IIIT-H

# BookAlive App

### Report

Rohit Girdhar
201001047

May 25, 2013

2

# Contents

# Introduction

BookAlive app is an educational mobile application, mainly targeted towards children to work as an aid in learning. It is a computer vision based application that the user can use to click an image of a page of a book to interact with it. The user will be able to interact with the image of the page using various interfaces like touch, speech etc.

## 1.1 Workflow of the application

The app is designed to work with specific books, that are BookAlive enabled. The data for each book would be packaged and distributed separately from the app.
The application starts to show a home screen. The user then fires up the camera application to click the image of a page of the book. The application will identify the page and its orientation, with respect to a stored representation of the page. After that, the user can interact with the image of the book.

### 1.1.1 Main Workflow

The user interaction begins with the system asking a question to the user corresponding to the page image clicked. The system takes the partial answers from the user, evaluates them and responds accordingly. The cycle continues till the user is able to answer the question.
After that, virtual hyperlinks appear on the page. These give more information of the objects. Such information may be further locked and require the user to complete some tasks to retrieve the information.
To make the experience more lively and exciting, we add light AR elements, like animations etc.
The application provides even more ways to interact with it in form of an Edit Mode. In this, the user can actually edit the entities in the clicked image of the book, like changing color of the shirt of a character, or even synthesize new objects.

# Main Features

The main features proposed for the application can be grouped into the following:

## 2.1 Questioning System

The questioning system is the first step of user interaction in the system. The questions can be of the various categories as described below.

### 2.1.1 Mapped to clicked image

The first category of questions include those that involve a mapping to the content of the image, along with the location of the content on the image.

**Technical Implementation**

The implementation of such questions requires knowledge of the homography between the clicked image of the page and its database image. The database image may not necessarily be an image, and can also be a set of feature vectors. Hence, we retrieve the clicked page using instance retrieval based on bag of words (details in demo section), and find the homography of the clicked image with the stored meta-image of the page. The interaction for the questions, (touch inputs and corresponding outputs use this information)

**Example Questions**

1. Locate an object in the image

   - Aim : To teach children to identify objects.
   - Task : To mark an object on the image. For e.g., locate the post box in the image. The user will have to draw a block around the required object in the image. The given coordinates by the user will be compared with the actual coordinates of the object in the image, and depending on the overlap, the response will vary. If the overlap is greater than some threshold, the system will accept the answer, else it will give the user hints on improving the answer.

- Method : The actual location of the object is stored in the system with coordinates on the original image, and the users coordinates are translated to the actual coordinates using the homography between the clicked and original image. The 2 polygons are then compared for the overlap.

2. Point the number of objects in the image

   - To teach student to identify objects and identify/search for occurrences of different types of same object
   - Task : The task in this question is to point out all occurrence of an object in the image, For eg. Point all the cars in the image. The user is then required to tap at every occurrence of a car in the image. The system responds with "CORRECT" if the tap was on a car image, else gives relevant suggestions. Once all the occurrences have been tapped, the system accepts the answer.
   - Method : Again the bounding box of the objects are stored on the system with the image, and taps by the user are translated to the taps on actual image using the homography. The system finds if the tap is inside the region or not. It accepts if inside, else gives relevant suggestions to the user. Once all the objects location have been tapped, the system accepts the answer.

3. Where a subject is happy/sad/angry

   - Aim : To teach children semantic concept of expression of feelings
   - Task : The task in this question is to identify the position of a specific character in the image, when the image clearly shows the character as angry/sad etc. Again the user will be required to draw a block about the object and the system will compute the overlap and accept/give suggestions accordingly.
   - Method : We store with every instance of the character, it's mood in that instance for comparison. Rest of the workflow is similar to above.

4. Map sound to the object image

   - Aim : To teach sounds of animals, and other objects (train, cars, etc)
   - Task : The task in this question is to match the sound made by an object (e.g., animals) to its image. The input for the user in this case will be an auditory signal. The user has to identify and mark over the object as in the Example 1. For e.g., <Sound of chirping> Which animal makes this sound ? The user must make a box about the image of birds in the image.
   - Method : This can be implemented by storing a small WAV file of the sound made by all the objects along with page information. The

WAVs can be reused over multiple pages, and can be stored for all varieties of objects (cars/trains/animals etc). The matching task is same as 1. The system can randomly pick any object from the page and play its sound, and ask the same question.

5. Where a conversation is angry/sad/happy

   - Aim : To understand semantics of conversations
   - Task : This question is more suited to comic books, where the user will be asked to point out conversations among characters that can be distinctly termed as angry/happy/sad etc. The user will be required to tap on the specific block of the comic with the conversation.
   - Method : Again, it can be implemented similar to the above by storing the conversation meta data along with each block of the page (if applicable).

## 2.1.2 Mapped to only content of clicked image

**Technical Implementation**

The implementation for this part is also similar to above, only it no longer requires the actual homography between the image and actual image, since we ask questions based only on the meta data associated with the specific page clicked. So, we can use the page recognition system, and load the metadata corresponding to that page.

**Example Questions**

1. Tell the count of some object (absolute/relative)

   - Aim : To teach counting, and comparisons
   - Task : The question will be framed like "Count the number of X" with a text input or "Compare number of X with Y" (less/more/equal).
   - Method : The counts of objects can be stored with the page meta data, and the question of above 2 types can be generated randomly for any/multiple objects

2. Tell what characters appear on that page (checkbox)

   - Aim : To enable memory based recognition
   - Task : The task in this problem is to recognize the characters on a given page, by ticking in a multiple options input form that appears. Based on user response, the system may give hints to recognize missed characters, and to remove incorrectly recognized characters.
   - Method : The characters on the image on page can be stored, and corresponding question can be generated after identifying the page.

3. Characteristics of a character based on the page

- Aim: To understand semantics of a character

- Task : To select the best matching characteristic of a character based on the page. For e.g., if a character is shown as angry on a given page, the question could ask about that character and give options to the user to chose from.

- Method : Similar to previous, we store the meta information about characters on each page.

### 2.1.3   Independent of the content of the image

- Arithmetic:

  Arithmetic could be taught to children using animations. For e.g., to teach addition, the system could ask for $5 + 4$ by showing apples, ask for an answer, and show an animation (combing the apples and assigning a number to each to get the final count) to explain the logic behind addition. Similarly for rest.

- Spelling

  Spelling can be taught by the system speaking out a word, and asking the user to write it out. The system will response with each character input, and guide the user to correct output. It could also give hints on breaking the word into syllables to guess its spelling etc.

- Vocabulary

  Vocabulary of the user can be improved by asking for words for a given situation or context. It can simply take input in as text, and give hints as the user gives his input. This could be displayed as a guess the characters game where some characters of the required word are known and the user has to guess the rest based on the context provided.

- Grammar

  Grammar could be taught by fill in the blank type questions where the user has to add vowels/adjectives etc.

- Traffic Rules

  This could be taught by developing a small game, where the user is a dot (maybe car/pedestrian), and has to go from one end to other following all traffic rules (the game canvas will have traffic lights, sign boards etc). The system will guide the user in case he breaks some rule.

## 2.2 INFO Mode

Once the user comes out of the question mode (after getting the answer accepted), he goes into INFO mode, where he can tap on any object to know about it. The aim of this mode is to encourage curiosity in the user, and encourage him to know about various objects he can see in the image. This can simply be done by tapping over the object, and an info snippet will appear. Also, some of the objects will be animated, and will do something when the user taps it (e.g.. a dog on being touched will bark, etc). (more in visual effects)

## 2.3 EDIT Mode

The edit mode is the most exciting part of the system. This mode will be available from the INFO mode. In this mode, the user can edit any of the characters on the book to be have characteristics of the user. These changes will persist for that application session. For e.g.

1. The user can replace the face of a character by his/her own face. The system will have a record of where all the character appears on every page and in what orientation. It'll click image of the user using the in built camera at multiple orientations, and replace the face at the location with the user's face.

2. Change Color of clothes etc: The user can customize the characters outfits by changing the color/type etc at one place and that will be affected at all places in that book.

3. Add new objects into the scene: The user can add more objects to the scene, that will be repeated across all images of that same location. for e.g., user can add a post box along a road in one snippet of a comic book image. The post box will be repeated across all images of the same place in the expected orientation. This will be implemented by user selecting an object from a predefined set, and adding. The system will have a map based model of the whole world in case of the book, and will have information of what snippets point to same place. So, those changes will be reflected at all such places.

## 2.4 Special Visual Effects

The application's main attraction is the Augmented reality applications. We can have multiple types of special effects:

- Static Animations : e.g., animated smoke coming out of cars, animations for fire, etc

- Moving animations : e.g., a car in a comic snippet actually moving away on being tapped (this can be implemented by playing a video, with the snippet as the first frame)

# Working Demo

A working demo for the app was developed for android. It had the following features implemented:

- Questions of type "mapped to image"

- Basic INFO mode

- Static special visual effects

### 3.0.1 Technical Details

- All meta information stored as XML files

- All audio/video/XML files stored separately from the app, so they can be distributed separate from the app.

### 3.0.2 Demo Video

Available here:
http://www.youtube.com/watch?v=p-LYdqaPaD8
http://www.youtube.com/watch?v=MulbfU1xEOQ