DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
THE UNIVERSITY OF TEXAS AT ARLINGTON

SYSTEM REQUIREMENTS SPECIFICATION
CSE 4316: SENIOR DESIGN I
FALL 2017

The Taraminator
By Visioneers

DANIEL STENBRO
MITCHELL SHELTON
BIPLOW GHIMIRE
JUSTIN HANCOCK
SHERVIN OLOUMI

Revision History

Revision	Date	Author(s)	Description
0.1	03.10.2017	DS, MS, BG, JH, SO	document creation
0.2	03.20.2017	DS, MS, BG, JH, SO	complete draft

CONTENTS

List of Figures

1 PRODUCT CONCEPT

This section of the document provides a high-level explanation of the product concept as well as the targeted audience. The Taraminator is a computer vision system that is based on Tara stereo camera. Our product captures two images and creates a point-cloud map that can be used to calculate depth. The point-cloud data would be available on the on-board microcomputer, which will be running a server. For instance, a robot arm in a factory can send frequent requests to the on-board microcomputer to read the data. Additionally, the system will be controllable through the cloud. This gives the users the ability to interact with the system directly and intervene in case of a miss-step by the robot arm. While the targeted audience includes mainly the manufacturing industry, Taraminator can be used by computer vision researchers, autonomous vehicles and other areas with a need for computer vision.

1.1 PURPOSE AND USE

The Taraminator shall be designed as an open source camera, in which will be ultized to aid hobbist and organizations for industrial use.

1.2 INTENDED AUDIENCE

The open source API along with the tara camera will provide researchers, manufacturers and robot enthusiasts a way to develop their own devices.

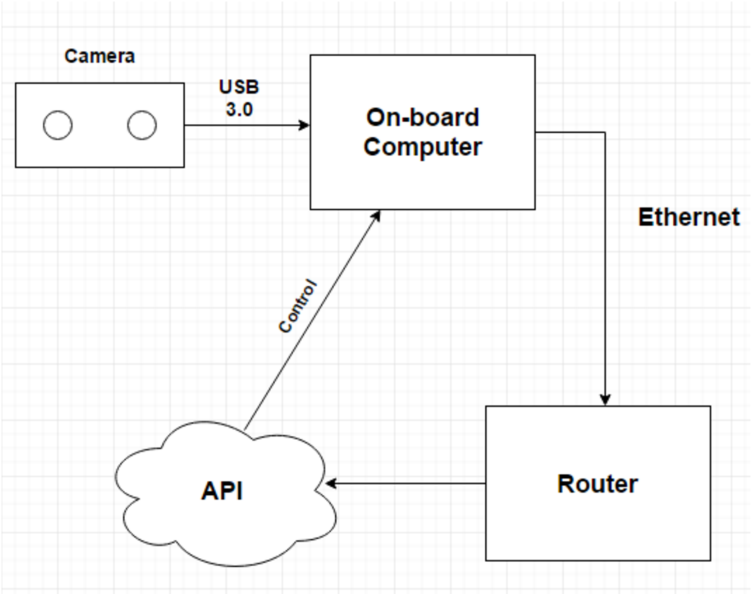

Figure 1: Ethernet Conceptual Drawing

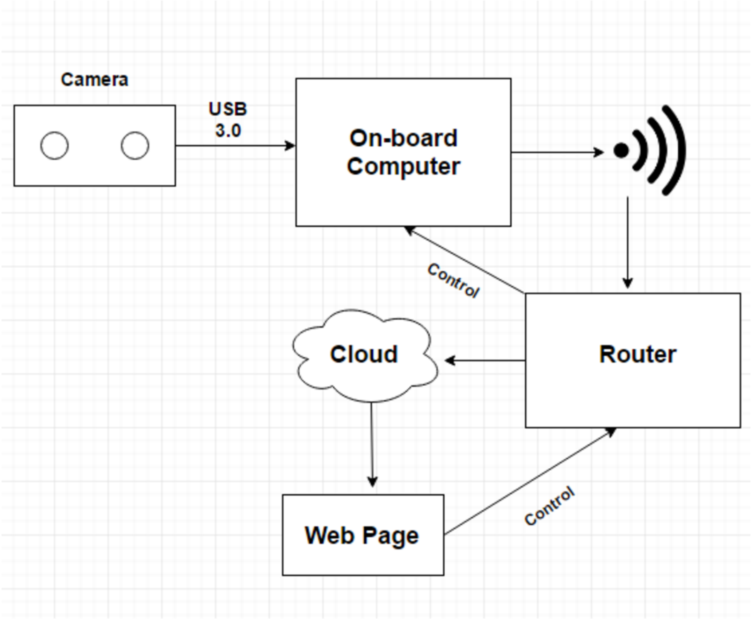

Figure 2: Wireless Conceptual Drawing

2 Product Description

Computer vision opens the doors for new tangible innovations that involve object detection, dimension calculation and calculating distance in objects. 16 years ago OpenCV made it more accessible to develop applications that involve computer vision. Our project will make the computer vision field even more accessible to people who may want to utilize depth cameras in their projects. Along with OpenCV libraries, the open source API shall add robust libraries that can improve performance of many robotic applications, camera applications or other computer vision applications.

2.1 Features & Functions

The open source API will be structured around the Tara development kit, and a webserver. This kit has an SDK which has embedded OpenCV libraries that will be built upon.

2.2 External Inputs & Outputs

Information gathered by the camera will be transferred through USB to an onboard computer. From there, the computer board will run a computation on the data points gathered. The information that is calculated will be sent through ethernet or wifi depending on the developers choosing. The end machine will then use that data to provide an input for an action or otherwise.

2.3 Product Interfaces

Because our product is an API so our primary user base will be developers. This would mean it would differ what the interface would be depending on the project it is utilized for. We will distribute our API via github.

3 Customer Requirements

The open source API will contain libraries that will allow the end developer to create their platform. The libraries will allow methods such as Object detection, color detection, and depth perception.

3.1 Open Source Platform

3.1.1 Description

The end developer will access the API through a GitHub. The content will constantly get updated with new features and bug fixes.

3.1.2 Source

The source of this requirement will be located on www.github.com

3.1.3 Constraints

The github project must be available and able to implement features for most of the camera functionality.

3.1.4 Standards

The ability for the general public to access the content and develop (or approved) or maintain the project via a web platform like github. This will allow the project to facilitate interoperability and data exchange among different products or services and are intended for widespread adoption.

3.1.5 Priority

- Critical

3.2 Ethernet Interface

3.2.1 Description

The ethernet interface will allow the information collected from the camera will be passed off to a external device i.e a robot or machine.

3.2.2 Source

Source

3.2.3 Constraints

Ethernet will allow the data gathered from the camera to be compressed and passed to an end machine. At this point, the end machine will interpret the data and display the content.

3.2.4 Standards

The Ethernet requirement shall meet the requirements defined under IEEE 802.3

3.2.5 Priority

- Critical

3.3 Web Server

3.3.1 Description

The on-board computer which will be used for all computation shall run in the background and handle any requests the end machine requests.

3.3.2 Source

Senior Design team requirement. The source of this requirement will be located on www.github.com

3.3.3 Constraints

The web server will serve as a focal point for all information that's translated from the camera to an external source.

3.3.4 Standards

This standard must meet team requirements to satisfy other tasks.

3.3.5 Priority

- Major

3.4 Image Compression Data

3.4.1 Description

The images gathered in the cameras point of view must be compressed into data points. These points will then be used to redisplay the picture on the external machine or medium.

3.4.2 Source

The source of the image compression standard can be found at cs.rutgers.edu.

3.4.3 Constraints

The compression algorithm that can be used for this must be able to complete the compression in a reasonable amount of time.

3.4.4 Standards

The image compression method in which is used must abide by The JPEG2000 Standard or The JPEG Standard

3.4.5 Priority

- Major

3.5 Point Cloud

3.5.1 Description

The images gathered in the cameras point of view must be in a point cloud format. This will allow the user to demonstrate what camera or robot sees when interacting with objects in the real world

3.5.2 Source

Senior Design requirement

3.5.3 Constraints

The data gathered will be on a 3D plane in coordinates such as X,Y,Z.

3.5.4 Standards

The standards used must be able to display in a 3D plane format and clearly visible to the user.

3.5.5 Priority

- Major

3.6 Depth Perception

3.6.1 Description

The images gathered in the cameras point of view must be in a point cloud format which can be used to display a depth for the robot or user. Along with this idea, the developer can also overlay two images together, calculate the depth between the overlay and display a distance differential. This can be used to sense certain distances between objects in the image.

3.6.2 Source

Senior Design requirement

3.6.3 Constraints

The data gathered will be on a 3D plane in coordinates such as X,Y,Z. Along with the point cloud, the image overlay should display the measurement gathered from the two images.

3.6.4 Standards

The units used to measure distance should match any formatting standards and possibly able to use multiple types of units (metric or imperial).

3.6.5 Priority

- Major

4 Packaging Requirements

The distribution of this product shall be provided on an open source platform. The open source API shall be accessed and also managed GitHub. Any changes to the implementation will be approved by the source provider, tested, and released on newer versions of the API.

4.1 Open Source access

4.1.1 Description

The end developers shall be able to access the open source API through github.com.

4.1.2 Source

Senior design team requirement

4.1.3 Constraints

The API shall allow intra-team changes but also allow the possibility of outside parties to change the content of the API. The changes push to the repository shall abide by the standards placed on the API and improve the overall functionality of the API.

4.1.4 Standards

The API shall abide by the industry standards for Open Source API content.

4.1.5 Priority

- Major

5 Performance Requirements

A major component of the Open source API should be performance. A major factor in performance will correlate with how much computation power, but also what kind of computer board is available for the camera development kit.

5.1 Image Compression Algorithm

5.1.1 Description

The image compression algorithm shall meet a linear time requirement or a logarithmic time requirement. This requirement is necessary to ensure the images that are sent to an end device shall interpret the information and display the images within a reasonable amount of time.

5.1.2 Source

Senior design team requirement and also image compression standards

5.1.3 Constraints

Depending on the algorithm that is used, the computing power will be a factor in each. All images gathered shall be compressed in a reasonable amount of time. Estimated at .01 seconds per image gathered. However the computer board that's used "TX1" shall provide reasonable resource computation for any image compression algorithm that's used.

5.1.4 Standards

The compression time should meet a reasonable time standard for any developer when visualizing the information.

5.1.5 Priority

- Major

5.2 Compact Computation board

5.2.1 Description

In order to handle the amount of computation retrieved from the camera, there must be a computer board that can process all the data available. The Tara development kit comes with a small computer board connected to the camera, however to ensure abundant resources are available, the TX1 board shall be used to add additional power.

5.2.2 Source

Senior design team requirement and also image compression standards

5.2.3 Constraints

"TX1" shall provide reasonable resource computation for any image compression algorithm that's used but also any instructions that may require more power then expected.

5.2.4 Standards

The board shall provide computation power for all actions and will fulfill any standards that require time (image compression)

5.2.5 Priority

- Major

6 Maintenance & Support Requirements

The open source API shall provide a solid support for any improvements or enhancements in which are pushed to the repository. Any changes, bug fixes and improvements shall be monitored on Github by the content creators but also any authorized contributors registered on the repository. Any future improvements on the repository shall be considered and reviewed. Any bugs that occur in the API during performance shall be documented and shall be submitted in repository.

6.1 GitHub Changes and Support

6.1.1 Description

All code shall be published and managed on an open source platform called GitHub. Any changes to the content in the repository shall be approved prior to making any changes. This allows all functionality to be managed by the content creators and review all changes that occur on the repository.

6.1.2 Source

Senior design requirement and also any Github restrictions

6.1.3 Constraints

Only the contributors are able to push and change anything in the master branch. Any changes that are made must be approved prior to a push to the master branch.

6.1.4 Standards

Must follow the open source library standards on Github

6.1.5 Priority

Major

7 Other Requirements

The requirements below are miscellaneous and shall be implemented alongside the major requirements. These requirements are necessary in order to obtain certain features of the product and may be included throughout the development process. Any features listed below are subject to change through the development lift cycle and throughout the lifetime on the open source platform.

7.1 Support on all platforms

7.1.1 Description

The Open API shall support development on all platforms including Microsoft Windows, Linux, and Mac Os

7.1.2 Source

Senior Design team requirement

7.1.3 Constraints

7.1.4 Standards

7.1.5 Priority

Minor

7.2 Development Language

7.2.1 Description

The language shall be in C++. The team is using this language since the Tara development kit is developed using C++ and also the OpenCV Libraries also support C++.

7.2.2 Source

Senior Design team requirement

7.2.3 Constraints

The main functionality of the application shall support C++ but also work with Node.js for the web server.

7.2.4 Standards

7.2.5 Priority

Critical

7.3 JavaScript for WebServer

7.3.1 Description

Since a Webserver shall be used to transmit and receive the data from the camera, the open source API must also support Node.js in conjunction with C++

7.3.2 Source

Senior Design team requirement

7.3.3 Constraints

Node.js functionalities shall be integrated with C++ to ensure no flow of data is interrupted.

7.3.4 Standards

7.3.5 Priority

Major

7.4 Wireless Transmission

7.4.1 Description

The point cloud data in which is gathered by the camera shall be transmitted through wifi to a designated end machine.

7.4.2 Source

Senior Design team requirement

7.4.3 Constraints

The ability to have multiple mediums to transmit data will allow different types of machines to interact with the information. Companies may move to a more wireless environment which would make this feature ideal in order to transmission data without the use of an Ethernet interface.

7.4.4 Standards

7.4.5 Priority

Minor

8 Future Items

Below are features that were considered/discussed and documented herein, but will NOT be addressed in the prototype version of the product due to constraints of budget, time, skills, technology, feasibility analysis, etc.

8.1 Robot Instruction

8.1.1 Description

With the ability to gather data from the camera, a certain robot can perform an action directed at the object located at a certain point. Ideally the open source API will only handle the object detection but functionality like this could possibly save a developer time.

8.1.2 Source

Senior Design group discussion

8.1.3 Constraints

8.1.4 Standards

8.1.5 Priority

Minor

8.2 Multi-camera Visualization

8.2.1 Description

The development taking place is based on one camera, however, having multiple cameras in sync with each other might benefit in the future. For example, in the VR industry, multiple cameras are used to display a 360 degree view for the user. This type of technology can be used on a robot in the warehouse industry which can improve object detection.

8.2.2 Source

Senior Design group discussion

8.2.3 Constraints

The cameras shall used with an Ethernet interface and connects all cameras together in a 360 degree formation.

8.2.4 Standards

8.2.5 Priority

Minor

REFERENCES

[1] Jernej Mrovlje, Damir Vrancic. *Distance measuring based on stereoscopic pictures, 2008*. `http://dsc.ijs.si/files/papers/s101%20mrovlje.pdf`

[2] Rubin, Kenneth S *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Addison-Wesley, Professional,2012

[3] e-con Systems.com *USB Stereo Camera | 3D Depth Camera(OEM), 2017*. `https://www.e-consystems.com/3D-USB-stereo-camera.asp`

[4] MVnet mvnet.fi *Image compression| Image compression, 2007*. *http://www.mvnet.fi/index.php?osio=Tutkielmat&luokka=Yliopisto&sivu=Image_compression*

[5] Pearson Pearson IT Certification *Network+ Exam Prep: Ethernet Networking Standards, 2009*. `http://www.pearsonitcertification.com/articles/article.aspx?p=1358404&seqNum=4`