The Social Camera: Recommending Photo Composition Using Contextual Features

Steven Bourke, Kevin McCarthy & Barry Smyth

CLARITY: Centre for Sensor Web Technologies,
School of Computer Science and Informatics,
University College Dublin (UCD), Belfield, Dublin 4, Ireland.
E-mail: firstname.lastname@ucd.ie

ABSTRACT

The digital camera revolution has changed the world of photography and now most people have access to, and even regularly carry, a digital camera. Often these cameras have been designed with simplicity in mind: they harness a variety of sophisticated technologies in order to automatically take care of all manner of complex settings (aperture, shutter speed, flash, etc.) for point-and-shoot ease. However, there is little or no support for the end-user when it comes to helping them to compose or frame a scene. To this end we describe a novel recommendation system application that is designed to assist the user by recommending relevant compositions given their current location and scene context. This application has been implemented on the Android platform and we describe its core recommendation technologies and user interface.

1. INTRODUCTION

The success of digital cameras means that the world of photography has changed forever. But, the first generation of dedicated point-and-click digital cameras represent only the beginning of a much broader revolution. Today many of us carry a digital camera with us everywhere we go; they are a common feature of a modern mobile phone. This has lead to an explosion of photographic content, which has been created and uploaded to a variety of photo-sharing services. In parallel, considerable research effort has been focused on assisting users when it comes to capturing and managing images. For example, in addition to the auto-exposure setting features of most modern cameras, new advances in face recognition are now being used to help users to improve portrait style photography by auto-focusing on faces in a scene [3]. Similar techniques are also being used to help users to better organise and catalog their growing image collections [12]. Indeed a wide range of classification technologies are being used for scene recognition and classification for improved collection management [2]. Recently, cameras started to become available equipped with location sensing technology and digital compasses, this introduces some interesting new opportunities when it comes to helping novice users to take better quality photos. Indeed, while modern cameras have sophisticated auto-exposure modes (to take care of aperture and shutter speed, flash, etc.) [4] there is little or no support for the end-user when it comes to helping them to compose or frame a scene. Simply put, modern point-and-shot cameras work well at fixing the exposure settings that are appropriate to a given scene but they don't help the user when it comes to choosing an interesting shot or framing the scene. This problem is referred to as the *composition problem* and choosing the right composition is a key ingredient when it comes to taking high-quality photographs.

We consider this composition problem as a novel type of recommendation opportunity whereby individual users are prompted, as they setup to take a photograph, with nearby examples of relevant, well-composed, previously taken photographs. In other words, we can recommend a short-list of high-quality, well-composed photographs to the user, based on their current location, lighting conditions, etc. in the hope that one of these compositions may usefully guide the photographer with respect to their current photograph. This represents an interesting recommendation challenge for a number of reasons, in particular the use of contextual information from the physical world (location, time, lighting, etc.) is related to recent work in *context-aware recommender systems research* [10, 13]. Moreover, this is an opportunity to introduce recommender systems into an existing and ubiquitous consumer technology, namely digital cameras and camera phones, where there is a pre-existing history of sophisticated assistive technologies and thirdly, this work is enabled by recent device advances (e.g. GPS and digital compass technologies) and online services such as geo-coded image repositories like Panaramio[1], without which this work would not be practically possible. We use the social ranking systems of these image services to provide an initial step in the filtering process of our recommendation system.

In this paper then, we describe our initial attempt at developing the *social camera*, which embodies our recommender system to provide compositional support. The application has been developed on the Android[2] platform and we describe its core recommendation technologies and demon-

CARS-2010, September 26, 2010, Barcelona, Spain.

This research is supported by Science Foundation Ireland under Grant No. 07/CE/11147 CLARITY CSET

[1] www.panoramio.com
[2] www.android.com

strate its effectiveness in a number of real-world scenarios.

2. BACKGROUND

The work described in this paper brings together ideas from a number of different areas of research including, recommender systems, image retrieval, context-based and context aware systems, mobile computing, computational photography, and the sensor web.

A core element of our social camera system concerns the ability to identify and retrieve relevant images and there has been considerable work to date on the image retrieval in general. In particular, content-based image retrieval (CBIR) approaches seek to understand the content of an image by using a variety of image analysis techniques to extract core image features as the basis for matching during retrieval. For example, colour histogram, edge detection, and shape extraction techniques have all been used as the basis for image retrieval; see for example [6]. On their own however these *intrinsic* properties of an image are rarely sufficiently informative to drive an effective image retrieval system and so researchers have begun to look beyond the image towards *extrinsic* forms of information that may be used during retrieval. For example, recently the work of Von Ahn [14] and others have demonstrated how image representations can be greatly enhanced with tags, and how people can be encouraged, and are willing to, provide tags that carry semantically rich information about images. In our work, we are focused on image retrieval in a mobile context and this provides additional sources of informative features that can be used to greatly constrain the retrieval task. For instance, the availability of accurate location information (via GPS sensors) is one obvious source of context information and this type of information has been harnessed as the basis for tag inferencing during image retrieval [11].

Indeed the availability of rich context information has led to an increasing interest in *context-aware recommender systems*; see [9, 13, 15]. Generally speaking there are different forms of context (as opposed to user preferences) that can be used to guide recommendation. For instance, the work of Lombardi et al [7] look at context in an e-commerce setting. They argue that the preferences of a user can, and should, be segmented into different contextual states. In other words, the preferences of individual users will change according to their buying context and thus recommendation accuracy can be improved by modeling their contextual states and transitions. The work of Baltrunas and Ricci [1] focuses on the potential for contextual information to influence traditional matrix-based collaborative filtering recommender systems. More relevant to the work in this paper is the role of context information that can be derived from the physical world as a by-product of recommender system usage. For example, the early work of Van Setten et al. [13] has specifically focused on the role of location information in a mobile recommender systems for tourism applications. Location information is clearly external, physical world factor that is generated by the movement of the user and that can be applied to the generation of recommendations, and it is this type of context that is quite relevant to the work of this paper. Generally, it is worth highlighting that researchers have begun to explore the different ways that context can be incorporated into the recommendation process; see [9, 15].

In this paper we are concerned primarily with a very familiar real-world scenario, helping people to frame photographs. We believe that the time is now right for recommender systems to play an important role in a new generation of connected cameras, especially when it comes to providing intelligent assistance to the user. Of course, digital camera manufacturers and users alike are already all too familiar with the important role that sophisticated computational techniques have played when it comes to supporting image capture and processing. For example, sophisticated object and face detection techniques can be implemented in real-time so that prominent objects can be identified within a scene for improved focus and exposure settings [8]. The dreaded "red-eye" effect can now be removed using a combination of face detection and local colour manipulation [5]. Images that have been blurred by unwanted camera shaking can be repaired to produce sharply focused images [16].

3. THE SOCIAL CAMERA

Our aim in this work is to develop a recommender system that is capable of recommending well-composed photographs to a user, which are relevant to the current location and setting, as a way to help the user take better pictures for themselves. This brings together a number of important recommendation ideas: (1) understanding the user's current context as the basis for a recommendation profile/query; (2) selecting a suitable set of candidate images from an online image repository; (3) ranking these candidates and selecting a short-list for recommendation to the end-user. In this section we describe the form and function of the social camera application.

3.1 Architecture

The overall social camera system is divided into 3 main components — the *camera component* or *social camera app*, the *recommendation engine*, and the *image server* — as shown in Figure 1. The camera component is the actual software that runs on the camera. This has been implemented on the Android platform and is responsible for handling the core image capture functions of the camera itself, as well as providing the primary interface between the user, the user's context and the recommendation service. Each time the user points the camera at a scene, the social camera app generates a set of *context features* from the current scene settings. These features include the current time, GPS coordinates, compass direction, lighting conditions, as well as the current camera settings, such as aperture and ISO speed; see Figure 2 for an example of these various context features. In short, these features provide a detailed representation of the current scene context. They represent not just the current location (GPS) but also the direction that the camera is pointing (digital compass) and photos that have similar location and compass features are likely to capture very much the same scene that the user is currently seeing. Features such as camera's ISO, aperture, and exposure time settings are set automatically by the camera device and they capture important information about the lighting conditions that currently exist; images that match in terms of their lighting conditions are therefore likely to be good matches for the current scene, from an exposure viewpoint.

In combination then, these context features provide the basis for image retrieval. In the case of social camera, we rely on a variety of online image repositories, such as Panaramio[3]

[3]www.panoramio.com

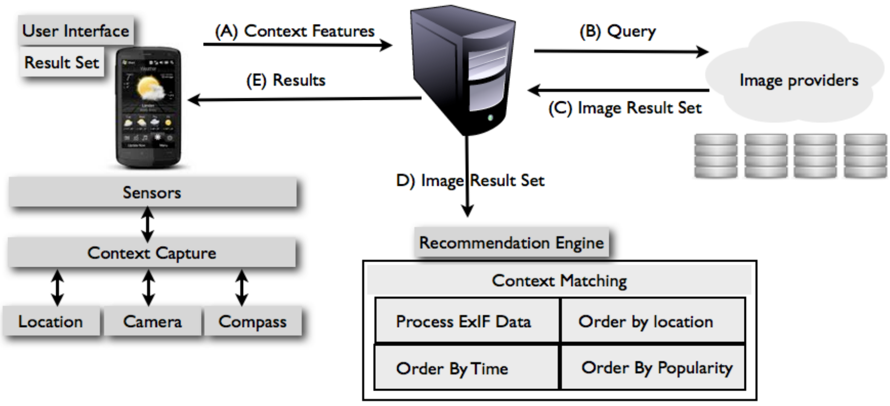

Figure 1: System Architecture of the Context-Aware Social Camera

where users have uploaded GPS-tagged photographs, complete with relevant EXIF[4] meta-data (ISO, aperture, etc.) we use the social ranking aspect of these repositories when ranking images which are then used in our recommendation technique. The recommendation engine selects relevant images based on a matching function that compares the current user context to the meta-data stored with the images (see Section 3.2). This provides a short-list of relevant, high-quality images that can be ranked and presented to the user through the social camera app.

```
<context id="12345" user="abcde">
    <feature name="DateTime"><value>2010:01:17 15:16:21</value></feature>
    <feature name="Lighting"><value>77/10</value></feature>
    <feature name="CompassDirection"><value>15.463</value></feature>
    <feature name="Location">
        <property name="longitude"><value>53-18-26.56N</value></feature>
        <property name="latitude"><value>6-13-21.08W</value></feature>
    </feature>
    <feature name="ISOSpeedRatings"><value>100</value></feature>
    <feature name="AperatureValue><value>37/10</value></feature>
</context>
```

Figure 2: Example context information.

As with any consumer-facing technology, the user experience is a vital success factor and special care and attention has been paid to the development of an simple but powerful user interface for the social camera app. There are three basic parts to the social camera interface: *photo recommendation*, *directional assistance*, and *framing assistance*. Obviously, the context capture functionality remains invisible to the user and is activated when they point the social camera app at a particular scene. But once a suitable set of recommendations have been located the user is given the option to review these as examples of high-quality images that have been take nearby; this is the photo recommendation component. If the user chooses an image that they would like then the interface provides on-screen directional assistance to the user to help them to better re-orientate themselves so as to be more closely aligned with the chosen image-scene. Once the user is in the correct photo-taking position they can receive framing advice, effectively overlaying the chosen image on the current scene as a transparent overlay so that the user can more precisely compose their own photograph. In what follows, we will describe the core recommendation algorithm in more detail and then present a brief walk-through of the social camera in action, focusing on these primary user interface features.

[4]www.exif.org

3.2 Recommendation Process

The summary recommendation algorithm is presented in Figure 3. The basic input to the recommender include the context profile (CP) and the number of images to return as recommendations to the end-user (k), typically we return 5 images. The first step of the recommendation engine is to locate a suitable set of images that match in terms of their location and direction properties. This is a key point. There is little benefit to presenting the user with photos, no matter how well composed they are, if the photos bear no resemblance to their current scene and location. Similarly, all things being equal it makes sense to prioritise photos from a given location that have similar directional information. For this reason, during this stage of the recommendation process we retrieve a set of n images (where n is typically 100) such that these images are within 50m of the current location and at a similar time of day; there is little advantage to presenting the user with a night-shot if they are experiencing bright sunshine. Next these images are then scored according to a combination of how close they are to the users current position and the angular difference between their direction and the users current direction, as shown in Lines 8-15. This provides a set of recommendation candidates that are likely to be recognisable within the view of the current user.

Next, we use further scoring functions in order to evaluate the utility of these recommendation candidates. First we score the images based on the date and time in the meta data (see Lines 16-22).Then we score the images based on how closely their light-related settings match the current user's context features (see Lines 23-28). This will allow us to give a preference to photos taken under similar exposure settings; it may be a particularly dull day leading to the need for a longer exposure time or a greater aperture setting, for example. Second, the images are also scored on the basis of their popularity rating; see (Lines 29-30). Many image repositories allow users to rate images and this information can be used by our recommender to give preference to images that seem to be well liked, on the assumption that such images are likely to be of higher quality.

We now have a set of images that have been taken in the vicinity of the current user, at a similar time of day and these images have been scored according to their precise proximity, exposure settings, and popularity. To produce a final set of k recommendations we used a weighted scoring function (see Line 31) in order to rank these images by the combination of these proximity, exposure, popularity scores. We assign equal weights to each score but as a matter of future work it would be worth considering alternative scoring regimes.

```
CB: Context Profile, rs: Result Set, q: query, ID:
Image Database, ,EA: ExIfAttributes, SA: Supported
ExIF attribute, DR: Date Rank, R: Searchable Range, L:
Location

1.   Define RecommendImages(CP, k)
2.     rs' ← RequestImages(CP, CP.Angle)
3.     rs' ← DateTimes (RS, CP)
4.     rs' ← ExIfData(RS)
5.     k' ← PopularityRanking(rs,k)
6.     Return k
7.   End

8.   Define RequestImages(CP, Angle)
9.     rs' ← ImageDatabase.Query(CP.Location,
     CP.Radius)
10.    rs' ← (img ∈ rs)
11.            IF(img Not Within Angle)
12.                rs.remove(img)
13.            End
14.    End
15.  End

16.  Define DateTimes(RS, CP.Time)
17.      rs' ← (img ∈ RS)
18.        IF(img.Time > (CP.Time + 5hours)
19.         rs.remove(img)
20.        End
21.      End
22.  End
23.  Define ExIfData(RS)
24.     rs' ← (img ∈ RS)
25.     if(img.ExIfattributes Not Compatible
     (CP.ExIfAttributes)
26.  Rs.remove(img)
27.  End
28.  End

29.  Define PopularityRanking(rs,k)
30.    OrderByPopularity(rs)
31.     rs' ← rs.GetTop(k)
32.              Weight.rs.score((EA/SA * 1.25)
                 Weight.rs.score(L * 1.4)
                 Weight.rs.score(CB*1.35)
                Return Weight.rs(k)
33.     End
34.  End
```

Figure 3: The recommendation algorithm.

4. EXAMPLE SESSION

Figure 4 presents screenshots of the social camera app in action. In this case the user is located near to Tower Bridge in London and in what follows we will summarize a brief walk-through of the assistive technologies in action.

4.1 Location Selection

When the social camera app is activated the user is first presented with a location selection screen; Figure 4(a). Very briefly, this encapsulates the recommendation search area according to user's current location. Typically the user quickly moves off this screen but it does provide an opportunity to adjust some of the recommendation settings if desired. For example, by default the social camera, as mentioned above, will focus on retrieving images that were taken from positions no more than 50m from their current position. This interface allows the user to easily adjust this default by either extending or contracting the location disc as shown; indeed the user can also use this feature as a way to relocate their current position manually, in order to review photo recommendations from other locations, for example.

4.2 Photo Recommendation

Once the user is satisfied with their location, the social camera retrieves a ranked list of images according to the recommendation strategy outlined in the previous section; see Figure 4(b). From an interface standpoint the user can simply cycle through these images until they find one that they like. In this example, let's assume that the user has selected the lower image in Figure 4(b) and wished to take a similar shot. Once the image has been selected the social camera app will adjust the camera's current exposure settings to match those of the current image, allowing for variations in lighting as appropriate; priority is given to aperture in the current system.

4.3 Directional Assistance

Having selected the image the user now needs to better position themselves in order to take a similar image. In Figure 4(c) we show a simple form of directional assistance provided by social camera. In the top left corner of the screen, the photographer positioning for the selected images are displayed on a dynamically changing positioning radar. The radar updates as the user moves about and helps the user to get to the correct location. To achieve this goal, the icon which represents the images location adjusts depending on the location and bearings of the user. Additionally, the user is guided to adjust their position so that the colour-coded recommendation icon is presented in the centre of the screen. Is the user's current position is too far off to the left, for example, then the icon will appear on the right-hand side of their screen, and as the adjust their positioning it will move towards the centre of the screen; in Figure 4(c) we can see that the user has adjusted their positioning so as to be aligned just slightly to the left of the retrieved image's position. Of course at any time the user can just decide to take a photo; they are not compelled to exactly replicate the recommended image.

4.4 Framing Assistance

The directional assistance feature is unlikely to produce a perfect alignment between user and recommended image. There are limits to the accuracy of current GPS location-sensing technologies which ultimately mean that the directional assistance feature is useful for some rough positional adjustments. The final feature of the social camera interface is designed to help with the final framing of the user's photograph. Basically, this allows for the current recommended image to be overlaid on the current viewfinder scene as a transparent overlay; the degree of transparency can be adjusted by the user with the on-screen slider, to allow for different lighting conditions. In this way the user can make some fine-tuned adjustments to their framing and position. In Figure 4(d) for example, we can see that the user has reproduced very closely the recommended image and is now ready to take their own photograph.

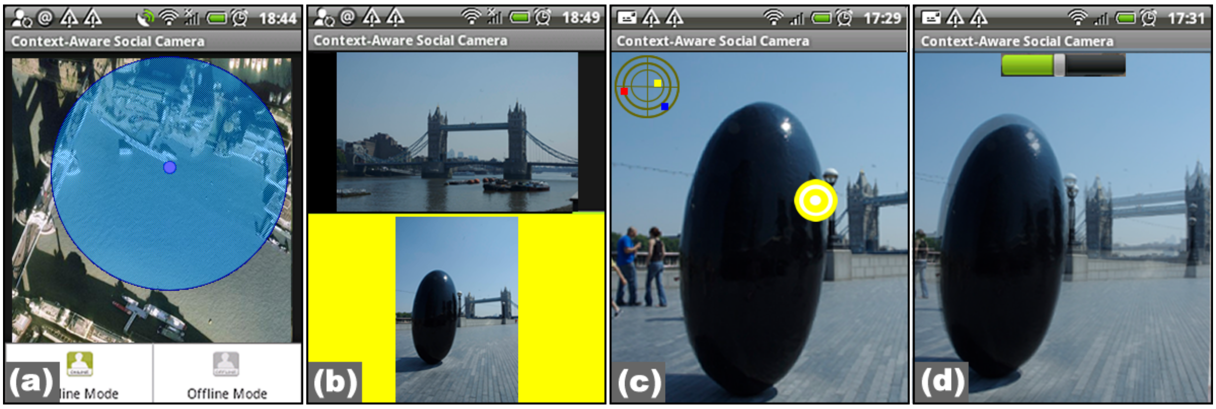

Figure 4: a) Selecting your location. b) Available images from the Photo Recommendation list. c) Viewfinder image with Directional Assistance. d) Increasing the opacity of the recommended image with the Framing Assistance tool.

5. CONCLUSION

In this paper we have argued the need for composition and framing advice so that users can learn how to compose a well-framed photograph. Current digital camera technologies are very much lacking in this regard. As such we have described the design and development of the social camera system. This has been fully developed as an Android app. It provides users with recommendations of well-framed photographs based on their current context (location, direction, lighting conditions) so that the user can easily emulate an image of their choosing and, in the longrun, improve their own photographic competence. Additionally, such an experience allows for users to frame expert pictures themselves while including their family or friends. There are some challenges when it comes to improving the user interface and overall user experience but we believe that this is a good first step towards a very interesting context recommender systems opportunity. Future work will involve a live user study examining the recommendation quality and user reaction to the interface.

6. REFERENCES

[1] L Baltrunas and F Ricci. Context-dependent items generation in collaborative filtering. *Proceedings of the 2009 Workshop on Context-Aware Recommender Systems*, 2009.

[2] Matthew Boutell and Jiebo Luo. Bayesian fusion of camera metadata cues in semantic scene classification. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:623–630, 2004.

[3] M Davis, M Smith, J Canny, N Good, and S King. Towards context-aware face recognition. *Proceedings of the 13th annual ACM international conference on Multimedia*, 2005.

[4] Frédo Durand and Richard Szeliski. Computational photography. *IEEE computer graphics and applications*, 27(2):21–2, 2006.

[5] M. Gaubatz and R. Ulichney. Automatic red-eye detection and correction. In *IEEE International Conference on Image Processing*, page 804âĂŞ807. Citeseer, 2002.

[6] Y Liu, D Zhang, G Lu, and WY Ma. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, 2007.

[7] S Lombardi, SS Anand, and M Gorgoglione. Context and Customer Behavior in Recommendation. *Proceedings of the 2009 Workshop on Context-Aware Recommender Systems*, 2009.

[8] David G. Lowe. Object Recognition from Local Scale-Invariant Features. *ICCV*, 1999.

[9] Umberto Panniello, Alexander Tuzhilin, Michele Gorgoglione, Cosimo Palmisano, and Anto Pedone. Experimental comparison of pre- vs. post-filtering approaches in context-aware recommender systems. *ACM Conference On Recommender Systems*, 2009.

[10] Moon-Hee Park, Jin-Hyuk Hong, and Sung-Bae Cho. *Ubiquitous Intelligence and Computing*, volume 4611 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[11] Benjamin Proß, Johannes Schöning, and Antonio Krüger. iPiccer: automatically retrieving and inferring tagged location information from web repositories. *ACM International Conference Proceeding Series*, 2009.

[12] Kerry Rodden and Kenneth R. Wood. How do people manage their digital photographs? In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 409–416, New York, NY, USA, 2003. ACM.

[13] Mark van Setten, Stanislav Pokraev, and Johan Koolwaaij. Context-Aware Recommendations in the Mobile Tourist Application COMPASS, 2004.

[14] Luis von Ahn. Human computation. *International Conference On Knowledge Capture*, 2007.

[15] GE Yap, AH Tan, and HH Pang. Dynamically-optimized context in recommender systems. *Proceedings of the 6th international conference on Mobile data management*, 2005.

[16] L. Yuan, J. Sun, L. Quan, and H.Y. Shum. Image deblurring with blurred/noisy image pairs. *ACM Transactions on Graphics*, 26(3):1, 2007.