



A G H

**AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE
FACULTY OF ELECTRICAL ENGINEERING, AUTOMATICS, COMPUTER SCIENCE AND
BIOMEDICAL ENGINEERING**

DEPARTMENT OF AUTOMATICS AND BIOENGINEERING

Master of Engineering Thesis

*Design and development of the system for data acquisition and analysis
for the mobile platform with a set of two manipulators.*

*Projekt i wykonanie systemu akwizycji i analizy danych dla mobilnej
platformy z zespołem dwóch manipulatorów*

Author: *Jakub Olesiński*
Degree programme: *Automatics and Robotics*
Supervisor: *dr inż. Paweł Rotter*

Kraków, 2014

Oświadczam, świadomy(-a) odpowiedzialności karnej za poświadczanie nieprawdy, że niniejszą pracę dyplomową wykonałem(-am) osobiście i samodzielnie i nie korzystałem(-am) ze źródeł innych niż wymienione w pracy.

Serdecznie dziękuję ... tu ciąg dalszych podziękowań np. dla promotora, żony, sąsiada itp.

Contents

Introduction.....	7
1. Mobile manipulation system design	9
1.1. Project background.....	9
1.2. Design requirements	10
1.3. Hardware components	10
1.4. Software architecture	13
2. Depth data acquisition techniques.....	15
2.1. Stereo vision	15
2.2. Time-of-flight	16
2.3. Structured light	17
2.4. Summary and hardware selection.....	19
3. Analysis of the depth data	21
3.1. Basic point cloud processing	21
3.2. The Random Sample Consensus algorithm.....	24
3.3. Descriptors for object recognition	25
3.4. The Iterative Closest Point algorithm.....	26
3.5. Object recognition pipeline	26
4. Implementation and testing	27
4.1. Hardware setting.....	27
4.2. Algorithm implementation	27
4.3. Testing environment and results	27
Summary.....	29

Introduction

The aim of this work is to design and implement a mobile manipulation robotic platform with a basic vision system for object recognition with a three dimensional camera. This work is a part of a project named "Mobile set of manipulators on a wheeled chassis". The project was realized in a three person team, as part of the Second Edition of ABB Students Scientific Association programme organized by ABB Corporate Research Center in Cracow.

In the first chapter, a design of the robotic system is presented. General concept of mobile manipulation is described and possible application areas are mentioned. Subsequently, design requirements, selected hardware components, and implemented software architecture is described. The second chapter provides information about modern depth map acquisition techniques, including stereo, time-of-flight and structured light cameras. For each method, its operation principle basics and general advantages and disadvantages are provided. The third section is focused on modern methods of depth image analysis. Concepts of point clouds and point descriptors are introduced. Moreover, general object-recognition pipeline is provided.

The final section presents an implementation of selected object recognition method. Test results in the real environment are provided. Possible future improvements to the algorithm are also mentioned.

1. Mobile manipulation system design

1.1. Project background

A mobile manipulation system (MMS) is a robotic system that is capable of both manipulating objects and locomotion. Typically, the system is composed of a robotic arm based on a robotic mobile platform. Such configuration enables the manipulator to operate in an unlimited workspace, thus providing great application opportunities.

Typically, mobile manipulators are designed to relieve people in hostile situations. They are, for example, widely used in the field of chemical, biological, radioactive or nuclear (CBRN) defense. Explosive materials and other hazardous substances can be disposed without exposing operators to any danger. Another example is space exploration, where manipulation systems are used in planetary rovers in unmanned exploratory missions on other planets. Substitution of human operators in such expeditions significantly reduce their expenditure and risk.

Augmenting the MMS design with a certain degree of autonomy, brings the possibility for the robot to be used as a human-assistant in the household. Most of current applications in this field refer to pick up and delivery services, that have the potential to improve lives of the elderly, injured and disabled people []. Furthermore, typical service robots in human environments are dedicated to accomplish only a single task, such as vacuum cleaning, lawn mowing, pool cleaning, window washing. Their operation is achieved by adjusting existing domestic appliances with a degree of autonomy. With a multi purpose robotic system such as a mobile manipulator, it is potentially possible not to replace existing devices but to replace the human operator. Autonomous gripping and transportation could also be used in the industry for designing flexible factory plants and intelligent warehouses, as discussed in [].

One of the most promising applications is the PR2 robot from the Willow Garage company. The PR2 robot has already proven successful in such dexterous experiments as opening doors, folding towels and serving beverages to people. Amazon competition.

In MMS robots, the unstructured environment and additional degrees of freedom complicate the control task. Furthermore, the workspace of a manipulator is often shared with people and other vehicles. Such work environment renders many potentially hazardous situations. Therefore, it requires a highly sophisticated control system, with vision based feedback.

1.2. Design requirements

Design requirements:

1. The robotic system should consist of two cooperating serial manipulators, based on a wheeled robotic platform.
2. The whole construction should be made of easily accessible components, preferably available on the consumer market, as it would ease the maintenance of the equipment in the future.
3. Both serial manipulators should be equipped with grippers for general object manipulation.
4. Their reachable workspace should allow to reach objects located on the ground.
5. The workspace of each manipulator should also intersect, to allow collaboration on manipulation tasks.
6. The steering mechanisms of the MMS chassis should be kept simple and convenient to control.
7. The platform is expected to move only on flat surfaces at indoor areas.
8. The MMS robot should possess a control server, that is expected to provide a web interface, developed in a commonly used and well supported standard.
9. That interface is required to provide methods for setting positions on manipulator joints, for setting speed of achieving those positions, and for setting velocity and direction of platform movement.
10. Processing power of the computing unit should be sufficient to analyse camera images in an online fashion.
11. The whole system should be powered by a source that could withstand at least an hour of continuous work.

1.3. Hardware components

Mechanical structure:

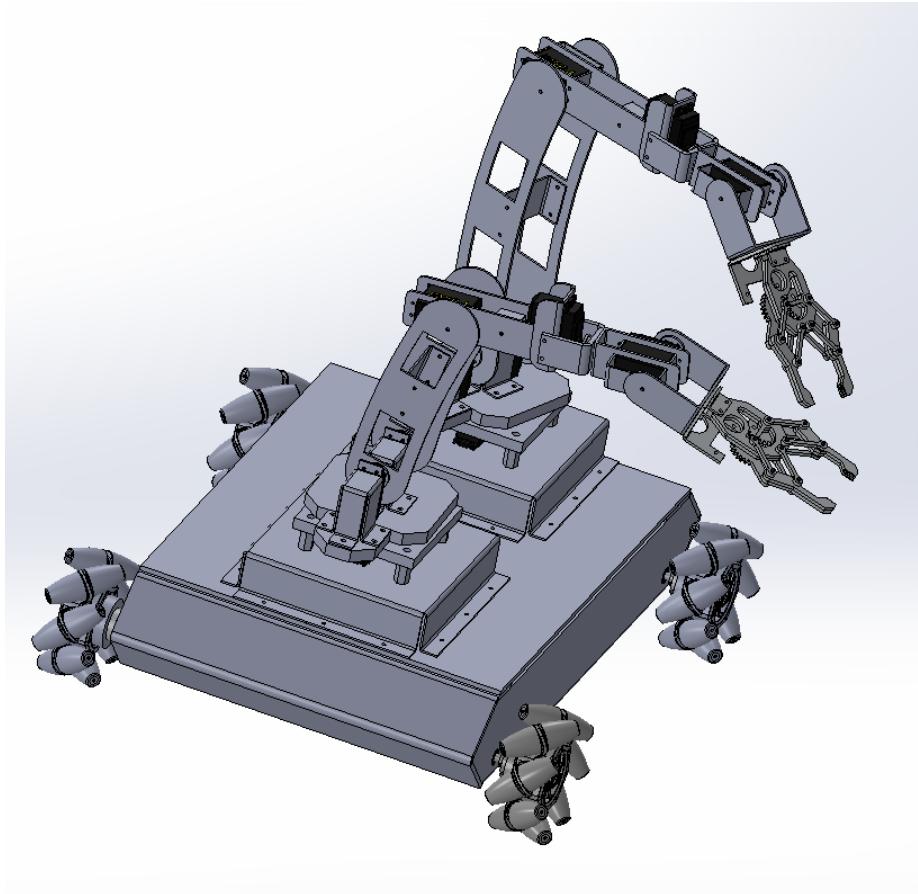


Figure 1.1: Model

The MPTM robot consists of two robotic arms based on a simple four-wheeled mobile platform. Both of robotic arms have 5 degrees of freedom and are in articulated structural configuration. To avoid complicating the mechanical design of the chassis with a steering mechanism, a special kind of wheels were used. Mecanum wheels, as they are called, are equipped with a set of rollers attached to their circumference, which allow a vehicle to move in any direction without turning the wheels.

By alternating wheels with left and right-handed rollers, in such a way that each wheel applies force roughly at right angles to the wheelbase diagonal the wheel is on, the vehicle is stable and can be made to move in any direction and turn by varying the speed and direction of rotation of each wheel. Moving all four wheels in the same direction causes forward or backward movement, running the wheels on one side in the opposite direction to those on the other side causes rotation of the vehicle, and running the wheels on one diagonal in the opposite direction to those on the other diagonal causes sideways movement. Combinations of these wheel motions allow for vehicle motion in any direction with any vehicle rotation.

Another advantage of such design is increased stability of the platform.

Basic components:

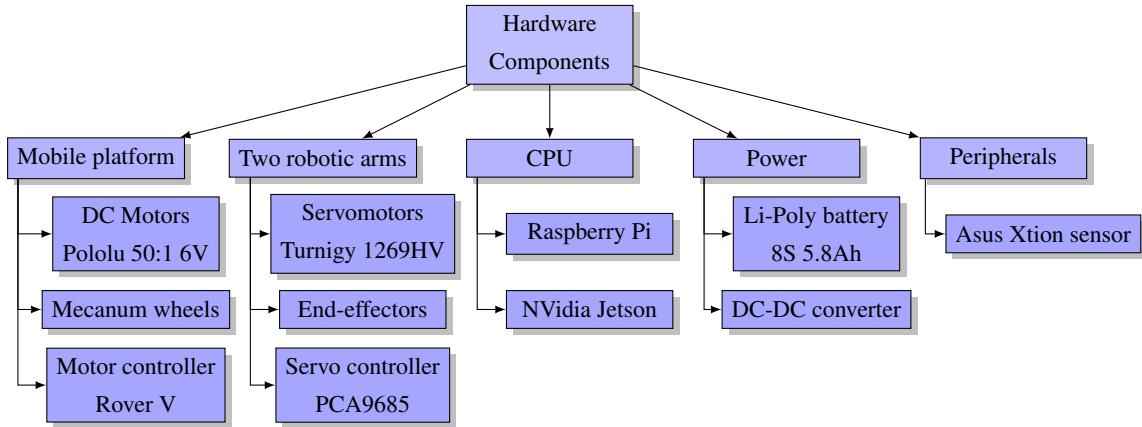


Figure 1.2: Components

The MPTM robot utilises two types of actuators, 4 DC motors for the mobile platform wheels and 12 servomechanisms for manipulator joints. All DC motors have 50:1 metal gearboxes. They achieve 200 RPM of no-load speed and 12 kg cm of stall current. Additionally, they possess 64 CPR encoders, which multiplied by the gear ratio provide 3200 counts per revolution.

Servomechanisms used are Turnigy 1269HV, with operating speed of

Actuators are driven by PWM signal generated by MSP430G2553 microcontrollers, one for each of manipulators and one for the mobile platform.

End effect:



Figure 1.3: Construction

1.4. Software architecture

2. Depth data acquisition techniques

Why use 3D data? Applications in robotics!

The first depth acquisition techniques emerged as a replacement for a contact-based coordinate measuring machines (CMM). CMMs were used in the industrial quality control applications and worked by recording the displacement of a probe tip sliding across a solid surface. Such method was time consuming and inadequate for fragile objects. Modern, contact-less 3D scanning apparatus overcome those limitations by using light to interact with the environment. The new technology had also extended the application area of 3D scanning to the field of robotic perception.

Methods of 3D data acquisition are classified by the light source they utilize to measure depth. Passive techniques rely only on the ambient light, whereas active ones operate by projecting illumination onto an object and recording the reflected beam. In the following sections, examples of both categories are presented, with a brief description of their principles of operation and a discussion of advantages and disadvantages.

Add active projection of artificial texture - complete depth info as in Ensenso cameras.

2.1. Stereo vision

Stereo vision is a passive depth acquisition technique, widely used in research and industry. Its principle of operation is based on human vision system and the biological process of stereopsis, where the disparity between two different projections of the world on the human retinas leads to the depth sensation. Analogically, in computer stereo vision technique, two (or more) displaced in space cameras concurrently acquire the scene view. From the captured images disparity, a scene depth information is then inferred. A typical scheme for stereo vision depth calculation after image acquisition is divided into two steps: the correspondence problem and triangulation.

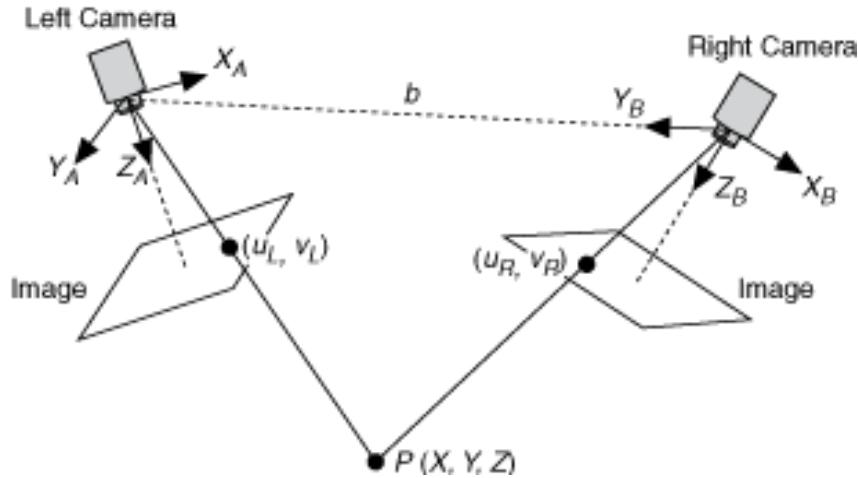


Figure 2.1: Typical Stereo Vision System

Correspondence problem is the most difficult part of the stereo vision. It can be stated as follows: given two displaced 2D images of the same scene, find points representing the same space location in those images. There are many point matching algorithms discussed in the literature. Some of them rely on computing point features based on point neighbourhoods, others compute statistical descriptors of characteristic areas in the image, and then minimise measured difference between analysed regions to find the best matches. The correspondence problem in computer vision is a wide and open research area, that is beyond the scope of this work. For a thorough solution of the problem, the Reader can refer for example to [1].

After obtaining the corresponding points, the point depth information can be derived by the means of triangulation (picture 1). if the imaging properties of the camera are known, two three-dimensional lines, from the camera's projection centres to the examined point can be drawn. The intersection of those lines is then used to infer the depth of the point.

The main advantage of the stereo vision system is that it can be built with easily accessible, standard 2D cameras. Such cameras allow to reduce the expenditure and provide relatively high resolution. On the other hand, the stereo vision systems have many drawbacks. Firstly, their performance is reduced in environments with low ambient light intensity, which makes the system impractical in some settings. Secondly, solving the correspondence problem is computationally expensive and often limit the depth acquisition frame rate. **Finally...**

2.2. Time-of-flight

Time of flight (ToF) cameras work on a completely different principle than stereoscopic systems. ToF systems are characterised with active illumination and are composed of an near infra-red (IR) emitter and IR camera. They work by illuminating the scene with a modulated IR beam and recording the

received light, as presented in the figure 5. The distance from the recorded object is then calculated by measuring the phase shift between the illuminated and reflected signals.

ToF cameras, due to their specific architecture, are prone to errors caused by radiometric, geometric and illumination variations. The power of the emitted IR signal limits their measurement accuracy. The light entering to the sensor has an ambient and reflected components, thus high ambient light intensity reduces the signal to noise ratio. Moreover, the material and color of the object surface cause variations in the amplitude of the reflected IR signal.

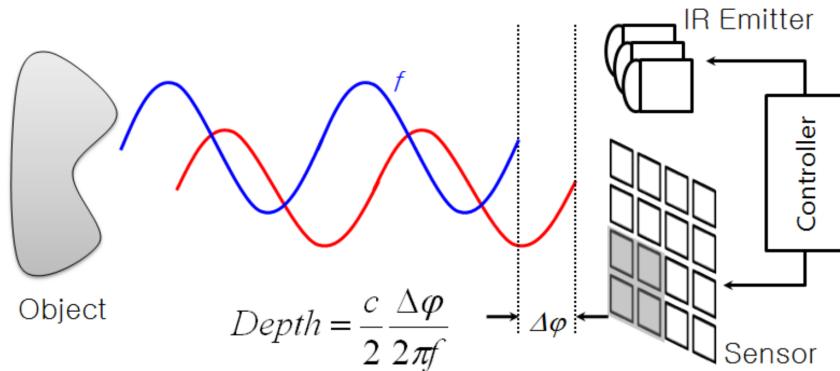


Figure 2.2: ToF principle

Rewrite this. The simplest version of a time-of-flight camera uses light pulses or a single light pulse. The illumination is switched on for a very short time, the resulting light pulse illuminates the scene and is reflected by the objects in the field of view. The camera lens gathers the reflected light and images it onto the sensor or focal plane array. Depending upon the distance, the incoming light experiences a delay. As light has a speed of approximately $c = 300,000,000$ meters per second, this delay is very short: an object 2.5 m away will delay the light by 16 ns.

Despite problems arising from ToF principle of operation, modern ToF cameras are distinguished by relatively low latency and fast scanning speed. The depth measurements are acquired directly from the hardware, so the speed is not limited by software. Unfortunately, most of the products available on the end-user market provide relatively low depth image resolution, typically QQVGA (160x120) and the pricing rises dramatically with the resolution.

2.3. Structured light

The structured light (SL) depth measurement technique combines some of the features of both time of flight and stereo vision principles. Similarly to ToF cameras, SL relies on active illumination, work in the near infra-red and is composed of an IR emitter and IR camera. As in the case of stereo vision, depth information is inferred from the disparity between two images by the means of triangulation. In this case, the projected IR pattern is compared with the image captured by the IR camera. The IR projector emits patterns of non-coherent light, which then appears distorted from the perspective of the camera. In

such settings, the projection of defined patterns makes explicit correspondences on the reflected image. A typical setting of the SL system is presented in figure 5.

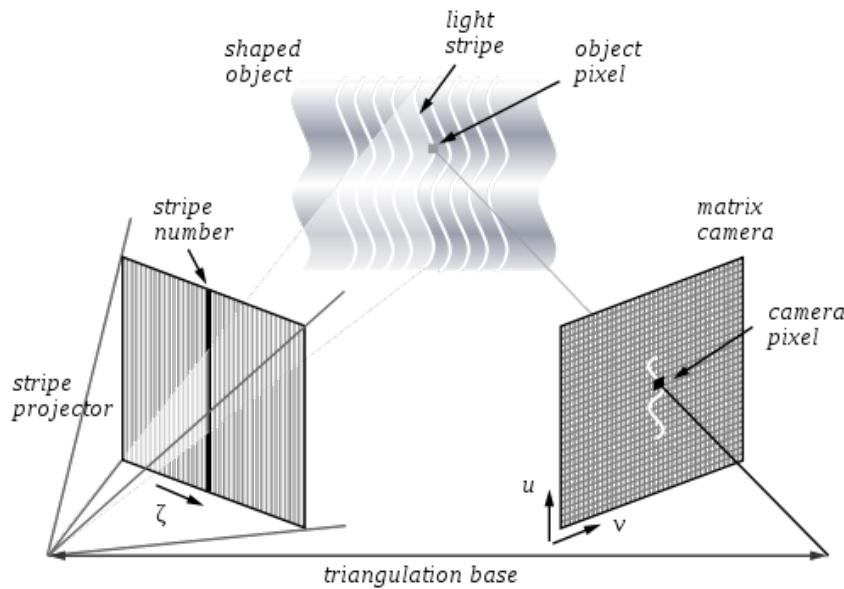


Figure 2.3: Structured light

There are many pattern strategies that allow for correspondence identification, including projections of grids, dots, vertical slits and multi-color patterns. In the literature, particular attention is paid to fringe patterns (figure 6), which are suitable to maximize the measurement resolution [sensors]. Moreover, to reduce the reconstruction artefacts, the measurement process is often extended into a sequence of different pattern projections. A comprehensive assessment of such codes can be found at [SPB04 from byo3d]

Depth measuring devices that employ the SL technique suffer from many drawbacks. Utilizing multiple pattern frames introduces high latency and makes the measurement ill conditioned for dynamic scenes. Moreover, the system performance is degraded in bright ambient light. On the other hand, popular SL devices are characterized with relatively high, VGA (640x480) depth map resolution and are easily accessible on the consumer market.

Particularly noteworthy application of SL based depth data acquisition device is the Kinect by Microsoft. In addition to an RGB camera and an array of microphones, the device includes an IR projector-camera pair from the PrimeSense Ltd. company, used for depth measurements. The SL light pattern used in Kinect is a non-periodic speckle pattern produced by the interference of partially coherent beams [TN-2011]. In this device, every pixel is identified in the IR image using a correlation window, after which the depth information is calculated, using triangulation. The Kinect device is produced as a gaming controller for the Xbox 360 console and it is widely available at the consumer market with a relatively low price. Moreover, there are many open source drivers for the Kinect device, which make it a perfect match for robotic and research applications.

PICTURE WITH KINECT AND THE SPECLE PATTERN

2.4. Summary and hardware selection

Three of the most popular depth map acquisition techniques were presented in this chapter, including stereo vision, time of flight and structured light. Each method has its advantages and disadvantages and they all have already been successfully applied in robotics. A comparative summary of each method characteristics is presented in Table 2.1.

Feature/Technique	Stereo Vision	Time of Flight	Structured Light
Depth data generation	Directly out of chipset	High software processing	Medium software processing
Latency	Medium	Low	Medium
Low light performance	Weak	Good	Good
Bright light performance	Good	Medium	Medium
Power consumption	Low	Medium - scales with distance	Medium - scales with distance
Resolution	Camera dependent	QQVGA, QVGA	VGA, 1080p
Accuracy	mm, cm	mm, cm	µm, cm
Scanning speed	Medium - limited by software complexity	Fast - limited by sensor speed	Fast - limited by camera speed

Table 2.1: Qualitative comparison of depth acquisition techniques [2]

In order to fulfil project assumptions, one solution providing depth map measurement had to be chosen. In the stereo vision technology, ready-made devices unfortunately did not fit within the project budget. Utilizing this technique would therefore require an own design of the stereo vision system. Such solution would probably be the least expensive, but require an extensive amount of work and time. Therefore, the author focused on finding a ready-made solution in the remaining technologies. The DepthSense 311 [3] device was found as a representative of the ToF technique, and the Asus Xtion Pro Live [4] was considered as an option from the SL method. Both devices are available in similar prices and provide the same frame rate of 30 frames per second. The Asus device, however, offers higher depth map resolution (VGA instead of QVGA) and has better support from the open source community. The Asus Xtion is based on the PrimeSense Ltd. depth sensing hardware, similarly to the popular Kinect device and is even compatible with the same open source drivers. Nonetheless, it was chosen over the Kinect device,

because it has smaller size and is powered directly from the USB port (the Kinect requires external power source). Main specifications of the Asus Xtion Pro device are provided in the Table 2.2.

Operating range	Between 0.8m and 3.5m
Field of view	58 °H, 45 °V, 70 °D
Depth map resolution	VGA (640x480) : 30 fps QVGA (320x240): 60 fps
Interface	USB2.0/ 3.0
Dimensions	18 x 3.5 x 5 cm

Table 2.2: Asus Xtion Pro Live specification

3. Analysis of the depth data

The acquired depth information can be stored in computer memory in two ways. The first is the depth map, which takes the form of a two dimensional array, similarly to the plain gray-scale image. In this case, however, the depth measurement is stored in place of the color intensity. Depth map is the simplest way to represent and store the acquired depth of the scene and it is usually obtained directly from the sensor driver. The main disadvantage of a depth map is its inflexibility. This representation is strictly bound to the camera point of view and thus it is inconvenient in further processing.

The second representation of a scene's depth information is a point cloud. Generally speaking, it is a set of data points in some coordinate system. Typically, the Cartesian system is used and the points are defined by their x, y and z coordinates. Point clouds are derived from depth maps and offer new capabilities, such as viewpoint transformation or cloud concatenation.

Furthermore, most software libraries provide an interface to interactively manipulate and visualize the point cloud, which is a useful tool when

This chapter presents the main tools used further in the implementation of the autonomous control mode for the MMS robot.

3.1. Basic point cloud processing

A frequently used operation during the processing of a point cloud is the affine transformation. Point clouds can be translated, rotated and scaled by multiplying a transformation matrix $A \in \mathbb{R}^{4x4}$ with data points in homogeneous coordinates $[x, y, z, 1]^T$. Basic transformation matrices are presented in the Figure 3.1. Presented transformations can be further composed by multiplication to produce more complex ones.

$$\begin{aligned}
A_t &= \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} & A_s &= \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & A_x &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta_x) & -\sin(\theta_x) & 0 \\ 0 & \sin(\theta_x) & \cos(\theta_x) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
\text{Translation by a vector} && \text{Scaling along } x, y, z \text{ by factors} && \text{Rotation around } x \text{ with} \\
t = [t_x, t_y, t_z]^T && S_x, S_y, S_z && \theta_x \text{ angle} \\
\\
A_y &= \begin{bmatrix} \cos(\theta_y) & 0 & \sin(\theta_y) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta_y) & 0 & \cos(\theta_y) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & A_z &= \begin{bmatrix} \cos(\theta_z) & -\sin(\theta_z) & 0 & 0 \\ \sin(\theta_z) & \cos(\theta_z) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
\text{Rotation around } y \text{ with } \theta_y \text{ angle} && && \text{Rotation around } z \text{ with } \theta_z \text{ angle}
\end{aligned}$$

Figure 3.1: Basic affine transformations

Preprocessing. Another useful type of operation is spatial filtering. Probably the most basic filter is the pass-through filter, which rejects all points outside a given range along a specified dimension. This procedure allows to focus the analysis process on the region of interest, i.e. the reachable workspace of a manipulator. Apart from limiting the cloud dimensions, the number of data points can be also reduced by downsampling. The voxel grid filter is typically used for this purpose. This filter creates a three dimensional regular grid over the input point cloud data and then, in each voxel, approximates all the present data points with their centroid. Such reduction is particularly useful when large point cloud datasets have to be processed online with limited computing resources. Finally, the filtration process has to cope with numerous measurement errors present in the raw data acquired from the 3D camera. Such measurement noise manifests itself in the form of sparse outliers, which corrupt the results of further processing, i.e. the surface normals estimation. The impact of those irregularities can be reduced by applying an outlier removal filter. The simplest form of such filter rejects all the data points which does not have enough neighbours within a specified radius. A more refined version is based on the neighbouring points distance distribution. Firstly, the mean distance from each point to all its neighbours is calculated. Next, based on the assumption that the resulting distribution is Gaussian, all the points whose mean distances lay outside of an interval defined by the global distances mean and their standard deviation are rejected from the dataset. The effects of statistical outlier removal are presented in the Figure 222.

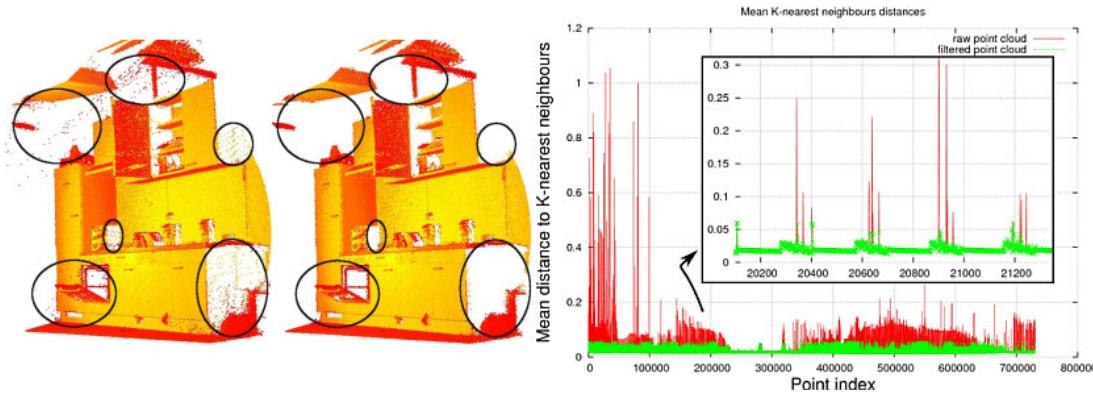


Figure 3.2: Statistical outlier removal [tutorials]

In a given point cloud, a single point by itself does not provide much information about the surface to analyse. Therefore, an essential concept in depth data analysis is the local neighbourhood of a point. For a query point p_q in the point cloud P its neighbourhood is given by:

$$d(p_q, p_i) \leq r \quad (3.1)$$

where $p_i \in P$ is the neighbouring point, d is the selected metric, typically Euclidean, and $r > 0$ is the neighbourhood radius. In practice, approximate methods are used, as direct application would require calculation of distance from the query point p_q to all other points in P . The algorithms used for neighbourhood search require a parameter k specifying the maximum number of points in the neighbourhood or parameter r , denoting the maximum search radius. Proper determination of those parameters is crucial in further analysis. Too small values will not provide enough information about the surface. Too large, on the other hand, will average the surface and skip small details.

Many of the algorithms used further during the analysis of a point cloud base on the notion of a surface normal, known from the 3D geometry. These vectors are also widely used for shading in 3D computer graphics and a variety of methods have already been developed to solve the surface normal estimation problem. One of the simplest uses the least-squares plane fitting to estimate the normal to a plane tangent to the surface, which approximates the desired vector [Rusu]. More specifically, for a given point on the surface, it's k -neighbourhood centroid \bar{p} is calculated as:

$$\bar{p} = \frac{1}{k} \cdot \sum_{i=1}^k p_i \quad (3.2)$$

The tangent plane is thereafter defined by the centroid \bar{p} and the sought normal vector \vec{n} . The latter is computed by minimizing the total (squared?) distance from every k -neighbour p_i to the tangent plane, given by $\sum_{i=1}^k (p_i - \bar{p}) \cdot \vec{n}$. The minimization problem can be solved by utilizing the covariance matrix $C \in \mathbb{R}^{3 \times 3}$, given by:

$$C = \frac{1}{k} \sum_{i=1}^k (p_i - \bar{p}) \cdot (p_i - \bar{p})^\top \quad (3.3)$$

The covariance matrix C is symmetric, positive semi-definite and possess three real eigenvalues $\lambda_j \geq 0, i = 1, 2, 3$. The eigenvector corresponding to the smallest eigenvalue is an approximation of the desired normal vector \vec{n} , disregarding the sign. Furthermore, if the viewpoint v_p is known, the normal \vec{n} has to be oriented towards v_p , which means that it has to satisfy the equation:

$$\vec{n} \cdot (v_p - p_i) > 0 \quad (3.4)$$

3.2. The Random Sample Consensus algorithm

The indoor human environment is abundant of regularly shaped objects that could be described with basic geometrical models, such as planes, spheres or cylinders. The plane model:

$$\theta_1 \cdot x + \theta_2 \cdot y + \theta_3 \cdot z + \theta_4 = 0 \quad (3.5)$$

is particularly useful, as the floor, walls or furniture is typically composed of flat surfaces. By knowing which points belong to the surface of a floor, the robot can autonomously navigate and avoid collisions. For this reason, a robust model fitting algorithm is a strongly desired tool in the analysis of the depth data. The point dataset received from the depth camera , however, consists of both points that belong to the model, the inliers, and a lot of other points in the scene, the outliers. Therefore direct usage of classic model fitting algorithms, such as the least squares method, would not provide the desired effect, as they try to fit the model into all the input data points, including outliers. As an alternative, the Random Sample Consensus (RANSAC) algorithm, can effectively cope with such problems. In its basic form, the RANSAC algorithm is essentially composed of two, iteratively repeated steps [Dummies,Wiki]:

1. Firstly, a minimal sample subset is randomly selected from the input dataset. The model parameters are computed using only the selected subset. The cardinality of the subset is the smallest sufficient to determine the model parameters.
2. Secondly, the remaining dataset points are tested to be consistent with the model computed in the sampling step. A data point will be considered as an inlier if it fits the computed model within a defined error threshold. The set of such elements is called a consensus set. If the consensus set contains enough points, the model is reestimated from all selected inliers and evaluated with the error of the inliers relative to the model.

This procedure is then repeated until a termination condition is met, which usually is a fixed number of iterations. The main advantage of the RANSAC algorithm is the robust estimation. This method is able to fit a model with high accuracy even if the data set contains a significant amount of outliers. On the other hand, the algorithm in its basic form has several disadvantages. There is no upper bound on the time needed to estimate model parameters and by limiting the number of iterations, the obtained solution is may not be optimal. Furthermore, the RANSAC algorithm can only estimate one model per dataset and if multiple models exist, it may fail to estimate any of them. Since the original RANSAC

was first published in 1981, it has been widely adapted by the image processing community and many modifications that address the RANSAC limitations have been proposed. A comparative summary of the recent extensions to the RANSAC algorithm can be found for example in [dunno].

3.3. Descriptors for object recognition

Descriptors what for. Difference between global and local. Itemize some global and local with brief description. Table with comparison. Reference to detailed info.

Object recognition - problem formulation. Point cloud matching etc.

Surface normals are the most basic representation of the geometry around a certain point. Even if coupled with surface curvature, they usually do not provide enough descriptive information for object recognition and pose estimation. To achieve better performance in such tasks, more complex and higher dimensional descriptors have been proposed in the literature [summary]. A descriptor is considered to be good if it is able to capture the same surface characteristics, regardless of rigid transformations, varying sampling density and noise. In general, 3D shape descriptors are divided into local and global. The former describe only the local geometry around a query point, while the latter represent the geometry of a whole object. A few selected descriptors will be presented further in this section.

The Point Feature Histogram (PFH) is a generalization of both surface normals and curvature estimates. It represents the relative orientation of normals between every point pair (p_i, p_j) in the neighbourhood of the query point p_q . For each point pair, using the surface normal n_i at p_i , a new coordinate frame u, v, w is constructed as follows:

$$u = n_i, \quad v = u \times \frac{p_i - p_j}{d}, \quad w = u \times v \quad (3.6)$$

where $d = \|p_i - p_j\|_2$ is the Euclidean distance between p_i and p_j . Using this reference frame, the difference between normals at p_i and p_j is expressed by the angular features α, ϕ, θ , given by:

$$\alpha = v \cdot n_i, \quad \phi = u \cdot \frac{(p_i - p_j)}{d}, \quad \theta = \arctan(w \cdot n_i, u \cdot n_i) \quad (3.7)$$

Finally, to create the PFH descriptor, the angular features are binned into a histogram. The angular ranges are typically divided into 5 subdivisions, thus receiving a $3^5 = 125$ binned histogram, that counts occurrences of any value combination for every point pair (p_i, p_j) .

The main disadvantage of the PFH descriptor is its $O(nk^2)$ complexity, where n is the number of points in the point cloud and k is the number of each point's neighbours. For large datasets, this is one of the major bottlenecks during online processing. To overcome this problem a simplification to the PFH formulation, called Fast Point Feature Histogram has been proposed (FPFH) [Proposal]. In the first step of the FPFH, the angular features α, ϕ, θ are computed only between the query point p_q and its k -nearest neighbours, as described in Equation 3.7. Those features produce a histogram, called Simplified Point Feature Histogram (SPFH). The SPFH is computed for every point in the cloud, and then, the FPFH

descriptor is formed as follows:

$$FPFH(p_q) = SPFH(p_q) + \frac{1}{k} \sum_{i=1}^k \frac{1}{w_i} SPFH(i) \quad (3.8)$$

where w_i is a distance between p_q and p_k in some metric space. The FPFH descriptor reduces the computational complexity of the PFH to $O(nk)$, while maintaining similar descriptive performance.

Figure: Influence diagrams

SHOT or VFH

3.4. The Iterative Closest Point algorithm

3.5. Object recognition pipeline

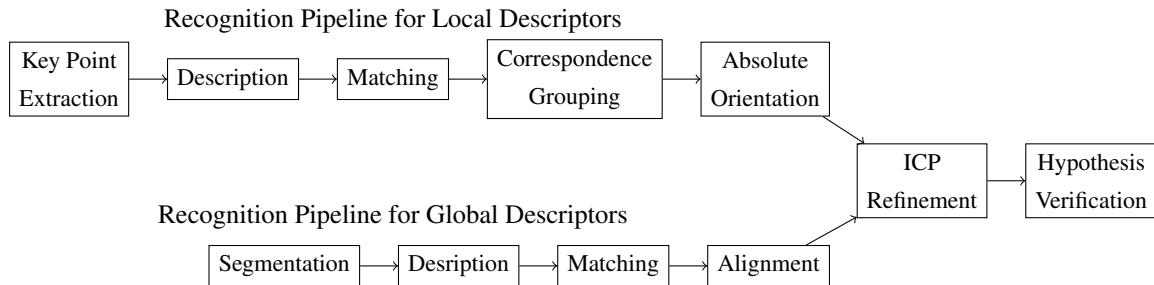
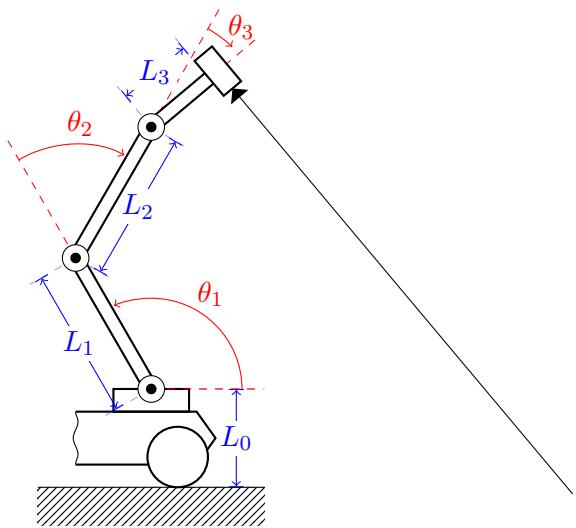


Figure 3.3: Object recognition pipeline [IEEE]

4. Implementation and testing

4.1. Hardware setting



4.2. Algorithm implementation

4.3. Testing environment and results

Summary

Bibliography