrimų apžvalga. **Metodinė dalis:** 3D spausdintuvų ir 3D skaitytuvų kontrolės ir tyrimų tipų apžvalga bei būsimo skaitytuvo prototipo pristatymas. **Tiriamoji dalis:** 3D skaitytuvo techninė analizė ir pagaminimo metodologija bei rezultatų apibendrinimas.

4. Reikalavimai ir sąlygos: lengvai naudojamo 3D skaitytuvo sukūrimas ir testavimas

5. Darbo pateikimo terminas 2015m. gegužės mėn. 29 d.

6. Ši užduotis yra neatskiriama baigiamojo darbo dalis.

Išduota studentei Eglei Dubauskei

Užduotį gavau Eglė Dubauskė

(studento vardas, pavardė)

(parašas, data)

Vadovas Assoc. Prof. M.Rimašauskas

(pareigos, vardas, pavardė)

(parašas, data)

Introduction	9
The main abbreviations used in the project	10
1. OVERVIEW OF 3D PRINTING AND 3D SCANNING	12
1.1. Most common 3D scanning types.....	12
1.2. Analysis of Point Clouds usability.....	16
1.3. Prototype of a prospective scanner	17
2. PHYSICAL SETUP AND WORKING PRINCIPLE OF THE DEVICE	20
2.1. Structure of a 3D scanner.....	20
2.2. Working principle	20
2.3. Photo Camera Specifications	28
2.4. Motor Specifications	28
2.5. Laser Mechanics	29
2.6. Scanner Code	30
3. TEST OF THE SCANNING QUALITY.....	33
3.1. Results.....	34
3.2. Modifications and improvements	36
3.3. Scanner prototype	39
Conclusions	40
References	41

INTRODUCTION

In the latest years, growth in computer technology, personal computers and graphics processors has improved, in consequence of this situation use of 3D models either in science or among the end users has become comprehensive. With the increasing demand for 3D models, high costs of point-based laser scanners and difficulty in processing the data obtained from these devices, has led to the development of low-cost alternative 3D scanning methods.[1]

Counting on this I have chosen for my Research Project - Development of 3D scanner with low hardware requirements and easy usability. I have made a low cost 3D scanner by using:

- Digital camera
- Rotational platform
- Laser line generator
- Code based on a laser triangulation method.

The main aim of this work – to develop a 3D scanner with low hardware requirements and easy usability intended for reverse engineering or replacing missing or older parts.

The objectives will be:

- 1) To analyze the market of 3D scanners
- 2) To design the 3D scanner targeted to reverse engineering
- 3) To present technical analysis of a 3D scanner and development methodology
- 4) To summarize the results
- 5) To familiarize with possible developments and new inventions

THE MAIN ABBREVIATIONS USED IN THE PROJECT

CT - Computerised Tomography
CAD – Computer Aided Design
CAM – Computer Aided Manufacturing
AM – Additive Manufacturing
STL - Standard Tessellation Language
NURBS – Non-uniform rational basis spline
RE – Reverse Engineering
IGES - Initial Graphics Exchange Specification
PDS – Product Definition Statement
FAI- First Article Inspection
FEA - Finite Element Analysis
BIM – Building Information Modeling
OBJ – an open data format that represents the vertices of polygons
DXF – Drawing Interchange File
SLDPRT – a native CAD format for SolidWorks
WRL (VRML) – Virtual Reality Modeling Language
STEP – Standard for the Exchange of Product model data
RP – Rapid Prototyping
SLA – Stereolithography
SLS – Selective Laser Sintering
FDM – Fused Deposition Modeling
UV – UltraViolet
COR – Centre of Rotation
CCD- Charge Coupled Device

LASER TRACKER - sends a laser beam to locate a reflective target held against the object to be measured. The beam reflects back to the tracker and calculates the distance and angle of the targets location. Laser trackers are a great option when you need extreme accuracy over larger measurement ranges.

COLOR MAP - a graphical display for visualizing dimensional differences between the measured shape of an object and its nominal CAD model; deviations are mapped to a color spectrum indicating location and magnitude. A reference key maps the deviations to values.

CMM - Coordinate Measuring Machine, a mechanical device that obtains 3D coordinates by probing, may be either touch probe based or non-contact, portable or stationary, or motorized or manual.

1. OVERVIEW OF 3D PRINTING AND 3D SCANNING

3D printer is used to make real objects from digitals. Opposite to the traditional subtractive manufacturing, 3D printing uses an additive process. It melts some materials such as metal or plastic to make a layer above the previous layer to construct an object gradually. In the recent years, 3D printing has entered the personal market from the industrial domain. The demand for 3D printing for entertainment and other personal use has dramatically increased. And lots of companies are making efforts to reduce the cost of the 3D printer to meet the demand.

Meanwhile, a complementary technology to 3D printing is also gaining more and more attentions. That is 3D scanning. It collects the 3D information of an object like its shape and appearance. In the domain of the entertainment industry, it is used for producing movies and video games, industrial design, orthotics, prosthetics and reverse engineering, prototyping, quality control/inspection and documentation of cultural artifacts [2]. 3D scanning has significant usages both on accelerating the design process of 3D model and on digitalizing realistic objects. This research project is focusing on how to make a low cost and practical 3D scanner. The combination of a 3D printer and a 3D scanner can realize the concept of "3D photocopy" which is to make a replicate of a real object. [3]

1.1. Most common 3D scanning types

Contact scanners analyze the surface of the object under the physical contact. Often they are used in industrial production. The best advantage of such devices are high accuracy. However, the mechanism of action by contact such devices have disadvantages. During the scanning time the object may be deformed or it may be even damaged. The principle of operation is relatively slow, compared with other scanning methods.

The easiest mode to think about Time of Flight laser scanning is to look into it as nothing more difficult as a laser range finder [13]. For this type of scanning there can be used diversified nature of radio waves, both light and ultrasound or X-rays. The most common scanners using laser light. In a Time-of-Flight system a laser stroke is sent out and a fragment of a pulse is echoed from a given exterior and returns to the unit. The distance to the exterior is determined from the time of the flight of the pulse. Since the speed of light c is known, the round-trip time determines the travel distance of the light, which is twice the distance between the scanner and the surface. If t is the round-trip time, then distance is equal to $c \cdot \frac{t}{2}$. [14] Scanner simultaneously measures the distance to a surface point thus the entire surface is consistently reviewed. This type of equipment is suitable to digitize the objects located at ideal distance from the main device. Consequently they are adaptable to scanning large structures

such as buildings, huge statues. Functioning on the long-distance the device loses the accuracy. The corners of the object being scanned are hardly recognizable, because the light dispersal are increased in these areas. Example of Time-of-flight camera, can be seen in Fig.1-1.



Fig. 1-1 Lidar Scanner

The contrary qualities has so-called triangulation laser 3D scanner. It is also classified as a non-contact active scanner. Its operating range is not high, but accuracy can be achieved by tens of micrometers. Triangulation scanners are composed of laser, digital camera and lens system. Depending on the laser beam location where it strikes the surface which is scanned, the reflected laser is fixed at a different locations of a camera field of vision. This technique is called triangulation because the laser dot, the camera and the laser emitter form a triangle. In most cases a laser stripe, instead of a single laser dot, is swept across the object to speed up the acquisition process. The National Research Council of Canada was among the first institutes to develop the triangulation based laser scanning technology in 1978. [5] Getting 3D information from structured light is based on the principle of triangulation, seen in Fig. 3. According to the triangulation method, it becomes possible to calculate the 3D coordinate of point (c), if the length (d) between laser source and camera and the two angles of the triangle is known. Fig.1-2 shows how magnitude of the (γ) angle affects the depth resolution. If the magnitude of (γ) increases, depth resolution increases as well. [6]

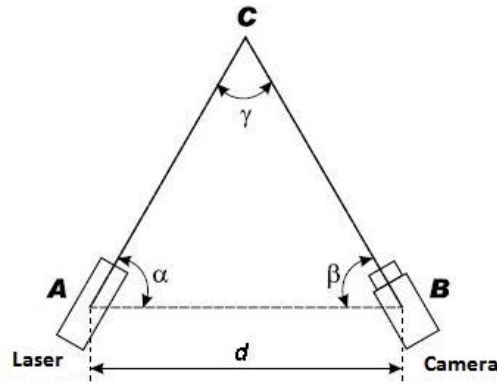


Fig. 1-2 Triangulation

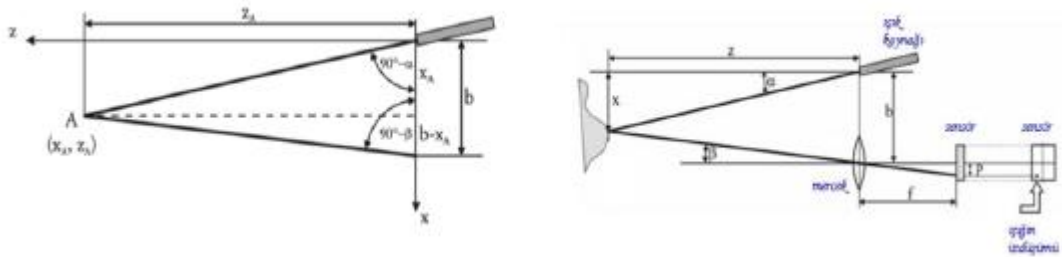


Fig. 1-3 Principle of Triangulation

$$z_A = \frac{b}{\tan(\alpha) + \tan(\beta)} \quad (1.1)$$

$$\tan(\beta) = \frac{p}{f} \quad (1.2)$$

Strengths and weaknesses

Time-of-flight and triangulation range finders each have strengths and weaknesses that make them suitable for different situations. The advantage of time-of-flight range finders is that they are capable of operating over very long distances, on the order of kilometers. These scanners are thus suitable for scanning large structures like buildings or geographic features. The disadvantage of time-of-flight range finders is their accuracy. Due to the high speed of light, timing the round-trip time is difficult and the accuracy of the distance measurement is relatively low, on the order of millimeters. Triangulation range finders are exactly the opposite. They have a limited range of some meters, but their accuracy is relatively high. The accuracy of triangulation range finders is on the order of tens of micrometers.

Time-of-flight scanners accuracy can be lost when the laser hits the edge of an object because the information that is sent back to the scanner is from two different locations for one laser pulse. The coordinate relative to the scanners position for a point that has hit the edge of an object will be calculated based on an average and therefore will put the point in the wrong

place. When using a high resolution scan on an object the chances of the beam hitting an edge are increased and the resulting data will show noise just behind the edges of the object. Scanners with a smaller beam width will help to solve this problem but will be limited by range as the beam width will increase over distance. Software can also help by determining that the first object to be hit by the laser beam should cancel out the second.

At a rate of 10,000 sample points per second, low resolution scans can take less than a second, but high resolution scans, requiring millions of samples, can take minutes for some time-of-flight scanners. The problem this creates is distortion from motion. Since each point is sampled at a different time, any motion in the subject or the scanner will distort the collected data. Thus, it is usually necessary to mount both the subject and the scanner on stable platforms and minimize vibration. Using these scanners to scan objects in motion is very difficult. Recently, there has been research on compensating for distortion from small amounts of vibration [7] and distortions due to motion and/or rotation [8].

When scanning in one position for any length of time slight movement can occur in the scanner position due to changes in temperature. If the scanner is set on a tripod and there is strong sunlight on one side of the scanner then that side of the tripod will expand and slowly distort the scan data from one side to another. Some laser scanners have level compensators built into them to counteract any movement of the scanner during the scan process.

Hand-held laser scanners create a 3D image through the triangulation mechanism described above: a laser dot or line is projected onto an object from a hand-held device and a sensor (typically a charge-coupled device or position sensitive device) measures the distance to the surface. Data is collected in relation to an internal coordinate system and therefore to collect data where the scanner is in motion the position of the scanner must be determined. The position can be determined by the scanner using reference features on the surface being scanned (typically adhesive reflective tabs, but natural features have been also used in research work [9][10]) or by using an external tracking method. External tracking often takes the form of a laser tracker (to provide the sensor position) with integrated camera (to determine the orientation of the scanner) or a photogrammetric solution using 3 or more cameras providing the complete Six degrees of freedom of the scanner. Both techniques tend to use infrared Light-emitting diodes attached to the scanner which are seen by the camera(s) through filters providing resilience to ambient lighting.

Data is collected by a computer and recorded as data points within 3Dspace, with processing this can be converted into a triangulated mesh and then a Computer-aided design model, often as non-uniform rational B-spline surfaces. Hand-held laser scanners can combine

this data with passive, visible-light sensors—which capture surface textures and colors—to build (or "reverse engineer") a full 3D model.

1.2. Analysis of Point Clouds usability

After scanning a real world object with three-dimensional scanner to utilize achieved point clouds are difficult and complicated. To carry out deformities, or any calculations (volume, height) with three-dimensional models is necessary to have it described in geometric objects (straight lines, curves, splines, triangles, squares, etc.).

1.2.1. Usability of a NURBS splines

Very often the surface of the model are described by splines . Due to its flexibility and the ability to form any kind of regular or irregular shape NURBS splines became very popular [15,16]. NURBS (Non-Uniform Rational B-Spline) - Non-unified rational B-spline basis Bezier curve. B-Spline is composed of several Bezier segments with a certain degree of continuity, localizing the impact of a control points for curve and adding weights. This type of curves are used for a convenience to calculate by the computer, stability, in respect of floating comma errors, low memory costs required for the storage of a construction and the possibility to obtain any form.

As NURBS splines practically describes any surface very well, it has become necessary to adjust this interpolation method not only for the development of new models, but also for a reverse engineering [17], where the three-dimensional models are obtained from the point cloud. There are methods that allows to reconstruct the object surface from point cloud by RE method and immediately get a CAD model. This method relies on a calculation of a surface curves, its boundaries establishment and NURBS surface applications. In order to create such suitable surface one should be aware of each surface point in space. The parameters of a points position are calculated using parameterization methods. At the beginning the base surface is obtained, which is adjusted in the following iterations. Depending on the complexity of reconstructed model may be necessary to combine several surfaces with different parameters with connection line. Therefore are possible to occurrence of distortions of restored surface.

1.2.2. Analysis of a Triangulation Methods

It is far better known and more often used the methodology to reconstruct surfaces from point sets are called triangulation [26]. Triangulation is a plane or surface polygon decomposition into a set of triangles. Typically, there are limitations, such that each side of the triangle shares two adjacent triangles (see Fig.1-4). It has been proved that each surface has

triangulation, but to create it may be required an infinite number of triangles [27]. Surface triangulation with a finite number of triangles is called a compact.

Point cloud triangulation problem could be solved in a very similar. Initially is calculated convex polyhedron, surrounding all point cloud points. Then its walls are divided into triangles by calculated parameters (eg. Surface curvature) to the required level, until is reached the surface accuracy, equivalent to a real-world object.

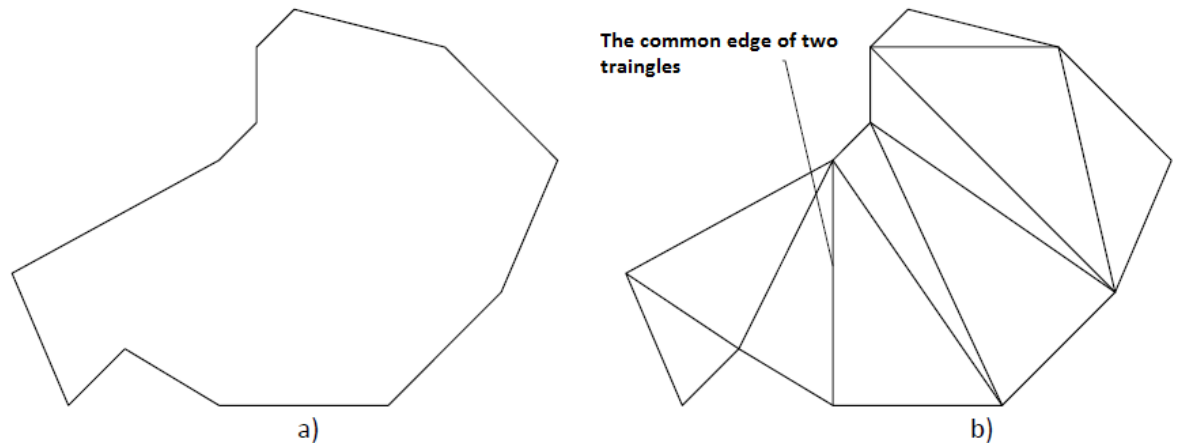


Fig. 1-4 Polyhedron triangulation

a) The initial polyhedron

b) A polyhedron subdivided into triangles

1.3. Prototype of a prospective scanner

DAVID Laser scanner is a software package for low-cost 3D laser scanning. It allows scanning and digitizing of three-dimensional objects using a camera (e.g. a web cam), a hand-held line laser (i.e. one that projects a line, not just a point), and two plain boards in the background. Its most distinctive feature is that the laser line can simply be swept over the object by hand (like a virtual brush) until the results are satisfactory. At the same time, DAVID Laser scanner generates 3D data in real time and shows them on the computer screen.

The resulting 3D mesh can be exported into well-known file formats and can thus be imported and processed in most 3D applications. The software is also able to grab the texture and "stitch together" scans made from different viewing directions.

Scanning process

An initial calibration is made to determine the lens parameters and location of the camera relative to the background boards, which are two vertical planes positioned at 90 degrees to one another behind the object to be scanned. When scanning, the camera must be able to see part of the laser line on each board. This enables the software to reconstruct the plane of the projected laser light. Once it has determined the two-dimensional plane that the

line laser is projecting, it is able to analyze the image of the laser line falling on the scanned object and resolve it into points in space

History

Development of DAVID Laser scanner started in September 2006 by German computer scientists Dr. Simon Winkelbach and Sven Molkenstruck, research associates of the Institute for Robotics and Process Control of the TU Braunschweig. The concept has been published as a research paper and has received the Best Paper Award at the German Association for Pattern Recognition (DAGM) on September 14, 2006, in Berlin.

DAVID Laser scanner was originally available for free but has increased in price since late 2008. The companion software, 3D Shapefusion, was originally sold for €25. DAVID Laser scanner now includes both products in the €329 price tag (as of March 2012) [4].

Laser Calibration

The basic principle of triangulation based systems discussed in the previous section, is that determination of the beam space, which is formed by camera, laser source and the object, as mathematically and establishment of the relationship between image and object coordinates. Thus, new 3D point coordinates can be calculated with the help of mathematical expressions. A background Fig.1-6 which has control points on it is employed at DAVID laser scanning system to establish the relationship between image and object coordinates. The laser ray, expanded to a plane by a cylindrical lens, has to intersect two things at the same time: the (unknown) surface, and the a priori known reference geometry (usually the background). The visible intersection with the background is used to calibrate the laser, i.e. to calculate the exact 3D pose of the laser plane Fig.1-5. Detailed information can be found at [11]. On the other hand, the camera should be calibrated. In this software, the calibration of the camera is done with the in [12].

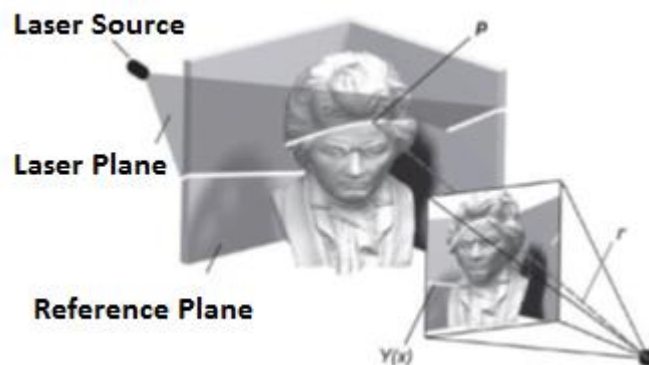


Fig. 1-5 Laser triangulation

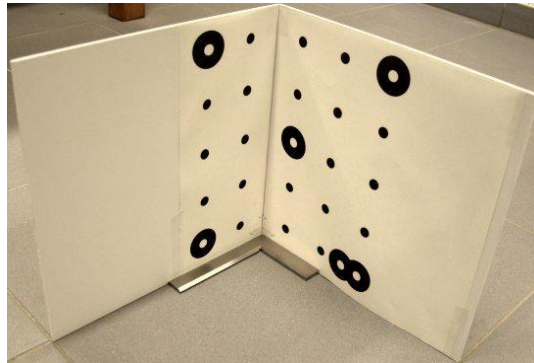


Fig. 1-6 Background for calibration

System Design and Object Scanning

In the process of scanning, the user generally sweeps the laser plane over the object multiple times manually. However, since it is difficult to move the laser properly, some data occlusions can occur at the final data. For this reason a control tool developed for this process. The control mechanism consists of a stepper motor, electronic equipment to control the motor and transmission system to reduce the rotation speed, can be seen in Fig.1-7.



Fig. 1-7 David scanner system

2. PHYSICAL SETUP AND WORKING PRINCIPLE OF THE DEVICE

2.1. Structure of a 3D scanner

To meet a requirements and task of a project I have chosen to reuse the old substances for my scanner. The only one that was bought – linear laser for €10, ordered online.

Composition:

- motor from an old microwave
- plywood base
- 4 corky pads
- old white CD
- linear laser module
- photo camera
- some models
- working Meshlab[28]
- VRmesh programme[29]
- and Some help

2.2. Working principle

Step 1: Setting the device

I have got the 4 corky pads attached to the plywood base by 4 bolts. Then the hole for the rotating motor coupling has been cut and the motor has been connected to the lower side of the base by bolts. An old white CD was used as table and it has been glued to the motor coupling. At the opposite site of the board linear laser module has been attached together with the external power source which consists of 3AA batteries of 1.5V and has been taken from old toy. Finally the photo camera has been set in the place where it makes an angle of 40° between it shooting range pointed to the turn table and a laser line[5] , can be seen in the Fig. 2-1.

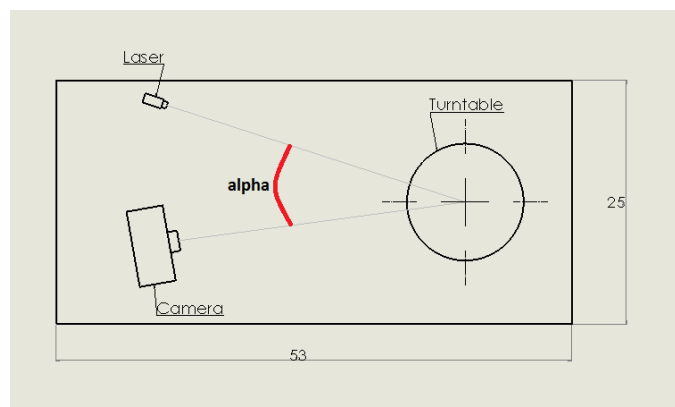


Fig. 2-1 Top view of scanner

Step 2: Figure capturing

Laser line module is on, also the turntable is connected to the power. To take a pictures I have used my photo cameras video function. As the turntable doesn't have a function to stop at our required time I have made a video of the figure turning few rounds. The video has been made in the dark that only laser line would be visible on the figure. After the video has been taken, it has to be exported to the computer, then converted to the .mpeg file.

Once the image was made with the camera, the 3D object must be converted into 2D model on the image plane. This image will be converted into a virtual 3D image by the build up program.

During this alteration a function, ϕ , was determined for this transformation. See Fig.2-2. This function, ϕ , consists of structural and auxiliary parameters similar to various impacting parameters during the image taking such as camera focal length, distances and some other camera properties.

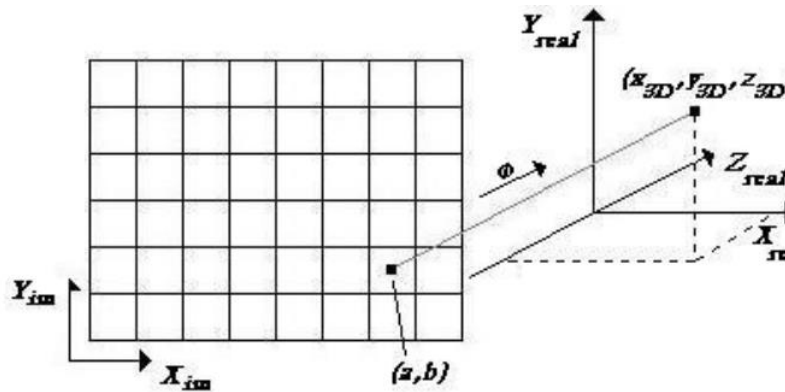


Fig. 2-2 A mapping image from coordinates to virtual world

In mathematical terms, function ϕ outlines the image coordinates to virtual coordinates as seen in (Fig. 12) and shown by the mathematical expression:

$$\phi = \mathbb{R}^2 \rightarrow \mathbb{R}^3 \quad (2.1)$$

It was calculated by using a well-known Least Square Fitting technique (LSF) [25]. The origin of the coordinates (0,0,0) was considered to be the center point of a circular platform surface.

ϕ was considered as a 3 x 3 transformation matrix.

$$\phi = \begin{pmatrix} \phi_{11} & \phi_{12} & \phi_{13} \\ \phi_{21} & \phi_{22} & \phi_{23} \\ \phi_{31} & \phi_{32} & \phi_{33} \end{pmatrix} \quad (2.2)$$

where ϕ_{ij} values are the coefficients calculated by LSF and determine the mapping parameters.

A point on 2D image was defined by

$$p_{im} = \begin{pmatrix} a \\ b \\ 1 \end{pmatrix} \quad (2.3)$$

and a point in 3D space was determined by

$$p_{3D} = \begin{pmatrix} x_{3D} \\ y_{3D} \\ z_{3D} \end{pmatrix} \quad (2.4)$$

ϕ was multiplied by p_{im} (to develop p_{3D})* for every obtained point on the 2D image plane. This was shown by:

$$\phi p_{im} = p_{3D} \quad (2.5)$$

During the scanning, the laser is projected along the yz plane with $x=0$. This was depicted in the function ϕ by the first row filled with zeros as shown here.

$$\phi = \begin{pmatrix} 0 & 0 & 0 \\ \phi_{21} & \phi_{22} & \phi_{23} \\ \phi_{31} & \phi_{32} & \phi_{33} \end{pmatrix} \quad (2.6)$$

Therefore p_{3D} which contained the 3D coordinates of the factual object for an individual scan line was calculated as:

$$\phi = \begin{pmatrix} \phi_{21}a+ & \phi_{22}b & \phi_{23} \\ \phi_{31}a+ & \phi_{32}b & \phi_{33} \end{pmatrix} \quad (2.7)$$

Step3: Calibration

ϕ_{ij} coefficient values were calculated by height calibration (ϕ_{2j}) along the y axis and depth calibration (ϕ_{3j}) along the z axis of the scanner system.

The depth calibration is in such way; The laser line was accident on a flat surface parallel to xy plane in front of the camera. This flat surface was placed along the z axis at 5 different locations and the image of incident laser line on each surface was recorded by the camera as shown in Fig.2-3. These laser line images were identified as calibration images and later processed by using LSF technique [2.12] to give the ϕ_{3j} and z value of p_{3D} .

The height calibration was similar to depth calibration with the exception that the flat surfaces were taken in y direction parallel to xz plane.

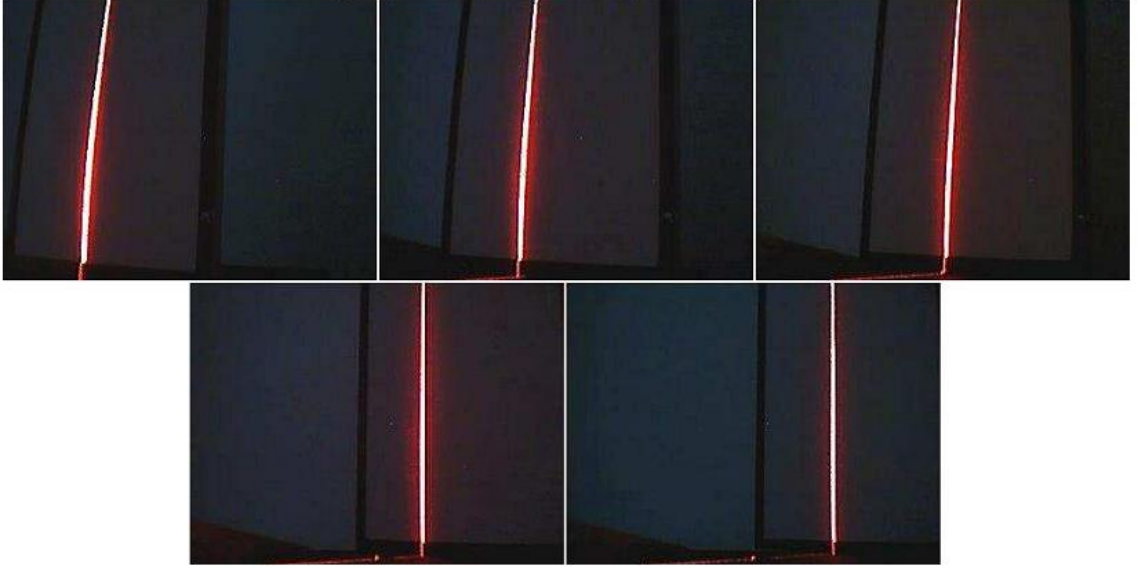


Fig. 2-3 5 images of laser of depth calibration

There were m calibration images each of which corresponded to a depth of z_k .

z_k stand for the distance of the calibration surface from the world coordinate origin and k was the index number of the calibration image. Single calibration image contains n_k bright pixels and every bright pixel has image coordinates of (a_{lk}, b_{lk}) where $l=1$ to n_k .

The sum of squares of the difference between the calculated depth $(\phi_{31}a_{lk} + \phi_{32}b_{lk} + \phi_{33} - z_k)$ and the actual depth z_k can be written as

$$Sum = \sum_{k=1}^m \sum_{l=1}^{n_k} (\phi_{31}a_{lk} + \phi_{32}b_{lk} + \phi_{33} - z_k)^2 \quad (2.8)$$

From LSF manner [25], the sum will be minimal when its partial derivatives are zero with respect to ϕ_{3j} . This was shown as:

$$\begin{aligned} \frac{\partial Sum}{\partial \phi_{31}} &= 2 \sum_{k=1}^m \sum_{l=1}^{n_k} a_{lk} (\phi_{31}a_{lk} + \phi_{32}b_{lk} + \phi_{33} - z_k) = 0 \\ \frac{\partial Sum}{\partial \phi_{32}} &= 2 \sum_{k=1}^m \sum_{l=1}^{n_k} b_{lk} (\phi_{31}a_{lk} + \phi_{32}b_{lk} + \phi_{33} - z_k) = 0 \\ \frac{\partial Sum}{\partial \phi_{33}} &= 2 \sum_{k=1}^m \sum_{l=1}^{n_k} (\phi_{31}a_{lk} + \phi_{32}b_{lk} + \phi_{33} - z_k) = 0 \end{aligned} \quad (2.9)$$

The equations can be rearranged as shown below:

$$\begin{aligned} \phi_{31} \sum_k \sum_l a_{lk}^2 + \phi_{32} \sum_k \sum_l a_{lk}b_{lk} + \phi_{33} \sum_k \sum_l a_{lk} &= \sum_k \sum_l a_{lk}z_k \\ \phi_{31} \sum_k \sum_l a_{lk}b_{lk} + \phi_{32} \sum_k \sum_l b_{lk}^2 + \phi_{33} \sum_k \sum_l b_{lk} &= \sum_k \sum_l b_{lk}z_k \\ \phi_{31} \sum_k \sum_l a_{lk} + \phi_{32} \sum_k \sum_l b_{lk} + \phi_{33} \sum_k n_k &= \sum_k n_k z_k \end{aligned} \quad (2.10)$$

They could be written in a matrix form as:

$$M \begin{pmatrix} \Phi_{31} \\ \Phi_{32} \\ \Phi_{33} \end{pmatrix} = Z \quad (2.11)$$

where

$$M = \begin{pmatrix} \sum_k \sum_l a_{lk}^2 & \sum_k \sum_l a_{lk} b_{lk} & \sum_k \sum_l a_{lk} \\ \sum_k \sum_l a_{lk} b_{lk} & \sum_k \sum_l b_{lk}^2 & \sum_k \sum_l b_{lk} \\ \sum_k \sum_l a_{lk} & \sum_k \sum_l b_{lk} & \sum_k n_k \end{pmatrix} \quad (2.12)$$

And

$$Z = \begin{pmatrix} \sum_k \sum_l a_{lk} z_k \\ \sum_k \sum_l b_{lk} z_k \\ \sum_k n_k z_k \end{pmatrix} \quad (2.13)$$

Since, Φ_{31} , Φ_{32} , and Φ_{33} values are required to calculate the object surface depth, the above matrix equation can be reformed as:

$$\begin{pmatrix} \Phi_{31} \\ \Phi_{32} \\ \Phi_{33} \end{pmatrix} = M^{-1} Z \quad (2.14)$$

The inverse matrix M^{-1} and Z matrix can be calculated by the methods of calculus [25] and finally Φ_{31} , Φ_{32} , Φ_{33} values can be derived as:

$$\begin{aligned} \Phi_{31} = & \left(\left(\sum_k n_k \sum_k \sum_l b_{lk}^2 - \left(\sum_k \sum_l b_{lk} \right)^2 \right) \sum_k \sum_l a_{lk} z_k \right. \\ & + \left(\sum_k \sum_l a_{lk} \sum_k \sum_l b_{lk} \right. \\ & \left. - \sum_k n_k \sum_l a_{lk} b_{lk} \right) \sum_k \sum_l b_{lk} z_k \\ & \left. + \left(\sum_k \sum_l b_{lk} \sum_k \sum_l a_{lk} b_{lk} - \sum_k \sum_l a_{lk} \sum_k \sum_l b_{lk}^2 \right) \sum_k n_k z_k \right) / |M| \end{aligned}$$

$$\begin{aligned}
\phi_{32} = & \left(\left(\sum_k \sum_l a_{lk} \sum_k \sum_l b_{lk} - \sum_k n_k \sum_l a_{lk} b_{lk} \right) \sum_k \sum_l a_{lk} z_{lk} \right. \\
& + \left(\sum_k n_k \sum_k \sum_l a_{lk}^2 - \left(\sum_k \sum_l a_{lk} \right)^2 \right) \sum_k \sum_l b_{lk} z_k \\
& \left. + \left(\sum_k \sum_l a_{lk} \sum_k \sum_l a_{lk} b_{lk} - \sum_k \sum_l b_{lk} \sum_k \sum_l a_{lk}^2 \right) \sum_k n_k z_k \right) / |M| \\
\phi_{33} = & \left(\left(\sum_k \sum_l b_{lk} \sum_k \sum_l a_{lk} b_{lk} - \sum_k \sum_l a_{lk} \sum_k \sum_l b_{lk}^2 \right) \sum_k \sum_l a_{lk} z_{lk} \right. \\
& + \left(\sum_k \sum_l a_{lk} \sum_k \sum_l a_{lk} b_{lk} \right. \\
& \left. - \sum_k \sum_l b_{lk} \sum_k \sum_l a_{lk}^2 \right) \sum_k \sum_l b_{lk} z_k \\
& \left. + \left(\sum_k \sum_l a_{lk}^2 \sum_k \sum_l b_{lk}^2 - \left(\sum_k \sum_l a_{lk} b_{lk} \right)^2 \right) \sum_k n_k z_k \right) / |M|
\end{aligned} \tag{2.15}$$

Where $|M|$ stands for the determinant of M .

A quantitative view can be derived from the following example with 5 calibration images ($m=5$) for depth measurements. A set of five bright pixels from each calibration image were considered and presented as:

- 1st image= {(4,3),(5,4),(5,20), (6,58), (7,70)}
- 2nd image= {(12,4), (12,5), (13,7), (13,18), (15,30)}
- 3rd image= {(33,5), (34,13), (36,75), (37,88), (37,90)}
- 4th image= {(42,95), (45,82), (47,18), (48,8), (49,6)}
- 5th image= {(60,70), (61,54), (61,66), (66,28), (68,8)}

The depth distances, (z_k), between the 5 calibration surfaces and the world coordinate origin are determined as -20cm, -10cm, 0cm, 10cm and 20cm respectively.

Using these models 2D calibration image points and their respective depth information, the elements of matrices M and Z in equations (2.12) and (2.13) were calculated. The results were modified in ϕ_{31} , ϕ_{32} , ϕ_{33} coefficients were again calculated similarly for the height calibration images.

Step4: Programs

The converted video.mpeg was used to make a burst of photos using a video capture utility VirtualDub [30]. Later on all the photos (about 350 of them) has been processed by Java program created by my colleague. Finally the program has computed the photos and created the required file test.asc, which has been uploaded to the Meshlab. Using Render functions => Bounding box => Show Vertex Dots I have got view visible in the Fig.2-4.

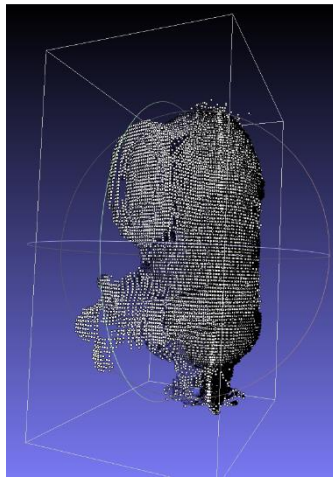


Fig. 2-4 Primary point cloud in Meshlab

When the point cloud has been made into 3D view, file test.ply has been processed in VRMesh to make it look more similar to the initial figure, see in Fig.2-5.

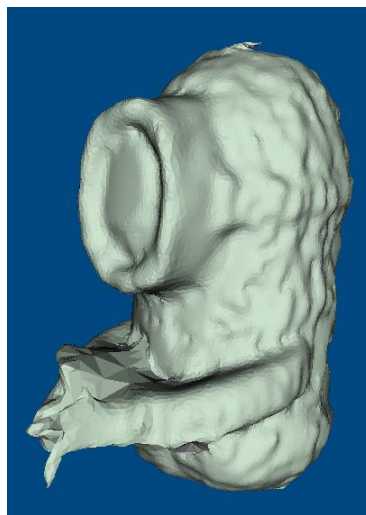


Fig. 2-5 Final 3D Model

As seen in Fig.2-6, all of the important parts were seated on a plywood base. For this certain variant, a wooden board was used to diminish intrusion. The video camera is set to the right side of the wood board and the linear laser module has been fixed to the reversed side of the board. All electronic parts were arranged beneath or behind the base, leaving the factual scanning area empty.



Fig. 2-6 Real set of the scanner

In Fig.2-7, the housing for the motor can be seen. The motor is connected to the lower side of the base by bolts. Above the motor (before it has been mounted) has been cut the hole for the rotational table.



Fig. 2-7 The housing of the motor

2.3. Photo Camera Specifications

Olympus SP-620 UZ

- 16MP - 1/2.3" CCD Sensor
- ISO 100 - 3200
- 25–525 mm F3.1 - F5.8 Zoom Lens
- 1280 x 720 video resolution

2.4. Motor Specifications

Key specifications of a motor “Permanent Magnet Synchronous Motor for Air Conditioner/Microwave Oven” [31]

- Rated power: 3 to 4W
- Rated voltage: 220V
- Starting voltage: $\leq 176V$
- Withstand voltage: 1500V/minute
- Rated frequency: 50 or 60Hz
- No-load current: $\leq 20mA$
- Draw in torque: $\geq 5kgf.cm$
- Insulation resistance: $\geq 100M\Omega$
- Rotation speed: 5 or 6rpm
- Change direction: CW/CCW
- With permanent magnet

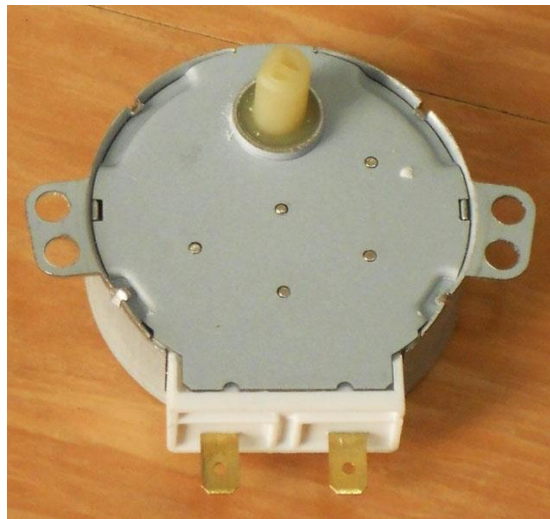


Fig. 2-8 Motor

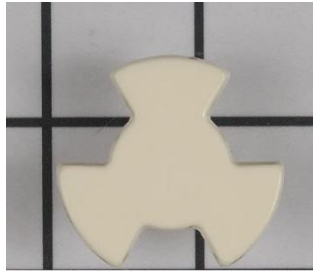


Fig. 2-9 Motor Coupling

Goes together with motor coupling on which I had attached the CD which serves for the table.

2.5. Laser Mechanics

This type of laser is easily built since all that is required is a suitable power supply. A simple current-regulator is required to energize the diode. These are fairly robust devices but are easily damaged by overcurrent, even for a short time, so a proper regulator is really needed. Inexpensive laser pointers do not even use an active regulator.



Fig. 2-10 Red line laser

Both red and infrared diode lasers can be used from old handheld barcode scanners, CD players, laser printers, laser pointers, etc. Once removed, hopefully they have numbers which can be cross-referenced the part number on the manufacturer's site to determine its characteristics - most importantly maximum current. [32]

The photodiode, incidentally, exists on virtually all laser diodes and is the third terminal on the laser diode package. It is designed for feedback to maintain constant light output as the device heats during operation. For simple non-stabilized applications it is not required although intensity of the laser beam will tend to wander as the laser heats during use. Note that the diodes in many cheap red laser pointers (i.e. the under \$10 variety) also lack photodiodes. One inexpensive unit, which ran from three small batteries, occupy nothing more than a surface-mounted resistor for current limiting - no active drive circuitry whatsoever.

The linear laser has been connected to the external power source which consists of 3AA batteries of 1.5V and has been taken from old toy.

- 5 mW
- Focuses on the radius (line)
- Power supply 3.5V ~ 4.5V
- Metal housing

2.6. Scanner Code

The code has created my colleague Marius Motiejūnas, group IFF-1, (using the program OpenCV library) by the example of „3D scanner project based on Arduino and Processing“[33]. The code has been used for the scanning coordinates rotation of the turntable with the video camera recording in spite of first make a sequence of photos. After recording this sequence of images as the item on the turntable rotates 360°, code is accomplished to create a 3D point cloud. A point cloud is just what it sounds like: a cloud of points in 3d space that contains one or more channels of data (lighting, occlusion, area, etc). In our case each having an x, y, and z coordinates. The code for the 3D scanner finishes here and outputs a “test.asc” file, which contains the point cloud. After the file is moved into a program Meshlab, which is able to process a large point cloud, and, by using distinct algorithms that makes small regions to connect each point, a 3D model is created. From this point it can be viewed with nearly any 3D program, and even printed with a 3D printer, as is the motive of this research project. The technique for recording the images is comparatively elementary: as the turntable rotates the object, the camera records a video, and later on it is converted into sequence of photos.

The following step is identification of a series of points from each image. For each image, the y-coordinate is known, as this is the given starting point for each point to be identified. The centre of rotation, or COR, is also known, as it is a part of the initial setup. The x-position of the laser line is established by finding the column with the most intense measure of red light for each row. The distance between the COR and the laser line gives the x-coordinate for each y-coordinate, as shown in Fig.2-11. If the laser angle is slightly inaccurate, it will cause an anamorphic transformation of the x-coordinates, which can be easily regulated.

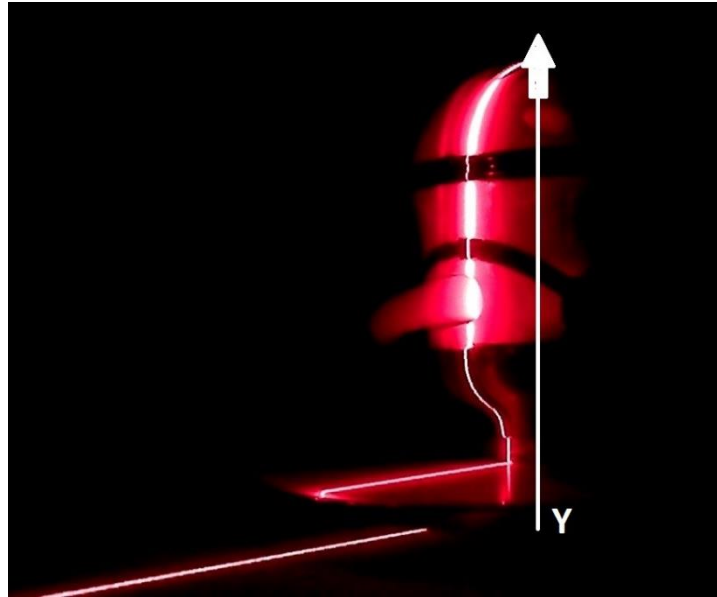


Fig. 2-11 Centre of rotation shown by y-axis, laser line along figure, x-coordinate as difference between y-axis and laser lines

Once x- and y-coordinates have been determined, the z-coordinates have to be added. First, a “cage” is created by rotating the 2D line data around the y-axis. This gives new points, as shown by the following equations:

$$z' = z \cdot \cos r - x \cdot \sin r$$

$$x' = z \cdot \sin r + x \cdot \cos r \tag{3.16}$$

$$y' = y$$

$$R_y(r) = \begin{pmatrix} \cos r & 0 & -\sin r & 0 \\ 0 & 1 & 0 & 0 \\ \sin r & 0 & \cos r & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.17}$$

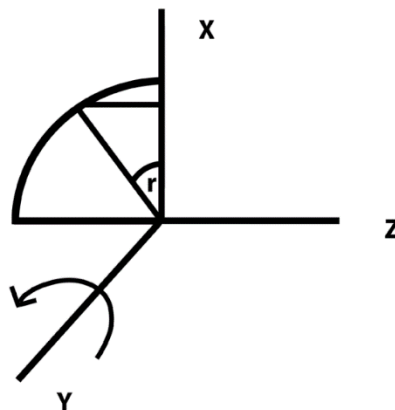


Fig. 2-12 Rotation on y-axis

Where $r = \text{angle of rotation around y-axis} = (360 / \text{number of images}) * \text{image number}$

These are basic formulas for 3D rotation around the y-axis.

However, the scanner always starts with a 2D image, where z is always 0, so the equations can be simplified to:

$$y = y$$

$$x = x(\cos(r))$$

$$z = -x(\sin(r))$$

The y-coordinate will be kept, as it is the point that the others are based off of (the rotation is around the y-axis). The x-coordinate that was previously recorded is equal to $x(\cos(r))$, as it takes the original x-position, and rotates according to Fig.2-12. It can be clearly seen how the x-coordinate becomes $x(\cos(r))$. Because the z-coordinate will always start at 0, all we care about is the new position that the x-coordinate has moved to. Again, it is clear that the z-coordinate is $-x(\sin(r))$, which can be easily derived from Fig.2-12. [34]

Once all of the points have been made to form the point cloud, they still must be made into a functional 3D model. The first process that appears is the formation of the mesh. Whereas there are many approaches to do this, the main idea is to create a region with every single point, such that the regions are all connected with exactly one point-per-region. Once a mesh has been made, many things can still be adjusted. One concern is occlusion; it is typical that a part of the object could be lost, usually because the laser is blocked by other, more in front parts of the object. This can be fixed by simple smoothing if there is a small part missing, or by being solved as 3D equations for the larger holes. Distortion is also another problem, which is easy to repair. As a rule it influences the whole model, so uniform transformations of the points can solve this problem. Smoothing is used for the whole model, so that it is not full of very small toothed points all over. Smoothing reduces differences between points, with neighbouring points making big differences, and points with increasing distance from the point being investigated making less of a difference. This is done to all points in the point cloud. All of these repair techniques are set into the MeshLab software and can be easily implemented.

3. TEST OF THE SCANNING QUALITY

To find out if the scanner is accurate I have measured the figure and then made final measurements in the last program VRMesh.

The results are quite satisfying.

The real figure (Fig.3-1, Fig.3-2) measurements are:

- Height - 8cm
- From back to the fingers – 7cm
- The diameter of an eyeglass – 2.5cm

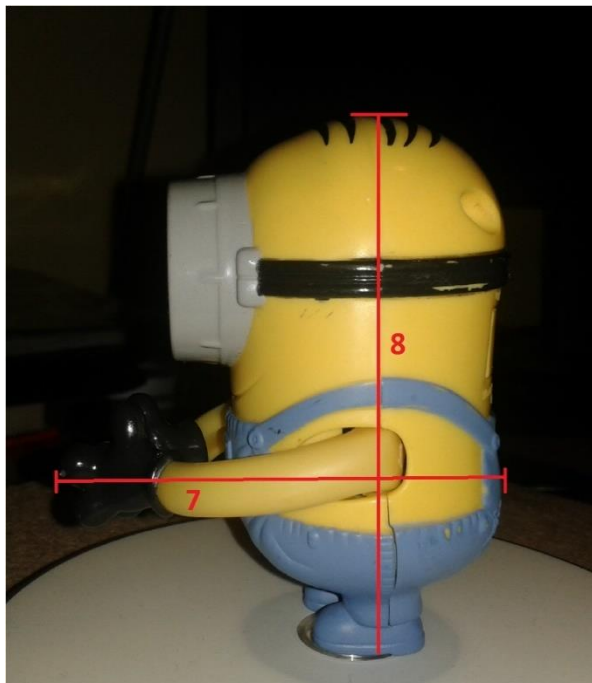


Fig. 3-1 3D rotation



Fig. 3-2 Real measurements of a figure

The measurements of the 3D model (Fig.3-3, Fig.3-4):

- Height – 8.577 cm
- From back to the fingers – 6.667 cm
- The diameter of an eyeglass – 2.137 cm

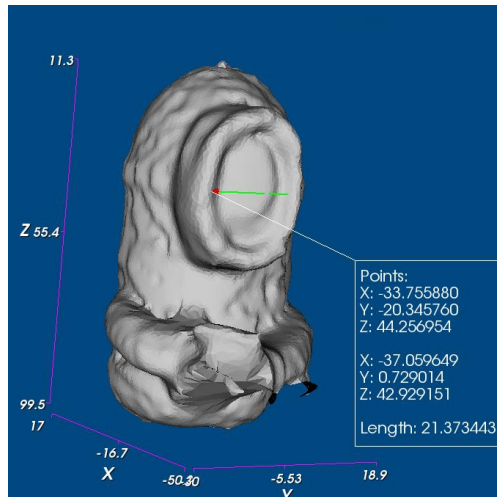


Fig. 3-3 Measurements of a 3D model No.1

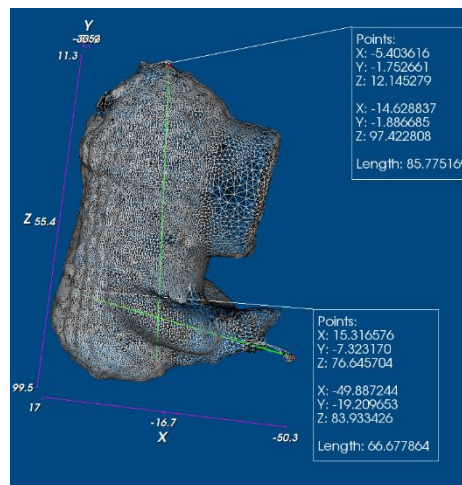


Fig. 3-4 Measurements of a 3D model No.2

2.1. Results

The result of the cleanest scan of the few done so far can be seen below in Fig.3-5. The scan was unable to clearly detect the center of the wheel due to the viewing angle of the camera, and therefore looks a little rough. However, it definitely detected the shape of the scanned object, and the final output looks fairly recognizable. This scan was created by importing the 3D point cloud into Meshlab, and then using the Bounding Box and Show Vertex Dots. This created a rough mesh. After this, I have used a demo program VRmesh for the remaining rough spots to clean and to create more solid body of the scanned statue.

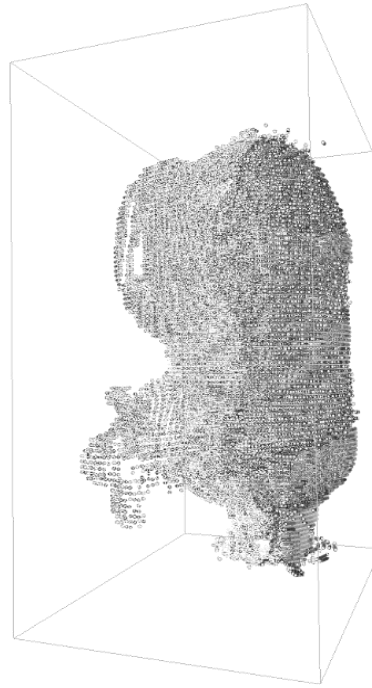
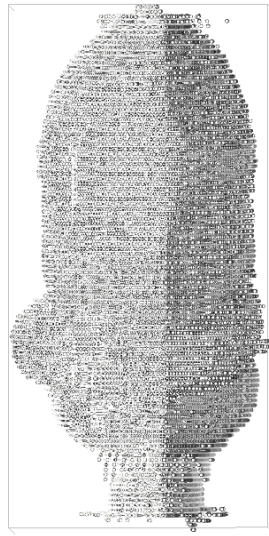


Fig. 3-5 Point cloud in Meshlab

In the Fig.3-6 it's shown the 3D model after solidifying the surface. The minion is clearly not the same shape as of yet, but further investigation of the Meshlab software and edition of the code will make this possible. In Fig.3-7 you can see the original scanned figure.

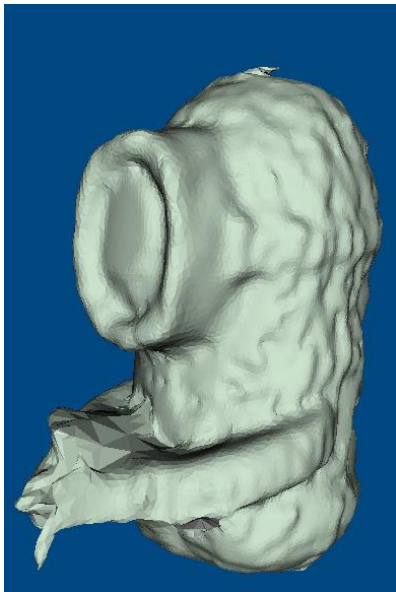


Fig. 3-6 Figure in VRmesh



Fig. 3-7 Original figure

2.2. Modifications and improvements

1. Scanning in the light.
2. Moving camera to different distances and angles.
3. Alternative to Meshlab program tried (Meshmix, GLC Player)

All modifications reflects on the scanning code. Even smallest step made in changing any detail (angle, distance, strength of laser, light) strongly affects the output and makes huge distortions.

Scanning in the light attracts a huge amount of needless pixels, which appears in the Meshlab. Also then the laser light weakens and the code needs to be greatly modified.

Changing the angle and distances of camera also requires to change the code on every step. Also the big minus while moving camera vibrations can not be prevented.

Alternative programs works in both similar and very different manner. While you can make small modifications of a processed files with Meshmix and GLC player, a point cloud conversions can only be achieved by Meshlab.

In the future works with this scanner I would try to use multiple beam laser.

A method to achieve a very high precision of measurement for surfaces with steps or holes, using a 3D multibeam scanner laser.

As mentioned before the projection of the laser on the surface is used to measure the 3D surface with several various methods. Of all known methods for 3D depiction, triangulation provides the most functional way for gaining a high speed-accuracy product [20, 30]. The triangulation method supplies many advantages but it is not immune to particular types of errors caused by occlusion or shading. The problem may be one of the two: either the laser light does not reach the area seen by the camera (Laser Occlusion, Fig.3-8) or camera does not see the area reached by the laser (Camera Occlusion, Fig.3-9). [35]

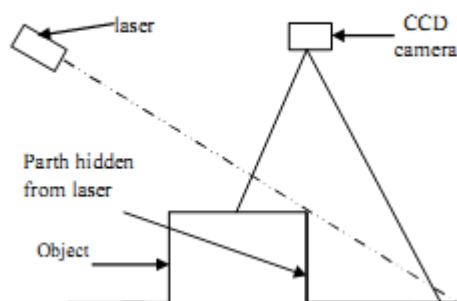


Fig. 3-8 Laser occlusion

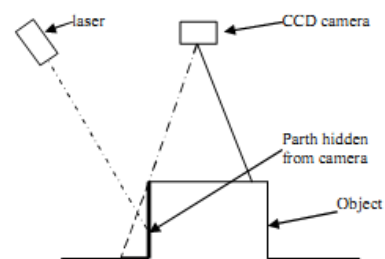


Fig. 3-9 Camera occlusion

Laser occlusion is obviated by guaranteeing that laser reaches all areas seen by the sensor or that the optical center of the laser lens is further from the scene optical center of the sensor. It can also be obviated with multiple laser sources or using a laser multibeam which has a considerable area of scope (Fig.3-10).



Fig. 3-10 Illustration of a multi beam laser and the emitted sheets of light and its area of coverage

Sensor occlusion can only be obviated by using multiple sensors. If two sensors are used, each viewing with baseline but individual sides of the laser, most of the sensor occlusion is avoided. The only remaining sensor occlusion comes from deep holes where both sensors are occluded.

Method of multiple beam lasers description

For image making, there could be used two cameras equipped with CCD sensors [18], which capture the light in small photocells included in a matrix, and a multibeam laser (Fig.3-11). The signals provided by the cameras are amplified and then applied to an ADC [19]. After acquisition, pre-processing operations has to be carried out, having the role to liquidate the noise or redundant information from the image. To remove the noises and to distinguish the surfaces, should be used filtering operation. After filtering, the image has to be divided in more classes by signing them individual pixels.

The CCD cameras should be mounted according to a “pinhole camera model” [20], in which the camera’s centre of projection is just in the centre of the coordinate system and the z-axis coincides with the optical axis of the camera, according to Fig.3-11. The cameras are first calibrated [21-23].

The laser projects on the surface 12 lines which need to be analyzed which of them must be dealt with. For solving this problem, the real surface is approximated with a small plane section. In this case, the coordinates of the features and the intensity distribution around the features must be similar in both images, which allows to restrict the search range and to match features through intensity cross-correlation.

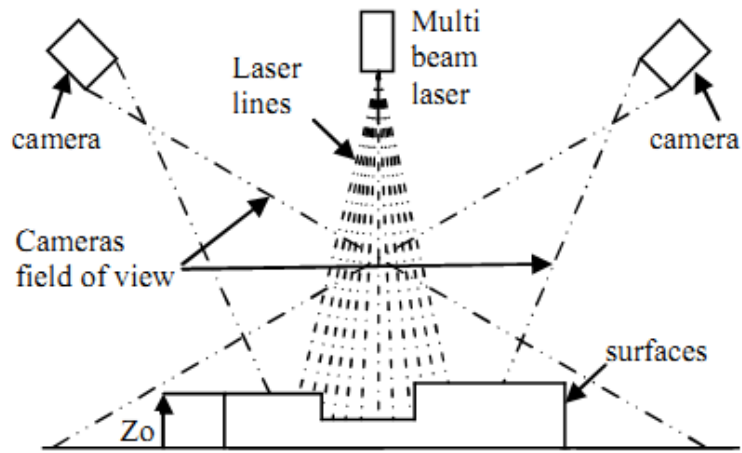


Fig. 3-11 Basic configuration of the multibeam laser method

The laser projects on the surface 12 lines which need to be analyzed which of them must be dealt with. For solving this problem, the real surface is approximated with a small plane section. In this case, the coordinates of the features and the intensity distribution around the features must be similar in both images, which allows to restrict the search range and to match features through intensity cross-correlation.

The measurement should begin with a primal evaluation of the position of the plane section. For each point, it ought to be attributed a value of light intensity. The pixel position should be calculated by bilinear interpolation in the image. The target center in image spread is calculated with sub pixel accuracy by least squares image matching. The 4-pixel neighbours of the target center are pondered for computing a 3D point which roughly conforms to the target center. The rigorous extraction of the 3D points bounded to the targets uses three restraints: intensity of laser light, Euclidian distance of the 3D points from the two 3D target centers, distance between cameras and laser.

The reference height is calculated based on the baseline part of the projected line. It is assumed that the object doesn't fulfil the whole image and therefore a small part of the line is always assured to be projected onto the ground. The depth to the point on the object is found as [24]:

$$Z = \frac{F \cdot L}{X + f \tan \theta} \quad (2.17)$$

where L is the baseline (triangulation distance) partition of the laser and the camera optical centres, f is the focal length of the camera lens, X is the image location within a row where a laser streak is detected, and θ is the projection angle of the camera central axis with

respect to x axis. When laser streak is going to be projected, all the points along the streak should be digitized using a particular algorithm. With smaller triangulation angles, a deduction of shading effects is inherently achieved and estimation of z_o of the object is done with good accuracy for a surface seen by the two CCD cameras and the laser.

3.3. Scanner prototype

The physical dimensions must be 40cmx37cmx5cm

Part of the enclosure is constructed of plastic. It allows the construction to be as light weight and stable as possible. A moving HD camera head with dual lasers and a rotating platform makes scanning three-dimensional objects easy, fast and accurate. After a one-time set up, simply place an object on the scanner, click a button to scan and watch it go. The scanner prototype can be seen in Fig.3-12.

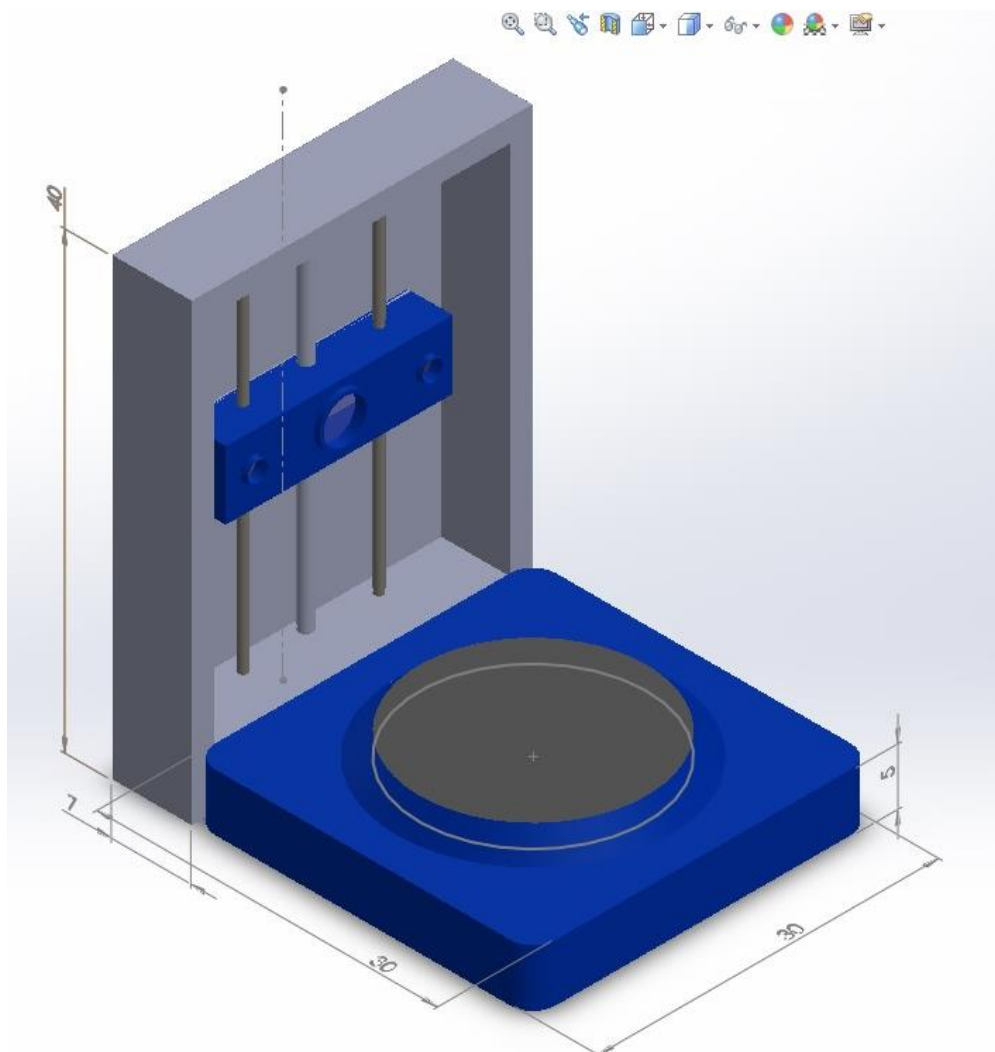


Fig. 3-12 Future Scanner Prototype

CONCLUSIONS

1. In the project I have introduced the main types of 3D scanners already working in the market.
2. I was able to develop a working 3D laser scanner in domestic conditions that costs approximately € 10. After taming various programs as Meshlab, VRmesh it was successfully made in a format that is recognizable in 3D printing programs, as Free CAD, Open SCAD. Initial scans have been completed,
3. Final 3D model is quite accurate for domestic conditions and low knowledge of the main programs. The quality of linear laser is quite low for that reason I think the object isn't in the best shape. However, using the stronger laser might correct the situation. Further investigation into the Meshlab software would also help in order to create cleaner results from what has already been scanned.
4. The final result is quite satisfying. Although the scanner was unable to clearly detect the center of the wheel due to the viewing angle of the camera, and therefore looks a little rough, however, it definitely detected the shape of the scanned object, and the final output looks fairly recognizable. Main peculiarity is that the code of the program is very sharp and can't take any variations from the ones that were provided in the original program code, which causes the biggest problems. Point cloud can't form as a solid body from the very beginning when uploaded to the Meshlab. Furthermore it needs to be processed with the program to make it recognizable to the human eye, and other computer programs.
5. In the future I would try to use a several lasers or multiple beam laser technique to avoid camera occlusion. Also the code must be strongly modified to detect the shape in less time and to receive the color of the figure.

REFERENCES

- [1] General information about David Laser Scanner <http://www.david-3d.com/en/>
[09/04/2014]
- [2] Kimberly Sheppard, Unabridged Guide to 3D Printing, Emereo Publishing, 2012-10-24 - 174 p. [09/04/2014]
- [3] Yi Li, Low cost 3D scanner, Tech.report, Department of Computer Science, Aalborg University, January 2012. [10/04/2014]
- [4] Simon Winkelbach, Sven Molkenstruck, and Friedrich M. Wahl, Low-Cost Laser Range Scanner and Fast Surface Registration Approach, Pattern Recognition (DAGM 2006), Lecture Notes in Computer Science 4174, ISBN 3-540-44412-2, Springer 2006, pp. 718-728. 2006.[03/04/2014]
- [5] Roy Mayer (1999). Scientific Canadian: Invention and Innovation From Canada's National Research Council. Vancouver: Raincoast Books. ISBN 1-55192-266-5.OCLC 41347212.
- [6] Zagorchev, L., Goshtasby, A.: A paintbrush lazer range scanner. Computer Vision and Image Understanding 101 (2006)
- [7] François Blais, Michel Picard, Guy Godin (6–9 September 2004). "Accurate 3D acquisition of freely moving objects". 2nd International Symposium on 3D Data Processing, Visualization, and Transmission, 3DPVT 2004, Thessaloniki, Greece. Los Alamitos, CA: IEEE Computer Society. pp. 422–9. ISBN 0-7695-2223-8.
- [8] Salil Goel, Bharat Lohani (2014). "A Motion Correction Technique for Laser Scanning of Moving Objects". IEEE Geoscience and Remote Sensing Letters: 225–228.[
- [9] K. H. Strobl, E. Mair, T. Bodenmüller, S. Kielhöfer, W. Sepp, M. Suppa, D. Burschka, G. Hirzinger (2009). "The Self-Referenced DLR 3D-Modeler" (PDF). Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2009), St. Louis, MO, USA. pp. 21–28.
- [10] K. H. Strobl, E. Mair, G. Hirzinger (2011). "Image-Based Pose Estimation for 3-D Modeling in Rapid, Hand-Held Motion" (PDF). Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2011), Shanghai, China. pp. 2593–2600.
- [11] Winkelbach, S., Molkenstruck, S., and M. Wahl,F., Low-Cost Lazer Range Scanner and Fast Surface Registration Approach, DAGM 2006, LNCS 4174, pp. 718–728, 2006, Springer Berlin Heidelberg 2006.

- [12] Tsai, R.Y.: An efficient and accurate camera calibration technique for 3d machine vision. In: IEEE Conf. Computer Vision and Pattern Recognition. (1986) 364–374. [03/05/2014]
- [13] General information about laser range finder http://guides.archaeologydataservice.ac.uk/g2gp/LaserScan_1-2 [15/04/2015]
- [14] David Zhang, Guangming Lu, 3D Biometrics: Systems and Applications, Springer Science & Business Media, 2013-05-31 - 302 psl.
- [15] Carl de Boor. A Practical Guide to Splines.: Springer, 1978.
- [16] Jankauskas, Kęstutis ir Tautvydas, Perminas. Gaminio modelio analizė ir koregavimas pritaikant CAM sistemoms. Kaunas : Kauno Technologijos Universitetas, 2006.
- [17] J.P. Kruth, A. Kerstens. Reverse engineering modelling of free-form surfaces from point. Elsevier, Elsevier, April 1998 m.
- [18] General information about Charge-coupled device http://www.axis.com/files/whitepaper/wp_ccd_cmos_40722_en_1010_lo.pdf [30/03/2015]
- [19] Definition of Analog-to-digital <http://www.thefreedictionary.com/Analog-to-Digital+Converter> [30/03/2015]
- [20] Stefano Tubaro, Novara (it); Augusto Sarti, Seregno (mi) (it); Luca Piccarreta, Milano (it); Marco Marcon, Bollate (mi) (it), “Method for determining the three-dimensional surface of an object”. Pub. No.: us patent 2007/0229493.
- [21] O. D. Faugeras and B. Mourrain ”On the geometry and algebra of point and line correspondences between n images” In Proc.ICCV, pages 951–962, 1995.
- [22] Rosario Ancini, Consolatina Liguori, Vincenzo Paciello, and Alfredo Paolillo “A Comparison Between Stereo-Vision Techniques for the Reconstruction of 3D Coordinates of Objects”, IEEE Transactions On Instrumentation And Measurement, vol. 55, no.5, october 2006.
- [23] R.K. Lenz and R.Y.Tsai, „Techniques for calibration of the scale factor and image center for high accuracy 3d machine vision”, IEEE transactions on patten analysis and machine intelligence, 10 (1988) (5), pp. 713–720.
- [24] M. A. Sid-Ahmed and M. T. Boraie, “Dual camera calibration for 3D machine vision metrology”, IEEE Trans. Instrum. Meas., vol. 39, no. 3, pp. 512–516, Jun. 1990.
- [25] F. R. Giordano, M.D. Weir, W.P. Fox. A First Course in Mathematical Modeling. Thomson Brooks/Cole, 2003.
- [26] Marshall Bern, David Eppstein. MESH GENERATION AND OPTIMAL tRIANGULATION. California, World Scientific, 1992.
- [27] Allen Hatcher, The Kirby torus trick for surfaces, 2013

- [28] General information about program Meshlab <http://meshlab.sourceforge.net/>
[08/02/2014]
- [29] General information about program VRmesh <http://www.vrmesh.com/>
[08/02/2014]
- [30] Information about VirtualDub program <http://virtualdub.org/> [08/02/2014]
- [31] Information about Permanent Magnet Synchronous Motor
<http://hefeirishang.manufacturer.globalsources.com/si/6008820799081/pdtl/Synchronous-motor/1049419215/Synchronous-Motor.htm> [20/03/2015]
- [32] General information about diode lasers
<http://technology.niagarac.on.ca/people/mcsele/lasers/LasersOther.htm> [13/02/2015]
- [33] Example of a code <http://www.instructables.com/id/Lets-cook-3D-scanner-based-on-Arduino-and-Proces/step8/Arduino-IDE/> [13/05/2015]
- [34] Information about 3D Rotation
<http://www.cs.helsinki.fi/group/goa/mallinnus/3dtransf/3drot.html> [13/05/2015]
- [35] Michel Bunnell, Dynamic Ambient Occlusion and Indirect Lighting,
http://http.download.nvidia.com/developer/GPU_Gems_2/GPU_Gems2_ch14.pdf
[13/05/2015]

Appendices

Appendix A: Part List

Part	What Needed For	Cost	Where to Buy
Linear laser module	outline contours	€ 10	http://sviesosblyksnis.lt

Appendix B: Programme Code

```

package image.processing;
import image.processing.Panel;
import java.awt.*;
import java.awt.image.BufferedImage;
import java.awt.image.DataBufferByte;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.PrintWriter;
import
java.io.UnsupportedEncodingException;
import org.opencv.core.Core;
import org.opencv.core.Mat;
import org.opencv.highgui.Highgui;
public class Panel {
    private static final long
serialVersionUID = 1L;
    Mat picture;
    /**
     * Converts/writes a Mat into a
BufferedImage.
     * @param matrix Mat of type
CV_8UC3 or CV_8UC1
     * @return BufferedImage of type
TYPE_3BYTE_BGR or TYPE_BYTE_GRAY
     */
    public static Image
matToBufferedImage(Mat matrix, int itr,
PrintWriter writer, int num) {
        int cols = matrix.cols(); //kiek yra
stulpelių (paveiksliuko plotis)
        int rows = matrix.rows(); //kiek yra
eilučių (paveiksliuko aukštis)
        int kampas = 50;
        for (int i = 1; i < rows-10; i++) {
//skanuojam eilutes
            int max = 0;
            float angleitr = (float)360/num;
//kiek vienas kadras apsisuka
            double fi = itr * angleitr;
//paskaičiuojam pasisuktą kampą (itr - kuris
paveiksliukas)
            int maxBrightPos=0, brightestCol
= 1, currentPos =0;
//algoritmas
            surasti ryškiausią pikselį eilutėje
            for (int j = 0; j < cols; j++) {
                double[] r = matrix.get(i, j);
                if (max <= Panel.avg(r)) {
                    currentPos = i * cols + j;
                    max = Panel.avg(r);
                    maxBrightPos=currentPos;
                    brightestCol = j;
                    double x, y, z, b, ro;
                    b=((maxBrightPos+1-
i*cols)-cols/2)/5-10;
                    ro=b/Math.sin(kampas* Math.PI
/ 180);
                    x=ro * Math.cos(fi* Math.PI /
180); //changing polar coords to kartesian
                    y=ro * Math.sin(fi* Math.PI /
180);
                    z=i/5;
                    if(y > 180 || x > 180 || z > 180) {
//truputi sumažina kiekį nesamoningų taškų
                        System.out.println("skip");
                    } else {
                        writer.println(Panel2.fmt(x) + ","
+ Panel2.fmt(y) + "," + Panel2.fmt(z)); //įrašo į
failą kordinates
                    }
                    return
                    Panel.toBufferedImage(matrix);
                    // Create a constructor method
                    public Panel() throws
FileNotFoundException,
UnsupportedEncodingException, IOException
                    //JFrame f = new JFrame();
                    String file = "";;
                    String dir = "images\\";
                    int num = new
File(dir).listFiles().length - 2; //kiek yra
paveiksliukų direktorijoje?
                    PrintWriter writer = new
PrintWriter("test2.asc", "UTF-8"); //pasiruošiam
rašyti į failą
                    for(int i = 1; i <= num; i= i+1) {
//iš eilės atidarinėjam paveiksliukus
                        file = dir + "pic"+
String.format("%04d", i) + ".jpeg"; //paveiksliuko
pavadinimą pasidarome
                        System.out.println(file); //į
konsolę išvedame kurį failą atsidarėme, jog
matytume kada programa veikia

```

```

        picture =
Highgui.imread(file); //atsidaro paveiksliuką ir
paruošia naudoti kaip matrica
        matToBufferedImage(picture, i,
writer, num);
        writer.close(); //uždaro failą
//main
        public static void main(String arg[])
throws FileNotFoundException,
UnsupportedEncodingException, IOException
System.loadLibrary(Core.NATIVE_LIBRARY_N
AME); //biblioteką užkraunam
        Panel a = new Panel();
        //funkcija kuri suranda vidurkį
masyvo elementų (šiuo atveju tai yra pikselio
spalvos RGB [0,0,255] - raudona, [255,255,255] -
balta, [0,0,0] - juoda)
        public static int avg(double r[]) {
            int cnt = 0;
            int sum = 0;
            for (double i : r) {
                sum += (int) i;
                cnt++;
            }
            if (cnt != 0) {
                return (int) (sum / cnt);
            } else {
                return 0;
            }
        }
        //funkcija kuri suformatuoja
skaičių į tinkamą formatą .asc failui
        public static String fmt(double d)
        return String.format("%.2f", d);
        public static Image
toBufferedImage(Mat m){
            int type =
BufferedImage.TYPE_BYTE_GRAY;
            if ( m.channels() > 1 ) {
                type =
BufferedImage.TYPE_3BYTE_BGR;
            }
            int bufferSize =
m.channels()*m.cols()*m.rows();
            byte [] b = new byte[bufferSize];
            m.get(0,0,b); // get all the pixels
            BufferedImage image = new
BufferedImage(m.cols(),m.rows(), type);
            final byte[] targetPixels =
((DataBufferByte)
image.getRaster().getDataBuffer()).getData();
            System.arraycopy(b, 0,
targetPixels, 0, b.length);
            return image;
        }

```

