TagPix: Automatic Real-time Landscape Photo Tagging For Smartphones

Hillol Debnath and Cristian Borcea
Department of Computer Science
New Jersey Institute of Technology
University Heights, Newark, NJ 07102, USA
Email: hd43@njit.edu and borcea@njit.edu

Abstract—This paper presents TagPix, a lightweight smartphone photo tagging app that provides good tagging accuracy, works in real-time, and protects user privacy. The main novelty of TagPix consists in leveraging the phone sensors and a place-tag database to tag landscape photos which include landmarks. GPS location is used to identify landmark tags in a given region. Then, TagPix computes the angular distance for the object in the camera focus using the orientation sensors. This allows the app to select a small subset of landmark tags for the photo. For further tag accuracy, we devised new usable methods to estimate the Euclidean distance between the user and the landmark in the camera focus. These methods employ simple user actions and lightweight trigonometric calculations. TagPix is implemented and tested using several Android phones and Google Places API. The app was tested in 8 cities across USA. Using only angular distance, TagPix achieves 86% tagging accuracy. Adding Euclidean distance estimation leads to 93% accuracy.

I. Introduction

John visited Paris during last Christmas, and he captured several thousand photos of different places, statues, museums etc. When he went back home after the tour, he found that he has lots of good memories to share with his friends on Facebook. He started to upload the tour photos, but soon realized that he cannot name many places in these photos. He has completely forgotten the name of some, and he did not know the name of others in the first place. Even for the places that he remembered, he was getting tired to tag the photos manually. He wished there was an app around that would tag the photos automatically.

The necessity of an automatic photo tagging tool has long been felt, but it has become more acute in the last few years with the widespread availability of smartphones and tablets equipped with cameras. There are more than one billion smartphones and tablets in the world [1] today, and they are used to generate a huge number of photos. Many of these photos are tagged and shared with our social networks. For example, 300 million photos are uploaded daily on Facebook [2]. Tagging all these photos manually is becoming almost impossible.

We have developed TagPix, a smartphone app that can automatically tag a significant number of photos: landscape photos containing landmarks. This app is very useful especially for tourists and for people who share their photos using social networking applications. We had four goals when designing this app:

High accuracy automatic tagging: We aim to tag each photo with just one tag: the name of the landmark in the camera focus. If there are multiple landmarks in a photo, the user may configure the app to include them. In most situations, the tagging should require no user actions.

User privacy protection: The app must not upload the photos to Internet tagging services such as Google Goggles. Thus, all tagging processing should be done locally on the phone.

Real-time tagging: To help users who do not know the name of the landmark they just photographed, the app should display the tag as soon as a photo is taken. Therefore, the processing should be fast.

Modest phone resource usage: The app should be lightweight and consume a modest amount of resources, with battery power and network bandwidth being the main concerns.

To the best of our knowledge, currently, there is no smartphone app that can satisfy all these goals. Most of the existing automatic tagging systems are content-based, pattern matching, and classification-based systems [3], [4]. These systems either work on servers (do not protect user privacy, and may not be able to provide real-time tags) or they consume a lot of CPU and energy if ported on the phones. Furthermore, they demand photos of moderately good quality (i.e., do not work well for night photos) and taken from specific angles. Finally, they require extensive training. Unlike these systems, TagPix satisfies all four goals by relying solely on phone sensors, a tag-place database, and lightweight processing on the phone. It also works with photos of any quality and does not require training.

The main novelties of TagPix are: (1) An algorithm that leverages the GPS and the orientation sensors on the phone to compute the angular distance between the object in the camera focus and the landmarks in the camera's angle of view, which then allows TagPix to accurately select the tag(s) for the object; and (2) Three usable methods, based on simple user actions and trigonometric calculations, to estimate the Euclidean distance between the user and the object in the camera focus, which allow TagPix to improve its tag selection accuracy.

To summarize, the contributions of this paper are:

- We designed a new real-time, privacy preserving photo

tagging application, which works locally on the phones and consumes little resources (e.g., battery). TagPix provides high tagging accuracy without requiring any any previous training or indexing. Since it does not use computer vision techniques, it works well independent of the photo quality.

- We devised three new and usable methods for estimating for estimating the Euclidean distance between the user and the landmark to improve tagging accuracy and remove false positive tags. Estimating this distance is difficult because phone cameras have digital zoom which cannot be used in classic optical physics formulas. Our methods require minimal effort from users (e.g., move a few steps and point the camera to the object again) and employ lightweight trigonometric calculations.
- We developed a lightweight, low energy consuming prototype for Android. The prototype works on several types of Android-based phones and uses the Google Places API [17] to retrieve landmarks in the proximity of the user. This prototype was successfully tested with 6 users across 8 cities in the USA. Only using tagging based on angular distance calculations (i.e., the best method in terms of usability), TagPix achieves an average accuracy of 86%. After applying the Euclidean distance estimation in conjunction with angular distance, the accuracy increased to 93%.

The rest of this paper is organized as follows. Section II presents an overview of TagPix. Section III discusses the related work. Section IV describes the design and implementation of TagPix. Section V presents the evaluation results. The paper concludes in Section VI.

II. TAGPIX OVERVIEW

To illustrate how TagPix works, let's consider the scenario from Figure 1, in which the user zooms in and takes a photo of Tour Eiffel. TagPix's goal is to determine one tag that best represents the photo. We assume a place-tag database stored on the phone; alternately, this database can be accessed over the Internet. This database provides landmark tags and their associated locations.

First, TagPix reads the GPS location of the user and retrieves the nearby landmark tags. If it just considers the distance between these landmarks and the user, the app would end up with the wrong tag (i.e., Trocadero which is the closest, but it's behind the user). Therefore, TagPix uses the orientation sensors on the phone to determine the angle of view for the photo (only the vertical angle of view, α_V, is shown in the figure). This operation helps to eliminate the Trocadero and Pont d'Iena tags; the first is at the back of the user, and the second is below the angle of view as the user is on a hill and zooms in the Eiffel Tour.

Next, we would like to compute the distance to the object in the camera focus and select the closest tag. Unfortunately, distance estimation based on standard optics formulas can be done only for objects within a few feet of the camera. This is because the phones come with cameras and lenses that have
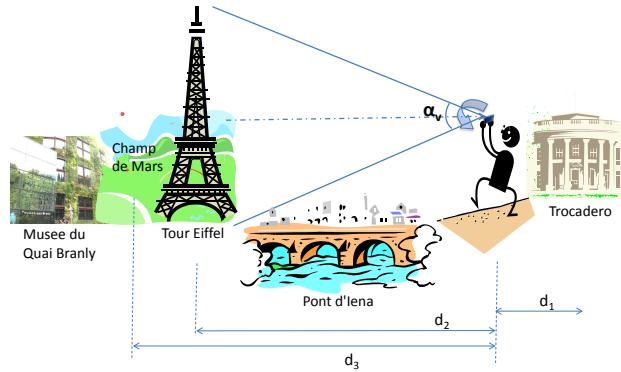

Fig. 1: How to Select the Correct Tag for a Photo?

minimal ability in sensor size and focal length (e.g., the digital zoom value is unusable in optics formulas, and the lens has a fixed and small focal length). Instead of Euclidean distance, TagPix computes the angular distance between the direction of the camera focus and each of the remaining tag-identified landmarks. This eliminates Musee du Quai Branly, and keeps Tour Eiffel and Champ de Mars which both have 0 angular distance in this example. In our basic solution, TagPix chooses Tour Eiffel because it's closer to the user position. In many situations this method provides very good accuracy, but it's still possible to tag incorrectly in a few situations (e.g., the landmark with the lowest angular distance is not the one in focus).

However, if the user is willing to do a bit of extra work, TagPix is able to estimate the distance to the object in focus with good accuracy and solve this problem. We devised two methods to do it: in one, the user is required to first point to the base of the landmark being photographed, and then TagPix uses simple trigonometric formulas to compute the distance. In the other, the user is required to take two photos of the landmark from two positions, which can be very close to each other (e.g., 10 feet). With these methods, we can guarantee the correct tag in our example, e.g., Tour Eiffel.

As a final observation, let's note that determining the tag through this distance estimation method can work with or without the angular distance. However, by combining both of them TagPix achieves the highest accuracy. For example, two tags could be at the same Euclidean distance, but one could be eliminated using the angular distance.

III. RELATED WORK

Besides the computer vision systems mentioned in the introduction, geotagging for multimedia content has been proposed in the literature as well [8]. Most of these projects are based on 3D modeling and object matching. Similarly, the work in [9] combines the 2D appearance of landmarks in photos with 3D geometric constraints to extract scene summaries and construct 3D models. This system uses "iconic scene graphs for clustering based on geometric constraints, and then uses structure from motion techniques to generate the 3D model of

the landmark. Thus, they are similar in nature to the computer vision techniques already discussed, which are not suitable for deployment on phones. TagPix provides a lightweight, real-time, and privacy preserving alternative.

A number of other projects have addressed photo tagging or used techniques similar to those in TagPix. In [5], the system expects the user to tag a set of photos manually, and then tries to correlate the other photos with this set using the time and location of the photos. It suggests existing tags based on these correlations. This system is lightweight, but it has limitations: it requires significant actions from users, and may have low accuracy in places with many landmarks. TagPix is able to overcome these limitations.

TagSense [6] is a smartphone automatic tagging system, which explored the opportunity of using the phone sensors to get a clue about what is being shot. The accelerometer, GPS, gyroscope, light sensor, and microphone are used to infer the context. The phones also collaborate to learn who may be in the captured photo. The main focus was to tag people and include keywords about context, such as "outdoor", "noisy", or "inside XYZ museum". While this system proposed a novel way of providing tags automatically based on sensor data collected from the user's phone, it still differs from TagPix in a very important aspect. This system did not try to identify the landmarks in the camera focus, and implicitly did not include a method to do so. As we explained already, this is not a straightforward problem: the landmarks in the photos may not be nearby, and selecting the most appropriate tag is difficult.

Argon [11] and Wikitude [12] are two augmented reality browsers which overlay tags for nearby landmarks or places of interest over the live camera screen. The main similarity with TagPix is that they use a place-tag database to find nearby tags. Unlike TagPix, they do not attempt to place the tags precisely over the landmarks they represent. They place info bubbles on the camera screen to let the user know what landmarks are nearby. Furthermore, Argon does not overlay tags for farther away landmarks.

A location-driven tag suggestion system is proposed in [7]. This system use sources such as a public Geographic Names Information System (GNIS) database, community tags from Flickr pictures, and personal tags shared through users' photo collections. Bags of place-name tags are first retrieved, clustered, and then re-ranked using spatial distance criteria. The community tags from photos taken in the vicinity of the input geotagged photos are ranked according to the distance and visual similarity to the input photo. Compared with this system, TagPix achieves much higher tagging accuracy because it leverages the orientation sensors to eliminate the tags that are not in the camera's angle of view. By using the Euclidean distance estimation, TagPix is able to further improve the tagging accuracy.

Similar with this system, TagPix could use multiple tag-place databases, especially community and personal tags. For example, folksonomies such as the one presented in the MarkIt game [10] could be added to TagPix. In the literature, there are also tagging systems based on visual folksonomies, generate tags for new visually similar photos. Such systems utilize collaboratively annotated image databases, and then analyze the images to find visually similar photos and propagate the tags for them. TagPix is different in nature from this system because it works on the phone and requires no (or minimal) user actions.

The work in [13] focused on accurate localization of distant objects, and used similar methods with TagPix for distance estimation. The user needs to focus the target object in the camera viewfinder and then take multiple photos from different nearby positions. The distance is then estimated based on 3D models of the captured photos. The major difference between this work and TagPix is in terms of usability. TagPix provides high tagging accuracy even without asking the users to move or take multiple photos. Furthermore, even when it does ask the users to move, it is only for a short distance and one additional photo as opposed to larger distances and many photos.

SmartMeasure [14] is a popular app in Google Play, which uses the camera to estimate the distance to a target object. It first asks the user to input the estimated height of the object. Then, it uses a simple trigonometric formula to estimate the distance. This method has some similarities with the methods used by TagPix to estimate the distance to an object. However, we do not require the user to input any value, which ultimately improves the usability and could potentially avoid errors due to wrong user input.

Google Goggles [15] is a visual search tool which require the user to upload the image to a server, which then returns relevant information. The servers supporting Goggles are trained with more than a billion photos [16]. While this system works relatively well for images containing text or logos, it does not work well with landmarks missing such features. It also needs very specific levels of light and resolution to work well. We tested it, and it failed to identify all the landmarks that TagPix tagged correctly by using the phone sensors and our algorithm. Unlike Goggles, TagPix works independent of the photo quality and protects the user privacy because photos are not uploaded to a server to be tagged.

IV. DESIGN AND IMPLEMENTATION

Figure 2 shows the system architecture for TagPix, which consists of three layers. The first is the Sensors layer, which is responsible for managing the sensors, e.g., when to turn them on and off, how to collect data from them, etc. The second is the Data Aggregator and Manager layer, which is responsible to do pre-processing and adjustment work (e.g., angular adjustment for magnetic declination) as well as merging operations on the collected sensor data. The third layer is the Tag Suggestion Generator, which is responsible for tagging decisions and providing suggestions to users. In other words, it computes the distances (angular and Euclidean) between the tagged landmarks and the object in the photo, and then selects the best matched tag (or tags if the user configures the app for multiple tags).
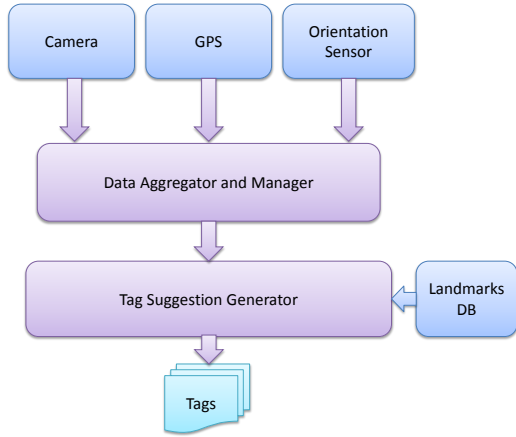
Fig. 2: TagPix System Architecture

Algorithm 1 TagPix Pseudo-code

1: gps = getGPSData()
2: landmarkList = callGoogleLocationAPI(gps, tagsNumber)
3: orientData = getOrientationData()
4: cameraState = getCameraInfo()
5: aov = angleOfView(cameraState)
6: coordinateRange = calculateTargetedArea(aov)
7: tags = angularDistFilter(landmarkList, coordinateRange, cameraState, orientData, AngThreshold)
8: **if** useEuclideanDist == true **then**
9:     angles = getUserLandmarkAngles()
10:     euclDistance = getEuclideanDist(angles, UserHeight, UserMovementDistance)
11:     tags = euclideanDistFilter(tags, euclDistance)
12: **end if**
13: **return** tags

Algorithm 1 presents the pseudo-code for TagPix. In the following, we describe each of the instructions in the pseudo-code based on our Android implementation; we used Android 2.3.3 (Gingerbread), but TagPix works on newer Android versions as well.

A. Acquiring Location Data

Once the user pushes the camera button, TagPix queries Android for the best location provider based on the accuracy of existing sensor hardware, environment, energy level, etc. We find that most of the time it provides GPS as the best provider. But, in case the GPS signal is not available with a decent accuracy or the battery level is too low to access the GPS sensor, Android provides other location sensing options such as WiFi, 3G, etc. TagPix registers a location listener to that location provider, listens for a "location change" event, and requests updates for a change in location of at least 1 meter. Whenever the "location change" event fires, TagPix fetches the longitude, latitude, and altitude for the current location. The location API also provides a way to cache the previously known location data in case it is unable to resolve the current location. As long as the camera is open, the location provider is updated once the user moves 20m from the previous place.

B. Fetching the Landmark List

Once the current location data is acquired, TagPix inquires the place-tag database about the nearby landmarks list. TagPix uses Google Places API [17] to retrieve tags associated with places and landmarks. In this way, TagPix is able to access a relatively up-to-date place-tag database. While TagPix sends the user location to the database, it does not send the photos; thus, it protects user privacy to a significant extent and reduces the bandwidth usage. To avoid sending the user location to the server, TagPix could store such a database on the phone and update it periodically. For example, the GeoNames project [18] provides a database of 8 million named places from all over the world and is available for off-line use.

Resolving the location from GPS and then acquiring the places list by calling the Google API is a fairly slow process compared to the other calculations, which may degrade the user experience. Therefore, we create an asynchronous task to invoke the Google Places API before TagPix starts any other computation. Thus, the places list is fetched in the background while TagPix calculates the phone orientation.

Since it is not clear what radius value to set for "nearby", we choose to ask for a fixed number of tags ordered by proximity to user location. The call returns a JSON array containing the information of interest (i.e., name/tag and location) about nearby landmarks and popular places. The JSON array is then converted to Java objects to be used in the app.
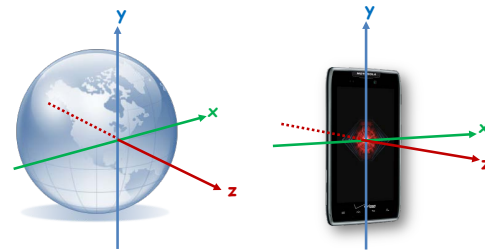
C. Calculating the Orientation

Fig. 3: The World Geomagnetic Coordinate System vs. The Phones Coordinate System

To determine the phone orientation, TagPix uses the software-based orientation sensor in Android, which derives its data from the accelerometer and the geomagnetic field sensor. It provides geomagnetic field strength values for each of the three coordinate axes. TagPix calculates the rotation and inclination matrices that allow us to compute the rotation around the three axes. However, there is one additional step before this computation: the app has to align the phone's coordinate system to the World Coordinate System (as shown in Figure 3). In the phone coordinate system, X and Y axes are tangents to the ground location of the user and Z is perpendicular on X-Y plane. TagPix uses the two matrices to find the azimuth angle (angle around the Z axis) and the tilt angle which are used to filter landmarks in a later stage.
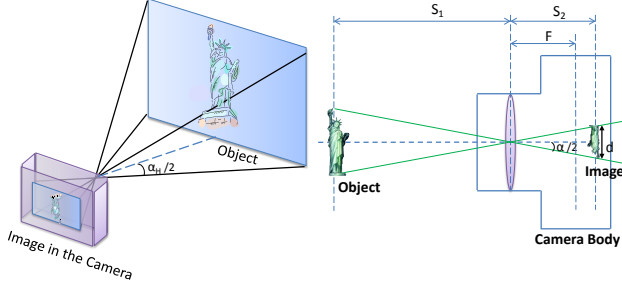
Fig. 4: Angle of View

D. Calculating the Angle of View

Calculating the angle of view (AoV) α is a standard procedure used in Optics and Photography. As shown in Figure 4, the AoV is defined as the angular extent of a given scene that is imaged by a camera and is computed using the following equation:

$$\tan \frac{\alpha}{2} = \frac{\frac{d}{2}}{S_2}; \tag{1}$$

For a well-focused photo, S_2 equals the focal length of the lens (F). When calculating the horizontal AoV (α_H in the figure), d is the camera sensor frame width (w); when calculating the vertical AoV, d is the camera sensor frame height (h). These sensor values are retrieved from the Android Sensor API. For example, the horizontal AoV will be:
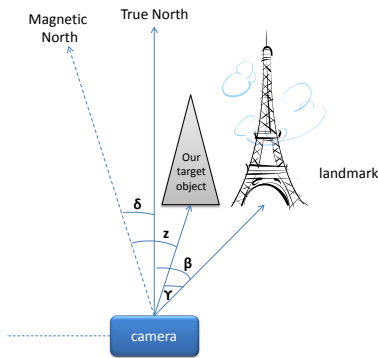
$$\alpha_H = 2 \arctan \frac{w}{2F}; \tag{2}$$

Fig. 5: Angular Distance

E. Calculating the Angular Distance and Selecting Tags

Using the AoV and the orientation angles, TagPix computes the angular distance between the object in focus and the retrieved landmarks. This operation allows TagPix to filter out many tags which are not located within the region targeted by the camera. Then, TagPix orders the remaining tags as a

function of their angular distance (i.e., the closest to the object in focus is the best).

Figure 5 shows the angles used to calculate the angular distance γ. The equation is:

$$\gamma = \beta - z + \delta \tag{3}$$

The angle z is the azimuth angle calculated from the orientation data. The angle β is the GPS bearing between two coordinates, which is defined as the horizontal direction of travel of the phone from the first coordinate to the second one. We can consider the bearing as the angle created by the connecting line of the two coordinates and the geographic Y axis. The GPS bearing is calculated using true North Pole as reference. However, for Android devices, the orientation sensor works with the magnetic north pole as a reference.

The difference among the true north and the magnetic north is known as the magnetic declination, δ, and varies from place to place; it also varies with time. The National GeoPhysical Data Center [19] provides a calculator to know the exact value of δ for a particular place. Android's SDK offers a specific class, GeoMagneticField, which provides utility methods to adjust the magnetic declination with orientation sensor data.

In the ideal case γ is 0 (i.e., our targeted object is aligned with the landmark). However, in many situations it is difficult to achieve such accuracy. Therefore, we use a threshold angle value θ to select the most appropriate tags: all the landmarks which have an angular distance $\gamma < \theta$ are returned as the suggested tag list. Keeping the threshold angle very low would return only the most accurate tags.

F. Estimating Euclidean Distance

As we will show in Section V, using only angular distance produces good results in practice with the major advantage of not requiring any action from the users. However, if the user is willing to do some simple actions, TagPix can further improve the tagging accuracy by estimating the Euclidean distance to the targeted object. For example, if two landmarks A and B fall within the angle θ and both have the same angular distance, the estimated Euclidean distance can be used to choose one of them (i.e., the one closer to the targeted object).

We devised three methods for estimating the distance: one requires the user to just point to the base of the landmark after taking the photo; the other two require the user to move short distances and take two photos. The first method is more convenient for users, but it works well only for objects located in the immediate vicinity of the user. At the cost of a slight inconvenience, the other methods achieve better accuracy for objects located farther away.

1) Method 1: This method does not require the user to move or take multiple photos. All it needs is user's height (which is inputted by the user when the app is configured) and the camera to be aimed at the base of the landmark after taking the photo. In this way, TagPix obtains the angle x (as shown in Figure 6) from the orientation sensor data, and then it computes the distance:
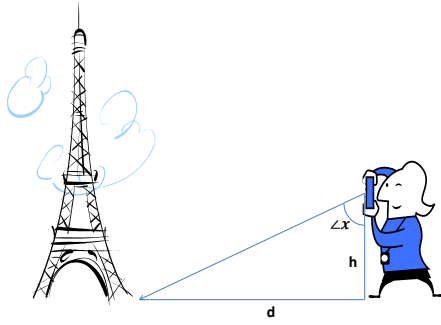
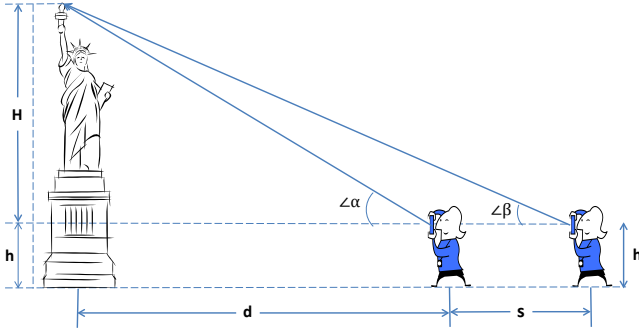Fig. 6: Method 1 to Estimate the Object Euclidean Distance

Fig. 7: Method 2 to Estimate the Object Euclidean Distance

Fig. 8: Method 3 to Estimate the Object Euclidean Distance

$$d = h * tan x \tag{4}$$

2) Method 2: This method needs two photos taken from two different locations, as shown in Figure 7. First, the user needs to aim the camera at the top of the landmark, and TagPix measures the angle of inclination β from the orientation sensor data. Then the user needs to move forward for a short distance (s) and repeat the procedure; TagPix measures the angle α. The longer the distance, the higher the accuracy of the method will be. From [20], we know that there is a high correlation between step length and height of a person. As we already know the height of the user, the app will just ask the user to move a fixed number of steps forward (e.g., 10). The distance is then calculated using this equation:

$$d = \frac{s * \tan \beta}{\tan \alpha - \tan \beta} \tag{5}$$

3) Method 3: This method can be used instead of method 2 if the user cannot move forward (e.g., she would end up in road traffic). In this method, the user moves sideways, and TagPix measures the angles α and β as illustrated in Figure 8. The distance is calculated as follows:
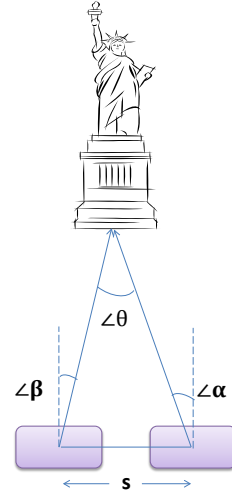
$$d = \frac{s}{\tan \alpha + \tan \beta} \tag{6}$$

We found that measuring α and β accurately is problematic due to the practical limitation of GPS and orientation sensor. A better solution is to calculate the angle created at the target by s (let us call it θ, measured in radians) from the two bearing values corresponding to the two locations of the user, which is comparatively more accurate. The user only needs to move about 10 meters to achieve good accuracy. The distance is then computed as follows:

$$d = \frac{s}{\theta} \tag{7}$$

We have tested all three methods by measuring the distance of real world objects. Method 1 has the lowest accuracy among them (about 20% average error), and is usable only for distances less than 40 meters. Method 1 also has problems with landmarks situated in elevated places. For method 2, the average error is 15%. For method 3, it is 14%. Both these methods maintain this level of accuracy for any practical distance measurement. Finally, methods 2 and 3 can adjust their calculations for elevated landmarks if we know the elevation of the landmark.

V. EVALUATION

TagPix was tested with 6 users and evaluated for many places and landmarks in 8 cities in USA: New York City; Newark, Harrison, Kearny, Jersey City, Morristown, and Parsippany-Troy Hills in New Jersey; and Columbus in Ohio. For ground truth, we used user's feedback in real time to verify the generated tags. For incorrectly generated tags, the users entered the right tags. We have tested the app for both very famous landmarks and more obscure ones, and it performed well in both cases. For example, we tested the app with university buildings, hospitals, churches, museums, libraries, schools, restaurants, coffee shops, etc.

For visual illustration, Figures 9 10 show examples of photos correctly identified by TagPix. Figure 9 shows two of the test photos where there were more than one landmarks
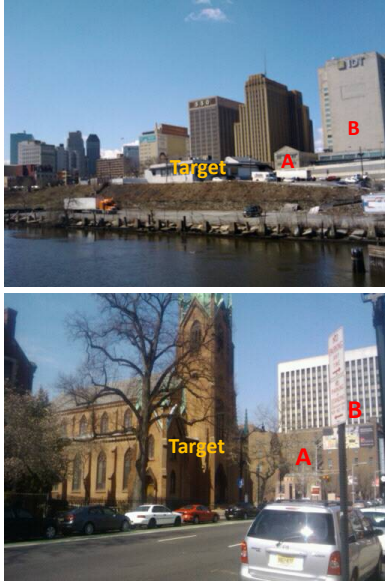
Fig. 9: Photos Taken During the Test Phase. Despite Having Other Landmarks in the Photos, the First Tag Correctly Identified Both of Them: Rio Rodizio Restaurant (Top) and Saint Patrick's Pro-Cathedral (Bottom)
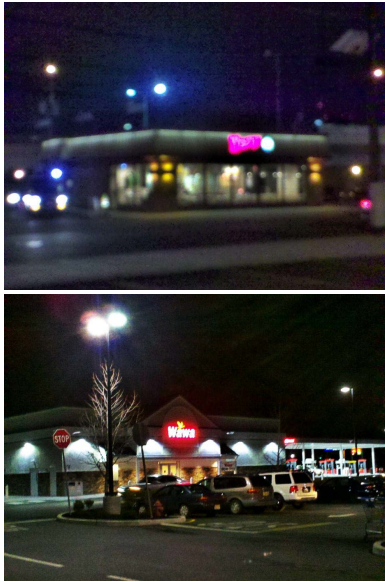

Fig. 10: TagPix Works Well with Blurred Photos or Photos Taken in Low Light. The First Tag Correctly Identified Both of Them.

in close proximity and also in close angular distance. In the top photo, the user's target was to capture the restaurant Rio Rodizio, while two other landmarks, namely The Berkeley College (A in the figure) and The IDT Energy (B in the figure), were present in the frame. Our angular distance based calculation ranked the target landmark (Rio Rodizio) as the top suggestion, and TagPix also filtered out a false positive tag (a tag which has small angular distance, but is incorrect) which was lying behind the target object by using the Euclidean distance estimation. The photo shown at the bottom of the figure illustrates a similar scenario: TagPix

TABLE I: TagPix Accuracy and Comparison with Google Places API

User #	# of photos	1st Tag Correct (%)	2nd Tag Correct (%)	Incorrect (%)	Accuracy %	Google Places Accuracy(%)
1	7	100	0	0	100	42.86
2	67	86.57	1.49	11.94	88.06	14.93
3	7	42.85	0	57.143	42.85	28.57
4	4	75	25	0	100	50
5	1	100	0	0	100	100
6	3	100	0	0	100	33.33
Total	89	84.27	2.25	13.48	86.52	21.35

ranked the Saint Patrick's Pro-Cathedral as the top suggestion, while the Newark Museum (A in the figure) and Horizon Blue Cross Blue Shield (B in the figure) are ranked second and third, respectively.

Figure 10 shows two of the test photos taken during night when there was very little available light to shoot. But, as TagPix does not rely on visual contents of the image, it worked perfectly for such scenarios.

Table I shows the tagging accuracy of TagPix for each user when only angular distance (angular distance threshold is set to 20 degrees) is used to select the tags. This is the method with the best usability as it does not require any action from the user. The table shows the accuracy when the first suggested tag is the correct one and when the second suggested tag is the correct one. For few photos, the landmark list was not fetched in time due to technical problems (e.g., GPS signal taking too much time or lack of Internet connectivity). We do not consider these photos as usable in our experiment. Overall, the results indicate 86.52% accuracy. Among the incorrect tags, 10.11% were just outside of our angular distance threshold.

To understand the extent of improvement due to TagPix, we also compare our results with the tag list returned by Google Places, ranked by distance to the user (i.e., the closest landmark is ranked first). We calculated the accuracy of Google Places by considering any of the first two tags in the list. The last two columns in Table I show that TagPix obtains a much higher accuracy due to our algorithm that leverages the phone sensors in the process of tag selection.

Figure 11 shows the overall distribution of tags. Each bar represents how many tags were in that class (e.g., first suggested tag, second suggested tag, etc.) For this graph, we considered the first 4 tags, while any other tags were deemed incorrect. The results show excellent accuracy for the first tag: 84.2%. Many users may actually accept multiple tags for their photos. In such a situation, if we consider all 4 tags, the tagging accuracy becomes 96%.

Next, we performed experiments to quantify the accuracy benefits of using Euclidean distance estimation together with angular distance. First, we have applied the distance estimation for some of the unrecognized photos from the dataset used in the previous experiments. We found that applying this method minimizes the number of false positive tags. A false positive tag is defined as a tag that falls within the angular
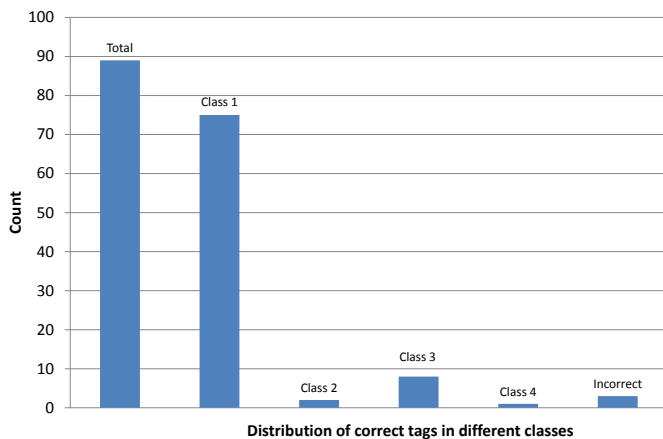
Fig. 11: Distribution of tags in different classes

Fig. 12: Comparison Between Actual and Estimated Distance Using Method 3

TABLE II: TagPix Results with Euclidean Distance Estimation

Photo #	Actual Dist.	Estimated Dist.	Dist. Accuracy	Correct Tag?	False Positive	False Positive Removed
1	59	52.1	88.3	Y	1	1
2	47	50.9	91.6	Y	0	0
3	74	77.9	94.7	Y	2	2
4	57	63.7	88.3	Y	2	2
5	24	26	91.5	Y	1	0
6	41	38.2	93.2	Y	0	0
7	39	44.3	86.5	Y	2	1
8	41	36.7	89.4	N	N/A	N/A
9	101	122	79.5	Y	2	1
10	75	64.5	85.9	Y	1	1
11	62	74.1	80.4	Y	0	0
12	17	19.8	83.3	Y	1	1
13	170	120	70.5	Y	3	3
14	21	19.1	90.9	Y	0	0
15	14	15.6	88.4	Y	1	1

distance, but it is not the correct one. For example, one of our photos contains a landmark named "Burger King", but the landmark was not in the middle of the frame. It has an angular distance of 47 degrees. Whereas two other nearby landmarks (not visible in the photo due to the distance, and thus false positives), "Cosmos Bar" and "Manor Bar & Grill", had angular distance of 21 and 22 degrees, respectively. If only angular distance is considered, then both Cosmos Bar and Manor Bar come before the Burger King in the suggested list; in fact, Burger King is not even considered because we selected only the first two tags. The ground truth distance between the user and Burger King is 47 meters, whereas Cosmos Bar and Manor Bar are 188 and 135 meters away, respectively. The best of our distance estimation methods has 86% accuracy. If we apply it to this scenario, TagPix can easily select Burger King as the best tag

Second, we have used TagPix with Euclidean distance estimation (method 3 in particular) for a new set of photos. TagPix adjusts the spatial distance +/-15% to take into account the observed average error. Specifically, we apply it for the tags returned after the filtering done using the angular distance in
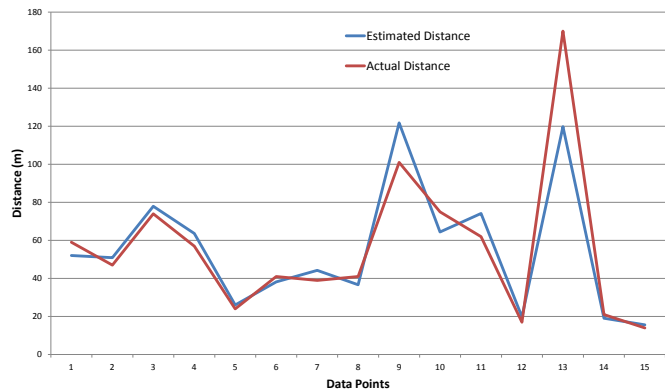
order to select the best one. Table II presents the results. The accuracy increases to 93%, compared to 86% when using only angular distance. Additionally, TagPix is able to remove 81% of the false positive tags.

For better illustration, Figure 12 shows the comparison between the actual distance and the estimated distance by method 3. As we can see, our estimated distances are good enough to be used in TagPix for most practical situations.

Finally, we tried to find the reason why some landmarks cannot be recognized by TagPix in a few occasions. We determined that this is mainly due to the place-tag database, which did not contain information for these landmarks. In a few other cases, the reason was the noisy nature of the geomagnetic orientation sensors. If two landmarks are very close, their correct order can be inversed due to this reason.

VI. CONCLUSION

This paper presented a smartphone app for real-time landscape photo tagging. This app, TagPix, leverages the phone sensors and a place-tag database to achieve very good tagging accuracy. At the same time, TagPix is lightweight as it consumes a modest amount of resources on the phones and protects users' privacy as the photos are not uploaded to servers for tagging. In the basic design, TagPix is completely automatic, thus offering very good usability. For even higher accuracy, users are asked to do simple actions such as moving a short distance and taking an additional photo. TagPix was implemented in Android and successfully tested for many photos in different cities.

REFERENCES

[1] M. T. Review, "Smart phones are eating the world," 2013. [Online]. Available: http://www.technologyreview.com/photoessay/511791/smartphones-are-eating-the-world/

[2] D. Tam, "Facebook processes more than 500 tb of data daily." [Online]. Available: http://news.cnet.com/8301-1023_3-57498531-93/facebook-processes-more-than-500-tb-of-data-daily/

[3] J. Li and J. Wang, "Real-time computerized annotation of pictures," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 6, pp. 985–1002, 2008.

[4] S. Lindstaedt, R. Mrzinger, R. Sorschag, V. Pammer, and G. Thallinger, "Automatic image annotation using visual content and folksonomies," *Multimedia Tools and Applications*, vol. 42, pp. 97–113, 2009, 10.1007/s11042-008-0247-7. [Online]. Available: http://dx.doi.org/10.1007/s11042-008-0247-7

[5] M. Naaman, R. Yeh, H. Garcia-Molina, and A. Paepcke, "Leveraging context to resolve identity in photo albums," in *Digital Libraries, 2005. JCDL '05. Proceedings of the 5th ACM/IEEE-CS Joint Conference on*, june 2005, pp. 178 –187.

[6] C. Qin, X. Bao, R. Roy Choudhury, and S. Nelakuditi, "Tagsense: a smartphone-based approach to automatic image tagging," in *Proceedings of the 9th international conference on Mobile systems, applications, and services*, ser. MobiSys '11. New York, NY, USA: ACM, 2011, pp. 1–14. [Online]. Available: http://doi.acm.org/10.1145/1999995.1999997

[7] D. Joshi, J. Luo, J. Yu, P. Lei, and A. Gallagher, *Using Geotags to Derive Rich Tag-Clouds for Image Annotation*, 2011, p. 239.

[8] J. Luo, D. Joshi, J. Yu, and A. Gallagher, "Geotagging in multimedia and computer vision–a survey," *Multimedia Tools Appl.*, vol. 51, no. 1, pp. 187–211, Jan. 2011. [Online]. Available: http://dx.doi.org/10.1007/s11042-010-0623-y

[9] R. Raguram, C. Wu, J.-M. Frahm, and S. Lazebnik, "Modeling and recognition of landmark image collections using iconic scene graphs," *International Journal of Computer Vision*, vol. 95, no. 3, pp. 213–239, 2011. [Online]. Available: http://dx.doi.org/10.1007/s11263-011-0445-z

[10] K. Patel, M. Ismail, S. Motahari, D. Rosenbaum, S. Ricken, S. Grandhi, R. Schuler, and Q. Jones, "Markit: Community play and computation to generate rich location descriptions through a mobile phone game," in *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*, 2010, pp. 1–10.

[11] "Argon browser." [Online]. Available: http://argon.gatech.edu/

[12] W. GmbH, "Wikitude." [Online]. Available: http://www.wikitude.com/app/

[13] J. G. Manweiler, P. Jain, and R. Roy Choudhury, "Satellites in our pockets: an object positioning system using smartphones," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*, ser. MobiSys '12. New York, NY, USA: ACM, 2012, pp. 211–224. [Online]. Available: http://doi.acm.org/10.1145/2307636.2307656

[14] "Smart measure." [Online]. Available: https://play.google.com/store/apps/details?id=kr.sira.measure&hl=en

[15] Google, "Google goggles." [Online]. Available: http://www.google.com/mobile/goggles

[16] D. Takahashi, "How google goggles works to deliver visual search results for mobile phones," 2010. [Online]. Available: http://venturebeat.com/2010/08/23/how-google-goggles-works-to-deliver-visual-search-results-for-mobile-phones/

[17] Google, "Google places api." [Online]. Available: https://developers.google.com/places/

[18] GeoNames, "The geonames project." [Online]. Available: http://www.geonames.org/

[19] NOAA.gov, "The magnetic declination calculator." [Online]. Available: http://www.ngdc.noaa.gov/geomagmodels/struts/calcDeclination

[20] M. Merlijn, "Gait analysis." [Online]. Available: http://www.geradts.com/html/Documents/gait.htm