Multimodal Human-Robot Interaction in an Assistive Technology Context

Elizabeth Harte and Professor Raymond Jarvis

Intelligent Robotics Research Centre
Department of Electrical and Computer Systems Engineering
Monash University, Clayton, Victoria 3800, Australia
ARC Centre for Perceptive and Intelligent Machines in Complex Environments
{ Elizabeth.Harte, Ray.Jarvis }@eng.monash.edu.au

Abstract

In this paper, we present a prototype robotic system that captures, processes and fuses speech, vision and laser-depth data to more accurately interpret and perform simple tasks in a domestic environment. We can never assume that any one of these inputs are completely accurate, but by using a combination, a more accurate interpretation could be found. For each speech, gesture recognition and object recognition input, an associated probability of correctness is calculated. Using these probabilities, a predefined associative map between specific words and gestures, and contextual information, we update the associated probability until some threshold has been reached. This contextual information includes the history of recently uttered phrases, as well as the information about known objects in the environment. Once an interpretation, complete or otherwise, is found, a response is formulated. This system has been developed using a Mitsubishi Heavy Industries robot, Wakamaru, as a platform[1].

1 Introduction

Having robots to assist people in their home is a goal that the robotics community has aspired to for many years. The proportion of people aged 65 years and older in Australia's population was 12% in 1999, but is projected to increase to between 24% and 27% in 2051 [ABS, 2000]. While assistive technologies may not replace human carers in the future, they could provide viable alternatives. A robot carer could assist an elderly person in a domestic environment, allowing that person to feel independent and remain in their home while still having some basic support and security.

A crucial requirement of a robot that interacts with humans is that it correctly interprets what is being communicated to it. To effectively do this, a robot should demonstrate both spatial and transactional intelligence. Spatial intelligence is being able to understand and navigate the space that a robot is in, as well as knowing object locations and how to manipulate those objects. Transactional intelligence is being able to meaningfully communicate with a person by speech or gesture recognition. These intelligences both use contextual knowledge, such as what was recently said by the speaker, where the robot and the speaker are in the space, and temporal information (e.g. time of day).

We can rarely assume that either the spatial or transactional knowledge is perfectly correct all the time, so to avoid misunderstandings in the interpretations, a probabilistic correlation of the data using an associative map is used to decide which response is to be made, especially if that response is to gain more information to clarify the interpretations further. This correlation means that simple techniques can be used for the data processing, and many errors will be suppressed [Oviatt, 2000].

The field of mutimodal integration began with the work in the seminal paper, [Bolt, 1980], and continued with works such as Virtual World [Codella et al, 1992], Finger Pointer [Fukumoto et al, 1994], VisualMan [Wang, 1995], Jeanie [Vo and Wood, 1996] and QuickSet [Cohen et al, 1997]. All these systems integrated speech and either pen or gesture based motion at a semantic level. A unification based approach was developed by [Johnston, 1998] that also used speech and pen based recognition, and has been well referenced since. [Wu et al, 1999] used an associative map to represent legal semantic combinations between all types of speech and pen based inputs.

The work in [Johnston and Bangalore, 2000] defined a finite state multimodal parsing system, which uses a finite state device to parse multiple input streams into a single semantic representation, but has not been fully evaluated. A salience based approach was proposed by [Eisenstein and Christoudias, 2004] but again has yet to be evaluated on real time extracted multimodal data. [Holzapfel and Nickel, 2004] have also developed a constraint based multimodal system for speech and pointing gestures which was evaluated on a small datasetm though they do not handle n-lists from multiple modalities. Finally, [Russ et al 2005] used a semantic network for their multimodal fusion, taking into account temporal alignment, the surrounding environment and the history of the preceding user's inputs.

Of these papers, only [Russ et al, 2005] seems to directly incorporate environmental and some contextual information into their system. We believe this information, is crucial to integrating the multiple modes of input from a user.

In Section 2, we will outline the hardware and software of the robotic system developed, and in Section 3 discuss each piece of the system in more detail. Section 4 will present some simple experiments and demonstrations the system is capable of and discuss these. Section 5 outlines our future work for the project and Section 6 will summarise

1 Mitsubishi Heavy Industries, Ltd. More information at: http://www.mhi.co.jp/kobe/wakamaru/english/index.html

our conclusions.

2 Outline

2.1 Hardware

Wakamaru has a wheeled base, a pair of articulated arms, each with four degrees of freedom, and a head with three degrees of freedom (see figure 1). It also has an array on onboard sensors, such as infrared and ultrasonic sensors and touch sensors on the shoulders and in the hands. The robot is able to localize itself with a set of infrared reflectors positioned at known locations on the ceiling using the panoramic camera on the top of its head. This robot has unarticulated hands, so our demonstrations merely show that Wakamaru can find an object, but not return it to the user. The retrieval task will not be dealt with until a pair of articulated hands is engineered.

The Wakamaru system included a Java API of all its programmable functionalities, which include detecting power levels, going to and leaving its charging base, giving its current location, and moving itself. While this API was documented in Japanese, through the use of AltaVista Babelfish online translation[2], we have been to access many functionalities to make our robotic system richer.

Processing of the speech, vision and laser depth data is done on a separate 2.66GHz, 512MB RAM laptop computer, which is attached Wakamaru's back and communicates with Wakamaru wirelessly via a wireless router.

A Point Grey Research BumbleBee stereo camera[3] was attached to the front of the robot's head. This camera can produce disparity and colour images at 15fps, and is used for all our computer vision processing. Wakamaru's onboard sensors have a 0.4m range, which is only useful for local obstacle avoidance. Using the disparity image, objects up to 3m away can be detected, which allows better navigation in a cluttered space.

Figure 1: Wakamaru with Bumblebee camera on head, HokuyoURG laser range finder at middle and laptop on back.

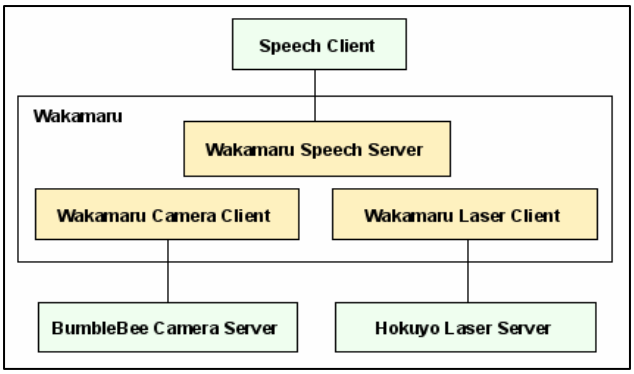

Figure 2: Client-server system design

Because it is difficult to find stereo matches on visually bland regions, we also use a HokuyoURG Laser Range Finder[4] to return a single stripe of depth data. The laser range scanner has a depth range of 0.02m to 4m, with a 270 degree angular range and an angular resolution per step of 0.352 (3d.p.) degrees. For obstacle avoidance, only a 129 degree angular range was scanned, ignoring obstacles that are on either side or behind Wakamaru.

2.2 Software

The developed system is distributed by design. The capturing and processing of the speech, vision and laser-depth data are all done in separate processes on the laptop, as it is a more powerful system than Wakamaru (see figure 2). The correlation and response system is a single threaded process running on Wakamaru. The speech system is a client which triggers action in Wakamaru's correlation system by recognizing and processing a spoken utterance. A single spoken utterance may return as many as five possible interpretations and estimates of correctness. These are all sent to Wakamaru for correlation. Once these interpretations are received by Wakamaru, it will request the last recognized gesture from the camera system server, if a gesture was recognized at all. The BumbleBee camera system is a server system that is continually detecting gestures while it waits for an instruction from Wakamaru to return the last found gesture, detect obstacles, or recognize objects. In this version of the camera system, only the dynamic gestures 'wave', 'come' and 'go' are detected. The HokuyoURG laser system waits for an instruction from Wakamaru to perform a scan and return valid depth values.

When a speech input is sent to the robot, it will then request the last recognized gesture from the camera server system, rather than detecting the current one. This is because in 93% of cases, a deictic word occurs during or after an associated gesture [Kettebekov and Sharma, 2001; Poddar et al, 1998]. While this system does not semantically recognize deictic words, it is assumed this statistic would apply to general key words in a spoken phrase. The speech and gesture inputs all have associated probabilities. Assuming that more than one possible speech input was recognized or more than one possible gesture was detected, then these ambiguities need to be clarified. A hypothesis is

2 More information at http://babelfish.altavista.com/

3 Bought from Point Grey Research, More information at: http://www.ptgrey.com/index.asp

4 Hokuyo Automatic Co. Ltd. 1-10-9 Niitaka, Yodogawa-ku, Osaka 532-0033, Japan.

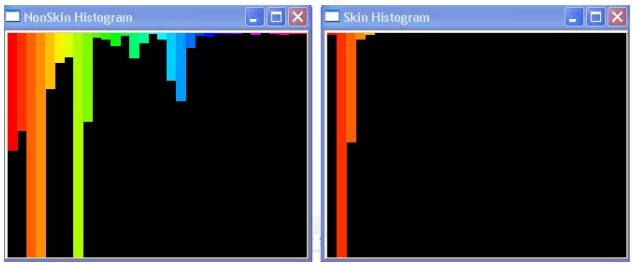

Figure 3: Non skin and skin histograms after normalizing

generated by correlating these speech and gestures inputs, updating the speech probabilities according to the gesture inputs, the recent hypotheses list and the known objects in Wakamaru's object database. The gesture inputs are only updated with reference to the speech inputs. Once a most likely hypothesis is derived, an active response can be taken by Wakamaru, either to attempt to perform the task requested, or perhaps to request a clarification of the hypothesis made for one of the systems. This hypothesis is added to the recent hypotheses list, which is referenced as contextual information when updating the speech input probabilities.

3 The System

3.1 Speech Recognition

For our speech recognition, we decided to use Microsoft Speech SDK 5.1[5] because it was free, well documented, trainable and had innumerous sample programs to illustrate its use.

Training the speech recognition engine required reading provided passages while the engine associates what is said with what is written. Approximately 7 hours of training was needed before this very American engine was recognizing most of the 60 words the system is interested in. A training file was also developed containing the specific phrases the system should recognize, such as "Bring me my blue cup."

Using [Chakkarandeep, 2004] as a guide, a simple speech recognizer was developed which would recognize a user's utterance as a phrase. Each utterance has at most 5 interpretations and each of these interpretations is parsed word by word, where each word has an associated confidence value from 0 to 50 000, as defined by the speech recognition engine. The parsing simply detects an action, a colour, an object type and/or a location based on predefined domestic-related lists of words. If a word is in the set of 60 the system is using, such as 'bring', 'blue' or 'cup', then the word's predefined unique ID number and its confidence value are stored in the statement structure, [*action, colour, object, location*], to be sent to Wakamaru. These confidence values are summed for each sentence and once all the alternatives have been processed, all the phrase confidence values are scaled so then all the values sum to 1.0.

5 Microsoft Speech SDK 5.1 is available for download from: `http://www.microsoft.com/speech/speech2007/downloads.mspx`.

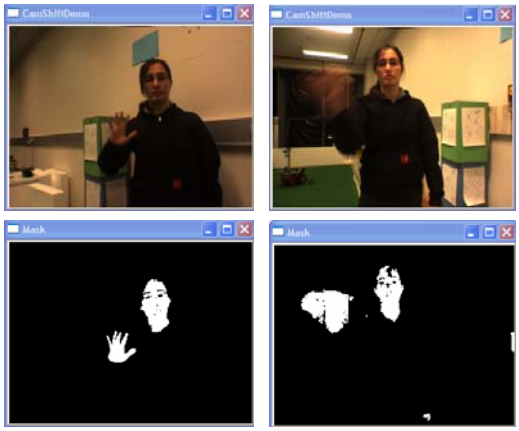

Figure 4: Two examples of skin segmentations

3.2 Gesture Recognition

The gesture recognition in the camera system recognizes dynamic, i.e. moving, gestures using OpenCV's [Intel, 2006] Camshift tracking algorithm and 3D geometry. The three dynamic gestures to be recognized are 'wave', 'come here' and 'go away'. Gesture recognition would only be performed if a frontal face was detected in the scene using OpenCV's Haar Object Detection algorithm, as it is assumed a person is looking at who they're gesturing to.

3.2.1 Skin Segmentation

The skin regions are segmented using a simple histogram based method defined by [Jones, 2001]. Two hue-saturation histograms with 32 bins are defined, one representing skin values and another one representing non-skin values. The skin histogram was trained using 500 images of skin pixel samples, each 320x240 in size, from Phung's segmented skin database, the results from [Phung, Chai and Bouzerdoum, 2001]. The non-skin histogram is trained on 50 images of the laboratory without people in it. This gave the histograms 7 130 275 skin pixel samples and 3 840 000 non-skin pixel samples for training. The histograms were then normalized to 10 000 each, and the skin histogram discards bins less than 10 in size. The resulting histograms can be seen in figure 3.

The captured colour image from the camera is converted to HSI colour format. The system decides if a pixel, hs, is skin if

$$\frac{P(hs \mid skin)}{P(hs \mid \neg skin)} \geq \Theta \qquad (1)$$

Where $P(\ hs \mid skin\ )$ is the bin value for hs in the skin histogram and $P(\ hs \mid \neg skin\ )$ is the bin value for hs in the non skin histogram. Pixels whose intensity value is above 50, where 0 <= intensity <= 255, are ignored as they would appear as near black, i.e. not skin. Θ is a threshold, defined as 0.4, based on the experimental results of [Jones and Rehg, 2001] comparing the number of correctly and incorrectly segmented pixels. Examples of skin segmentations with no skin coloured objects in the background are shown in figure 4.

Figure 5: The backprojection of the skin histogram on the hue-saturation channels (left) and the resulting region selections (right)

3.2.2 Region Tracking

OpenCV's implementation of the Camshift algorithm was used to track the hand region in a series of frames. The Camshift algorithm uses the backprojection of the skin histogram onto the colour image and a mask of all the segmented skin pixels to identify its initial search windows, one per skin region, in the frame. In the next frame, it assumes part of the tracked region will still overlap with one of the search windows. If this is true, then the bounding box of the tracked region in the next frame is returned, otherwise the region is considered lost for that frame. The center of the bounding box is considered the (x,y) coordinate of a region for that frame. Figure 5 shows a backprojection and the corresponding tracked regions in boxes.

For each tracked region of each frame, the center (x,y) and the average disparity value of the tracked region are translated into (X, Y, Z) camera coordinates using Point Grey Research's Triclops library and are stored. Tracking lasts for 10 frames which at an average 3 – 5 fps lasts 2 or 3 seconds. The region with the most variance in its set of 10 frames is assumed to be the dynamic gesture, as all other regions are probably static background.

There are some issues with the Camshift implementation, particularly with gesture detection. If the hand moves too near the face, CamShift loses the hand as more of the face region will still overlap with the current bounding box than the moving gesture, so Camshift selects the face as the tracked region. To avoid this, the dynamic gestures are performed to one side of the person, rather than in front, as is natural. It was decided that this was acceptable as having an ideal gesture recognition system was not a goal of this project.

Another issue is if the search window is too big, multiple regions could be detected in the one window, resulting in inaccurate search windows for the following frames, but also loss of detailed tracking of smaller regions. It was decided to make the search window as large as the region it was tracking, which also aids in losing small noisy regions.

3.2.3 Features and Recognition

When Wakamaru sends the instruction to return the most recently received gesture, then lines are approximated in the XY and XZ planes for the last stored data set for the region with the highest variance in both XY and XZ planes. It is assumed that the region that has moved the most is the gesture region, as only moving gestures are being recognized. The gradients of these lines and the ratio of the variance in XY and XZ directions are the features of the recorded gesture. However, a region of the background may

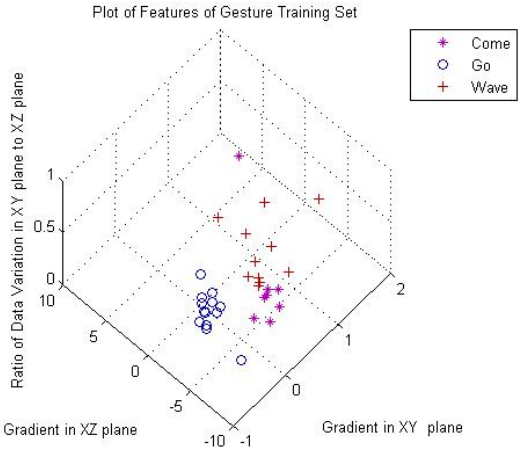

Figure 6: A 3D plot showing the gestures training database in terms of their three features.

'move' more than the gesture resulting in a misrecognition. To avoid this, the background is meant to be free of skin coloured objects that could be accidentally segmented.

Using OpenCV's k-Nearest Neighbour (kNN) implementation, a kNN classifier was trained on between 8 and 15 samples of each dynamic gesture. A plot of these gestures are in figure 6. This classifier is used to classify the recorded gesture, returning the five best matches, and the distances from the recorded gesture to the nearest matches. These distances are summed for each class (wave, come and go), then normalized so they sum to 1.0. The camera server then sends the probability for each class through to Wakamaru, as response to its request.

3.3 Object Recognition

When Wakamaru sends an instruction to the camera system requesting an object, the hypothesis in our system is 'find this object'. The requested object class and colour, if stated, are sent to the camera system by Wakamaru. The object classes in this system are 'cup' (which includes mug and glass), 'bowl', 'plate', 'purse' and 'case', as in a spectacle's case. (see figure 7)

This simple 2D object recognition system uses invariant moments and skeleton shapes as the most important features to be collected and compared. The invariant moments are initially used to normalize the size and translation of the gesture within the image space based on the steps described in [Barton and Delmas, 2002].

To handle the training and classification of the skeleton shapes of the objects, the ShapeMatcher (SM) program [Macrini, 2003] was used. Macrini developed this program as part of his Master's at the University of Toronto. It initially creates a skeleton of a binary image and then stores indexed segments of it in a shock graph, "a hierarchical, directed acyclic graph representing the decomposition of a 2D shape silhouette into primitive parts" [Macrini, 2003]. SM prunes away excess branches to represent a simplified skeleton of the object. This resulting graph is stored in a database for later reference. This algorithm is immune to segmentation noise and deformation to allow it to handle viewpoint changes. This was an appealing feature of

Figure 7: The five of the six objects recognised. The sixth is a blue cup of the same shape and size as the green one.

Macrini's system. To train the SM database, between 6 and 15 images of each object class from varying viewpoints were manually segmented to create the binary images. These images were then normalized before being processed by SM and added to a database file.

The objects are mostly solid coloured objects, simplifying the segmentation problem. The camera system segments objects from an image using the colour provided by Wakamaru. It was assumed that there are no brightly coloured regions that are not objects in Wakamaru's view when it is looking down at a table surface except the green carpeted floor surrounding the table. It was also assumed that the table is a visually bland light coloured surface, such as white.

If no colour is sent with the object class then the system will segment the image for each colour separately, matching the extracted regions to the database. While slow, this allows unique separation of the objects easily without implementing a new method. A better colour segmentation method will be introduced in a future version of the system.

Using OpenCV's implementation of contour detection, each contour is extracted as a binary image, normalized then compared to the training database using the SM's built in matching algorithm. The matching algorithm is two phase, initially selecting a small number of candidate objects from the database that may classify the image. Each of these candidates is then matched to the query using hierarchical structures, returning a similarity value.

By altering parameters, SM returns the 10 best matches, and their similarity values, for the extracted region when it's compared to the training database. These similarity values are summed for each class (cup, bowl, plate, purse and case), as the region may match more than one object class. Since we are always provided with a desired object class by Wakamaru, the system will ignore any extracted region that doesn't have this object class as a possible match.

For all the regions that have the desired object class as a possibility, their similarity values are normalized so they sum to 1.0. For each possible region using the disparity image, an (X, Y, Z) coordinate of the object is calculated with reference to the camera. The camera server sends the probability for each class and the (X, Y, Z) coordinates through to Wakamaru, as response to its request.

3.4 Obstacle Avoidance

A variety of sensors were used to build a rich environment model of our domestic environment. Wakamaru has 3 infrared sensors about 100mm from the ground and a wide ultrasonic sensor about 300mm from the ground. The HokuyoURG laser range finder is mounted approximately 500mm from the ground with the Point Grey Research Bumblebee Stereo Vision camera is attached to Wakamaru's head.

The laser data can return various error messages, but mostly commonly the error message is 'too close' or 'too far'. All laser range finder data within the 129 degree angular range that isn't an error message is what is added to the obstacle dataset.

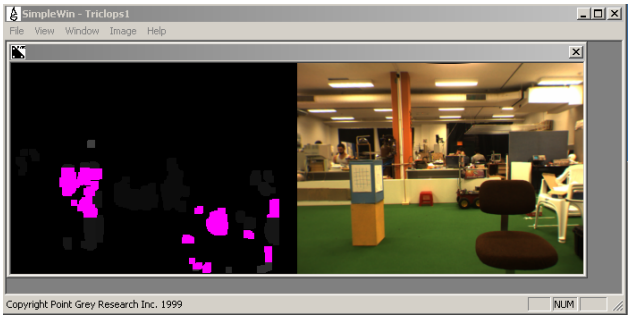

Figure 8: Obstacle avoidance using disparity (left). Pink pixels are considered obstacles to be avoided.

For the disparity data from the BumbleBee camera, pixels that may have been mismatched are ignored and noisy pixels are removed by applying some simple morphologicals. This step would sometimes remove small object pixels, such as a thin pole, but we considered this an acceptable loss as the other sensors may be able to pick it up.

Using Bumblebee's Triclops software, each valid pixel (x,y) and its associated non-zero disparity value are translated to (X, Y, Z) camera coordinates. Only obstacle pixels that are below Wakamaru's eye level (to ignore the ceiling) and above 400mm (to ignore the floor, close and far) are considered obstacles. A 2D obstacle map is generated in the XZ plane and the XZ pairs are then translated to polar coordinates, so they can be added to the same obstacle dataset as the laser's obstacles. In the obstacle dataset, each cell in the occupancy grid has an associated probability, the probability that a cell is being occupied by an obstacle at a particular time. This probability is calculated based on the cell's prior probability and whether an obstacle was detected by either the laser or camera at this time.

The navigation algorithm used is a distance transform based method with static obstacle avoidance using Manhatten distance to calculate the distance field, as developed by [Gupta and Jarvis, 2007]. The steepest descent path to the destination is determined based on the calculated distance transform. If no path is found, then the failure is declared. As the robot is traversing its path, it continually detects possible obstacles from both the camera and the laser. If a obstacle is detected on its path, it will recalculate the distance transform including the new obstacles, and plot a new path using its current location as the start point. If the robot successfully makes it to its destination, it will tell the user and, if required, continue to carry out its tasks.

3.5 Correlation and Robotic Response

If the speech and gesture recognition systems each return more than one input, then the correlation system needs to decide which input pair is the correct one. To find complimentary pairs between a speech input, i.e. phrase, and a gesture input, an associative map was manually developed [Wu et al, 1999]. The rows of this map each represent a gesture, where the columns each represent a spoken word from our list of 60 words. The map illustrates

relationships between a gesture and an individual word. Initially, this map was manually constructed with either 2's, 1's or 0's, indicating whether the corresponding inputs strengthen each other, are unrelated or are contradictory to each other (see Table 1). The associative map values are decided based on the context of a domestic environment and common sense. For example, the gestures 'come here' and 'go away' are contradictory to the word 'stop', so their associative map values are 0. Also, the phrase "What's the time?", simply listed as 'Time', is not complimentary or contradictory to these gestures, so the values are all 1's.

Within each system, either speech or gesture, each input would have a probability of correctness, all of which sum to 1.0. The values of the inputs would be updated iteratively until one is greater than some specified threshold. The speech input values are updated per word using:

$$P(x_m^{v+1}) = wt \times P(x_m^{v}) \quad (2)$$

$$wt = wtMap + wtHist + wtObj + wtRisk \quad (3)$$

$$wtMap = \sum_{n=1}^{N} q_{mn} \quad (4)$$

$$wtHist = \sum_{h=1}^{H} hist_{mh} \quad (5)$$

$$wtObj = \sum_{d=1}^{D} obj_{md} \quad (6)$$

$$wtRisk = \sum_{r=1}^{R} risk_{mr} \quad (7)$$

Where x_m is a word m of speech input x which has M possible inputs and v is the iteration index. *wtMap* is the weighting of x_m with respect to the gesture input n using the associative map, where q_{mn} is the element of the mth row and nth column in the associative map of N x M size. *wtHist* is the weighting of x_m with respect to the recent correct interpretations, where $hist_{mh}$ is a value between 0 and 1 if x_m and the previous word x_h matches. Only objects and locations are currently considered in this weighting for simplicity. *wtObj* the weighting with respect to the database of known objects, where obj_{md} is a value between 0 and 1 if input x_m and an object x_d matches by type, and perhaps colour and location. *wtRisk* is the weighting with respect to the predefined risk matrix, where particular words, such as 'stop' or 'help', has a higher risk of being misrecognised.

If x_m is the gesture system, the formula would only be with reference to the associative map, *wtMap*, because it is assumed that a 'wave' gesture is not more likely now if it was done before, such as could be recorded in the conversational history. The gesture history could be relevant if it were part of a user model, however that has not been developed for this project. In addition, a gesture cannot be directly associated with an object description, and our current set of gestures do not have any that would require risk values, though perhaps in a future extension that would be needed such as for the gesture 'stop'.

Once all the values have been updated for each word, those values are averaged to become a value for the whole speech input. For each a system, the input values are put through a sigmoid function which squashes the values to between 0.0 and 1.0, but also slows the rate of increase as the input's value approaches the specified threshold, as is often used in neural networks. The outputs from the sigmoid function are normalized for each system so they sum to 1.0 again. This process repeats until either both a speech input and gesture input satisfies a threshold, i.e. is selected, or the maximum number of iterations is reached.

The resulting interpretation of what was communicated by the user can be either be considered clearly understood or ambiguous. A few simple common-sense rules were defined to outline ambiguous statements.

- the selected speech and gesture inputs are conflicted or unrelated according to the associative map,
- no speech input is selected,
- no action word was recognized in the selected speech input, or
- any necessary information is missing from the selected speech input, i.e. action is to 'get' but no object is defined

These rules are checked for once both a speech and gesture input have been selected but while we're still inside the iteration loop. If the inputs are ambiguous the system resets the values in each system so they are all equivalent and then the iteration process continues using the new input values. This handles cases where one of the inputs may have initially satisfied the threshold without iteration, but may not have been the best input. If this does not resolve the ambiguity, the robot would verbally request more information from the user. However, the robot should not ask for clarifications too repeatedly, as this may be annoying to the user, so there should be limitations to the number of enquiries made. In future implementations, this limitation could be varied depending on the user model.

An additional extension for future versions of the system could be a 'double check', where if the interpretation is considered clear, but the speech is not above some second threshold, then the robot could simply check with the user that its interpretation is correct with a simple yes/no question to be sure. While this could also be considered an annoyance to the user, it would give the system confirmation of its interpretations, which would be particularly useful for user modeling.

Temporal and localization information would also be included in this section in future implementation, as they would weight different inputs to get a more accurate interpretation.

Table 1: A sample of the associative map

Gest/Speech	Hello	Time	Come	Stop	Help	Kitchen
Come	1	1	2	0	2	2
Go	1	1	0	0	2	2
Wave	2	1	1	1	1	1

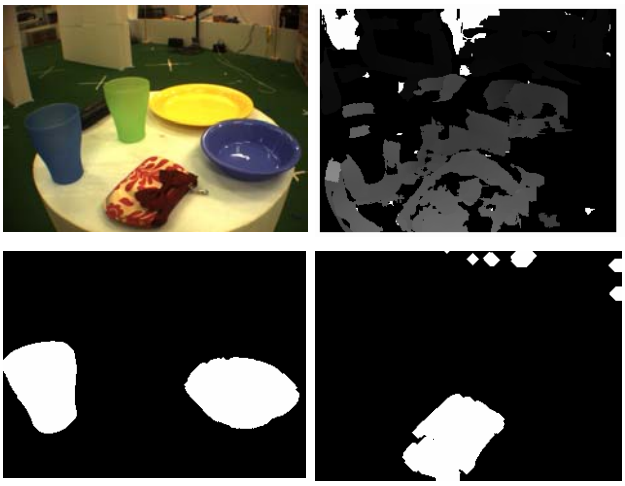

Figure 9: A captured colour image and its corresponding disparity image (top). The bottom row shows two different colour segmentations, for blue (left) and red/pink (right). You can see the shapes of the cup, bowl and purse.

4 Experiments and Discussions

4.1 Speech Recognition

By reading typical sentences that the system should recognize and recording the first recognition result according to the Microsoft Speech Recognizer v5.1, the number of incorrectly recognized words, either by deletion, substitution or insertion, was counted. For a sample set of 82 words making up 25 phrases, 11 words were substituted for incorrect recognitions, 1 word was deleted and 2 words were inserted. This gives the system a word recognition rate of 82.93% if only the first recognition result was used, such as the first row of table 1.

However, the developed speech system developed takes into account the first 5 recognition hypotheses, as shown in Table 1. The average word recognition rate for all the recognition hypotheses is 87.20%, 4.27% higher than if only the first recognition was used. Based on the 82 words spoken, if the most accurate hypothesis is picked from the 5 recognition hypotheses each time, the word recognition rate would be 89.02%.

4.2 Gesture Recognition

The gesture recognition system was tested by the user standing in front of the camera, approximately 2 meters away, performing the 'come', 'go' and 'wave' gestures right-handed, but to the side of the body, so then the hand skin region will not overlap the face skin region. These conditions also applied to the training data.

Seven samples of each gesture were recorded in the laboratory with the robot and the user in different positions and orientations around the room, but still facing each other and being 2 meters apart. This gave the images varying light conditions, but also varying backgrounds to test the gesture recognition on. The recognition rate was 57.14%, which is far from ideal.

Twice the hand moved too close to the face and the regions were merged. This will have to be avoided in future by being more careful with the performance of the gesture. Five times regions of the ceiling were segmented as skin regions and had more variation than the moving gesture itself, so were incorrectly recognized as the dynamic gesture. Perhaps by varying the gain in the camera, then the ceiling will appear whiter, which it is, than coloured. Only once did the hand region misrecognized as an incorrect gesture, and that was because the region would get 'stuck' on a patch of wall that was segmented as skin coloured and changed the dataset too much.

While this dynamic gesture recognition is far from ideal, improving its robustness is definitely a high priority for this project. The system will also be extended, using different methodology, to recognize static gestures, such as 'stop', 'thumbs up', and 'ok'.

Table 2: Showing 5 speech recognition hypotheses and their confidence values

Speech Recognition Hypothesis	Confidence Value
1. bring the blue cut from the kitchen	151180
2. bring the blue cap from the kitchen	151180
3. bring the blue cut from the kitchen to	151180
4. bring the blue come from the kitchen	151180
5. bring the blue cup from the kitchen	212275

4.3 Object Recognition

The object recognition system was tested by sending 10 different requests to the system and recording the results. The requests included "blue cup", "green cup", "yellow plate", "blue bowl" and "black case", as in a spectacles' case. The remaining requests were for an object class, such as "cup", without specifying a colour. Three different views of a table with the objects arranged around were used to test the object recognition, as in figure 9. The object recognition system could return more than one possible recognition result, if there's more than one region that could be the requested object. Each of these results would have an associated probability, as well as an (X, Y, Z) coordinate with reference to the camera. Using these (X, Y, Z) coordinates, it could be seen which object the recognition result was refering to on the table.

When the colour was provided in the request, the recognition rate was 77.8% over 18 requests. There were only three incorrect alternatives, one of which had a higher probability than the correct result. When colour was not provided in the request, the recognition rate was 88.9% for 18 requests. However, 50 incorrect alternatives were generated by the various requests. Six of these were generated by requesting an object type which was almost completed occluded, and an additional 21 were for when the request was to find a "purse". Since the purse shape is not geometrically simple, it was more likely to match to various segmented regions in an image. The overall recognition rate for any request is 83.3%, if the alternatives are taken into account. However, if only the region with the highest probability was returned, the recognition rate would be 55.6% because of the simple nature of the training, segmentation and recognition.

4.4 Data Correlation

The correlation system was tested by using the speech and gesture recognition results and correlating all possible combinations of these inputs to see how the system would react. The threshold to select an input was 0.7 and the number of iterations was capped at 300. 12 gesture inputs and 22 speech phrases were used, giving 264 possible hypotheses results. The maximum number of iterations possible was 79200. These results are either clearly understood or ambiguous, based on the rules defined in section 3.5. In this dataset, 219 test cases were interpreted as clearly understood while 45 were ambiguous using only 26720 iterations.

Of the 219 clearly understood results, 167 had a complimentary gesture selected and 52 had no gesture selected. This latter case would occur if the spoken phrases were not complimentary to any of the gesture inputs, such as with the queries 'What's your name?' and 'What's the time?'. This also occurred when the phrase action word was 'come' and the gesture inputs were 'go' and 'wave', where 'go' had an initial probability above the threshold. The 'go' gesture and 'come' word conflict, so all the input values were made equal and the iteration began again. A 'come' phrase did satisfy the threshold, but a gesture never did.

Out of the 45 ambiguous results, 43 of these were expected. 12 because the speech input did not have an action word so all cases were ambiguous regardless of the gesture input. 24 cases were because two speech inputs did not have words in our list, so there was no information to process. 7 ambiguous cases occurred because only a single gesture possibility was returned and the gesture 'go' only complemented 5 of the 12 speech phrases.

Two additional cases of ambiguity occurred unexpectedly where no speech or gesture was selected. The 2 phrases had 'come' as the action word and the gesture inputs were 'wave' and 'go' with similar input values (0.47 and 0.53 respectively). Neither of these gestures compliment the action word 'come' so neither gesture satisfied the threshold. However, no speech input was selected as an interpretation either. After further investigation, a cyclic process was noticed where a 'come' speech input was selected, then the 'wave' gesture input. But because they conflicted, their input values were reset, did not satisfy the threshold and the iteration continued selecting a 'come' speech and the 'wave' gesture again. This ambiguous result is similar to the clearly understood 'come' action word with the 'go' and 'wave' gestures not selected, except the initial input value of 'go' was at least 0.4 higher than the input value of 'wave' so the cyclic process happened to finish when the speech was selected but not the gesture. This cyclic relationship is unstable and needs to be resolved. One possibility is if a gesture is detected, a gesture then must be selected for interpretation regardless of the threshold. This would be a more sensical response, as when people communicate, they would not ignore someones gestures if they don't fit with the words spoken – they try to make sense of it, perhaps by asking for a clarification.

With regards to the number of iterations, only 88 of the 264 test cases iterated the maximum 300 times. Of the remaining cases, the maximum number of iterations was only 6. In future implementations, we may either raise the threshold level from 0.7 or we may drop the maximum number of iterations, which would speed up the algorithm for ambiguous cases.

For each case where an object and an object colour was part of a spoken phrase, there would be at least one alternative which is missing either the object type, i.e. 'cup', or the object location. However, because of the object database information, the phrase with the most object information – type, colour and location (if spoken) – is always selected in one iteration. Any additional iterations is to select a corresponding gesture.

From these results, while some are expected, more testing is required to fully evaluate the usefulness of using the recent hypotheses and the known object database in a multimodal system. This system has been developed using live data results, though has not been thoroughly tested onboard a robotic platform. This is the next goal for the project.

5 Future Work

An immediate goal of this project is to finalize a real-time demonstration using the Wakamaru robot. The tasks, at this stage, are simple, but handling more complex tasks only involves scaling up the approach, as well as extending the vocabulary of the speech and gesture systems. We also hope we can train the associative map between speech and gestures through experienced based learning. The system should also be able to be triggered by gestures alone, such as a wave, as well as speech. Temporal information will also be included as a correlation factor, so that 'lunch' would be more likely if it was midday, and user models should also be developed as contextual information. Environmental information, such as the location of Wakamaru or the location of the user, should also be taken into account to strengthen or weaken input probabilities.

Ideally, this system should also include other modalities, such as eye gaze, head and body posture, facial expressions and speech analysis to complement the interpretation process further at the semantic level. With the unique sensors now easily available for purchase, many of these novel modalities are possible.

We also plan to extend this system further by incorporating particle filter based people tracking [Chakravarty et al, 2006], 3D face recognition [Axnick and Jarvis, 2005], and a more robust navigation algorithm [Gupta and Jarvis, 2007], as developed by members of the Intelligent Robotics Research Centre at Monash University. We should also integrate the DORIS project (Dialogue Oriented Roaming Interactive System) as a spoken dialogue module which is being developed by [Zukerman et at, 2006].

6 Conclusions

The system described has great potential to be a useful robotic system to perform simple tasks in a domestic environment. The system uses only simple techniques to capture and process speech, vision and laser-depth data and calculate their probabilities of correctness. This information is correlated using a predefined associative map in a simple probabilistic way, but still generates correct interpretations

despite the unstructured-ness of the environment and the errors in the capturing and interpreting processes. The use of recent hypotheses and a known object database greatly aids the correct hypotheses, and more contextual and environmental information could only strengthen the results further.

Acknowledgements

We would like to thank Mitsubishi Heavy Industries Ltd for the use of the Wakamaru robot for this project. Thanks are also due to Alan Zhang and Punarjay Chakravarty for their valuable discussions during the development of this project.

References

[ABS, 2000] Australian Bureau of Statistics. Population Projections: Australia 1999-2101. Canberra: ABS, Catalogue No. 3222.0, 2000.

[Axnick and Jarvis, 2005] Karl B. J. Axnick & Ray Jarvis. *Face and Pose Recognition for Robotic Surveillance.* Australasian Conference on Robotics and Automation (ACRA), December 2005.

[Barton and Delmas, 2002] Barton, G. and Delmas, P, *A Semi-Automated Colour Predicate for Robust Skin Detection*, CITR at Tamaki Campus, Computer Science Department, University of Auckland, 2002.

[Bolt, 1980] R. A. Bolt, "Put that there: Voice and gesture at the graphics interface," *Comput. Graph.*, vol. 14, no. 3, pp. 262-270, 1980.

[Chakkaradeep, 2004] C.C. Chakaradeep. *A Simple Speech Application Using SAPI 5.1 SDK*. [Cited 2007 January]. Available from: `http://www.codeproject.com/audio/SAPI.asp`.

[Cohen et al, 1997] P. Cohen, M. Johnston, D. McGee, S. Oviatt, J. Pittman, I. Smith, L. Chen, and J. Clow. QuickSet: Multimodal interaction for distributed applications, *Proceedings of the Fifth Annual International Multimodal Conference*, ACM Press: New York. November, 1997

[Codella et al, 1992] C. Codella, R. Jalili, L. Koved, J. Lewis, D. Ling, J. Lipscomb, D. Rabenhorst, C. Wang, A. Norton, P. Sweeney, and C. Turk, Interactive simulation in a multi-person virtual world, in *Proc. ACM Conf. Human.* Factors in Computing Systems (CHI'92), pp. 329–334. 1992.

[Eisenstein and Christoudias, 2004] Jacob Eisenstein and C. Mario Christoudias. A Salience-Based Approach to Gesture-Speech Alignment. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pp.25-32. East Stroudsburg, Pennsylvania, May 2-7 2004.

[Fukumoto et al, 1994] M. Fukumoto, Y. Suenaga, and K. Mase, "Finger-pointer: pointing interface by image processing," *Comput. Graph.*, vol. 18, no. 5, pp. 633–642, 1994.

[Gupta and Jarvis, 2007] Gupta O. K. and Jarvis, R. A. *Multi-sensory Fusion and Understanding of Human-Robot Interaction in Assistive Robotic Technology Environments: Navigation and Hand-Eye Coordination*, Transfer Report, ECSE, IRRC, Monash University, Australia. 2007.

[Holzapfel and Nickel, 2004] Holzapfel, H., Nickel, K., and Stiefelhagen, R. Implementation and evaluation of a constraint-based multimodal fusion system for speech and 3D pointing gestures. In *Proceedings of the 6th international Conference on Multimodal interfaces* (State College, PA, USA, October 13 - 15, 2004). 2004.

[Intel, 2006] Intel, *Open Source Computer Vision Library*, v1.0. `http://www.intel.com/technology/computing/opencv/` 2006.

[Johnston and Bangalore, 2000] Johnston, M. and Bangalore, S. Finite-state multimodal parsing and understanding. In *Proceedings of the 18th Conference on Computational Linguistics - Volume 1* (Saarbrücken, Germany, July 31 - August 04, 2000). International Conference On Computational Linguistics. Association for Computational Linguistics, Morristown, NJ, 369-375. 2000.

[Jones and Rehg, 2002] Michael J Jones and James M Rehg. Statistical Color Models with Application to Skin Detection. *International Journal of Computer Vision*, 46(1):81-96, 2002.

[Ketteberov and Sharma, 2001] Ketteberov, S., Sharma, R. Toward Natural Speeach/Gesture Control of a Large Display. Lecture Notes in Computer Science, vol 2254, *Proceedings of the 8th IFIP International Conference on engineernig for Human-Computer Interaction*, pp221-234, 2001.

[Macrini, 2003] Diego Macrini, *Indexing and Matching for View-Based 3-D Object Recognition Using Shock Graphs*, in *Graduate Department of Computer Science*. 2003, University of Toronto.

[Oviatt, 2003] Oviatt, S., "Advances in robust multimodal interface design," *Computer Graphics and Applications, IEEE* , vol.23, no.5, pp. 62-68, Sept.-Oct. 2003

[Phung, Chai and Bouzerdoum, 2001] SL Phung, D Chai and A Bouzerdoum, A Universal and Robust Human Skin Colour Model Using Neural Networks. *Proc INNS-IEEE Int'l Joint Conf Neural Networks.* vol 4, pp 2844-2849, July 2001.

[Poddar et al, 1998] Poddar, I., Sethi, Y., Ozyildiz, E., Sharma, R. Toward Natural Gesture/Speech HCI: A Case Study of Weather narration. *Proceedings of the Workshop on Perceptual User Interfaces, PUI98*, San Francisco, CA, pp1-6, November 1998.

[Chakravarty et al, 2006] Punarjay Chakravarty, David Rawlinson and Ray Jarvis. *Person Tracking, Pursuit & Interception by Mobile Robot*, Australasian Conference on Robotics and Automation (ACRA), Auckland, New Zealand, December 2006.

[Russ et al, 2005] Russ, G., Sallans, B. and Hareter, H. Semantic based information fusion in a multimodal interface. *HCI'05, International Conference on human-computer interaction*, Las Vegas, Nevada, USA, 20-23 June 2005, pp 94-100.

[Vo and Wood, 1996] M. T. Vo and C. Wood, "Building an application framework for speech and pen input integration in multimodal learning interfaces," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Atlanta, GA, pp. 3545–3548. 1996.

[Wang, 1995] J. Wang, "Integration of eye-gaze, voice and manual response in multimodal user interface," in *Proc.*

IEEE Int. Conf. Systems, Man and Cybernetics, pp. 3938–3942. 1995.

[Wu et al, 1999] Lizhong Wu, Sharon L. Oviatt, and Philip R. Cohen. Multimodal integration—A statistical view. *IEEE Trans. Multimedia*, vol. 1, pp. 334–341, Dec. 1999.

[Wu et al, 2002] Lizhong Wu, Sharon L. Oviatt, and Philip R. Cohen. From Members to Teams to Committee — A Robust Approach to Gestural and Multimodal Recognition. *IEEE Transactions on Neural Networks*, vol. 13, no. 4, July 2002.

[Zukerman et al, 2006] Ingrid Zukerman, Michael Niemann and Sarah George. *Probabilistic, Multi-staged Interpretation of Spoken Utterances*" *cimca*, p. 194, International Conference on Computational Inteligence for Modelling Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce (CIMCA'06), 2006.