

THE PUZZLE PROJECT: A CASE STUDY IN MULTIMEDIA SIGNAL PROCESSING

Jacob Griesbach, Julie Wiejaczka, Radu Frangopol, Fransiska Harsono, and Delores Etter

Department of Electrical and Computer Engineering
University of Colorado, Boulder USA
<http://sig2.colorado.edu/~puzzle>

ABSTRACT

The Puzzle Project is an interactive software system that solves jigsaw puzzles. The voice interface includes speech synthesis and word recognition. The attributes of the puzzle pieces are determined using image processing techniques and wavelet decomposition. Two algorithms are used to solve the puzzles: an expert system and fuzzy logic. This paper describes the steps required to find the solution to the puzzle from image processing to decision-making algorithms. It also explains the techniques involved in designing the voice interface.

1. INTRODUCTION

The objective of the Puzzle Project is to design and develop an interactive system that solves jigsaw puzzles using signal processing techniques. A sophisticated Graphical User Interface (GUI) is used to run the Puzzle Project system. Additionally, the user can choose a voice interface that incorporates both voice recognition and speech synthesis. Therefore the system will be able to recognize certain key words spoken by the user and is also capable of "talking" to the user. There are also a number of logic engines being developed for the Puzzle Project to solve the puzzles, including fuzzy logic and a built-in expert system. This paper will focus on the signal processing algorithms developed for the voice interface and logic engines.

2. SPEECH PROCESSING

There are several aspects in speech processing that are common to speech synthesis and word recognition. Among these are the endpoint detection and the Linear Predictive Coding (LPC) algorithms. The endpoint detection algorithm keeps only necessary samples of an utterance by rejecting undesired beginning and ending sections. It works according to energy, zero-crossing, and amplitude threshold constraints [1]. The LPC algorithm is

an effective way of extracting speech parameters for use in many algorithms.

In the Puzzle Project, a user has the option of enabling the system to perform speech synthesis and/or word recognition. The two subjects should be considered to be independent, but it is important to realize their interdependence at times.

Speech Synthesis

Once the user enables the speech synthesis in the Graphical User Interface, the computer has the ability to "talk". Through speech synthesis, the computer asks the user for his/her name, takes the user through its thought process, and also summarizes the Puzzle Project's many options. This interface provides beginners with an easy way to understand the program and as a result serves as a nice link between computer and user.

The speech synthesis system for the Puzzle Project relies upon a database of words. Through a sound editor (and microphone) an utterance is sampled and saved in memory. Once saved, an utterance can be imported into MATLAB and analysis is ready to begin. From the word database, a sentence is formed by concatenating different words. The words are retrieved from memory and the sentence is played back to the user.

Concatenation of previously recorded utterances to produce speech synthesis does not sound like too difficult a problem, but in reality a lot of small intricacies arise. Periods of silence at the beginning and end of every word must be eliminated so that the concatenated sentence flows smoother. Continuity in spectral characteristics between joined words is necessary if we want smooth transitions between concatenated words. Endpoint detection and algorithms using digital filters are used to correct for both of these errors [1].

Due to an extensive database and a storage requirement of 128 kbits/sec (sampling rate of 8 kHz and 16 bit quantization), storing speech is yet another problem encountered. This problem can be reduced by

This project is funded in part of the NSF Optoelectronic Computing Systems Center.

compressing our speech segments using the standard Linear Predictive Coding (LPC-10) algorithm. LPC-10 focuses on extracting specific parameters related to frames of speech. The speech parameters include pitch period, a voiced/unvoiced decision, a gain, and vocal tract filter model coefficients. By storing a few parameters for each frame as opposed to every data point, LPC-10 not only returns recognizable speech, but also decreases storage requirements by a factor ~ 50 .

Word Recognition

Word Recognition enables the system to understand certain keywords spoken by the user so that certain actions requested by the user can be performed.

So far, two algorithms have been used in this project. The first one is called the 'Quickly Recognizer' Algorithm [2]. The signal is filtered into logarithmically-scaled frequency bands, then cut into a fixed number of time segments. The average of the data for each band in each segment is stored as the reference pattern. The unknown word is recognized by comparing its pattern against all patterns in the training set. When implemented, this algorithm gives about 50% recognition accuracy, which is not acceptable for a robust system.

The second algorithm is Vector Quantization [3], [4]. One codebook is used for each word, which works well with a small vocabulary. To build a speaker-independent system, the training vectors are collected from multiple trainers. These training vectors are constructed from the cepstral coefficients, which are derived from the LPC coefficients [4].

The LPC order has to be large enough to maximize the performance of the recognizer, but small enough to minimize the time needed to do the computations. Sixth order LPC and ninth order cepstral representations are used to represent the speech signal. The data windows for the LPC computations are 30-ms long with 50% overlap. The Lloyd algorithm is used for finding the best set of centroids for the 3-bit codebooks [4]. Using these parameters, the system gives a 90% accuracy for the small-vocabulary recognizer.

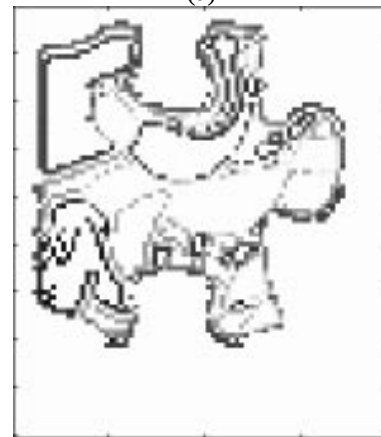
3. IMAGE PROCESSING

Formatting

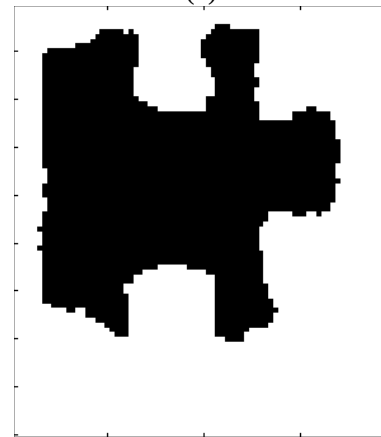
The first step in actually putting the puzzle together is to obtain the images of the puzzle pieces. Each piece is scanned with a specially selected background color (see Figure 1a). The background color is chosen as dark as possible to eliminate shadows but to be different from any of the colors on the edges of the puzzle piece. It is important to determine which parts of the images are the background color and which are the actual puzzle piece.



(a)



(b)



(c)

Figure 1: (a) scanned image of a puzzle piece. (b) edge detection of puzzle piece image. (c) filled image after background removal

Therefore, an edge detection algorithm was applied. Edge detection determines any large changes in color throughout the image. This is done by applying two-dimensional high-pass filters to the image which detect

the high frequency color changes. When the filters find a significant change, the program records a binary one at that location. The result, after edge detection, is a binary image with pixels where the colors changed dramatically (see Figure 1b). The next step is to determine the outermost points of the edges which define the physical edge of the puzzle piece.

To eliminate any missing pixels in the outer edge of the piece, morphing operations were applied to the image. This consisted of two-dimensional low pass filtering and finding and replacing specific patterns, such as gaps in the image. This results in a completely connected boundary with no inner lines. The final step was to use a recursive algorithm to fill in the inside of the boundary (see Figure 1c). This formatting process gives us the necessary information to begin solving the puzzle.

Attributes from Binary Images

Once the filled binary image is obtained, a number of attributes can be determined for each piece. First it is necessary to define each edge of a piece to be a tab, hole or border (assuming that the puzzle pieces are from a standard jigsaw puzzle, and not from one with unusual shapes.) The attributes are determined by looking at the edges of each side, and searching for certain patterns. A hole pattern is characterized by a number of zeros surrounded by ones. A tab pattern is characterized by a number of ones surrounded by zeros. To determine if a side is a border, it is fit to a line and the number of points outside the line is calculated. If this number is below a certain threshold, the side is determined to be a border.

Attributes from Wavelets

Wavelets give a different approach to finding edge attributes. Using wavelets, one can extract levels of detail from an image leaving an approximated coarser image. This image can easily be analyzed to find attributes.

Four 2-dimensional wavelet filters, which are combinations of low and high pass filters in the vertical and horizontal directions, are applied [5], [6]. After each filter, the results are decimated so that the images are exactly half the size of the original image. Thus, all four images together are the same size as the original. The low pass filter in both the vertical and horizontal planes results in a coarse image (see Figure 2), while all the other filters extract detail in their respective planes. This can be done multiple times recursively, applying the four filters to the coarse image each time.

To determine the edge types, we search for key patterns on the four sides of the coarse image. By applying pattern windows along the coarse edges of the piece, the probabilities of each edge attribute are

calculated. Pattern windows are slid through each position of the edge to be analyzed and the results are “or”ed together. For example, a tab pattern may be a zero-one-zero pattern. The resulting probability will be the likelihood that there is a tab, hole, or border on that edge. This is done for each pattern, and the pattern with the highest probability is chosen to be the type for that edge.



Figure 2: Level-3 Wavelet Reduced Image of Puzzle Piece from Figure 1.

4. LOGIC ENGINES

The next step in putting the puzzle together is to develop the logic engines. We are currently working on two different strategies, each of which is discussed below.

Expert System

An Expert System is a software program that addresses problems with similar intelligence to the way people do [7]. The program is based on a set of rules determined by experts in the given field of study. A control strategy is used to select which rules apply and to give an output. An Expert Puzzle System was developed based on a controlled study of humans putting puzzles together. These participants were observed while putting puzzles together and were asked to give feedback as to how they were putting the puzzle together. There were two main rules in the study. The participants were not allowed to see the entire image and the participants were not allowed to actually put puzzle pieces together. Thus they needed to devise a strategy to figure out if two pieces go together. To solve the puzzle the participants could lay the pieces out in the arrangement they thought was right. This was designed to emulate the constraints that the computer has. The computer does not have the entire image with which to compare. Furthermore, it is not desirable for the computer to physically put puzzle pieces together because of distortion introduced by the formatting process.

Color attributes were also determined to be used in the expert system. Since there are millions of colors available to be used in each piece, it was decided to limit the number of colors in a puzzle. Each puzzle was limited to 40-50 colors, and each individual piece was approximated using these colors. Next, a small band, slightly inside the edge of the puzzle piece, was analyzed and the six main colors were determined. This analysis was done for each of the four edges of each puzzle piece. The color analysis was performed in this way because the most important information for matching pieces is on the edge of the piece.

To simulate the experts, the system was designed to put together the border first. The basic strategy used by the Expert Puzzle System to put together the border is as follows: (1) Select a border piece at random. (2) Look at attributes for one side of the chosen piece. (3) Determine possible matches based on edge attributes. (4) Look at color information and determine if a decision can be made. (5) If a choice can be made, use new piece and go to step 2, else go to step 1.

Thus far, only small puzzles have been used. However, the results of the Expert Puzzle System have been promising. The expert system has a success rate ranging from 70% to 100%, depending on the puzzle used and the random choices made in the expert system.

Fuzzy Logic

Fuzzy logic provides a means to analyze and control mathematically challenging or ill posed systems. Fuzzy sets are created by assigning degrees of membership to set members. It is possible to be partially in more than one set. To form a hard decision, the member with the highest membership is selected.

To actually determine which puzzle pieces fit together, the edges of each piece are compared with the edges of other pieces. A fit would align the holes and tabs together with a good correlation. With wavelets, this is done first at very coarse levels of detail to form groups of pieces that look like good matches. This is done by sliding the edges of pieces over each other and looking for smooth continuity between the pieces. To check for continuity, the wavelet images are added together where the images overlap. The images are then normalized, so that if the images match well, the overlapping regions will contain ones. A fuzzy logic pattern is applied on the overlapping sections of the puzzle pieces that searches for this 2-dimensional block of ones. If the puzzle pieces fit, the result will be a high degree of membership for a fit between the two pieces.

The use of wavelet reduced images in this algorithm is key. Ideally, the entire algorithm could be done at a

highly reduced wavelet level. This provides fast computation because the images are smaller, and also much less dependent on the fine rotation differences introduced by scanning. Unfortunately, the pixelization associated with the wavelet reduction greatly reduces the detail level of the image, resulting in mismatched pieces. This can be compensated for by looking for matched groups at the lowest wavelet level, and then adding back a single level of detail to increase the resolution of the image and repeating the algorithm using only the matched group. This can be done repeatedly until the original level of resolution has been achieved, or a single match has been isolated. Alternatively, other algorithms can be used to isolate a match from a group, such as color matching, since this algorithm currently focuses only on shape.

5. CONCLUSIONS

This project is unique in many aspects. First, it is a combined effort of four master's level students each focusing on a certain part, but all working towards the common goal of the puzzle project. The best of a wide range of signal processing techniques are being applied to the project to offer a very high performance result.

There is still a lot of work to be done on this project. In the speech synthesis, a multi-pulse excitation model for the LPC algorithm is being studied to try to minimize the unnatural or buzzy characteristics often associated with standard LPC. For the word recognition system, Hidden Markov Models will be implemented, which will allow a larger vocabulary and yield greater accuracy.

For the Expert Puzzle System, larger puzzles will be attempted, and more intelligence will be added for better accuracy. In the Fuzzy Logic, an algorithm will be developed to allow the system to make decisions to solve the puzzle. These two logic engines are also going to be combined to create a more robust system.

6. REFERENCES

- [1] J. Taboada, S. Feijoo, R. Balsa and C. Hernandez, "Explicit Estimation of Speech Boundaries," *IEEE Proc-Sci. Meas. Technol.*, vol. 141, no. 3, 1994.
- [2] D. Danforth, "Quicky Recognizer Sketch," ftp://svr-ftp.eng.can.ac.uk/pub/comp.speech/info/DIY_SpeechRecognition, 1996.
- [3] R.M. Gray, "Vector Quantization," *IEEE ASSP Magazine*, April, 1984.
- [4] L. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*, Englewood Cliffs, NJ, Prentice Hall, 1993.
- [5] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley, MA, Wellesley-Cambridge Press, 1996.
- [6] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*, Englewood Cliffs, NJ, Prentice Hall, 1995.
- [7] S. Lindsay, *Practical Application of Expert Systems*, Wellesley, MA, QED Info. Sciences Inc., 1988.