ARShop: A Cloud-based Augmented Reality System for Shopping

Chao Wang†, Yihao Feng§, Qi Guo†, Zhaoxian Li‡, Kexin Liu†, Zijian Tang‡, Anthony K. H. Tung†, Lifu Wu‡, and Yuxin Zheng†
†‡Interactive & Digital Media Institute, National University of Singapore
§Computer Science, Dartmouth College
{wangchao, qiguo, liukexin, atung, yuxin}@comp.nus.edu.sg†, yihao.feng.gr@dartmouth.edu§, {lizhaoxianfgg, tangzj77, wulifu20hao}@gmail.com‡

ABSTRACT

ARShop is a one-stop solution for shopping in the cyber-physical world with the help of crowd knowledge and augmented reality. Its ultimate goal is to improve customers' shopping experience. When a customer enters a physical shop and snaps a shot, the enriched cyber information of the surroundings will pop up and be augmented on the screen. ARShop can also be the customer's personal shopping assistant who can show routes to the shops that the customer is interested in. In addition, ARShop provides merchants with a web-based interface to manage their shops and promote their business to customers, and provides customers with an Android App to query using images.

1. INTRODUCTION

The field of Augmented Reality (AR) has existed for several decades. Since the concept was first formally defined in a survey [1], this field has grown rapidly. The basic goal of an AR system is to enhance the user's perception and interaction with the real world overlaid with computer generated virtual objects that appear to coexist in the same space. For example, a simple demonstration is to display a 3D model by scanning a QR code. The animated model stands on the surface of the QR code which seems like it exists in the real world.

While talking about the applications of AR in shopping, it is far from new. However, most of the applications focus on allowing customers to visualize and interact with products via online stores, or displaying some details of products when customers scan a QR code with a smart phone. There are few interactions among merchants and customers.

Our novel concept aims to improve users' shopping experience by integrating simple AR with images and annotations data, and to make people acquire and share information more intuitively from what they see in the real world. Before introducing our system, we describe an application scenario below. Consider a case where a tourist Chris visits a large shopping mall for the first time. To find out what each store sells and locates the commodity he wants to buy are difficult. Even though he finds the item in his shopping list, he might have little ideas about the user experience of the item or the service quality of the store. The goal of our AR system is to help customers like Chris by exploiting both the preprocessed data uploaded by the merchant and the data submitted by other customers. What Chris needs to do is to enter items in the shopping list (Figure 1b) of the ARShop App, takes a photo inside the mall, and the details of each store and information related to his shopping list will be shown. Figure 1a demonstrates how we display annotations on an image and the annotations may include store names, what they sell, and the reviews written by other customers. ARShop will provide route suggestion by showing the photos of shops that a customer is likely to pass on the way to the shops of interest sequentially.

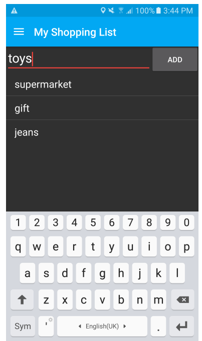
In this paper, we introduce ARShop[1,2], a cloud-based augmented reality system for shopping which aims to improve customers' shopping experience. When customers take a photo in a shopping mall and query our system, the related information on the images and surroundings will be displayed on the screen.

To do so, numerous functionalities need to be provided for both merchants and customers. For merchants and shopping mall owners, the functionalities to be provided include: (1) Shop Management: basic operations like creating shops by uploading images, deleting shops, and viewing different representations of shops. (2) Annotations Management: easily creating and deleting annotations for stores. Merchants are required to annotate a large amount of similar images to promote themselves to customers. ARShop has a feature called Annotation Transfer, where annotations can be transferred to similar images taken from different angles to save merchants' effort to create annotations. For customers, the functionalities to be provided include: (1) Query by Image: uploading an image to retrieve the relative annotations, which will be displayed on the screen. (2) Query by Location: viewing images of nearby shops and annotations. (3) Route Suggestion: generating a path using images of shops which leads to users' possible destinations, e.g., the shops selling items included in users' shopping list.

Proceedings of the VLDB Endowment, Vol. 10, No. 12

[1]ARShop Website: http://shopbyar.com/
[2]Demo Video: https://youtu.be/rzBmtTOy0Nw

(a) Information overlaid on a query image



(b) Shopping List

Figure 1: Annotations and Shopping List.

These functionalities cannot be easily achieved with trivial solutions. Most of them are difficult research problems related to the database area. First, Query by Image requires matching a customer's image with thousands of shops' images, which needs to be precise and conducted in real-time. In addition, the data provided includes not only images, but also the metadata of the phone when taking photos. The problem can be reduced to indexing and managing heterogeneous data, not to mention that image matching itself is a difficult problem in indexing and searching high-dimensional data. ARShop retrieves GPS and Wi-Fi metadata when taking photos. The metadata are used in building spatial clusters for indoor localization to improve the accuracy. Inside each spatial cluster, we further use visual features to build visual clusters. We refer this as two-level clustering. Using the two-level clustering, ARShop will conduct spatial feature matching first for the submitted query image and localize it among all the Wi-Fi cluster of images. Then the visual feature matching will be done only within the assigned cluster of images instead of all images of the model to speed up the query processing.

In addition, ARShop can suggest routes leading to the shops that a customer is interested in based on his/her shopping list. This is actually a keyword search problem. By matching the content in a customer's shopping list, we can know the shops of his/her interest. Then we use the two-level clusters to infer the shops' locations and retrieve the relevant images along the way. From the relevant images, we further select the representative images as references. We model this problem as a reachability problem, in which we select images whose maximum pair-wise distance is minimized.

Contributions — The contributions of this work include:
1. A demonstration system which provides necessary functionalities for both the merchants and customers. Our system improves the shopping experience, helps new customers to find the commodity they want to buy easily, and enable sharing of information like reviews. The novel solutions we provide make the management of shops and annotations data easier, and make the image query process real-time and precise. Besides, the indoor localization and route suggestion by images is worth highlighting.
2. A useful AR system which has the potential to be used in solving numerous similar problems. For example, the art galleries and museums normally provide limited information for the artworks. By using ARShop, users only need to take a photo and the details will be shown immediately. This application scenario has the potential to work with or replace the current audio tour system in the museums and galleries.

2. ARCHITECTURE AND INTERFACES

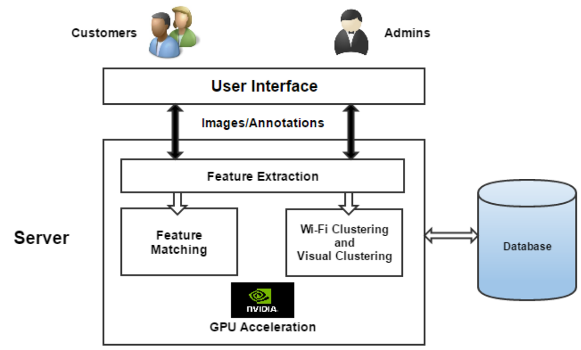


Figure 2: Overview of system architecture.

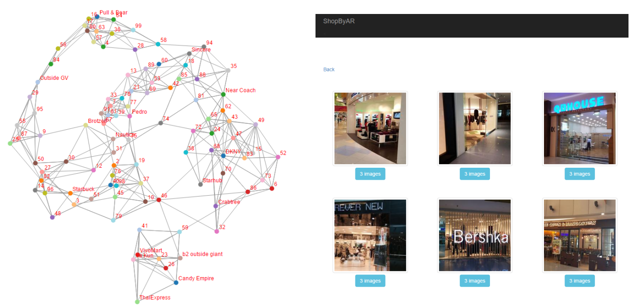
A high-level view of our system's architecture is shown in Fig. 2. The system has two frontend interfaces receiving input data. One is the web interface for admins or merchants to create a shopping mall model by uploading images and corresponding metadata. The other is an Android App which is used by customers to submit queries. In order to handle customers' queries, the procedure mainly consists of three steps, i.e., model building, feature matching and annotation generation. The first step is offline while the last two steps are online. During model building, admins need to submit images and metadata taken using our Android App through the web interface. When a customer takes a photo using the app and submits to the server, the spatial features and visual features will be extracted from the metadata and image. Once the feature matching is done, the annotations from the models and images uploaded by other customers will be transferred to the query image and displayed on the screen. In what follows, we will introduce the workflow of each step. The technical details of each step will be discussed in Section 3.

2.1 Web End

Mall Model Building: The interfaces of model building are shown in Figure 3. Model building is the primary step required to be done by admins or shopping mall owners. First, they need to take photos using our app as it will record metadata including GPS and Wi-Fi signal. The photos need to be taken from various angles in order to make the feature matching more precise. Then merchants need to upload them through the web interface. The model building procedure includes a Two-level Clustering which will be discussed in Section 3.1 in more details. The clustering results will be able to be viewed through the web interface.

2.2 Client End

The Android App allows users to query the system using images. First, users can register an account or login using Facebook account. After logging in successfully, ARShop will retrieve the GPS information and detect the nearest shopping malls. Customers are able to take photos of



(a) Wi-Fi Clustering

(b) visual Clustering

Figure 3: Web UI for Admins.

the stores inside the mall and view returned annotations by interactive exploration such as dragging and zooming or add their annotations or reviews. The annotations added by the customers are also available to be viewed by other customers. Hence the sharing of information and shopping experience are improved. In addition to image query and annotation creation, our system also supports functionalities like: (1) Nearby Shops: viewing images and annotations of nearby shops. (2) Shopping List: notifications of nearby shops that selling the items in customers' shopping list will be popped up.

Route Suggestion: By matching a user's shopping list with the annotations created by the merchants, we can get photos or shops of the user's interest. A natural question is asked by the user: how can I go to those shops?

With the query image and its metadata, we can infer the user's location using the pre-generated spatial clusters. With the user's interested photos, we can also get the location of the user's destination. ARShop can suggest the route to the shops of interest by showing the user the shops that he/she is likely to pass along the way. More details of route suggestion will be explained later in Section 3.3.

3. TECHNOLOGY HIGHLIGHTS

In this section, we will briefly introduce the technical highlights in our system including Two-level Clustering, Annotation Transfer, Route Suggestion, and GPU Acceleration.

3.1 Two-level Clustering

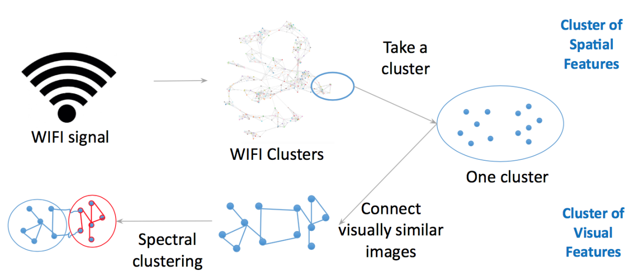


Figure 4: Two-level Clustering.

First, we introduce an innovative two-leveling clustering which considers both spatial features and visual features (Figure 4). The two-level clustering aims to conduct indoor localization and speed up the image query processing and feature matching.

Wi-Fi Clustering — We also call it spatial clustering which clusters all images based on spatial features extracted from the metadata. The features we consider include the Wi-Fi scan list, BSSID (MAC address), signal strength, and frequency. By using the ItCompress technique [2], the images are clustered into different groups based on their spatial features. This is the first level of our image query system and is conducted offline when the shopping mall owners create the mall by uploading images. The Wi-Fi cluster graph is able to be viewed and manipulated by the admins through our web interface. For example, admins can select an area of clusters and view the shop images within them. The admins can also create and change names of each cluster.

Visual Clustering — In order to further cluster images within each Wi-Fi cluster, we use the scale-invariant feature transform (SIFT) algorithm [3] to detect and extract visual features from all images. We use SIFT rather than other algorithms because it is more reliable when performing matching of images taken from different angles of an object or scene, which is necessary for our application scenario. The kNN matcher is used to calculate the distance between each pair of key points. From that, a distance matrix among images is generated and the hierarchical spectral clustering algorithm is applied to the matrix. Thus, images within each Wi-Fi cluster are further clustered according to the visual features. Our web interface provides the functionality to view the clustering result as well. The visual clustering not only expedites query by image, but also works in Annotation Transfer.

When customers submit an image, the spatial features are extracted from the metadata and the feature matching will be done prior to the extraction of visual features. Only the best matching cluster of images will become the candidates for visual feature matching. Therefore, the Two-level Clustering makes the image matching and retrieval more efficient.

3.2 Annotation Transfer

Annotation Transfer is a useful functionality provided for the shopping mall admins to manage the annotations with automatic transfer and manual adjustment. Consider the scenario that the admins need to add annotations for a brand new item. As the image dataset may contain a large amount of similar images taken from different angles and in order to make the customers get the best query results, the admins have to add annotations for all images contain the item one by one. This is time-consuming apparently. Our innovative solution which relies on the results from Two-level Clustering solves this problem. First, the admins need to select a rectangular area on the most representative image which contains the item. Then a two-level kNN matching will be conducted on images within the same visual cluster as the chosen image. This saves a considerable amount of time and makes sense because the images contain the item must be similar to each other and in a same visual cluster. The two-level kNN matching consists of kNN SIFT feature matching followed by kNN image matching. Top k similar images will be returned and further filtered by finding homography. Based on the coordinates of the rectangle area drawn by admins, the coordinates are transformed perspectively to the

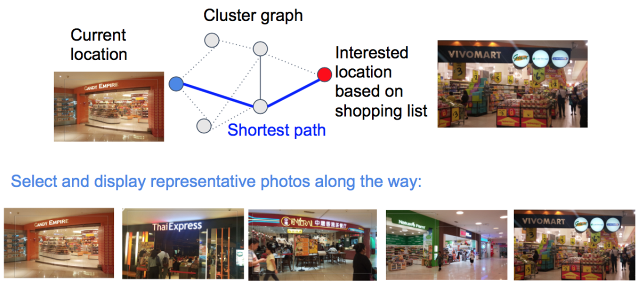


Figure 5: Route Suggestion.

candidate images. If transformed coordinates are not inside the image, this candidate will be removed. The final results will be shown to the admins as a pop-up window on web interface. Besides, the candidate images can be viewed and the annotation area can be adjusted or removed. With Annotation Transfer, admins can create, remove, and update a batch of same annotations easily.

3.3 Route Suggestion

Figure 5 shows the procedure of route suggestion. It contains two steps: (1) We find the shortest path between the starting point (the query location) and the destination (the shop of interest) in the Wi-Fi cluster graph. (2) After we find the shortest path, we select and display representative photos from the nodes in the shortest path.

Step (1) is straightforward. We can simply use the Dijkstra's algorithm to find the shortest path. Next, we discuss Step 2 in details. The intuition for the photo selection is that (1) the selected photos are not from the same shops, and (2) the selected photos should not be totally distinctive to avoid showing photos that are too far way to each other. For example, an ideal result is that the adjacent images on the path have common features and these two shops are physical adjacent with each other. In order to solve this problem, we can reduce it to a graph reachability problem. First, we can build a similarity graph of images. Given a distance threshold θ, a photo can reach another photo if their distance is within θ. Suppose k photos are to be returned, we select k photos satisfying (1) the starting point can reach the destination via these k nodes (2) the maximum pairwise distance is minimized. By doing so, we choose and show these k representative photos that the user is likely to pass on the way to their destination. To speed-up route finding and prevent generation of similar images, edge contraction can be done to eliminate duplicate images from the same shop. The distance of edges smaller than ϵ is considered as same and will be contracted.

3.4 GPU Acceleration

Creating the shopping mall model is computationally expensive because of the extensive calculation in SIFT feature extraction and kNN matching in finding similar images. For example, one of our shopping mall datasets contains around 1500 images and the computational cost is over 18 hours using CPU. It will be more costly for a larger shopping mall. Thus, GPU optimization is necessary. We incorporate a SIFT implementation using CUDA for GPUs and OpenCV GPU kNN matcher. The GPU model we use is NVIDIA GeForce GTX TITAN X. After optimization, a 40x speedup is achieved compared to the CPU version.

4. DEMO SCENARIO

In this section, we are going to demonstrate two application scenarios, i.e., Shopping Mall Scenario and Gallery Scenario. The shopping mall scenario aims to allow the conference attendees to interact with our system as admins or shopping mall owners and to create models using our web interface. The gallery scenario aims to demonstrate how our app displays annotations using query by images.

4.1 Shopping Mall Scenario

Our basic application scenario is for shopping malls. Prior to the conference, we will gather data inside the conference venue (consider the university as a shopping mall). The dataset which contains photos along with the metadata files will be provided to the attendees. They can use our web interface to create a "mall" model by uploading the dataset to our server and view the clustering results. Besides, the attendees can create annotations using Annotation Transfer. If the attendees are interested, they can download and install our Android app. With the model built previously, our app can be a guide who shows attendees around the campus.

4.2 Gallery Scenario

The novelty of our system is that it can be used in various settings (e.g. Art Gallery). For example, in Singapore Art Gallery, there does not have detailed descriptions for the artworks. For tourists, especially those who are interested in the artworks may want to know more details. Hence, the admins of the gallery can build a model using our system and add descriptions for each artwork. When visitors use our app, they can know more by just snapping a shot. Thus, in order to make conference attendees understand the application scenario, we will build the model for some simple paintings and print out the paintings so that the attendees are able to use our app to take a photo, submit a query and view the details of the painting.

5. ACKNOWLEDGEMENTS

The work by Anthony K. H. Tung was partially supported by NUS FRC Grant R-252-000-370-112 and NRF2016NRF-NSFC001-113. This research was carried out at the NUS-ZJU Sensor-Enhanced social Media (SeSaMe) Centre. It is supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the Interactive Digital Media Programme Office.

6. REFERENCES

[1] R. T. Azuma. A survey of augmented reality. *Presence: Teleoperators and virtual environments*, 6(4):355–385, 1997.

[2] H. Jagadish, R. T. Ng, B. C. Ooi, and A. K. Tung. Itcompress: An iterative semantic compression algorithm. In *Data Engineering, 2004. Proceedings. 20th International Conference on*, pages 646–657. IEEE, 2004.

[3] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.