Relevance Metrics for Coverage Extension Using Community Collected Cell-phone Camera Imagery

Aman Kansal, Lin Xiao, and Feng Zhao
Microsoft Research
One Microsoft Way, Redmond, WA.
{kansal,lixiao,zhao}@microsoft.com

Abstract

The rapid increase of mobile phone cameras has enabled users to easily take and share pictures. This has created a potential for mobile device driven sensing of our world at a previously unachieved spatio-temporal granularity, enabling a variety of new applications. The data collection activity is highly uncoordinated and hence, a key issues in effectively using such imagery is understanding the relevance value of each image. Having such a value can not only streamline the resource usage in sharing the image data but also support the development of incentive mechanisms for users to contribute worthwhile data. We discuss the problem of assigning relevance values to images from mobile devices with respect to an application's existing image data-set. We describe a general information theoretic framework for computing relative relevance and discuss specific value computation for a coverage based metric. We also develop a practical algorithm to compute relevance and describe methods to make our computation scalable to large data sets. Finally, we present our prototype implementation demonstrating our methods on real world data .

1 Introduction

A large fraction of cellular phones is now embedded with cameras. Since images provide a rich sensing modality, the widespread adoption of camera phones has created a potentially omnipresent sensor network that can provide fine grained measurements of the physical world. This capability can enable a variety of applications. For instance, users carrying cell-phone cameras may provide instantaneous news coverage of an unplanned interesting event. Images of sidewalks taken by users in their local regions may be used to enhance or update the street side imagery of the global road network. User groups may collect interest specific images documenting conditions of their interest, such as the maintenance status of a community park after a thunderstorm. Many other applications of sensing via mobile devices have been previously discussed [1, 2].

This mobile device centric sensor network is different from traditional wireless sensor networks. Rather than having dedicated sensing devices, this network piggybacks on user carried mobile devices. This leads to several design challenges, one of which is the extremely uncoordinated behavior of the network. In this paper we address one of the implications of uncoordinated behavior: that on data collection. The sensing devices may be present at different locations at different times and may never all be connected. This makes centralized coordination of data collection activities very difficult. We discuss how the applications using the data collected by such networks may exercise some control on the collection and sharing of data.

We propose to achieve a semblance of coordination in the data collection activity through assigning relevance metrics to data provided by mobile devices. If we provide a relevance metric that captures the utility of the data for the application, the application may extract the most useful data from the plethora of images provided by a large number of uncoordinated mobile devices. This can help streamline the use of system resources, such as bandwidth and storage space. Further, the relevance metrics can be used to design incentive mechanisms that persuade users to contribute more relevant data, thus leading to distributed coordination in the data collection activity. Also, if uploading the data from the mobile device is costly, relevance values may help select appropriate images to be uploaded.

1.1 Problem Description

We use the following system model (Fig. 1). Multiple contributors carrying mobile cameras upload images, using cellular or Internet connections. The images are processed by our relevance computation engine to assign application specific relevance metrics. Multiple applications may then access this data filtered by the relevance metric of their interest.

Relevance may be measured among several dimensions based on the application. Relevance may be temporal. For instance, for live coverage of an event, the latest image or the image taken at the correct time instance may be the most worthwhile. Relevance may also be spatial, measured in terms of the extent of spatial coverage provided by the image, its resolution, or its image quality, including the inten-

WSW'06 at SenSys'06, October 31, 2006, Boulder, Colorado, USA.

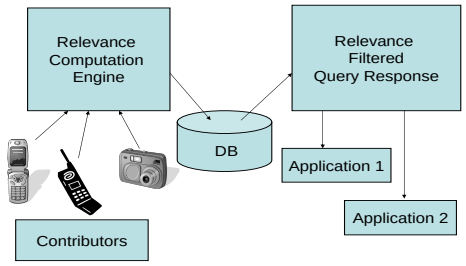

Figure 1. A block diagram of the system.

sity histogram, contrast, blur, or device sensor noise. Other dimensions of relevance include the utility of an image for 3D reconstruction of the scene, which may depend on the perspective provided by the image relative to other images. Sophisticated relevance metrics may also be based on the semantic content of the image, measured in terms of the objects of interest captured by it.

In this paper, we focus on a spatial relevance metric based on the coverage of the physical world provided by an image. Relevance of an image is then measured in terms of novel coverage provided with respect to a previously obtained set of images. Such coverage based relevance is useful for several applications such as enhanced or updated imagery of streets, highways or other public spaces such as parks, or coverage of community specific areas, such as a park after a thunderstorm. Measuring coverage based relevance for mobile camera imagery has certain issues arising out of the uncoordinated nature of this data collection activity. Unlike controlled imagery collected by satellite, airplane, or vehicle mounted cameras currently used by online mapping services [3, 4], the spatial relationships among the images from multiple mobile devices are unknown and uncontrolled. Arbitrary variation may exist in perspective, lighting conditions, image size, resolution, or noise quality, due to multiple users and mobile devices being involved. This implies that matching pixels across images will not yield the coverage overlaps. Also, since the number of images may be very large, such as thousands, the incremental processing required for each additional image should be scalable. Our methods address many of these issues.

2 Information Theoretic Image Ranking

Before discussing coverage specific relevance computation, we describe a general framework to determine the relevance of data, based on information theory.

2.1 Entropy

Information theory defines the information content of a data value based on the probability of occurrence of the data value. Suppose data is represented by a parameter x and takes values in the set $\mathcal{X}$ of finite cardinality. Suppose the probability that x takes a particular value $\mathcal{X}_i$ in the set $\mathcal{X}$ is denoted $P(x = \mathcal{X}_i)$. Then, the entropy $H(X)$ with respect to the probability distribution of x, is defined in as [5]:

$$H(X) = E[-log_2 P(x)] = -\sum_{x \in \mathcal{X}} P(x) log_2 P(x) \qquad (1)$$

where $P(x)$ denotes $P(x = \mathcal{X}_i)$. Consider now a second parameter y, taking values in the set $\mathcal{Y}$, that has some information about x. The novel information in x when the value of y is already known may be modelled as the conditional entropy of x given y, $H(X|y = \mathcal{Y}_i)$. This can be calculated using the same formula as used for entropy but applied to the conditional probability of x:

$$H(X|y = \mathcal{Y}_i) \quad = \quad E[-log_2 P(x|y = \mathcal{Y}_i)] \qquad (2)$$

The above entropy concepts can be applied to image data to measure the coverage information provided by an image. Suppose an image is represented by a vector $\mathbf{x} = \{x_1, x_2, ..., x_n\}$ where n is the number of pixels in the image[1]. Entropy can be computed for a vector variable $\mathbf{x}$ by using its joint probability distribution in equation (2). Suppose next that a set of images $\{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_m\}$ has already been obtained, and the novel information in $\mathbf{x}$ is to be determined with respect to these images. The conditional entropy $H(\mathbf{x}|\mathbf{y}_1, ..., \mathbf{y}_m)$ may then be used to measure the novel information in image $\mathbf{x}$ given the set of previously obtained images.

2.2 Coverage Based Entropy Measure

The definition of the probability measure used for $\mathbf{x}$ and for its conditional probability given the previously obtained images will determine what aspect of the information contained in the image is being used for characterizing relevance. We now develop coverage based metrics for these quantities.

Suppose the set of all images from which $\mathbf{x}$ is taken is represented by $\mathcal{A} = \{\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_N\}$, where N is the total number of possible images. Then, the probability of image $\mathbf{x}$ taking a particular value $\mathbf{a}_i$ is denoted $P(\mathbf{x} = \mathbf{a}_i)$. Note that:

$$\begin{aligned} P(\mathbf{x} = \mathbf{a}_i) &= P(x_1 = a_{i1}, x_2 = a_{i2}, ..., x_n = a_{in}) \qquad (3) \\ &\geq P(x_1 = a_{i1}) P(x_2 = a_{i2}) ... P(x_n = a_{in}) \qquad (4) \end{aligned}$$

where we restrict all images to a size of n pixels. The inequality in (4) is due to the fact that pixel values are typically not independent in an image. Modelling the interdependence among pixels is computationally intractable in a general sense since it depends heavily on the scene content of the image. As a heuristic, we use the probability computed with the independence assumption as an exact metric for probability and use it for computing image entropy. This is in keeping with the maximum entropy assumption used in signal processing when the exact interdependence among signal components is unknown. Suppose each pixel may take one of B values, where B may depend on pixel bit depth. The individual pixel probability, assuming uniform distribution over all possible pixel values becomes $P(x_j = a_{ij}) = 1/B$ and is same for all $j \in \{1, 2, ..., n\}$. Also, the number of possible images $N = B^n$. Thus,

$$H(\mathbf{x}) \quad = \quad -\sum_{\mathbf{x} \in \mathcal{A}} P(\mathbf{x} = \mathbf{a}_i) \log_2 P(\mathbf{x} = \mathbf{a}_i)$$

[1] We use boldface letters to represent vectors throughout.

$$
\begin{aligned}
&= -\sum_{i=1}^{B^n}\left[\prod_{j=1}^{n}P(x_j=a_{ij})\right]\log_2\left[\prod_{j=1}^{n}P(x_j=a_{ij})\right]\\
&= -\sum_{i=1}^{B^n}\left(\prod_{j=1}^{n}\frac{1}{B}\right)\log_2\left(\prod_{j=1}^{n}\frac{1}{B}\right)\\
&= n\log_2(B) \qquad (5)
\end{aligned}
$$

The result is intuitive as the coverage is found to be proportional to the number of pixels. Now, to define conditional entropy with respect to previously obtained images, we need to define the conditional probability for image $\mathbf{x}$ with respect to previously known images such that coverage information from previously known images is taken into account. To this end, we define a new vector $\mathbf{z}$, called the overlap vector, that contains all pixels of $\mathbf{x}$ that correspond to the same physical region as contained in the previously known images $\mathbf{y}_i, i \in \{1,2,..,m\}$. Note that the pixel values in $\mathbf{z}$ need not be equal to values of corresponding pixels in the previously known images but only the represented physical region needs to be the same. This means that same pixel value arising due to different objects in the view will not cause the novel information in $\mathbf{x}$ to be lost, nor will variations in pixel values due to changes in lighting, sensor noise, or perspective for the same physical region cause a pixel to be missed from $\mathbf{z}$. Suppose vector $\mathbf{z}$ is computed, and has k pixels. Re-order the pixels of $\mathbf{x}$ such that the pixel coordinates that do not occur in z are indexed $\{1,...,(n-k)\}$ and the remaining ones are indexed $\{(n-k+1),...,n\}$. Then, the conditional entropy becomes:

$$
\begin{aligned}
H(\mathbf{x}|\mathbf{y}_1,\mathbf{y}_2,...,\mathbf{y}_m) &= H(\mathbf{x}|\mathbf{z})\\
&= H(\{x_1,...,x_n|x_{n-k+1},...,x_n\})\\
&= H(\{x_1,...,x_{n-k}\})\\
&= (n-k)\log_2 B \qquad (6)
\end{aligned}
$$

again using the independence assumption and maximum entropy principle. Note that n and k depends only on the size of image $\mathbf{x}$ and not the sizes of previously obtained images. Since, $\mathbf{z}$ and k can be determined for $\mathbf{x}$ with respect to arbitrarily sized images, all images in the database need not be the same size. We normalize the conditional entropy with respect to n, obtaining the relevance metric, $r(\mathbf{x})$:

$$r(\mathbf{x}) = (1-k/n)\log_2 B. \qquad (7)$$

3 Algorithm for Overlap Computation

Clearly, to use the above formula for relevance assignment, a crucial step is the computation of vector $\mathbf{z}$ containing the pixels of $\mathbf{x}$ that correspond to regions overlapping with previously obtained images. We now discuss how this can be computed for real images.

3.1 Design Considerations

The method to compute $\mathbf{z}$, should be robust to changes in lighting, sensor noise, image size or resolution, and small perspective changes. Directly matching pixel values is obviously not a good strategy in terms of robustness. Instead we take an approach based on key features. Key features are certain invariants in the image which are robust to pixel value changes due to the above mentioned factors. Several key feature detection methods have been proposed and we use the one from [6].

However, detecting and matching key features directly does not yield the entire overlapping physical region in the image views since key features only occur at certain peculiar scene features such as object corners and texture rich spaces. Also, errors may sometimes occur in matching key features across images leading to false positives.

While the key features do not directly yield the overlap regions, the pixel locations of the key features of the input image that are found to be matching with a previous image can be used to find some of the pixels which represent the same physical region across the two images. Computing the convex hull of the pixel locations of the matched key features then suggests what portion of the input image overlaps with the previous image. However, as key features depend on the scene texture, the convex hull found may be much smaller than the actual overlap region, leaving significant textureless overlap regions undetected. To overcome this limitation, we assume that the fraction of texture rich areas in the image is similar for both overlapping and non-overlapping regions. Thus, rather than computing the number of pixels in the overlapping convex hull, we consider the number of key features in the overlapping convex hull divided by the total number of key features in the image as an indicator of the overlapping fraction in the image. Hence, when no textures exist in a large part of the image, the denominator of the fraction is appropriately reduced, leading to a normalized estimate of the overlap. This fraction is then used instead of the ratio k/n in equation (7). This also normalizes the overlap fraction with respect to varying image sizes.

To overcome false positives in matching key features, we use majority logic based outlier rejection. We assume that the affine transformation of the input image required to align the overlapping region with the corresponding region in a previously known image is similar for all matched key features. We approximate this transformation by a difference vector and among all the matched features take only those P percent that lead to most similar difference vectors. The remaining $100-P$ percent matching features that have significantly varying difference vectors are rejected. The value of P is chosen high, such as 90% assuming that the outliers will be much fewer in number than the correct matches. Also, since false matches may occur between unrelated images, we threshold on the number of matching features to reliably detect if there is indeed any overlap.

3.2 Scalability Concerns

In addition to the above issues, the relevance computation method should also be scalable, since the number of mobile devices and the resultant number of images taken can be very large. We enhance scalability as follows.

First, we cluster the image data set into virtual neighborhoods. A virtual neighborhood (VN) is defined as a collection of images corresponding to a common region of the physical world. If the number of images within a single VN grows very large, it may be split into multiple VN's. The VN of an image may be determined from the metadata associated with the image. Metadata includes the device identity,

such as the mobile phone number or login ID, from which the image came, any tags applied to the image by the contributor, such as the application name, a file name, or context information, and also the GPS coordinates if the device provided them. For the purpose of this paper, we assume that an appropriate method to determine the VN is available; developing specific clustering methods is part of our ongoing work. An image is compared only with the images in its assigned VN for relevance computation, drastically reducing the required computation.

Second, we restrict the number of key features extracted from each image. Since the number of key features in an image depends on its size, we scale down each image to a standard size such that the number of key features extracted stays manageable. Also, when key features are extracted from a new image to compare against previously obtained images, we cache this key feature information along with the image. Thus, feature extraction needs to be performed only once for each image.

3.3 Relevance Ranking Algorithm

The above methods are made more precise in the following algorithm to compute vector $\mathbf{z}$ and then assign relevance using equation (7).

Algorithm ASSIGN_RELEVANCE

Inputs: Input image: X. Previously obtained images in VN: $Y_1, ..., Y_m$. Cached results for this VN: struct C.

Outputs: Relevance metric: $r(X)$. Updated VN image set: $Y1, ..., Ym, X$. Updated cache: struct C.

Initializations: $\mathbf{z}$ = null vector. $i = 1$.

Step 1: Scale Image. Resize image X such that its longer dimension is reduced to L_{std}.

Step 2: Extract Key Features. Extract key feature vectors: $f_1^x, ..., f_p^x$ where p is the number of key features of X.

Step 3: Match Features. From struct C, for $i-$th previously known image Y_i load feature vectors $f_1^y, ..., f_q^y$, where q is the number of feature vectors of Y_i. For each feature vector f_j^x of X, if a match is found with any of the features $f_1^y, ..., f_q^y$, add the pixel corresponding to the location of f_j^x in the image X to vector $\mathbf{z_{tmp}}$. If the number of pixels added to $\mathbf{z_{tmp}}$ is less than threshold δ_m skip to step 5.

Step 4: Reject Outliers. For all matched key feature pairs across images X and Y_i, compute the two dimensional difference vector between the pixel coordinates of the respective key features. Compute the mode of the difference vectors. Retain in $\mathbf{z_{tmp}}$ pixels from only P percent of the matched key features whose difference vectors are closest to the mode. Set $\mathbf{z} = \mathbf{z} \cup \mathbf{z_{tmp}}$.

Step 5: Iterate. Set $i = i + 1$. If $i \leq m$, goto Step 3.

Step 6: Compute Texture Normalized Overlap. Compute the convex hull of the locations of pixels in $\mathbf{z}$. Count the number of feature vectors among $f_1^x, ..., f_p^x$ that have a location within the convex hull, denote the count as k. Compute relevance metric $r(X) = 1 - k/p$. The factor $\log_2 B$ is ignored since all images in our data set have the same pixel-depth and this constant will only scale all relevance metric values by a constant amount; the relevance metric is thus a fraction between 0 and 1.

Step 6. Update Cache.: Add the feature vectors $f_1^x, ..., f_p^x$ to the caching struct C, add the new image to the VN's image data set, and increment $m = m + 1$.

4 Prototype Implementation

We now describe the realization of methods discussed above in a prototype implementation. We took several images using mobile devices at different locations, spread across two VN's. The values of the various parameters used in Algorithm ASSIGN_RELEVANCE were: $L_{std} = 400$, $P = 90\%$, and $\delta_m = 5$.

As an illustration, Fig. 2 shows six of a set of $m = 22$ images taken outside an office building, forming one VN. Note that some of these images are covering the same physical region though there is little exact pixel value match.

Figure 2. A few sample images from the data set.

Consider images (5) and (6) in the above set. A significant region of image (6) captures the same world view as already available in image (5), though from a different perspective and zoom. Thus, from a coverage point of view, a portion of image (6) is redundant[2]. Matching key features and the corresponding convex hulls are shown in Fig. 3. Note that while the perspectives are different, the convex hull overlap regions in the two images capture the same portion of the physical world. For this specific VN, assuming the images arrive in the order of their indices, relevance value is calculated for each image with respect to all images with a lower index. The values computed are shown in Fig. 4. One of the applications of the relevance value is selecting which images to use out of a given data set. Consider a query asking for what all is present in the scene - the most relevant images then are those that cover a significant portion of the

[2]The image may have different relevance value when the application considers other basis for relevance such as image resolution, or perspective difference for 3D reconstruction.

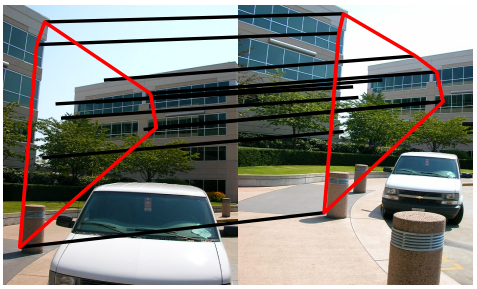

Figure 3. Overlap found between images (5) and (6).

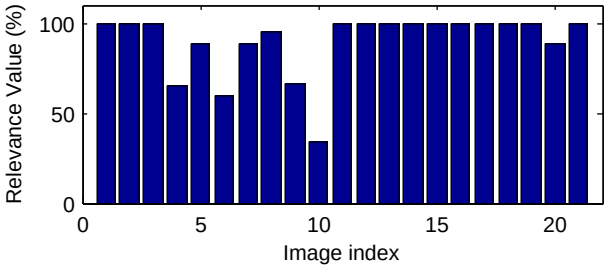

Figure 4. Relevance values assigned in one VN.

scene. Relevance values computed above can thus be used as a threshold to control the size of data that the application downloads to obtain images of interest. The relationship of data size to relevance is plotted in Fig. 5 for two VN's – the outdoor one shown above and an indoor one with photographs of an office kitchen facility.

5 Related Work

Applications of data collected by mobile devices have been considered before [2]. Methods focusing on sharing images or video from camera-phones were considered in [7, 8, 9]. We provide methods to share images with application specific relevance that may be used in the above projects to enhance the usefulness of data collected. The use of metrics similar to relevance has been previously made for selecting the value of information for a user in different parts of an image [10]. Information theoretic measures for quantifying the value of measurements have also been used in robotics and computer vision [11, 12]. We developed relevance metrics for coverage applications of image data.

6 Conclusions

There are currently no well-understood mechanisms to collect and use uncoordinated data from sensor networks of phone cameras. We proposed an information theoretic framework for assigning application specific relevance to the images provided by such networks, enabling applications to use the uncoordinated data in a resource-efficient manner.

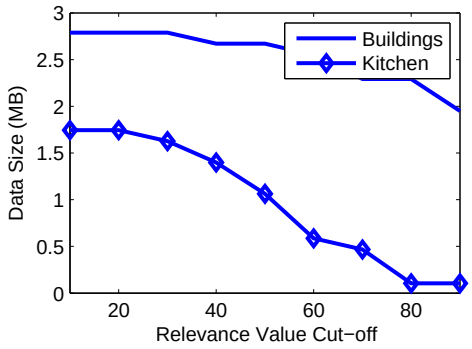

Figure 5. Data download size when images above a threshold relevance value are downloaded.

We presented a specific realization of this framework for a coverage based metric. All our methods were demonstrated on real image data and an illustration of how relevance can be used to control the download data size was also provided.

Our methods can further be used to design incentive mechanisms for data contributors such that more relevant data is collected. We also used virtual neighborhoods to cluster images for computation scalability. The automatic clustering of images into such neighborhoods is an open problem. We are addressing these issues in our ongoing work.

7 References

[1] E. Paulos. Mobile play: Blogging, tagging, and messaging. In *Ubicomp*, October 2003.

[2] A.J.B. Brush, T.C. Turner, M.A. Smith, and N. Gupta. Scanning objects in the wild: Assessing an object triggered information system. In *UbiComp*, Sept. 2005.

[3] Virtual earth. http://www.microsoft.com/virtualearth/.

[4] A9 Maps beta. http://maps.a9.com/.

[5] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. John Wiley & sons, 1991.

[6] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[7] R. Sarvas, E. Herrarte, A. Wilhelm, and M. Davis. Metadata creation system for mobile images. In *ACM Mobisys*, Boston, MA, June 2004.

[8] M. Davis, N.V. House, J. Towle, S. King, S. Ahern, C. Burgener, D. Perkel, M. Finn, V. Viswanathan, and M. Rothenberg. MMM2: mobile media metadata for media sharing. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 1335–1338, New York, NY, USA, 2005. ACM Press.

[9] N. J. McCurdy and W. G. Griswold. A systems architecture for ubiquitous video. In *ACM MobiSys*, pages 1–14, Seattle, Washington, 2005.

[10] N. Oliver and E. Horvitz. Selective perception policies for guiding sensing and computation in multimodal systems: a comparative analysis. *Comput. Vis. Image Underst.*, 100(1-2):198–224, 2005.

[11] B. Grocholsky, A. Makarenko, T. Kaupp, and H.F. Durrant-Whyte. Scalable control of decentralised sensor platforms. In *Information Processing in Sensor Networks*, pages 96–112, 2003.

[12] A.J. Davison. Active search for real-time vision. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, pages 66–73, Washington, DC, USA, 2005.