Use Cases

1 Introduction

1.1 General

Use cases should be based on everyday situations where people with strong visual impairment ("user") try to overcome visual challenges. This is accomplished by conversation with an equipment ("system") with capability to see environment (perception), construct a world model (comprehension) and predict future events (projection).

The conversation is either situation or user triggered.

1.2 Situation Triggered

System triggers speech in following situations:

Situation	Description
New {thing} or {stuff} enters the scene.	Name of {thing} or {stuff} class. For example, "Car". Upon system start, all classes are spelled out. (Same as answer to "What do you see?" question).
Probability of collision is significant and may cause severe injury or damage.	"Danger! {Thing}/{Stuff} will collide. Move {direction}." For example, "Danger! Wall will collide. Move left." The direction is based on making the collision probability lower.
Probability of collision is low but may cause severe injury or damage.	"Warning! {Thing}/{Stuff} may collide. Move {direction}." For example, "Danger! Car may collide. Move left." The direction is based on making the collision probability lower.
Probability of collision is significant but causes no or minor injury or damage.	"Caution! {Thing}/{Stuff} may collide. Move {direction}." For example, "Caution! Cat may collide. Move left." The direction is based on making the collision probability lower.

1.3 User Triggered

User triggers speech by giving following questions and commands:

Question/Command	Description
What do you see?	System list all thing and stuff classes in view. It also provides count for each class. For example, "3 cars, person, motorcycle. Probably also car. Clothes, road, sky, tree, wall." "Probably" refers to confidence level.
Where is {thing}?	The answer is formed for each instance of {thing} class. If not present, "No {thing}". The answer identifies distance and direction of object centre relative to the camera. For example, "Distance 6 meters, right 2 degrees, up 4 degrees."
Where is {stuff}?	The answer is formed for each instance of {stuff} class. If not present, "No {stuff}". The answer identifies minimum and maximum distance, and direction of object span relative to the camera. For

	example, "Distance 2.1 to 3.2 meters, from 10 degrees left to 30 degrees right."
Repeat answer	System keeps answering to the previous question until "Stop repeating" command is given, or enough time has passed.
Stop repeating	System stops answering to the previous question.
What colour is {thing}/{stuff}?	The answer is formed for each instance of {thing} or {stuff} class. If not present, "No {thing}". For example, "Red, yellow".
Where is {colour} {thing}/{stuff}?	Can be used to restrict instances for questions "Where is {thing}?" and "Where is {stuff}?". For example, "Where is blue car?".
Is there {thing}/{stuff}?	If not present, the answer is no. If present, the count is also given. For example, "Is there bus?", "Yes, two".
Follow {stuff}	For stuff classes "Road" and "pavement" the system keeps giving directional advice to keep the class stuff in middle enabling the user to follow path. For example, "Left, left, ahead, ahead, ahead, right". The initial position must be on {stuff}.
Stop following	Cancels "Follow" command.
Go to {thing}	System keeps giving directional advice to reach the {thing}. For example, "Go to bus", "Left, left, ahead, ahead, ahead, right". If there are multiple instances of {thing}, a warning is given.
Is {thing} moving?	The answer is formed for each instance of {thing} class together with speed and direction (approaching/leaving left/right). If not present, "No {thing}". If not moving, "No" otherwise speed. For example, "Yes, 57 km/h, approaching, left to right.".
Read text	System identifies all view areas including text and reads them.
Be quiet	Disables situation triggering.
Speak to me	Enables situation triggering.
How many {thing}/{stuff}?	Count of {thing}/{stuff} class instances. For example "How many cars?","4".
Horizontal list	List {thing}/{stuff} instances from left to right with general location (left/right) and distances. For example, "Left, car 13 meters, bus 15 meters. Right, car 22 meters."
Vertical list	List {thing}/{stuff} instances from up to down with general location (up/down) and distances. For example, "Up, bowl 0.9 meters, bowl 0.8 meters. Down, fork 0.7 meters."
Is {thing}/{stuff} on/beside/under {thing}/{stuff}?	Based on location proximity and orientation of objects. For example, "Is person on chair?", "Yes" or "Is bottle on table?", "No".
What size is {thing}?	System gives estimated height, width and depth of the item. For example, "What size is bottle?", "Height 15 cm, width 5 cm, depth 5 cm.".

1.4 List of Use Cases

Use Case	Description
Basic Description	Overall description of environment for creating spatial understanding.
Item Search	Enables user to search for an item.
Route Guidance	System gives directional orders for the user to approach an objective. Warns for obstacles and collisions.
Item Identification	System tells the user about the item, including attributes like colour and size.
Road Crossing	System tells the user if it is safe to cross a road. Warns for obstacles and collisions.
Price Tag	System locates price information and tells it to user.
Reading	System reads a book, magazine, computer text, etc to user.
Free Seat Search	System identifies a free seat and guides the user to approach it.
Shopping Assistance	System lists items, guides user to approach the selected item and tells the prize.

2 Basic Description
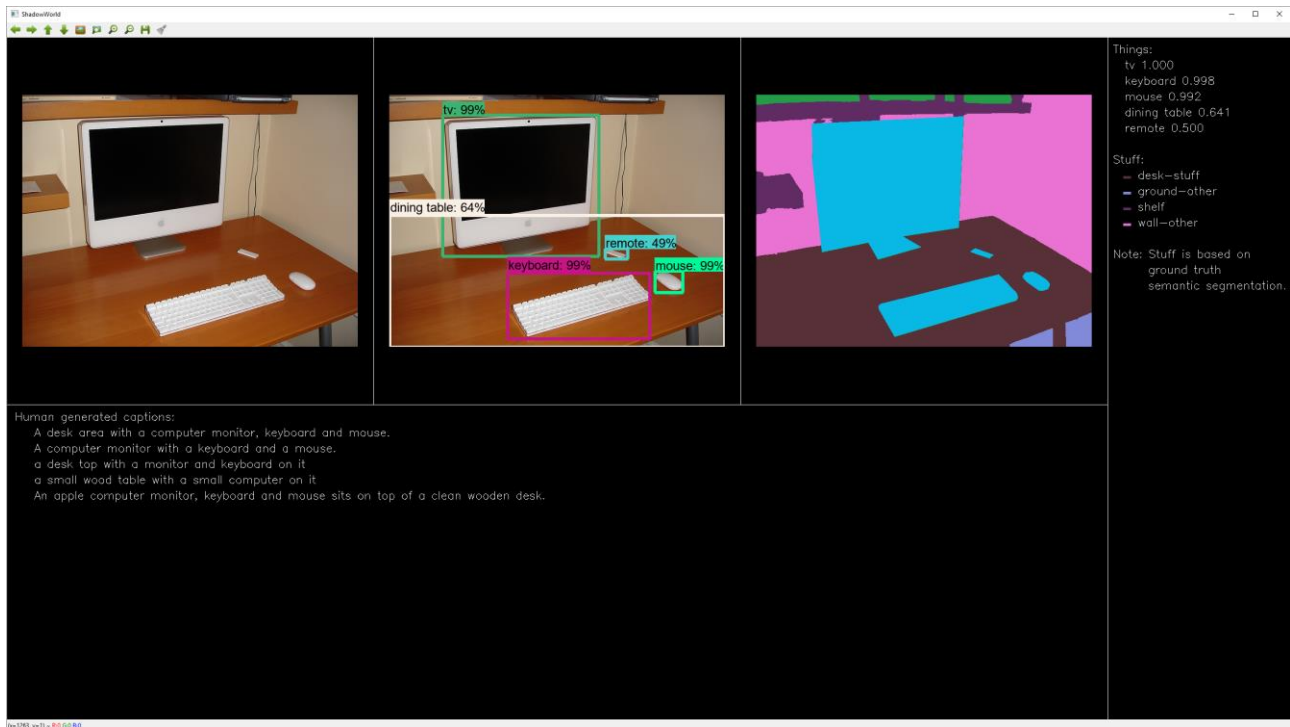
User: "What do you see?"

System: "Tv, keyboard, mouse. Probably also table, remote. Desk, shelf, wall."

User: "Horizontal list"

System: "Left, tv 1.8 meters. Right, keyboard 1.2 meters, remote 1.4 meters, mouse 1.3 meters."

3 Item Search

4 Route Guidance

5 Item Identification

6 Road Crossing

7 Price Tag

10 Shopping Assistance