|  |  |
| --- | --- |
| Injection Molding Machine Pin Detection System | Abstract  This Research Project is about finding a possible solution to detect the failed retraction malfunction that occurs in the injection molding machine's ejection pins during its operation. By exploring various technologies and implementations, a proof of concept has been developed to potentially solve the problem using Depth Surface Analysis. An implementation of this concept has been developed using the MATLAB and Python platforms. With the achieved results of the proof of concept, a recommendation for this real environment has been produced.  PrePared BY  Shreyas Macwan  Gabriel Stewart  PROJECT MANAGER  Bassim Elhassan  PRINCIPLE INVESTIGATOR  Hadi Adibi |

Table of Contents

[1 - Introduction 3](#_Toc38384042)

[1.1 Project Summary and Explanation 3](#_Toc38384043)

[1.2 Aim and Objective 3](#_Toc38384044)

[2 - Problem Analysis and Possible Solutions 4](#_Toc38384045)

[2.1 Problem Statement 4](#_Toc38384046)

[2.2 Initial Research Findings 4](#_Toc38384047)

[2.2.1 Thermal Imagery Analysis 4](#_Toc38384048)

[2.2.2 Image Recognition / Machine Learning Analysis 5](#_Toc38384049)

[2.2.3 Point Cloud Analysis (PCA) 5](#_Toc38384050)

[2.3 Development Approach and Explanation 6](#_Toc38384051)

[2.3.1 What is a Point Cloud? 6](#_Toc38384052)

[2.3.2 What is a 3D Camera? 6](#_Toc38384053)

[2.3.3 PCA Strategy 6](#_Toc38384054)

[3 - Proposed Research and Solution Methodology 7](#_Toc38384055)

[3.1 Implementation approach 7](#_Toc38384056)

[3.1.1 POC Using Mock Mold 7](#_Toc38384057)

[3.1.2 POC Using Training Mold 7](#_Toc38384058)

[3.1.3 Utilized tools 9](#_Toc38384059)

[4 - Research Results and Findings 10](#_Toc38384060)

[4.1 Observations from Mock Mold (Section 3.1.1) 10](#_Toc38384061)

[4.1.1 Point Cloud representation of Mock Mold (Reference) 10](#_Toc38384062)

[4.1.2 Point Cloud representation of Mock Mold from Camera 11](#_Toc38384063)

[4.1.3 Point Cloud Alignment 11](#_Toc38384064)

[4.1.4 Anomaly Detection 12](#_Toc38384065)

[4.2 Observations from Training Mold (Section 3.1.2) 13](#_Toc38384066)

[4.2.1 Results from MATLAB tool 13](#_Toc38384067)

[4.3 Time Complexity 14](#_Toc38384068)

[4.3.1 MATLAB Environment 14](#_Toc38384069)

[4.3.2 Python Environment 14](#_Toc38384070)

[5 - Research Summary and Conclusion 15](#_Toc38384071)

[5.1 Recommendations 15](#_Toc38384072)

[5.1.2 Trigger Mechanism 16](#_Toc38384074)

[5.1.3 On-Site Computation 16](#_Toc38384075)

[5.1.4 3D Camera 17](#_Toc38384076)

# 1 - Introduction

## 1.1 Project Summary and Explanation

Injection molding is a widely used manufacturing process for producing a large volume of plastic products. The Vuteq Company has a gigantic plastic production environment setup. In which a wide range of Injection Molding Machines are being used to manufacture interior panel components.

Vuteq has recently experienced rare but catastrophic problems with their ejection systems in the molding machines. One or more ejection pins used to push completed components out of the mold cavity will become stuck in a partially retracted position while the machine starts a new cycle. This stuck ejection pin will then be forcefully crushed into the reciprocating surface of the mold causing serious damage to the mold surface and potentially shutting down production.

## 1.2 Aim and Objective

This project aims to investigate a means of detecting an occurrence of this problem and warning the operator before damage can occur. The project explores multiple technologies and their potential application to this issue. The end result is a proof of concept demonstrating a selected technologies capable enough to accomplish the goal of mold surface obstruction detection.

# 2 - Problem Analysis and Possible Solutions

## 2.1 Problem Statement

The molding machine uses a molten plastic to fill a mold cavity (space between Cavity Block and Core Block) before cooling. When cooled the plastic retains the shape of the mold cavity it occupies. In the final stage of the molding process, the core block is retracted and several hundred Ejector pins extend to push the completed plastic part out of the mold cavity. This completed part is then collected by a robotic pick and place machine and the ejector pins then retract and the cycle repeats.

The problem occurs very rarely. One or more ejector pins will suffer a catastrophic failure somewhere between the Ejector Retainer Plate and the surface of the Core Block. This failure will cause the damaged pins to remain partially extended when the cycle repeats. If the machine is not stopped before the two mold surfaces make contact in the next cycle, the damaged pin will be forced into the surface of the cavity block.  This causes indentation and texture damage to the surface of the mold which will need to be removed from the machine and re-tooled before the operation can continue.

## 2.2 Initial Research Findings

Below is the list of potential technologies we initially considered researching and implementing into a feasible solution.

* Thermal Imagery Analysis
* Image Recognition / Machine Learning Analysis
* Point Cloud Analysis(PCA)

### **2.2.1 Thermal Imagery Analysis**

We considered the possibility that thermal footage of the mold surface would reveal damaged pins. We wondered if the ejector pin shafts would be cooler than the surface of the core block, and due to this difference in temperature it would appear as a cooler (bluer) color in the thermal footage. To test this concept, we requested thermal footage from Vuteq. Upon review of the footage however we could see no significant difference in temperature between the pin shafts and the surface. As such this approach was ruled out.

**Pros:** Output from thermal camera which basically is a thermal image does not get affected by the ambient light and provides accuracy.

**Cons:** Temperature difference between ejector pin shafts and faces is too insignificant to be detectable by thermal camera.

### **2.2.2 Image Recognition / Machine Learning Analysis**

The idea was to introduce artificial Intelligence (AI) using Image recognition technique in which a neural network trained on properly retracted mold surfaces can be trained to detect the broken pins. AI would be able to detect the difference in images taken when the pins had been properly retracted and when a pin had failed.

**Pros:** Machine learning process can be quickly adapted to variety of mold surfaces.

**Cons:** Training on very small differences between stored model and live data would prove difficult. Detection on such small differences would be largely unreliable. Deviation in position of mold surface as it shifts cycle to cycle would add complications.

### **2.2.3 Point Cloud Analysis (PCA)**

Having had prior experience using depth cameras, we considered applying this knowledge to the problem. Depth cameras are capable of generating point clouds to accurately model subjects (conceptually to 3D scanning technology). We considered that if we could create an accurate point cloud representing the current surface of the mold, and compare it to what we know the mold should look like, differences could be detected and highlighted.

**Pros:** This type of analysis provides a detailed profile of a surface which can be digitally compared to other 3D file formats.

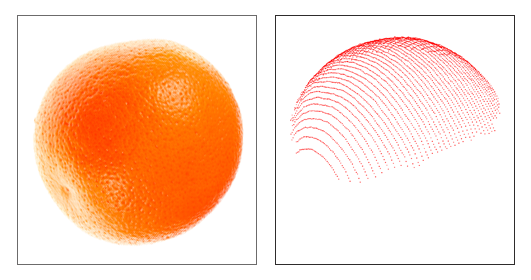
* Point Cloud Processing or Analysis have a vast community and efficient algorithm support for making such application by utilizing them.
* Application Based on point cloud processing can be optimized and scaled up according to the need.

**Cons:** Even though this analysis provides very promising support, determining which type of 3D hardware should be used and the algorithms should be considered can be time consuming in the initial phase.

## 2.3 Development Approach and Explanation

This section provides a fundamental explanation of the technology used in the proof of concept.

### **2.3.1 What is a Point Cloud?**



A point cloud is a set of data points in three-dimensional space representing the profile of a surface. Point clouds are generally produced by 3D scanners or cameras in this case.

### **2.3.2 What is a 3D Camera?**

The basic principle of 3D camera is to measure depth of objects calculating how far the objects are from each other as well as camera perspective. 3D camera uses types of different techniques to resolve the distance/depth. Such as Time-of-flight, stereo vision.

### **2.3.3 PCA Strategy**

In the point cloud analysis approach, point cloud generated from 3D camera and from a reference CAD model of the mold surface is aligned and compared with each other such that the profiles and their corresponding point cloud data are identical. This alignment and comparison procedure is called Point Cloud Registration. Pins in the extended position will appear as a series of dots on the Live Point Cloud which do not correspond to points on the stored CAD generated point cloud. This deviation can be detected and used as a means of detecting not only pins that have failed to retract, but also detect the presence of any foreign material of enough size.

# 3 - Proposed Research and Solution Methodology

After considering the technologies discussed in **section 2.2** and distinguishing their feasibility, we decided that the best technology for this proof of concept was the Point Cloud Analysis (section 2.2.3).

## 3.1 Implementation approach

In order to effectively test the technologies feasibility for this problem, we needed to assemble a method for testing the proof of concept without interfering with operations at Vuteq. In order to accomplish this, we created a plastic surface with cylindrical protrusions to mimic a mold surface. This will be referred to as the **Mock Mold** in this report.

### **3.1.1 POC Using Mock Mold**

The Mock Mold was created using a 3D printer in our Lab. The Mock Mold has a smooth semi-reflective surface and a series of contours that roughly mimic the reflectiveness and complexity of the real mold surface at Vuteq. This Mock Mold was used during the initial stages of research in 3D surface analysis for our early exploration into point cloud creation and registration.

**A picture containing table, sitting, black

Description automatically generated A picture containing sitting, black, water

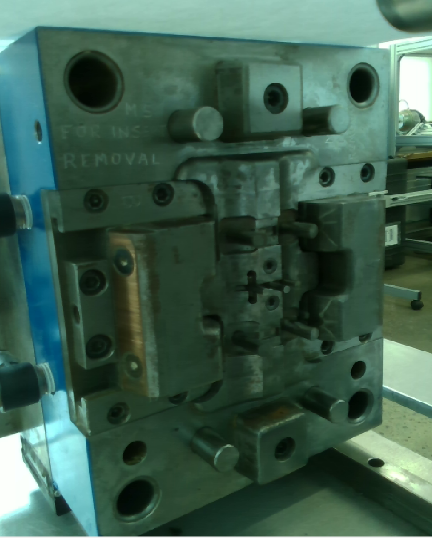
Description automatically generated**

**[TOP VIEW] [DIAGONAL VIEW]**

Using the stored CAD model and live data gathered using this 3D printed mock mold, we successfully performed a registration and deviation analysis test. (Please see Chapter 4 for results of this testing).

### **3.1.2 POC Using Training Mold**

After demonstrating the early proof of concept on our manufactured Mock Mold, we were given access to a training mold used by Vuteq. This training mold more closely resembled the actual mold surface of the Molding Machine and would hence be a better surface for our proof of concept development. Additionally, we were able to design and build mounting solutions for the training mold that allowed us to create a more routine registration process.

**** ****

**[CAMERA MOUNTED DIAGONALLY] [VIEW FROM CAMERA]**

In order to accomplish PCA strategy defined above, two different technologies are used to develop proof of concept.

1. Python/Open3D.
2. MATLAB.

Both above mentioned computational environment follow below described steps and implements PCA strategy.

Step 1: Capture depth frame from Intel Real Sense D435 Depth Camera. (This frame contains the XYZ point cloud data of the scene).

Step 2: Crop Region of Interest (ROI) from the captured frame, removing other peripheral objects detected.

Step 3: ROI contains large amount of data points. Therefore, to improve performance, perform down sampling and de-noising.

Step 4: Align both sets of point cloud data within the environment so that deviation can easily be seen through a process called point cloud registration.

Step 5: **Show anomaly** in the result.

3.1.3 Utilized tools

In the creation of the Proof of Concept, several tools were used:

* The Mock Mold and Training Mold (Please refer to section 3.1).
* The Intel Real Sense 3D camera module



* MatLab computation environment.
* Python / Open3D computation environment.

# 4 - Research Results and Findings

After successfully utilizing the tools and implementation approach, we have acquired satisfactory results which includes the point cloud interpretation of an object (Mock and Training mold’s surfaces to be observed), Alignment and pin detection based on the computational environment.

The results are shown in the following sections.

## 4.1 Observations from Mock Mold (Section 3.1.1)

The following images are point cloud captures of the Mock Mold used during the initial proof of concept. This Mock Mold was later replaced by the Training Mold (Section 4.2).

### **4.1.1 Point Cloud representation of Mock Mold (Reference)**

This point cloud has been generated from a CAD file which contains the geometry of the Mold Surface with no pins extended. This is the target point cloud.

**A picture containing computer, table

Description automatically generated**

**[POINT CLOUD VIEW]**

### **4.1.2 Point Cloud representation of Mock Mold from Camera**

This point cloud was generated from the 3D camera in real-time. It represents a snapshot of what the camera sees in a given frame. This is the actual point cloud.

A picture containing computer, dark, black, table

Description automatically generated

**[CAMERA GENERATED POINT CLOUD VIEW]**

### **4.1.3 Point Cloud Alignment**

This point cloud represents the registration view after Target point cloud (section 4.1.1) and Actual Point cloud (section 4.1.2) have been aligned through the registration process.

**A picture containing table, circuit, monitor

Description automatically generated**

**[POINT CLOUD REGISTRATION]**

### **4.1.4 Anomaly Detection**

This image shows the detection and highlighting of the deviated points (red) which are interpreted as potential ejection pin failures. Due to the lightning condition in our Lab environment where the entire mechanism has been set and performed, and such limitations with the camera, in the final output (please refer the image below) we can notice red points nearby the edges of the mold surface which can be eliminated further by introducing smooth brightness around the camera and mold surface and also using high-resolution camera (please refer section 5.1.4).

**A picture containing table, sitting, dark, computer

Description automatically generated**

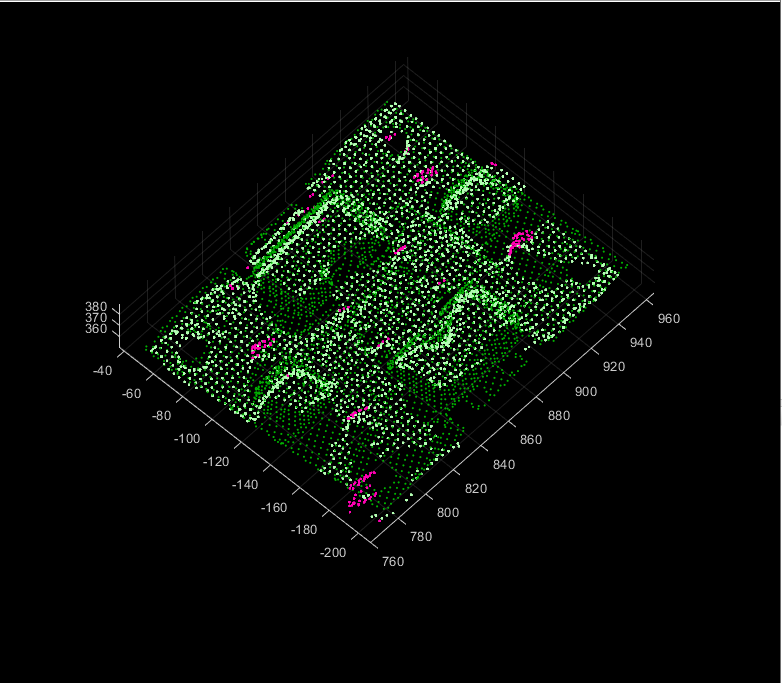
**[RED POINTS DEPICT PINS AND EDGES]**

**[GREEN POINTS DEPICT MOLD SURFACE]**

## 4.2 Observations from Training Mold (Section 3.1.2)

The following images are point cloud captures of Training Mold used in the second iteration of the Proof of Concept. The Proof of Concept was performed with this new Training Mold as it more closely resembled the Thermoplastic Molding Machines surface.

### **4.2.1 Results from MATLAB tool**

****

**[DEVIATION DETECTED AFTER REGISTRATION]**

**[Unreacted pins are depicted in pink color]**

# 4.3 Time Complexity

Tables below in section 4.3.1 and 4.3.2 illustrates time taken by each computational environment i.e. Matlab and Python respectively. Due to the fact that the entire procedure has to be near or under ~1 second time span, we have designed our applications such a way that the entire PCA steps should be processed staying close to the time limit.

### **4.3.1 MATLAB Environment**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Process** | **Initial Startup** | **CYCLE 2** | **CYCLE 3** | **CYCLE 4** | **CYCLE 5** |
| **Time in seconds** | **4.0 – 7.0** | **0.65** | **0.48** | **0.40** | **0.41** |

### **4.3.2 Python Environment**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Process** | **Initial Startup** | **CYCLE 2** | **CYCLE 3** | **CYCLE 4** | **CYCLE 5** |
| **Time in seconds** | **1.2 – 1.3** | **0.87** | **0.84** | **0.83** | **0.84** |

***\*All the values in table above are approximated.***

***These values are subject to change upon camera intrinsic performance, number of cloud points generated from camera, CPU performance (including Cross-platform library support), and the ambient lighting conditions.***

In the table above, Initial startup refers to the phase where the entire system is turning on for the very first time and do the operation. In order to enable the camera and the system as a whole, it takes a few extra seconds. Whereas the word cycle refers to the number of time the system or application has performed the entire PCA steps.

# 5 - Research Summary and Conclusion

Since the beginning of the Project this Term, a great deal of progress has been made. Various technologies have been researched and considered for their feasibility in creating a Proof of Concept. 3D surface analysis through point cloud registration was determined to be the most fruitful of our available technologies and was therefore developed into a working analysis concept on both a plastic Mock mold and a realistic Training mold. An application was developed to roughly demonstrate our thoughts on what a working system for surface analysis may look like, and surface analysis was performed in two environments (MatLab and Open3D) achieving desirable results.

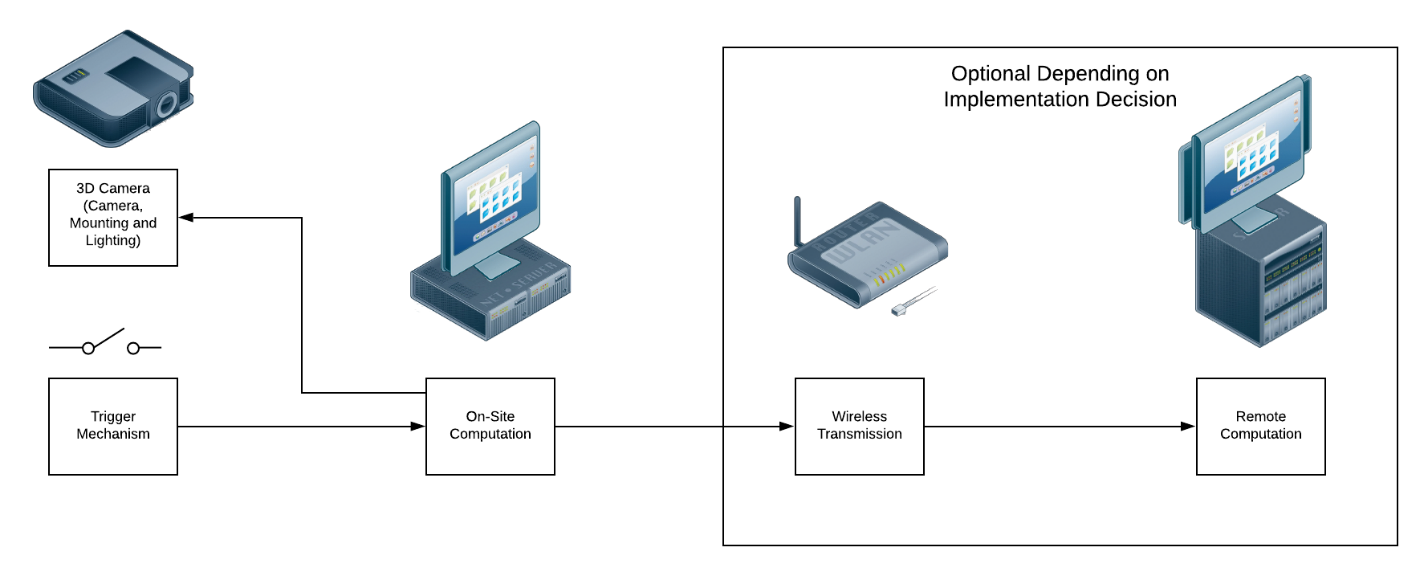
As this term comes to an end, we have made tentative plans for what next steps can be taken as we strive to develop a completed solution to the ejection pin problem. Developing a solution will require a large amount of research and testing on the actual injection molding machine. Specifics of this proof of concept have to be determined before testing begins.

## 5.1 Recommendations

In the next stage of the project, we recommend running the proof of concept on the actual mold surface. We recommend this because the current proof of concepts software and hardware performance is only known in the testing environment we’ve created over the past few months. We need to determine which cameras will be suitable for the application in the actual environment as well as how this camera will be mounted. We do not currently know how our current camera (Intel D435) will react to the Injection Molding Machines environment, the surface material of the mold, the limited lighting, the noise or the vibration of the machine. It is not likely that this camera will be used in a final solution, but it will be helpful in isolating the exact shortcomings we need to rectify in the selection of a new camera (Section 5.2). We also need to determine the locations for mounting of the cameras used on the mold surface. Since there will likely be multiple cameras, a series of mounting locations and solutions will need to be determined. Finally, the decision between computation on-site and wireless transmission before computation remotely needs to be made.

### **5.1.1 Hypothetical System Design**

We have determined what components we believe we be necessary in the next stage of development of this project. The following diagram details the interaction between the individual system components, and an explanation of each component can be found below the diagram.



**[HYPOTHETICAL SYSTEM DESIGN]**

### **5.1.2 Trigger Mechanism**

We believe that the trigger system should be a limit switch installed on the molding machine. There is already a series of such switches installed for the purpose of verifying alignment, if this signal could also be used to trigger analysis it would be a reliable cost-effective method. The signal generated from the signal would be received by an **On-Site Computation** system which would signal the camera to capture the live point cloud data of the mold surface. Since the time period for data capture is so brief, a physical trigger system seems to be the most reliable.

### **5.1.3 On-Site Computation**

When the limit switch is triggered, its signal will need to be responded to in order to trigger the camera. The analysis of the point cloud data needs to be performed, and its results produced, within a very brief period of time (~1 second). The possibilities for accomplishing this are still up for debate. A system could be installed at the machine itself to determine the results and present them. Conversely, point cloud data could also be transmitted via **Wireless Transmission** to a **Remote Computation** system for processing and analysis. These issues will need to be discussed further upon the project’s continuance.

### **5.1.4 3D Camera**

The camera, mounting and lighting also needs to be determined. We have selected a series of 3D cameras which we believe could be suitable for this application. In addition to the camera, lights may also be required to get optimal results. This is dependent on the camera chosen as various 3D camera technologies require varying levels of lighting. The camera mounting solution will also need to be researched further. We understand that mounting locations are limited on the machine, so further collaboration between the Conestoga team and Vuteq will be needed to find a suitable mounting solution.

Below listed are the camera suggestions which can be utilized in order to successfully fulfill project requirements. The reason these cameras have been chosen is that they are equipped with quality hardware and technologies that provide the highest quality 3D output. Based on preliminary research we believe that the following cameras represent a subset of possible cameras and technologies that could suit the purpose of this proof of concept.

These camera selections are not final by any means, and additional cameras or technologies may be selected in further development.

**INTEL REAL SENSE L515 LIDAR**



This camera is Intel’s first flagship enabling the LiDAR technology to provide accurate depth sensing with its compact design. Through its laser scanning technology it provides 3D representation of the scene with high resolution. Due to its small form factor and less in weight it can be mounted easily.

|  |  |
| --- | --- |
| DEPTH TECHNOLOGY | LiDAR |
| DEPTH FIELD OF VIEW(FOV) | 70° × 55° (±2°) |
| DEPTH STREAM OUTPUT RESOLUTION | Up to 1024 x 768 |
| DEPTH OUTPUT FRAME RATE | 30 fps |
| MIN – MAX DEPTH DISTANCE | Approx. 0.25 m – 9 m |
| SCANNING TECHNOLOGY | Laser Scanning |
| CAMERA DIMENTION | 61 mm x 26 mm |
| MOUNTING MECHANISM | - One 1/4-20 UNC thread mounting point - Two M3 thread mounting points - Tripod |
| USE ENVIRONMENT | INDOOR |
| PRICE | $349 |
| SOURCE | <https://www.intelrealsense.com/lidar-camera-l515/> |

**ZIVID ONE+ LARGE**



Using one of the advanced technique called structured light projection, Zivid’s One+ captures high quality point cloud data with color details. With its unique high dynamic range (HDR) capabilities the camera is able to produce advanced point cloud of challenging shiny surfaces and also of dark objects.

|  |  |
| --- | --- |
| 3D TECHNOLOGY | STRUCTURED LIGHT |
| FIELD OF VIEW,  MIN – MAX DEPTH DISTANCE | 843 x 530 @ 1.2 m  2069 x 1310 @ 3.0 m |
| POINT CLOUD OUTPUT | 3D(XYZ) + COLOR(RGB) + CONTRAST(C) FOR EACH PIXEL |
| BRIGHTNESS | 1/4 TO 1.8x, 1x = 400 lumens |
| KEY APPLICATIONS | Medium to large objects, Reflective metal surfaces |
| SOFTWARE SUPPORT | Zivid SDK |
| API + LIBRARIES | C++, **.**NET , Python, ROS |
| PLATEFORM | WINDOWS, LINUX |
| SOURCE | <http://www.zivid.com/zivid-one-plus-large-3d-camera> |

**PHOTONEO - PHOXI 3D SCANNER XL**



Coming with industry’s best sensor for unbeatable overall performance, Phoxi 3D scanner / camera module covers large field of view with high pixel confidence in terms of depth. Its integrated GPU provides precise depth details by eliminating ambient light noise.

|  |  |
| --- | --- |
| 3D TECHNOLOGY | STRUCTURED LIGHT |
| DEPTH RESOLUTION | Up to 3.2 Million 3D points |
| DEPTH SCANNING RANGE | 1680 – 3780 mm |
| OPTIMAL SCANNING DISTANCE | 2326 mm |
| 3D POINTS THROUGHPUT | 16M points per second |
| DIMENTIONS | 77 x 68 x 941 mm |
| API + LIBRARIES | C++ AND OTHER 3RD PARTY |
| SOURCE | <https://www.photoneo.com/products/phoxi-scan-xl/> |

**HELIOS 3D CAMERA**



Helios 3D camera marketed by LUCID vision labs features time of flight technology supported by Sony’s Depth sensor provides high intensity 3D point clouds which can be processed in real time.

|  |  |
| --- | --- |
| 3D TECHNOLOGY | Time Of Flight |
| DEPTH RESOLUTION | 640 x 480 px |
| MIN – MAX DEPTH DISTANCE | 0.3 to 6.0 m |
| FRAMERATE | 15 fps / 30 fps |
| SOFTWARE SUPPORT | Arena SDK |
| API + LIBRARIES | C, C++, C#, Python |
| PLATEFORM | Windows and Linux |
| Source | <https://thinklucid.com/product/helios-time-of-flight-imx556/> |