MANTECH Publications

Journal of Software Engineering & Software Testing
Volume 6 Issue 3
ISSN: 2457-0516 (Online)

Generating Dynamic Indoor Environments with Virtual Reality Markups

W.A.U.Y.S. Wickramasinghe[1], P.S.R.S.De Saram[2], C.P Liyanage[3], L.N.R. Rangika[4]

Lecturer[1], Student[2, 3, 4]

Department of Information Technology

University of Moratuwa, Sri Lanka

Corresponding Authors' E-mail: *yasasw@uom.lk[1], ruwasaram@gmail.com[2], cheraniliyanage@gmail.com[3],liyanage.ruwan27@gmail.com[4]*

Abstract

This research explores the capacity of dynamically generating interactive virtual environments using Virtual Reality Mark-up Framework to build interior environments that can effectively simulate immersive interior environments. The proposed system for this study includes multiple functionalities to create virtual worlds by using Virtual Reality Mark-up Language. Users of this solution can describe the 3D environment they want to build by using natural language or inserting a sketch of the objects to include in the virtual indoor environment, increasing the useability. The main focus of allowing dynamic 3D object generation is to allow users to create virtual environments as in their imagination without using dedicated 3D authoring tools. The virtual environment's users' expectations are extracted using co reference resolution and dependency parsing from text format. On the other hand, the sketches representing 3D objects drawn by users are identified using image processing techniques such as contour approximation and edge detection. Once the virtual 3D environment is created using those inputs, the system allows the user to manipulate virtual objects in the 3D environment using hand gestures and morphing techniques. This functionality enhances the immersion of the virtual reality experience. Moreover, the system allows navigating within the environment as an avatar using voice commands in the Sinhala language. Even though there had been some research and applications developed using Virtual Reality (VR) with 3D modelling, this

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

solution is unique and much more dynamic in generating and manipulating the 3D environments than the systems currently available.

Keywords: *Virtual Reality Markup Framework (VRML), Virtual Reality (VR), Image Processing, Natural Language Processing (NLP), Voice to Text Processing*

INTRODUCTION

This project implements an interactive three-dimensional (3-D) application that can be used for modelling interior architectural environments. This incorporates the concept of Virtual Reality [1] (VR), giving the user a real-world experience. Virtual Reality is a trending technology that is gaining popularity due to the user experience it gives and its effectiveness. The project contains four major components; adding objects to the environment using the instructions given by the user in natural language (ex: - add a chair to the living room by describing the chair), adding objects by giving an image of the object to be added (ex: - add a table by scanning an image of the table), moving the objects navigating in the environment using hand gestures and using voice commands in Sinhalese.

The 3-D application will be developed using VRML [2] (Virtual Reality Markup Language), which is a language for describing three-dimensional (3-D) image sequences and possible user interactions to go with them. Using the application developed using VRML, the user can interact with the virtual environment by viewing, navigating, moving, and rotating objects within the environment. Thus, it gives the user a feeling that he is living in the environment.

The user can control the 3D environment either by using hand gestures or Sinhalese voice commands. The usage of Sinhalese voice commands will make the application much more familiar. The hand gestures will be captured using a set of sensors. The voice commands will be captured using a microphone and Natural Language Processing [3] (NLP) techniques will extract the user command.

BACKGROUND AND MOTIVATION

Virtual Reality (VR) and interactive applications are trending research and development areas in the Information Technology (IT) field. These areas are becoming popular since they give an

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

immersive real-world experience to the users. Moreover, 3-D modelling is extensively used in industries such as architecture, automobiles, and medicine due to its effectiveness. Therefore we were motivated to combine those technologies to develop an interactive 3D application that can be used for interior design.

Even though there had been research and applications developed using Virtual Reality (VR) and 3D modelling, the concept of user interaction with the 3-D virtual environment is not implemented in the interior design field. Also, we were motivated to use Sinhalese voice commands since they will be more familiar for the users and because Sinhalese voice commands have been rarely used in interactive applications.

Our application will enable the users to design the interior of rooms by moving around objects such as tables and chairs using hand gestures and voice commands. Furthermore, the user can move around the room using avatars and see how it looks. Due to the 3-D environment and the ability to control the objects using hand gestures and voice commands, the application will be more effective and familiar to the users than a 2-D application.

PROBLEM IN BRIEF

Interactive virtual environment modelling with fundamental life user interactions is still not a very well developed area, and there are no practical applications from this area for the usage of the general public. The occasions like interior designing, indoor household item arrangements, simulation of architectural design are done with 3D drawings or using a graphical authoring tool like Photoshop. Nevertheless, it is just a primary way of doing it since there is no opportunity to expose those lively to customers and get a better requirement gathering.

Especially in interior designing and arrangements, customers love to see it and then give feedback to the design rather than listening to a description or looking at a drawing. When discussing things with a designer, they might face some difficulties explaining things and understanding things.

AIM AND OBJECTIVES OF THE RESEARCH

This research aims to develop a framework to enhance the data gathering, designing, navigating and visualizing interior designing using cutting edge virtual reality technologies like VRML (virtual reality mark-up language) incorporated with Natural Language Processing, Image

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

Processing and Voice Recognition. Ultimately users of this system will be able to experience virtual interior designing, modify them or rearrange the objects inside the space with hand gestures, indoor navigation and experience a virtual experience [4]. The following objectives have been identified currently to be included in the research.

- To generate an indoor environment and objects inside it from natural language inputs from the user.
- To allow actual user gestures to manipulate VRML objects in the virtual environment. Some of those manipulations are dragging objects here and there, pulling and pushing, and releasing objects. Moreover, the objects will support effects like collisions.
- To allow navigation inside the virtual environment as an avatar with voice commands from the user. Users can command the Sinhala Language. Avatar will support rotations, forward and backward navigation, turning to left and right, fling effects and jump.

  To evaluate the research with actual users and get feedback
- To fine-tune the system and allow practical usages.

PREVIOUS RELATED WORK

The proposed solution is to develop a Virtual Reality Mark-up Framework for Generating Interactive Indoor Environments. We proposed a solution based on Virtual Reality Mark-up Language (VRML).This implementation creates an indoor environment virtually and lets users explore it interactively. One of the novel features in this research is to investigate the possibility of generating virtual 3D objects by processing natural language gathered from the user. Users can input a sketch of objects as an image to the system and generate a 3D view of the sketch. VRML models will be articulated based on those analyses and generated in the environment.

This research is comprised of Hand Gesture Recognition with Image Computer Vision technologies like Real-Time Web Camera Video Feed Analysis. Users can use their hand gestures to manipulate VRML objects in the space. Touching and grabbing objects, releasing them, pulling and pushing objects based on hand gestures will let users arrange the indoor environment interactively. This will bridge the gap between the virtual and natural worlds and deliver a novel user experience. The Sinhala voice recognition module lets users navigate the virtual indoor

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

environment using voice commands. Those voice commands will let an avatar turn around, flying and a lot more other actions. This Sinhala voice recognition component will be implemented and trained to have higher accuracy, and it will support many voice commands and visualize them in the system.

Generally, virtual reality (VR) systems require 3D computer graphic models that can be interactively experienced by participants [1]. Well-developed immersive VR systems will provide a three-dimensional environment where users can directly perceive and interact with objects. The underlying factor that motivates most VR research leads to a more natural and effective human-computer interface [2].

In a virtual environment, there are three dimensions, and a user can interact with the objects in the virtual environment. Hand gestures can be used to interact with virtual environments. The first way is generating an event or command for the interaction. The other way is controlling objects directly. Several systems use hand gesture recognition to control objects and for generating commands. Some systems use 3D menus and cursors for generating commands in a 3D environment. Several methods have been mentioned in the literature used to create virtual worlds and interact with them. The mentioned below are some of the researches conducted.

Various sensing methods are used to get gestures for pattern recognition [4] to identify hand gestures. The available sensing methods in the previous studies can be classified into three classes. They are movement-based, electromyogram-based (EMG-based), and vision-based approaches [4]. The movement-based technique uses various sensors to measure movement. Glove-based is the mainly used movement-based approach, and it gives a good performance, especially in sign language recognition. In addition, data gloves like the Cyber Glove are used to track different hand gestures [6, 7].

Vision-based methods can efficiently detect, track, and recognize hand gestures [4]. These methods are sensitive scale, rotation, background texture, colour, and lighting conditions. Vision-based hand gesture recognition can be implemented and used for human-computer interaction effectively, particularly for applications in 3D Virtual Environments [9]. Video cameras can be used as the input device. Using the input of the video camera, tracking and detection of hand gestures can be done. Most of the existing vision-

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

based hand gesture recognition systems use a colour glove or a marker. It makes hand gesture tracking easy. [10] Another essential part of this project is generating virtual environments. Different methods have been used and mentioned in the literature for creating and interacting with virtual worlds using VRML.

Virtual Reality Markup Language (VRML) is a textual language that defines objects and behaviours in three-dimensional worlds. WRML is mainly used to enable 3D views for users on the Internet using different VRML plugins. As with languages such as HTML or XML, VRML is expressed verbatim. Although there are a variety of tools that allow the graphic definition, interactive 3D objects that can then be exported as a VRML text file, behaviour definitions must be specified textually, usually through short programs written in a language of Script such as JavaScript linked to 3D objects using the instructions of the VRML files.

Shapour Arjomandy Trevor J. Smedley Dalhousie [5] has proposed a model such that behaviour or animation in VRML is visually attached to objects in a 3D scene. Their efforts are directed towards exploring how the routing of events between the different parts of a VRML program can be described visually. They have suggested another way of using textual VRML coding for creating behaviours in 3D worlds. Messages are passed between nodes for creating behaviours.

1. Generating VRML using text descriptions from users

Natural Language Processing (NLP) techniques extract information from text written in natural languages. Extracting information from natural language text has become challenging due to its highly informal and ambiguous nature and different forms of writing. The following NLP techniques are commonly used when extracting information.

Part of Speech (POS) tagging

In this technique, a text corpus is used to train the tagger to assign the part of speech to each word in a sentence. The part of speech is assigned based on the relationship with adjacent words in the sentence. Therefore, the tag assigned to a word can change based on its context. In other words, the same word can be assigned different tags in different sentences. Part of the time, speech taggers are implemented using the Hidden Markov Model (HMM).

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

Tag	Description
CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential there
FW	Foreign word
IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal
NN	Noun, singular or mass
NNS	Noun, plural
NNP	Proper noun, singular
NNPS	Proper noun, plural
PDT	Predeterminer
POS	Possessive ending
PRP	Personal pronoun

Tag	Description
PRP$	Possessive pronoun
RB	Adverb
RBR	Adverb, comparative
RBS	Adverb, superlative
RP	Particle
SYM	Symbol
TO	to
UH	Interjection
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non3rd person singular present
VBZ	Verb, 3rd person singular present
WDT	Whdeterminer
WP	Whpronoun
WP$	Possessive whpronoun
WRB	Whadverb

Figure 1 Set of POS Tags

Hidden Markov Model

This is a supervised machine learning classifier. HMM is used to assign the POS tags for words in a given sentence or a paragraph by avoiding ambiguities. This model is implemented by considering the probability of a particular word belonging to a specific part of speech.

Anaphora and Cataphora Resolution

Anaphora resolution consists of two parts; anaphor, the noun that refers to a particular entity and the antecedent, the entity referred to. For example, Saman is not home yet, but he will come soon". Here Saman is the antecedent, and he is the anaphor. Cataphora resolution is also similar to this. The difference is that the entity referred to is mentioned at the end of the description.

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

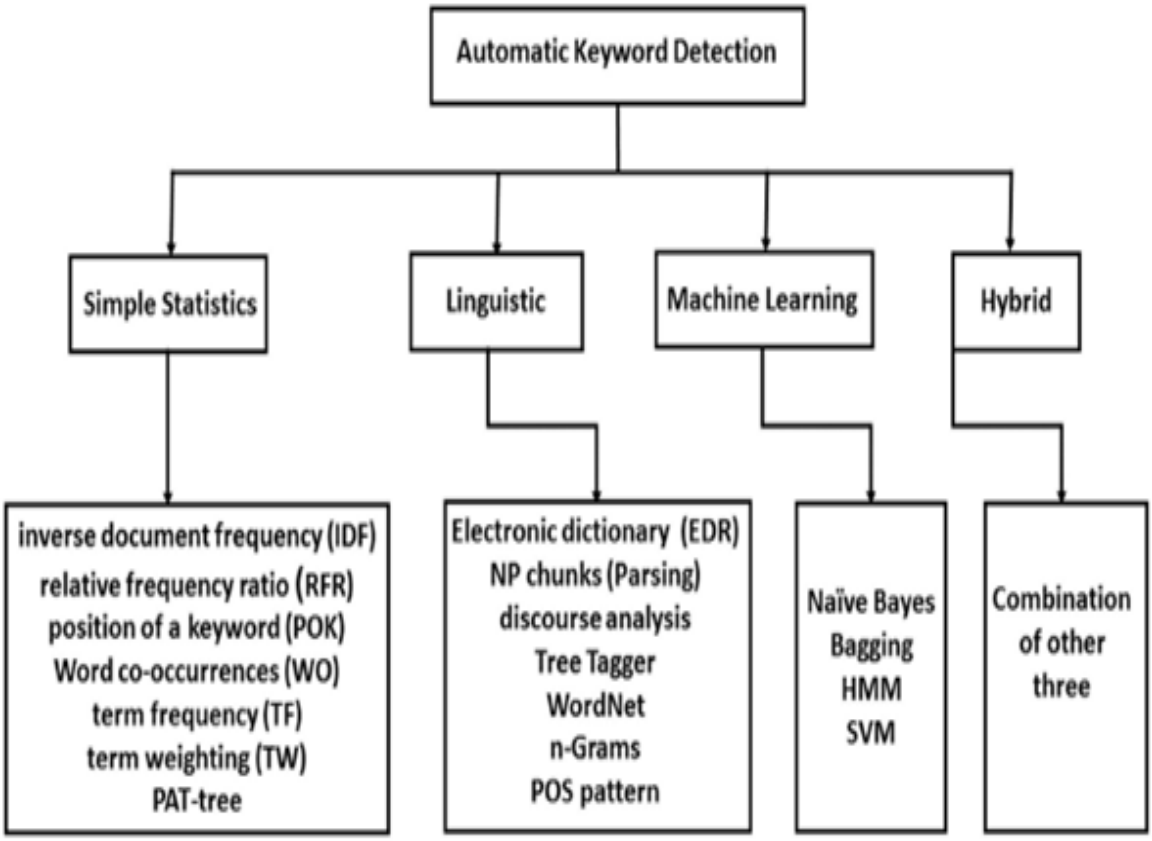


Figure 2 Keyword extraction with Natural Language Processing

NLP approaches have been used in the graphic designing field to generate scenes according to user descriptions. Words Eye automatic text-to-scene system by Coyne and Sproat [20] is one such system. This system consists of an object database that stores the essential elements of a 3D scene. The database currently stores around 2000 3D objects, and it allows the users to add new 3D objects to the database.

When the user enters a description, firstly, the description is broken down into sentences and tagged. After that, the sentences are parsed and converted into a dependency structure. Then the dependency structure is converted to a semantic representation using semantic interpretations.

Finally, these are converted to low-level 3D objects in the graphical environment. The depicters resent the objects in the scandal and heir attributes such as colour, posture, and location. Examples of a semantic representation and a depicter are given below.

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

```
(("node2" (:ENTITY :3D-OBJECTS ("mr_happy")
           :LEXICAL-SOURCE "John" :SOURCE SELF))
 ("node1" (:ACTION "say" :SUBJECT "node2"
           :DIRECT-OBJECT  ("node5" "node4" "node7")...))
 ("node5" (:ENTITY :3D-OBJECTS  ("cat-vp2842")))
 ("node4" (:STATIVE-RELATION "on" :FIGURE "node5"
           :GROUND "node7"))
 ("node7" (:ENTITY :3D-OBJECTS
           ("table-vp14364" "nightstand-vp21374"
            "table-vp4098" "pool_table-vp8359" ...))))
```

Figure 3 Semantic representations of objects

1. Entity: *cowboy*
2. Entity: *bicycle*
3. Entity: *store*
4. Attribute:
   Subject: *<element 2>*
   Property: *red*

Figure 4 Depicter used to represent objects

This system allows the user to generate 3D scenes without using sophisticated graphic packages. The main drawback of this system is that it stores a large amount of 3D objects (around 2000) in a database, requiring a large amount of storage space. Also, the object needs to be stored in the database to use it in the scene. It does not allow the users to create custom objects by giving instructions.

There have been several types of research for automatically identifying objects and their attributes from descriptions given by users using NLP in Object-Oriented Analysis and Design (OOAD). Identifying objects and their attributes act as the initial step in generating UML diagrams in OOAD.

Abbot[21] suggested a simple approach where nouns in the description are considered objects or classes and verbs indicate their behaviours. This is the most basic approach in identifying objects and their properties in a description using NLP.

Cockburn [22], in his work, suggested that adjectives are used to describe properties of objects or entities, and they can be used as the attributes of the objects identified using nouns.

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

However, these techniques rely on manual object verification rather than fully automatic extraction of objects from a text description. Some NLP based tools have been developed for the automatic extraction of objects. One of such NLP systems is LOLITA, proposed by Mich [23]. It has built on a large scale Semantic Network (SN). In this system also every noun is identified as an object, and the relationships among the objects are identified using links. The drawback of this system is that this does not differentiate between classes, attributes, and objects like any other SN system. Delisle et al.[24] in their project, DIPETT-HAIKU suggested an approach to identify the subjects, objects and verbs in a sentence using the Subject-Verb-Object (SVO) structure in a sentence and thus identifying objects in a description. Harmain and Gaizauskas[25] has developed an NLP based CASE tool, CM-Builder, which can be used to create a class model using text descriptions.

Elbendak[26] has suggested a framework for object identification using NLP techniques. In this approach, Use-Case Descriptions (UCDs) obtained from the users are taken as the input to the framework and the descriptions are parsed using Memory-Based Shallow Parser (MBSP), which creates noun phrased and verb phrases. Parsed Use Case Descriptions (PUCDs) are generated based on the output from the parsing tool. PUCD is used to identify nouns, verbs, adjectives and adverbs from use case descriptions to identify objects/classes and relationships in the description using a set of rules. The output is refined manually by an expert.

The process is shown on next page (See Figure: 5)

In most proposed approaches, the POS tagging technique tags the words in a sentence as nouns, verbs, and adjectives. POS tagging tools available can provide a high level of accuracy for grammatically correct descriptions. However, it will give inaccurate results when grammatically inaccurate descriptions are given. The descriptions given by users are not always grammatically correct, and thus this aspect should also be considered when extracting the objects from a user given description. Nanomiya and Mozgovoy [27] have suggested improving POS tagging for grammatically incorrect phrases. Most of the POS taggers use a machine learning approach. It is trained using a POS tagged corpus consisting of grammatically correct sentences. Their research has selected two POS taggers; the SVM tool and GCTag.

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

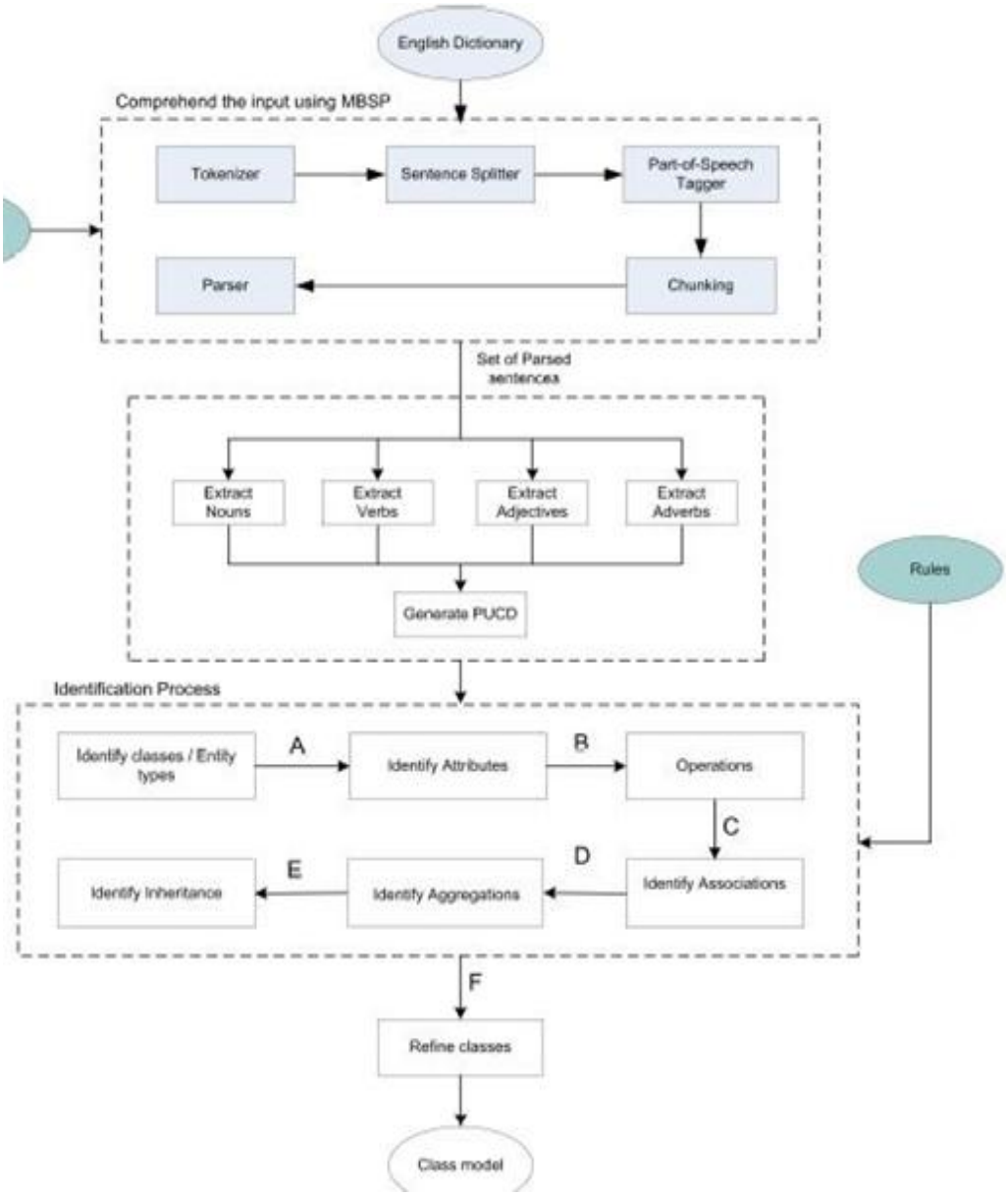


Figure 5 Process of PUDC framework

They have used corpora with ungrammatical sentences to train the POS tagger. For that, they have introduced three types of errors;

1. Wrong verb type
2. Wrong singular/ plural form
3. Missing article

In this approach, only these three types of errors are handled, and the tagger will not handle other types of grammatical errors. According to the results, when ungrammatical sentences are used to train the POS tagger, the tagger will give more accurate results for grammatically incorrect inputs. However, using a corpus

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

with grammatical errors will reduce the accuracy of grammatically correct inputs. The experimental results obtained are as follows;

			Errors in the test set	
			0	1, 2, 3
Errors in the training set	0	SVMTool	94.56	91.28
		GCTag	91.11	86.74
	1, 2, 3	SVMTool	82.95	94.06
		GCTag	79.39	89.99

Figure 6: Experimental results of training POS Tagger using ungrammatical corpus

2. VRML object creation using image processing

The creation of VRML objects is one of the significant parts of the project. In VRML, objects are defined from geometric objects such as cubes, cones, cylinders, and spheres.

Identifying shapes in a sketch drawn by the user is an important task. In this project, the user is given a chance to draw sketches of the objects such as chairs and tables. They can either draw the top view or the side view.

If they draw the top view, they are asked to use basic shapes such as circles, squares and rectangles. If the side view is drawn, the pre-defined symbol will represent an object. Shape identification and object identification is an essential part of the project.

Different methods have been used in researches to detect shapes and objects. In addition, several edge detection techniques also have been used.

Sobel edge detection

The Sobel operator is applied to the image to detect the edges in this method.

The vertical mask of the Sobel operator is

-1	0	1
-2	0	2
-1	0	2

The horizontal mask of the Sobel operator is

-1	-2	-1
0	0	0
1	2	1

Figure 7: Sobel Operator

The two masks can be applied separately to the image, and Gx and Gy can be calculated. Then the magnitude can be calculated using,

$$G=|Gx| + |Gy|$$

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

If G is more significant than a threshold value, that particular pixel can be identified as an edge. The threshold value should be found experimentally.

Canny Edge Detection

This method contains four significant steps. First, this is a second derivative based technique. So can locate the edge pixel using the zero-crossing property.

First, the noise in the image is removed using the Gaussian function.

Then the magnitude of the gradient and the direction of the gradient is found using a suitable mask. Successive difference operators also can be used for this purpose. Thin edges can be found using non-maxima suppression. Double threes holding is used to find the other edges.

Researchers have proposed various techniques to detect shapes. For example, some use generalised Hough transform [12] and template matching [13]. However, both mentioned methods are sensitive to noise and sampling artefacts. To overcome this problem, M. Kass proposed active contour models [14]. However, this method is more complex and requires higher computational time.

Following are some other techniques used for shape detection mentioned in the literature.

Object Shape Recognition in Image for Machine Vision Application

Mohd Firdaus Zakaria, Hoo Seng Choon, and Shahrel Azmin Suandi [11] have proposed identifying different shapes such as triangles, squares, and circles. The proposed algorithm considers the intensity value of the input image. Then the binary image is obtained by thresholding the input value by Otsu's meth finally, and noise is eliminated in the image using median filter and edges are detected using the Sobel operator.

In order to reduce the false detection thinning method is used. This method removes the unwanted edge pixels. Otherwise, those will be counted in the parameter estimation algorithm. The compactness of the region decides the shapes. This method has achieved 85% accuracy, according to the experimental results. As described above, the algorithm used is a simple algorithm with low computational time.

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

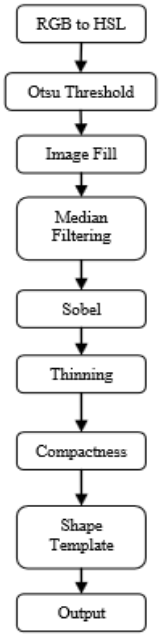


Figure 8: Algorithm for automatic object shape recognition in machine vision

Nidhal El Abbadi and Lamis Al Saadi [15] introduced a new approach for identifying 2D shapes in an image, and also, the algorithm recognizes the type of the shape.

The algorithm recognizes the known shapes by segmenting images into regions corresponding to individual objects and then calculating the shape factor.

The algorithm is tested with many images with different shapes, identifying all. The diagram below shows the steps of the proposed algorithm.

The proposed algorithm detects the shapes in the image. The image can be either coloured or non-coloured.

This algorithm works with gray scale images. So first, an image is needed to be converted into a gray scale image. Contrast-limited adaptive histogram equalization (CLAHE) is applied for contrast enhancement of the image. Usually, CLAHE is applied to small regions in the image. The histogram equalization is used to enhance the contrast of small regions. "CLAHE" technique should be applied to a gray scale image.

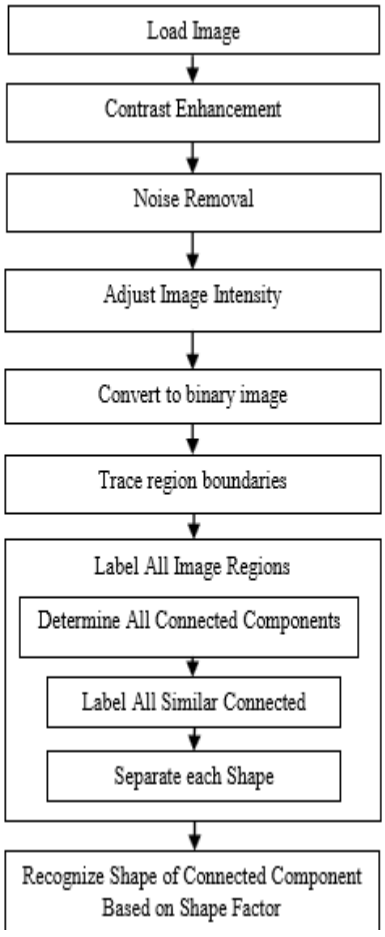


Figure 9: Algorithm for Automatic Detection and Recognize Different Shapes in an Image

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

Table1: Shape factors of different shapes

Shape Type	Shape Factor (SF)
Circle	0.7 < SF < 0.8
Square	0.484 < SF < 0.55
Rectangle	0.2 < SF< 0.3
Triangle	0.44 < SF < 0.483
Oval	0.32 < SF < 0.34
Diamond	0.36 < SF < 0.38

Noise will be there in any image. In this method, a Gaussian filter is used for noise reduction of the image which the contrast is enhanced. The contrast of the output image resulted is also enhanced. The image is converted into a binary image after enhancing. For this process, a threshold value is used.

Pixels with a value greater than the threshold are assigned one, and pixels with a value less than or equal to the threshold are assigned 0.The algorithm detects the exterior boundaries of the objects. Finally, each shape will be recognized and labelled by calculating the shape factor. Shape factors for particular shapes are mentioned in the table above.

Singh et al. 2012. [16] Has proposed a novel approach to detect shapes from tactile images by touching the surface of different real-life objects. In this method, four kinds of objects are considered. They are planar surfaces, object having one edge, cubical objects (objects having two edges) and cylindrical objects. These are used to recognize shapes. Multilevel thresholding has been used to segment the surface edges denoted by the high-pressure regions. High-pressure regions are unique to various object classes. In order to describe the unique high-pressure regions, regional descriptors are used. The regional descriptors are used as the features required for classification. Object shapes have been classified using the Linear Support Vector Machine classifier. This classifier gives 92% accurate results in

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

noise-free environments. The classifier's accuracy in noisy environments has been tested using SNR (signal-to-noise) ratios.

Schindler and Suter (2008) [17] have proposed a method for detecting object classes in images based on global shapes. In this method, a distance measure is derived for elastic shape matching, invariant to rotation and scale, and robust against non-parametric deformations. First, the image is over-segmented. Then the segments are explored to find potential boundaries of objects that are more similar to the shape template of the object class to be identified. Experimental evaluations have been conducted to evaluate the performance and accuracy of this method. This method has achieved a detection rate of 83%-91% at 0.2 false positives per image on three challenging data sets.

There are some projects mentioned in the literature which generates VRML views. They are mentioned below.

Nowadays, with the development of the web, people can view 3D virtual worlds and explore virtual space. Creating and viewing the VRML is relatively easy and possible in most browsers. There are only two ways of creating 3D virtual scenes.

The first way is coding the scene directly using VRML. The other uses existing CAD and modelling software and saves the created world in VRML format or converts it to VRML from another format. Both of the mentioned methods are time-consuming.

Igarashi et al. [14] used a pen interface to drive browsing in a 3D world by projecting the user's marks on the ground plane in the virtual world.

"Sketch that Scene for Me" by Ellen Yi-Luen [18] is a technique that generates a VRML world for freehand drawings by improving the idea of Igarashi. It analyzes what the user draws and gives the relevant 3D view.

3. Manipulation of Virtual Objects with Real-Time Gesture Tracking

The VRML framework can manipulate virtual objects that are added by processing user entered natural language text or image processing with the sketch fetched into the system. However, this concept has been used in very few published types of research. Under this section, those approaches will be analyzed and evaluated according to the similarities and differences with this module and their pros and cons.

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

Hand gesture tracking can be implemented mainly using three types of techniques.

1. Movement Based sensing
2. Electromyogram based (EMG)sensing
3. Vision-based sensing

The movement-based method is widespread in research that requires a data-glove-like device for sensing gestures. The advantage of following this method is the accuracy of the input data and real-time movement tracking, but this costs a lot compared to the vision-based technique.

Therefore, accelerometers are commonly used in these data gloves, which as available as commercial products.

The electromyo gram-based approach is easier to use than the movement-based approach, but it is still lagging in accuracy levels and is not yet feasible to use in industrial environments or commercial applications.

The vision-based approach is practical and less costly than the other two techniques. The human-computer interactions are captured using a digital camera, and the gestures are identified using image/video processing middleware.

A review of other works consists of two somewhat similar types of research and related to our solution. First, the following research is done under Movement-based gesture tracking technology.

Research on Movement Based Hand Gesture Recognition

Previous research was there, which is more focused on loading 3D architectural models, saving attributes of it and manipulation, translation and rotation manipulation and scaling objects with gesture commands. [4] This approach is similar up to some extent with this component.

They also have used hand gestures as input to the system in a virtual environment. However, they use a hand button mounted input method to get additional support for tracking the gestures and user inputs.

According to the inputs with the button and gestures, they have been processed with different system states. Table 2 shows the states which can be identified with this button mounted hand gesture tracking approach.

“ON BUTTON” refers to the collision of a finger with a finger button. Fingers from thumb fingers were numbered from 1 to 5.

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

Table2: States and functions according to the button behaviour

ON BUTTON	Interaction mode	Defined function description	Related state
B1	Select	release contacted object if selected previously	RELEASE
B2	Select	select touched object	SELECT
	Manipulation	change translation according to hand cursor position change	TRANSFORM
B1B2	Manipulation	change rotation according to hand cursor orientation change	ROTATE
	Navigation	change viewpoint to the collision detected position	GOTO
B2	Select	Activate selected object	ACTIVATE
B1B2B3	Manipulation	change rotation according to hand cursor movement related to object center	OBJECT ROTATE
B1B2B3B4	Manipulation	change translation and rotation according to hand cursor movement	MANIPULATE

When a finger collides with the finger-button area, the focused object state changes to "Manipulation" mode. This approach is extendable to select objects in a virtual environment, similar to the ray-casting approach [5].

Even though this research approach was focused on different state transitions with several pre-defined gesture models, this requires several special devices. One is Cyber Glove™ which is a commercial portable device for capturing hand gesture inputs. It is an electronic hand glove capable of detecting hand centric movements in real-time as a data feed.

This device lets smoothness and accuracy of the system, but it limits the natural hand gesture inputs and interactivity.

Moreover, another device is required for their research which is named Polhemus Fastrak™, which is a 6-axis compatible tracking device, which is very expensive and needs extra technical knowledge to set up and train the device as well. It acts as a 3D-Digitizer and motion tracker with the data glove mentioned above.



Figure 10: Polhemus Fastrak™ and Cyber Glove™ devices in others work approach

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

In our research, the hand gesture tracking component is not based on any electronic hand glove but rather a natural hand with the one-colourized usual glove for accurately tracking hand gestures. There is no electronic link with hand. After processing the video feed, all the inputs will be generated, captured with a web camera in real-time.

There is no need for a high-end full HD web camera as well. Our image and video processing implementations are smart enough to produce better inputs to the VRML framework with considerable accuracy.

The accuracy rate of identifying hand gestures is currently at approximately 78%, and this can be further increased with modifications to the image processing algorithm.

In the other's research approach, they are using "Virtual Research V6 HMD as a display unit to render the virtual reality model with Performer™, a graphics framework with 3D modelling libraries. Our research is currently capable of rendering models in any computer display using VRML enabled web browser.

Research on Vision-Based Hand Gesture Recognition

This way of identifying hand gestures is highly subjected to lighting conditions of the environment, camera sensor sensitivity and background colours. Using a video camera, hand gestures can be tracked in human-computer interaction modes for the virtual environment. There was research done under this technique [6].

Their approach is vision-based real-time gesture tracking based on four-hand templates loaded into the system, as shown in figure 10:

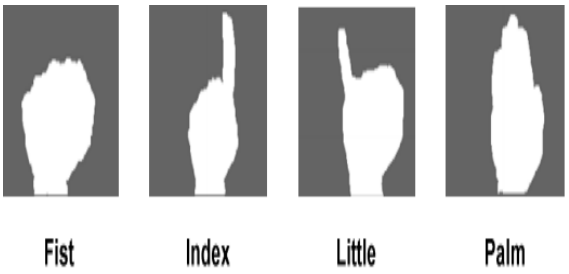


Figure 10: Hand model templates

Each frame that is captured by the video camera is compared with these models, and it tries to recognize the gestures under the HSV colour model. They have successfully eliminated accuracy inconsistency with this approach. However, still, there is a tolerance to errors due to occlusions and lighting conditions.

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

In our solution, we follow the RGB colour model and always detect the shape of the hand in each frame to detect the index and middle finger. Moreover, we are using Open CV real-time processing to deliver faster results than analyzing each video frame with pre-defined templates. So, it guarantees efficiency and low processing power. Further details about our approach will be discussed in the next chapters.

4. Voice Recognition for manoeuvring avatar in Virtual Environment with Voice Commands

Automatic Speech Recognition (ASR) is a popular field of research where different language models and applications are involved. There are many ASR-related types of research out, but there is no Sinhala Voice Recognition application targeting VRML avatar manoeuvring.

However, according to research [28] done for Sinhala, voice recognition has been used recently but not directly with the VRML markup framework. Moreover, another related to others' research is using the English language for voice commanding an industrial VRML environment[29].In their approach, they have used a context-sensitive speech-driven interface, but in our approach, we focus on the specific command set programmed to execute VRML avatar actions like Run, Stop, Turn Left and Turn Right, Speedup and Slow down.

In their research, they have invested lot of research effort for noise cancellation and enhancing accuracy level due to the application of the research. Since they are focusing on Industry Level usage, it is essential to have those implementations. However, it is not very much essential in our scope since we are experimenting with this framework in the classroom and laboratory environment. In this approach, they are manipulating 3D objects with voice commands, but in our approach, we need to command VRML avatar, but not the virtual objects in the virtual environment. In our implementation, avatar navigation attributes are shown in figure 12.

It is still a novel approach to implement Sinhala voice commanding to manoeuvre VRML avatar in a virtual environment, and there are significantly fewer research approaches have been carried out.

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

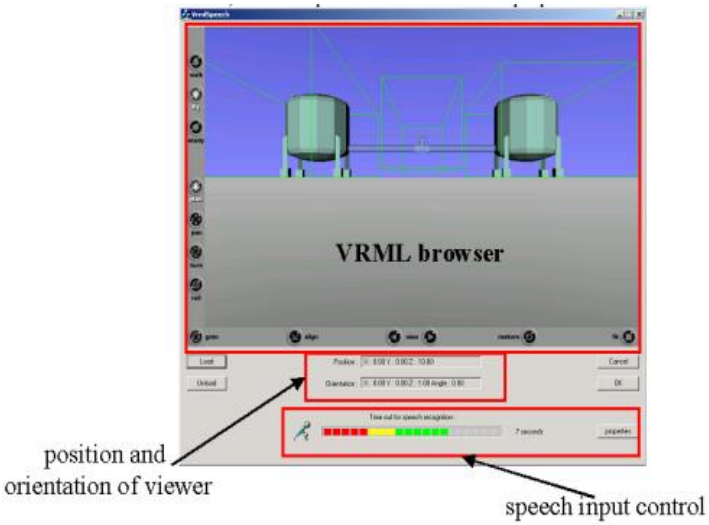


Figure 11: Controlling Industrial 3Denvironment with voice commands

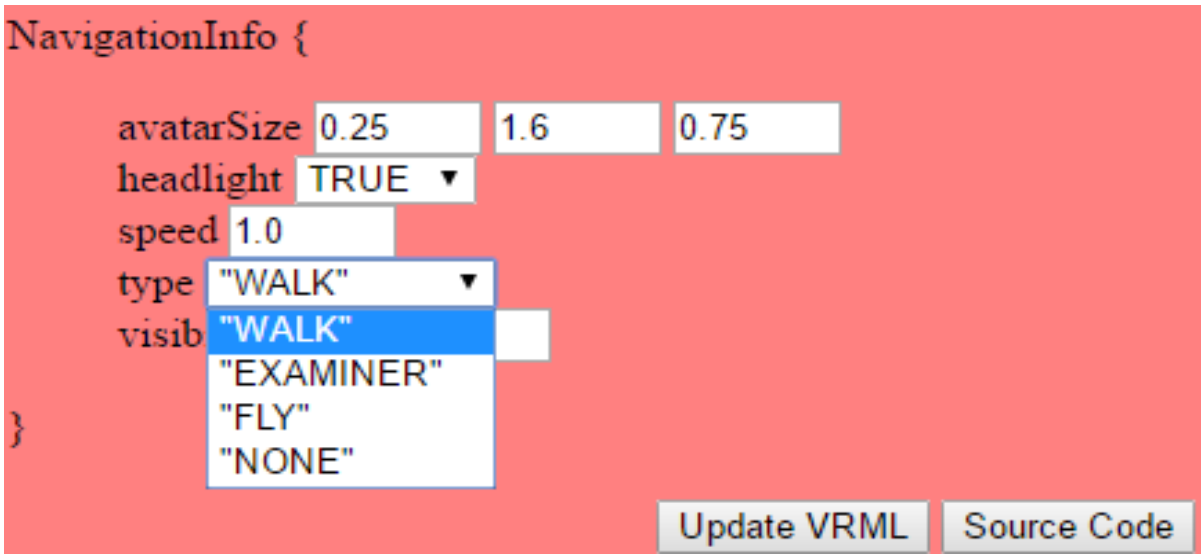


Figure 12: Few Navigation types of VRML Avatar

METHODOLOGY

This chapter explains the progress up to now and future approaches that will be engaged in to achieve the research targets and analysis requirements and the design which is followed to implement the research solution.

All the involved technological and conceptual aspects of project approaches can be described as the following four components.

1. VRML object generation using text descriptions-This module takes descriptions from the user as the input and generates the VRML environment consisting of 3D objects accordingly

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

2. VRML object generation for a sketch drawn by the user-This module takes the sketches created by the user from a drawing tool such as paint, identify the objects in the sketch and create the VRML environment accordingly.
3. Manipulation of virtual objects using real-time gesture tracking- This module can take the VRML environments created using the above two modules or another VRML environment as the input. It will identify hand gestures and allow the users to move the objects in the virtual environment using hand gestures
4. Voice recognition for manoeuvring avatar in a virtual environment with Sinhala voice Commands –This module can also take the VRML environments created using the above two modules or another VRML environment as the input. This function explores the virtual environment by moving an Avatar using Sinhalese voice commands.

Even though there has been previous work carried out in each of the above research areas separately, no adequate system has been developed by combining all the above areas in creating an interactive environment. In that sense, this system will give more interactivity and user-friendliness to the users.

The functionality of each module will be described in detail in the following sections.

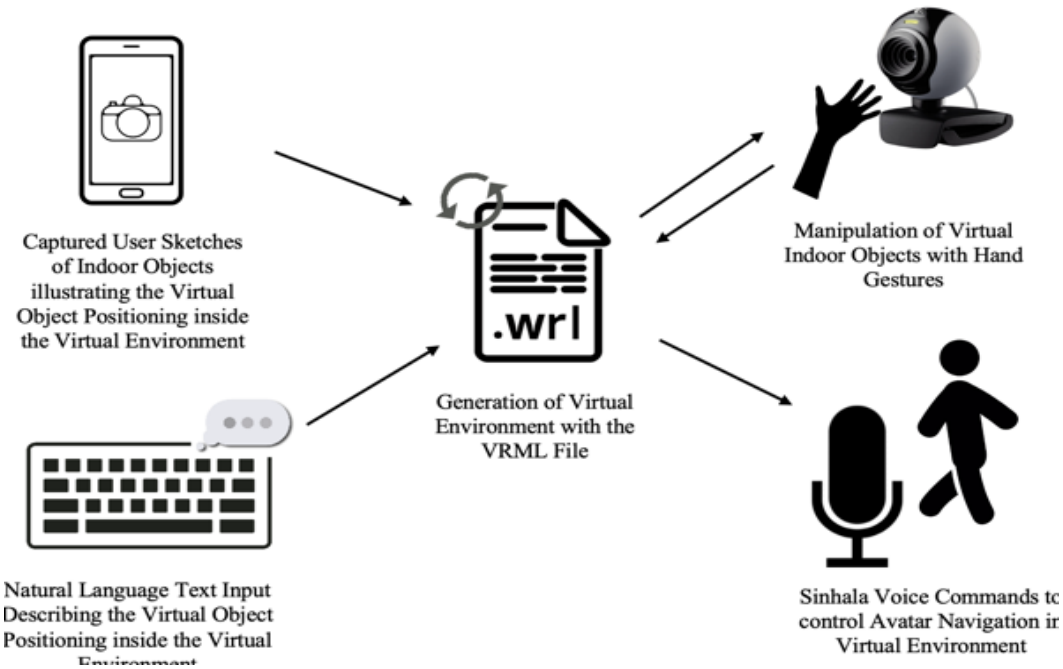


Figure 13: VRML markup framework system inputs

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

Figure 13 shows how these system inputs integrate into the VRML framework where all the processing happens and output the Virtual Environment with 3D models in it. Different modules in the system process these inputs accordingly and give output to the central VRML render enabled browser. The following subsections describe the approach and design aspects of each module.

1. Generating the VRML objects using text descriptions

The objective of this module is to create a VRML code to generate the 3D objects described by the users in the English language. In this chapter, we will explain the technologies and methods adopted by us in order to accurately extract the objects and their attributes described by the users. Figure 14 depicts the basic flow of this module.

As the initial step, we will identify the nouns as objects and the adjectives prior to a particular noun as the attributes describing that noun. The objects identified can be basic shapes such as a sphere, cylinder, and cube or they can be predefined objects such as table, chair and bed. The attributes identified can be colour, size, and the object's relative location in the 3D environment.

As we proceed with the project, we will implement anaphora and cataphora resolution and add tags to grammatically incorrect sentences as described in the previous chapter.

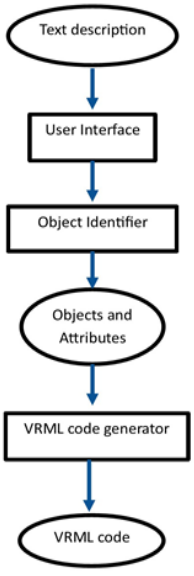


Figure 14: Process of generating VRML using a text description

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

The objects identified (nouns) and their attributes (adjectives) will be stored in a map, and these data will be passed along with their attributes to the VRML generator sub-module. In this sub-module, all the attributes of each object will be analysed, and this will identify which adjective represent which property of the object, such as the colour, size and location of the object.

The object name, colour, size and location will be passed as parameters to the method that will generate the VRML code. This method will generate the VRML code as a text file by adding the VRML code segments needed to generate the objects passed as parameters to this method.

2. Generating the VRML objects for a sketch drawn by the user

The steps of the process are mentioned below.

- Loading the image
- Resizing it
- Getting the greyscale image
- Smoothing it to reduce noise
- Thresholding the image
- Revealing shapes by identifying contours
- Generating the VRML file

Users can draw a sketch using paint and give it input to the system. Either the user can draw the top view or the side view. If auser is drawing the top view he can use the basic shapes. The basic shapes are assigned to objects. (Rectangle for a table, square for the chair, circle for a round table). The program will identify the shapes. If the user is drawing the side view, specific shapes are defined for the objects such as chairs, tables. The system will detect the object by analyzing the shape of the edges. After analyzing the object, the relevant VRML view will be generated.

VRML (Virtual Reality Mark-up Language) can construct 3D virtual worlds. It constructs objects using basic geometric objects such as cubes, cylinders, spheres, cones.

The followings tasks need to be completed in the future

- When there are multiple objects, the program should arrange them according to the coordinates.
- Try to generate the VRML object by analyzing the shapes in the sketch drawn without identifying the object at once.

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

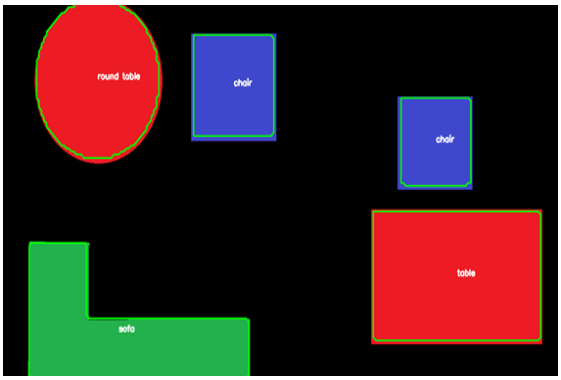


Figure 15: Detecting multiple shapes from user input

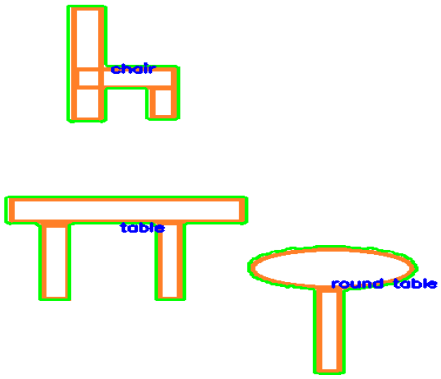


Figure 16: Objects extracted from the image

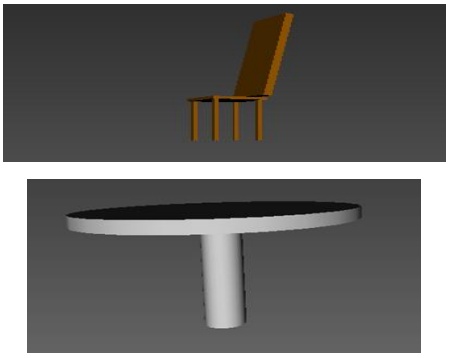


Figure 17: VRML models

3. Manipulation of Virtual Objects with Real-Time Gesture Tracking

The primary purpose of this component is to deliver a user experience of manipulating 3D objects in a Virtual Environment (VE) with real-time hand gestures. When the house's architectural design is generated, users will be able to move some objects inside the building. For that, there are a need of having "Grabbing" objects, "Moving", and "Releasing" objects. By considering these states, we propose some object states naming "Normal User Mode", "Selection Mode", "Object Manipulation Mode", as well as "Avatar Navigation Mode".

First, users need to tune their camera feed by reducing the capturing noise by adjusting filters that we are proving under our design. We are presenting GUI controls for this. They will be able to sharpen video frames and lower the noise levels. If there is too much noise, a text label will be displayed to guide the user. Figure 18 shows the alert which is displayed for high noise levels.



Figure 18: Displaying alerts to user if too much noise

If there is high noise, we are implementing several visual modes for projecting camera

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

view to screen with contour tracking techniques to detect the boundary of objects, ideally from a colourful background. Figure 19 shows the contour tracing mode. By looking at the visual, users will adjust dilute and blur settings for a sharpened boundary.



Figure 19: Contour tracing for identifying hand objects from the camera feed



Figure 20: Contour tracing with noise filter applied

Figure 21 is the visual mode, a blend of contour tracing and noise filter applied with a green filter for a realistic view. Users will get more opportunities for tuning the image for higher accuracy



Figure 21: Aligning contour tracing and morphing in actual camera feed

We are already identifying hand gestures, and it is available for manoeuvring pc controls like a mouse pointer. Our approach is to build an interface between these hand detection modules with the VRML Virtual Environment in the near future.

4. Voice Recognition for manoeuvring avatar in Virtual Environment with Voice Commands

In our approach, one of the objectives we are targeting is to let users navigate the virtual environment controlled by voice commands. Here, an avatar is used as a character controller, and the Sinhala language is used to input voice commands into the system.

For this, we used a voice recognition library called “CMU Sphinx”, which is free and open-source under the BSD license. The Sinhala voice phrases will be captured using a microphone and those clips are segmented into chunks by We are using the following key terms for our indoor navigation control for different actions.

- Run
- Stop
- Speedup
- Slow
- Turn Left
- Turn Right

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

According to those voice commands, the avatar behaviours vary. For that, we are in the progress of implementing an interface for VRML core functionalities.

DESIGN

This chapter describes our analysis requirements and the design which we are following to implement the research solution. The VRML Markup framework for architectural interior design has system inputs as follows.

- User typed natural language texts describing the interior items
- User sketched image of interior items
- Video camera input feed
- Human voice commands in the Sinhala language

Figure 22 shows how these system inputs integrate into the VRML framework where All the processing happens and output the Virtual Environment with 3D models.

Different modules in the system process these inputs accordingly and give output to the central VRML render enabled browser. The following subsections describe the design aspects of each module.

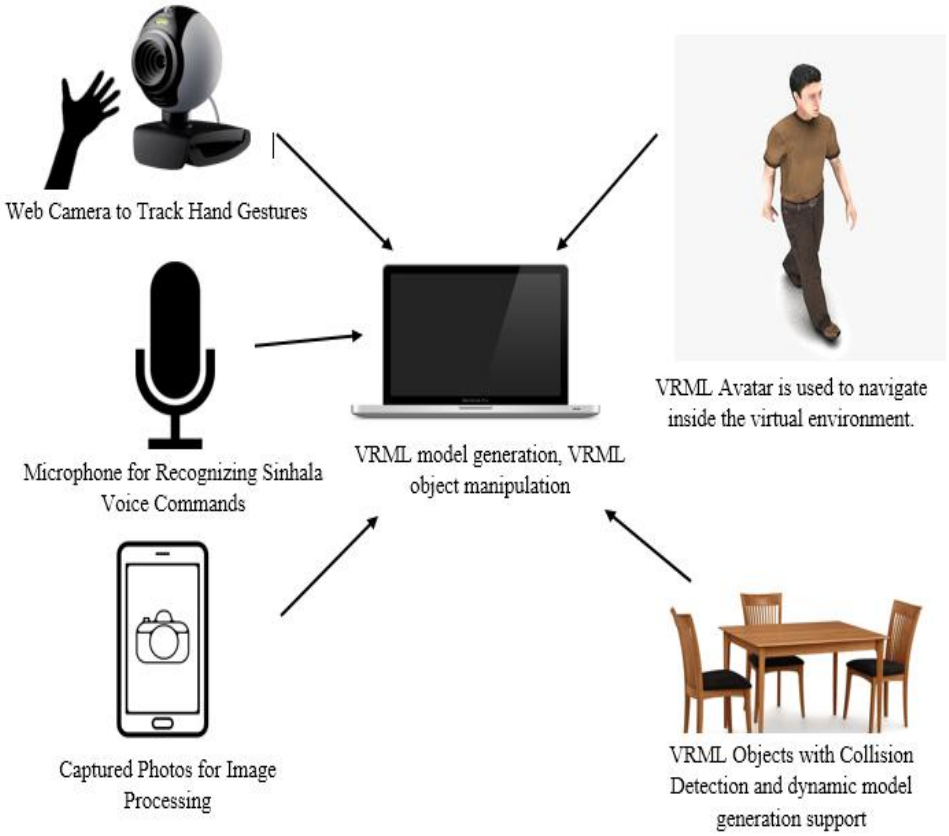


Figure 22: VRML markup framework system inputs

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

1. Generating the VRML objects using text descriptions

This section describes the main components of the object generation using the text description module, the inputs and outputs of each component and how the components interact with each other. The following diagram depicts the overall architecture of this module. The inputs, outputs and processes of each of the above sub-modules are described below.

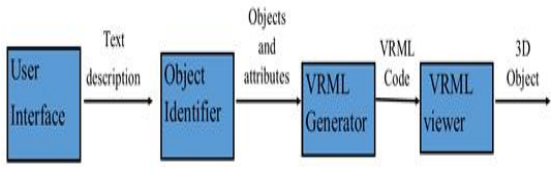


Figure 23: Architecture of object generator using a text description

User interface

This is a java swing application that allows the users to input a description using the English language. This model does not carry out any processing. Instead, it outputs the user input directly to the "Object identifier" module.

Object Identifier

This module takes the text description as the input and identifies the nouns and the adjectives describing the nouns in a paragraph using a POS tagger. Then, it will output the objects along with their attributes to the VRML generator module

VRML Generator

This module will take the objects and their attributes as the parameters and create the VRML code to represent the objects and the attributes given as the parameters. The output of this module will be a text file containing the VRML code.

VRML viewer

VRML objects generated from the previous module can be viewed from the web browser. Cortona 3D plugin will be used for this purpose.

2. Generating the VRML objects using image processing

This part describes the analysis and design of the VRML object generation using image processing.

This module has mainly two parts. They are the identification of the object and generation of the VRML world.

.

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

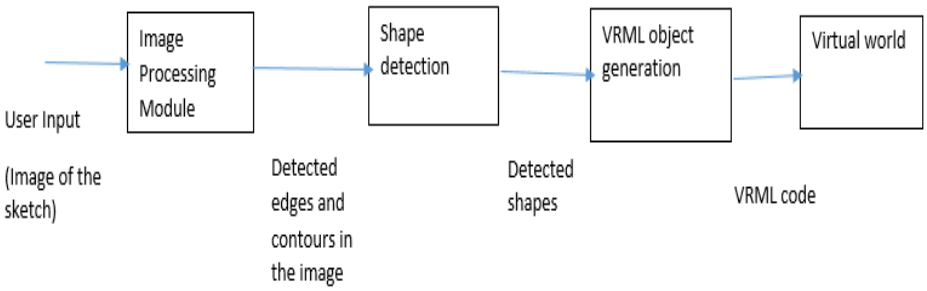


Figure 24: Architecture of object generator using image processing

The user's input that is the sketch is the input for the image processing module. By analyzing the image, it identifies the edges of the image.

In the image processing module,
First, the image is converted to the greyscale image

The Gaussian function is applied to remove the noise in the image. Then it is thresholded to find the edges of the image. The accuracy of edge detection can be checked by changing the threshold value. The contour approximation method is used to identify the shapes. Contour approximation is based on the Ramer-Douglas-Peucker algorithm. The contour approximation method is described in the implementation chapter. Another task is to develop the 3D model in VRML by analysing the shapes in a sketch of the object. An algorithm needed to be developed to generate the VRML code according to the identified shapes of the sketch. Also, another algorithm should be developed to place the objects drawn by the user in the sketch in the 3D world according to their coordinates. (If the user draws many objects)

Testing the accuracy

Accuracy can be tested by changing the threshold value. Also, using different sketches of the same object, accuracy can be tested. You can check whether the object is identified correctly.

By placing the objects in the sketch in different locations, one can check whether the objects are placed correctly in the virtual world.

3. Manipulation of Virtual Objects with Real-Time Gesture Tracking

This module is mainly coupled with the video camera, which is used as the input device to get signals into the component. Open CV library is used to analyze the input feed, and we are applying morphing

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

and contouring image processing techniques to reduce the noise at the second stage. Users also can manually tune the video feed as described in the above chapter with the controllers given. Then the component can track the hand gestures under "Track able Mode", which describes in the 3rd phase. As in figure 25, the four-phase is the interfacing section of this module with the virtual environment. Finally, the component output can access system controls like pointers and do operations like "Selection", "Move", and "Release" of the objects.

When describing the process of detecting hands from the video feed, it follows an algorithm. The video stream is segmented into frames, and those frames are sampled. Each frame is converted into its binary representation, and contouring is applied.

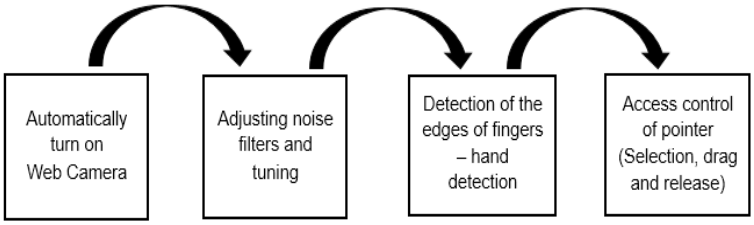


Figure 25: Flow of inputs and output of each stage in hand gesture detection module

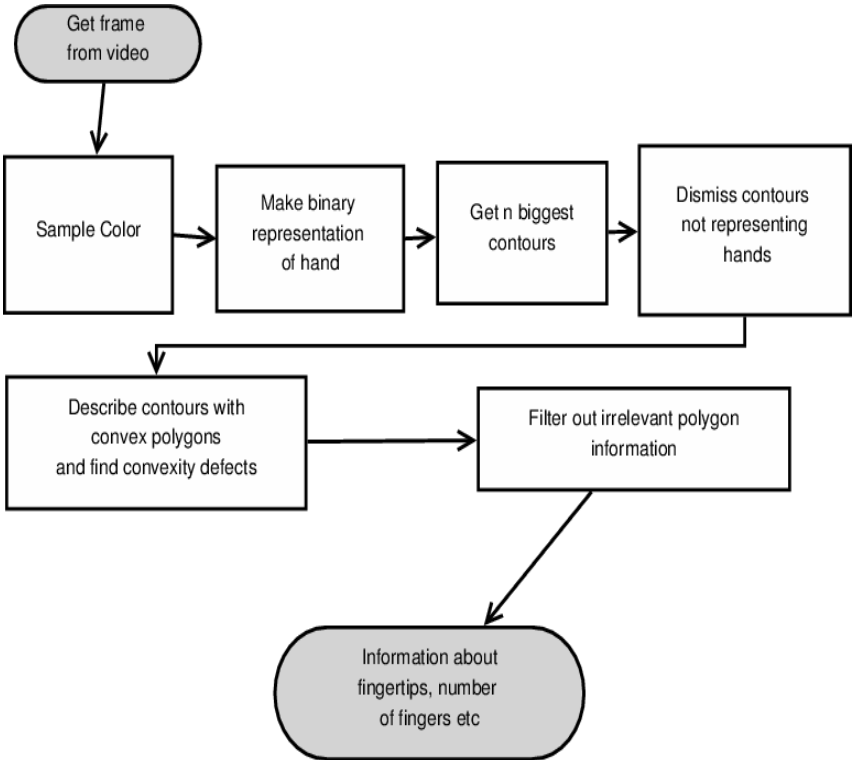


Figure 26: Flow of processing the hand detection algorithm

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

4. Voice Recognition for manoeuvring avatar in Virtual Environment with Voice Commands

This component has a language input in Sinhala. Stereo or Mono based voice input is taken into the module, and it decomposes the sound into chunks. Our design for this component is mainly under two sections named "Acoustic Model" and "Language Model". The acoustic model is the major component of detecting phonemes in speech, and the Language model can detect the connection between words in a sentence.

Acoustic Model

This acoustic model of the component receives speech input as a recording and phonetic strings as scripts to compile and transform into a statistical representation of speech sounds. The process is shown in figure 27.

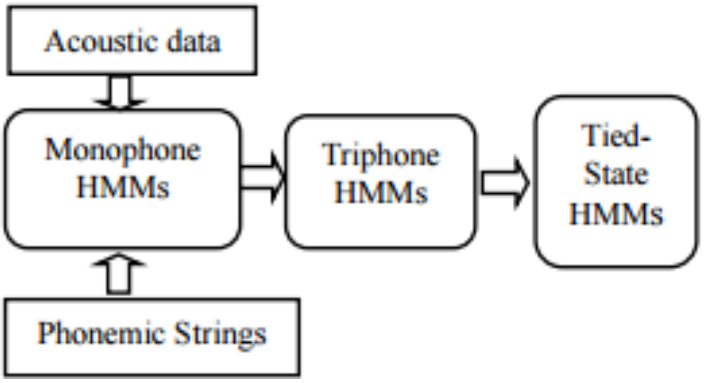


Figure 27: Acoustic Model and its design flo

Language Model

A language model is a pronounceable word dictionary. We are developing this language model by training dataset and currently if was developed up to a certain level. Therefore, we refer it to as the bigram language model. The process of language model and training data is shown in figure 28.

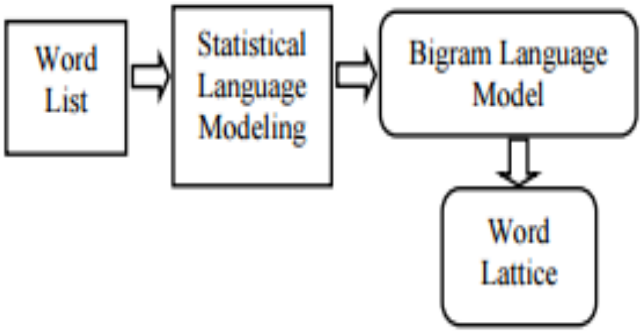


Figure 28: Language Model and its design flow

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

Training the module

The acoustic data of the training data set is processed for feature extraction, and the output will be the input for acoustic model training. The training dataset script (transcript) is the language model's input. Figure29 shows the process of training, and the trained model will be tested, and results are obtained to make some adjustments for enhanced accuracy.

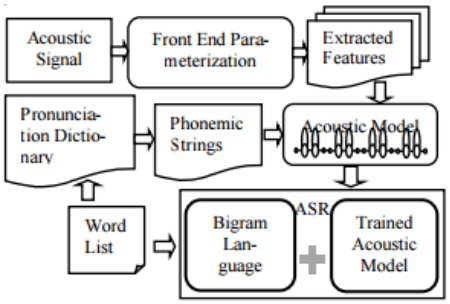


Figure 29: Training Process

IMPLEMENTATION

This chapter presents details of the implementation of each module in our solutions, which are mentioned in the above sections. And also, this will include the progress of each module at the interim level of the research.

1. **VRML Object Generation using text description**

This module is implemented using Apache Open NLP natural language processing library. Firstly the user is allowed to enter a description using a simple Java swing application. As we proceed with the project we will develop a desktop application integrating the four main modules, and the user input will be obtained using that application.



Figure 30: Interface for user input

After obtaining the user input, it will be passed to the object identifier sub-module. Here each word will be tagged with their respective Part Of Speech (POS) using the POS tagger available in the Apache Open NLP library.

The Java code which is used to implement the POS tagger and extract the objects and their attributes are as follows;

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

```
public class POSTagger {

    public String tagContent(String input) throws IOException{
        String output="";
        ArrayList<VRMLObject> objectArray = new ArrayList<VRMLObject>();
        int counter =0;

        POSModel model = new POSModelLoader()
        .load(new File("en-pos-maxent.bin"));
        PerformanceMonitor perfMon = new PerformanceMonitor(System.err, "sent");
        POSTaggerME tagger = new POSTaggerME(model);

        ObjectStream<String> lineStream = new PlainTextByLineStream(
            new StringReader(input));
        perfMon.start();

        String line;
        while ((line = lineStream.read()) != null) {

            String whitespaceTokenizerLine[] = WhitespaceTokenizer.INSTANCE
                .tokenize(line);
            String[] tags = tagger.tag(whitespaceTokenizerLine);


       POSSample sample = new POSSample(whitespaceTokenizerLine, tags);
       String[] taggedWords=sample.toString().split(" ");
       String[] current;
       for(String a:taggedWords){
           VRMLObject currentElement = null;
           System.out.println(a);
           current=a.split("_");
           if(current[1].equals("JJ")){
               currentElement.attributes.add(current[0]);
           }
           if(current[1].equals("NN")){
               currentElement.name=current[0];
               objectArray.add(counter, currentElement);
               counter++;
           }
       }

       System.out.println(sample.toString());
       perfMon.incrementCounter();
   }
  perfMon.stopAndPrintFinalResult();

       return output;
   }
```

Figure 31: Code to implement the POS tagger

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

The output obtained from the POS tagger is as follows;

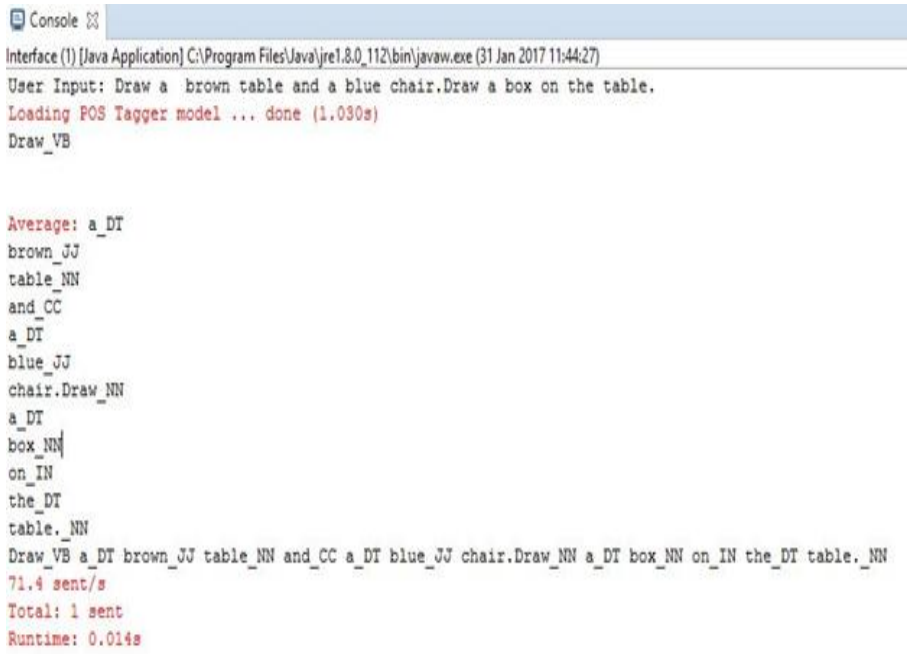


Figure 32: POS tags for user input

After identifying the tag of each word, each tag is inspected, and the words with the tag "NN" (noun) will be identified as objects (ex: - table in the above user input), and the words with the tag "JJ" (adjective)

After identifying the tag of each word, each tag is inspected, and the words with the tag "NN" (noun) will be identified as objects (ex: - table in the above user input), and the words with the tag "JJ" (adjective) is identified as an attribute of the noun following that adjective (ex:- brown is an attribute of a table in the above user input).

After identifying the objects and their attributes, the VRML code is generated accordingly.

```
#  Table top
#
Transform {
    translation 0.0 0.615 0.0
    children [
    Shape {
        appearance DEF Brown Appearance {
        material Material {
            diffuseColor 0.6 0.35 0.0
        }
        }
        geometry Cylinder {
        radius 0.7
        height 0.03
        }
    }
    ]
}

#
#  Table leg
#
Transform {
    translation 0.0 0.3075 0.0
    children [
    Shape {
        appearance USE Brown
        geometry Box {
        size 0.09 0.57 0.09
        }
```

Figure 33: VRML code generated for table

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved



Figure 34: VRML model for table

2. VRML Object Generation using Image Processing

Image processing techniques are used to identify the shapes in the sketch drawn by the user. Open CV library is used for this purpose. Relevant VRML file is generated after identifying the object. In this part implementation of the module is described.

```
def detect(self, c):
    # initialize the shape name and approximate the contour
    shape = "unidentified"
    peri = cv2.arcLength(c, True)
    approx = cv2.approxPolyDP(c, 0.04 * peri, True)
```

Figure 35: Defining the shape detector

Contour approximation is used to perform shape detection. Contour approximation is an algorithm for reducing the number of points in a curve with a reduced set of points. That is why it is named approximation. This algorithm is commonly known as the Ramer-Douglas-Peucker algorithm or simply the split-and-merge algorithm.

Contour approximation assumes that a series of short line segments can approximate a curve. This results in an approximated curve consisting of a subset of points defined by the original curve.

Contour approximation is implemented in Open CV (cv2.approxPolyDP method).

First, the perimeter of the contour is calculated then the contour approximation is made.

```
    # if the shape is a triangle, it will have 3 vertices
    if len(approx) == 3:
        shape = "triangle"

    # if the shape has 4 vertices, it is either a square or
    # a rectangle
    elif len(approx) == 4:
        # compute the bounding  box  of the contour and use the
        # bounding box to compute the aspect ratio
        (x, y, w, h) = cv2.boundingRect(approx)
        ar = w / float(h)

        # a square will have an aspect ratio that is approximately
        # equal to one, otherwise, the shape is a rectangle
        shape = "square" if ar >= 0.95 and ar <= 1.05 else "rectangle"

    # if the shape is a pentagon, it will have 5 vertices
    elif len(approx) == 5:
        shape = "pentagon"

    # otherwise, we assume the shape is a circle
    else:
        shape = "circle"

    # return the name of the shape
    return shape
```

Figure 36: Implementation of contour approximation

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

A contour consists of a list of vertices. The number of vertices can determine the shape of an object.

Ex:-if the contour approximated contains three vertices, it should be a triangle.

If a contour has four vertices, it can be either a square or a rectangle. In order to determine whether it is a rectangle or square aspect ratio is calculated. That is simply getting the width and the height of the contour bounding box and dividing the width by the height.

If the aspect ratio is ~1.0, the shape is square (since all sides have approximately equal length).Else the shape is a rectangle.

If a contour has five vertices, it is a pentagon. Otherwise, the shape we are examining is a circle.

Finally, we return the identified shape to the calling method.

```
# load the image and resize it to a smaller factor so that
# the shapes can be approximated better
image = cv2.imread(args["image"])
resized = imutils.resize(image, width=300)
ratio = image.shape[0] / float(resized.shape[0])

# convert the resized image to grayscale, blur it slightly,
# and threshold it
gray = cv2.cvtColor(resized, cv2.COLOR_BGR2GRAY)
blurred = cv2.GaussianBlur(gray, (5, 5), 0)
thresh = cv2.threshold(blurred, 60, 255, cv2.THRESH_BINARY)[1]

# find contours in the thresholded image and initialize the
# shape detector
cnts = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL,
    cv2.CHAIN_APPROX_SIMPLE)
cnts = cnts[0] if imutils.is_cv2() else cnts[1]
sd = ShapeDetector()
```

Figure 37: Shape detection with Open CV

First, the image is loaded and then it is resized. Then the image is converted to the greyscale image. The gaussian function is applied to eliminate the noise of the image. Then the image is thresholded to find the contours of the image. The threshold value should be found experimentally. From the contours, shapes can be determined.

The thresholded image is looked like this.

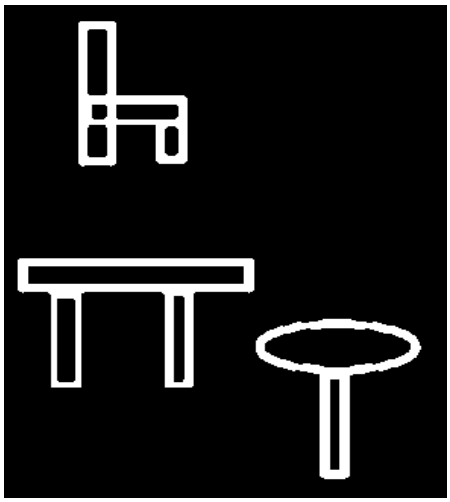


Figure 38: Threshold image

Finally, the relevant VRML file will be generated. In VRML, objects are defined using geometric objects.

3. Manipulation of Virtual Objects with Real-Time Gesture Tracking

This module is implemented using Open CV 2 image processing library with C++ programming language in Visual Studio integrated development environment (IDE).

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

Other than the Open CV image processing library, we used the "highgui library" to add rich user interfaces to the application. Track bars, windows and other GUI components were added to provide user-friendliness to the module. And also, this library helps to remember the states of controls.

```
#include<opencv2\highgui\highgui.hpp>
#include<opencv2\imgproc\imgproc.hpp>
```

Figure 39: Including Open CV libraries

When executing this module, it accesses the web camera drives installed in the machine and displays the live camera view on the screen. The Video Capture cap(0) method is used in Open CV to capture the camera image. Here "0" in parameter refers the inbuilt camera of the system. If the user wants to use an external camera, our framework easily gives that option by changing this parameter into "1". Then we are using in show function as follows to display the camera video to the user.

```
if (show changed frame)
in show("Camera", frame);
else
in show("Camera", range frame);
```

Figure 40: Image showing off the camera feed

The next important step of this module is to convert the actual colour video feed into gray scale for tuning and increasing accuracy levels. For that, we are following cvt Color (frame, grey, CV_BGR2GRAY), which converts the colour model into gray scale in each frame captured by Video Capture cap(0) as described above.

Moreover, there is a control window to adjust attributes of this camera feed to tune the video into less noise level, which increases the accuracy rate of the gesture recognition. There we have considered the attributes like Hue, Saturation and Values under the HSV colour model. Moreover, the parameters like erode, dilate, blur, and thresh are helpful for sharpening the moving object within the camera feed separating from background colours.

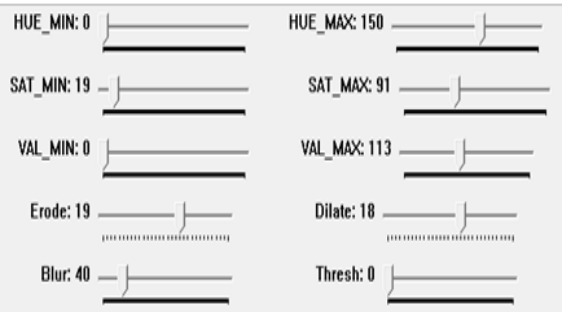


Figure 41: Camera feed noise reduction controls

Track bars have been used to adjust those controls as necessary. In create Track bars () and on_ track bars () methods are used

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

for implementing this control in multiple occasions.

Image morphing algorithm is used to enhance the quality of the video feed, which is computed with the web camera.

Under this morphing, the system manipulates video frames one into another for seamless transitions. This reduces the eye sensitive lagging of the visuals as well. Since we are using OpenCV, it supports this feature, and we used inbuilt library support.

4. Voice Recognition for manoeuvring avatar in Virtual Environment with Voice Commands

Since we are using the Sphinx library, we included the relevant header files to the source code, as shown in figure 41.

```
import java.io.File;
import java.io.FileInputStream;
import java.io.InputStream;

import edu.cmu.sphinx.api.Configuration;
import edu.cmu.sphinx.api.SpeechResult;
import edu.cmu.sphinx.api.StreamSpeechRecognizer;
```

Figure 42: Adding libraries into module

After initialising the environment, we used multiple wave files to define the dictionary, as shown in figure 41. The acoustic model used to identify phrases and words from a set of sentences depends on the training data set. We use software named “praat” to record these data with a 16 kHz Mono channel. Those recording files are kept in .wave format.

As described in our approach, we usea Bi-gram model lattice to represent the spoken words in statistical representation. For that, we implemented the dictionary for several Sinhala words, as shown in the following figure.

Pronunciation dictionary contains words with phonetic representations. There are two levels of transcripts that are involved in this, called phone level transcription and word-level transcription. So we started with word-level transcription first before phone level transcription. It was done by executing Perl custom script prompts2mlf in the HTK toolkit.

CONCLUSION

The proposed VRML solution, including all four modules, was tested with different data inputs and evaluated by comparing them with the expected outcomes.

In summary, the dynamic 3D model generation using the text description model expressed an accuracy rate of 77% when

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

the provided text descriptions are in modest complexity. However, this rate becomes lower when the complexity levels are high.

On the other hand, creating 3D models based on hand-drawn sketch inputs and image processing techniques such as thresholding and contouring showed an 80% accuracy level when it comes to identifying the shapes in the hand-drawn sketch (if the image is not overexposed or underexposed).

The hand gesture recognition module to manipulate virtual objects depicted accuracy up to 78%when the module automatically assigned the default threshold value. This accuracy value was increased up to 82%, depending on the manual adjustments of this threshold value.

With the combination of all modules, the solution expressed an adequate level of user-friendly and immersive indoor environment modelling solution with the ability to interact with indoor objects. However, further research can be carried out to improve the realism and accuracy levels by incorporating improved algorithms.

REFERENCES

1. Woodrow Barfield and Thomas A. Furness III, "Virtual Environments and Advanced Interface Design," Oxford Univ. Press, 1995.

2. Mark R. Mine et al. "Moving Objects in Space: Exploiting Proprioception in Virtual-Environment Interaction," Proceedings of SIGGRAPH 97, pp.19-26, August, 1997.

3. Hand Interface for Immersive Virtual Environment Authoring System by Motion Information Team, VR Center, Electronics and Telecommunications Research Institute (ETRI).

4. Hand Gesture Interaction with a 3D Virtual Environment by Nasser H. Dardas and Mohammad Alhaj.

5. Visual Specification of Behaviours in VRML Worlds University, Halifax, Nova Scotia, Canada by ShapourArjomandy Trevor J. Smedley Dalhousie.

6. Q. Chen, A. El-Sawah, C. Joslin, and N.D. Georganas. 'A dynamic gesture interface for VE based on

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

hidden Markov models', Proc. IEEE International Workshop on Haptic, Audio and Visual Environments and their Applications (HAVE2005), pp.110–115, 2005.

7. T. Metais, and N.D. Georganas. 'A glove gesture interface', Proc. Biennial Symposium on Communication (23rd), Kingston, Ontario, Canada, 2004.

8. S. Mitra and T. Acharya. Gesture Recognition: A Survey. IEEE Transactions on Systems, Man, and Cybernetics, Vol. 37(3), 2007, pp. 311-324.

9. Y. Wu, and T. Huang. 'Hand modeling analysis and recognition for vision-based human computer interaction', IEEE Signal Processing Magazine, Special Issue on Immersive Interactive Technology, Vol. 18, No. 3, pp.51–60, 2001.

10. Joslin, A. El-Sawah, Q. Chen. and N.D. Georganas. 'Dynamic gesture recognition', Proc. IEEE Instrumentation and Measurement Technology Conference (IMTC2005), Ottawa, Canada, 2005.

11. Object Shape Recognition in Image for Machine Vision Application by Mohd Firdaus Zakaria, Hoo Seng Choon, and Shahrel Azmin Suandi in International Journal of Computer Theory and Engineering, Vol. 4, No. 1, February 2012.

12. R. O. Duda and P. E. Hart, "Use of the hough transformation to detect lines and curves in pictures," Comm. ACM, vol. 15, pp. 11–15, 1972.

13. R. Brunelli, "Template matching techniques in computer vision: theory and practice," Wiley, ISBN 978-0-470-51706-2, 2009.

14. M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: active contour models," International Journal of Computer Vision. vol. 1, no. 4, pp. 321–331, 19874.

15. Automatic Detection and Recognize Different Shapes in an Image by Nidhal El Abbadi and Lamis Al Saadi in IJCSI International Journal of Computer

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

Science Issues, Vol. 10, Issue 6, No 1, November 2013.

16. Singh G., Jati A., Khasnobish A., Bhattacharyya S., Konar A., Tibarewala D.N., Nagar A.K., 2012. "Object shape recognition from tactile images using regional descriptors ", Nature and Biologically Inspired Computing (NaBIC), 2012 Fourth World Congress on, pp: 53 – 58.

17. Schindler Konrad, Suter David, 2008. "Object Detection by Global Contour Shape", Article published in Pattern Recognition, pp: 1-30.

18. Sketch that Scene for Me: Creating Virtual Worlds by Freehand Drawing Ellen Yi-Luen Do.

19. Igarashi, T., R. Kadobayashi, et al. (1998). Path Drawing for 3D Walk though. UIST 98. San Francisco, ACM: 173-174.

20. Bob Coyne, Richard Sproat .Words Eye: An Automatic Text-to-Scene Conversion System. AT&T Labs—Research.

21. R. J. Abbott. Program design by informal English descriptions. Commun. ACM, 26(11):882–894, 1983.

22 Cockburn, "Using Natural Language as a metaphorical Basis for Object Oriented Modeling and Programming", IBM Technical Report TR-36.0002, 1992.

23 L. Mich and R. Garigliano. A linguistic approach to the development of object oriented systems using the nl system lolita. In ISOOMS '94: Proceedings of the International Symposium on Object-Oriented Methodologies and Systems, pages 371–386, London, UK, 1994. Springer-Verlag.

24 Delisle S. Barker K. Biskri I. "Object-Oriented Analysis: Getting Help from Robust Computational Linguistic Tools." The Fourth International Conference on Applications of Natural Language to Information Systems (OCG Schriftenreihe 129), Klagenfurt, Austria, 1999, 167-171.

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved

25 H.M. Harmain and R. Gaizauskas, "CM-Builder: An Automated NL-based CASE Tool", In Proceedings of the 15th IEEE International Conference on Automated Software Engineering (ASE'2000), 2000, PP. 45-53.

26 Mosa Emhamed Elbendak. Student Research Workshop, RANLP 2009 - Borovets, Bulgaria, pages 23–28.

27 Daisuke Ninomiya, Maxim Mozgovoy. HCCE'12, March 8–13, 2012, Aizu-Wakamatsu, Fukushima, Japan.

28 Thilini Nadungodage and Ruvan Weerasinghe, Continuous Sinhala Speech Recognizer, March 2015, University of Colombo School of Computing, Sri Lanka.

29 Stuart Goose, Ingo Gruber, Sandra Sudarsky, Ken Hampel, Brent Baxter, Nassir Navab, 3D Interaction and Visualization in the Industrial Environment, Multimedia Department, Imaging and Visualization Department Siemens Corporate Research 755 College Road East, Princeton, NJ 08540, USA.

Page 34-75 © MANTECH PUBLICATIONS 2021. All Rights Reserved