2012 IEEE International Conference on Robotics and Automation
RiverCentre, Saint Paul, Minnesota, USA
May 14-18, 2012

Efficient on-line data summarization using extremum summaries

Yogesh Girdhar and Gregory Dudek

***Abstract*— We are interested in the task of online summarization of the data observed by a mobile robot, with the goal that these summaries could be then be used for applications such as surveillance, identifying samples to be collected by a planetary rover, and site inspections to detect anomalies. In this paper, we pose the summarization problem as an instance of the well known k-center problem, where the goal is to identify k observations so that the maximum distance of any observation from a summary sample is minimized. We focus on the online version of the summarization problem, which requires that the decision to add an incoming observation to the summary be made instantaneously. Moreover, we add the constraint that only a finite number of observed samples can be saved at any time, which allows for applications where the selection of a sample is linked to a physical action such as rock sample collection by a planetary rover. We show that the proposed online algorithm has performance comparable to the offline algorithm when used with real world data.**

I. Introduction

This paper addresses the problem of producing a concise summary of the set of observations collected by a robot during a mission. In particular, we are interested in image-based summaries that recapitulate the statistically most distinctive images seen or representative locations visited.

The summarization problem can be defined in different ways, however, for applications such as surveillance and exploration, we are interested in summaries which are representative of the full range observed measurements, and not just focus on the mean. Popular clustering-based summarization techniques such as k-medoids or other clustering based techniques are not useful in this context as they ignore the outliers. This work poses the summarization problem as a sampling problem, where we would like to identify samples in a summary, which minimize the cost defined as the *maximum* distance between any observation point and the closest summary point. This corresponds to solving the k-center problem, which like k-medoids is also known to be NP hard [3]. Intuitively, this minimizes how surprised we will be by the data set, once we have seen the summary.

An online algorithm which can approximate the solution of the k-center problem, without saving all the observed samples has many applications. Consider the surveillance problem: we would like a robot to continuously monitor an area, and maintain a summary of what has been observed. This small summary of these observations could then be used by a human to quickly analyze the current situation, instead of having her continuously monitor what is being observed by the robot. Another example is the sample collection problem: we would like a robot to collect k rock samples for further analysis. We would like to have the k samples be representative of all the diversity of rocks observed, and include any outliers observed in the summary. Due to physical restrictions, we might not be able to go back to a previously visited location to pick a previously discarded sample. The online algorithm proposed in this paper is suitable for application in both the scenarios described above, and many more.

An offline algorithm often outperforms an online algorithm, but use of an offline algorithm is often infeasible in practice due to resource constraints and issues of timeliness. Moreover, an online algorithm can provide an anytime solution. In this paper, we use the offline algorithm to define an ideal result, and compare the performance of the proposed

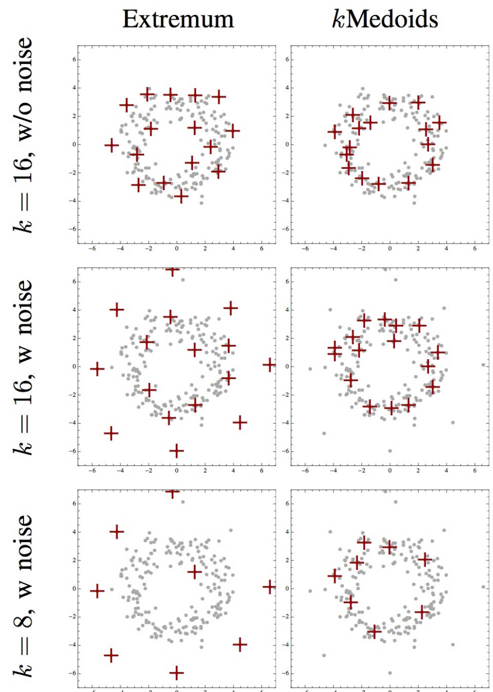


Fig. 1. Extremum vs k-Medoids Summaries. The dataset consists of 200 points generated randomly around a circle in $\mathbb{R}^2$. The summaries generated by the two algorithms are shown in the first row. Since there are no outliers in the dataset, the summaries seem similar. In the second row, we add 8 extra samples from a different distribution, which are all outliers in the context of the other points. Adding these outliers highlights the differences between the two strategies. We see that extremum summary favors picking the outliers, whereas the k-medoids summary ignores these outliers completely. In the last row, we reduce the summary size and see the differences exaggerated even more. The extremum summary is almost entirely made up of the outliers, whereas the k-medoid summary is only representative of the mean.

Y. Girdhar and G. Dudek are with Center of Intelligent Machines, McGill University, 3480 University Street, Montreal, QC, H3A2A7, Canada. yogesh@cim.mcgill.ca, dudek@cim.mcgill.ca

online technique.

Although the techniques discussed in this paper are applicable to almost any kind of quantifiable observation data, we primarily focus our efforts on image observations from a camera, and location data from a GPS sensor.

The main research contributions of this work are the following:

- A formulation of the offline summarization problem as an instance of the k-centers problem, and present a 2-approximation heuristic solution.
- An online strategy which approximates the offline solution, and analysis of upper bound on the number of samples picked by the algorithm in a given amount of time.
- Two different strategies to cull the samples in the summary to keep them of desired size, and show the difference between them.
- A novel way of combining multiple descriptors for online comparison of an incoming observation with the samples in the summary, even when the distance function between them is of unknown scale.
- Experiments with several different real world datasets which show that the performance of the online algorithm is comparable to the offline solutions.

II. Related Work

Generating summaries with observed image data is related to the problem of identifying landmark views in a view based mapping system. Ranganathan and Dellaert [13], have worked on the problem of identifying such set of landmark locations, and then build a topological map using them. They modelled each observed image using a SIFT bag of words histogram, which they use to learn a Multivariate Polya posterior and prior model. The KL diverence between these two distributions gives us the “Bayesian Surprise” [10]. If this surprise score is a local maxima above the mean, then the images was selected. This algorithm is an offline algorithm and hence not suitable for the online applications such as surveillance and exploration.

Another example is the work on view based maps by Konolige et al. [11]. In this work, the goal was to identify a set of representative views and the spatial constraints among them. These views are then used to localize the robot. One could consider the views as a summary, however it is not practical for online, life-long operation.

The seminal work by Cummins and Newman [2] on loop closure involves the ability to identify common geographic origins of images, which might not be completely similar; thereby producing a meaningful clustering of images, which could be used to generate summaries. This would indeed produce very detailed summaries, however, our focus is on producing a much sparser sampling of the observed data, and do it online.

Methods for video summarization have generally put emphasis on summaries for video data created by humans where significance is related to both temporal extent and audio content. For example, in Gong and Liu [6] video summaries were produced by exploiting a principal components representation of the color space of the video frames. They used a set of local color histograms and computed a singular value decomposition (SVD) of these local histograms to capture the primary statistical properties (with respect to a linear model) of how the color distribution varied. This allowed them to detect frames whose color content deviated substantially from the typical frame, as described by this model. Truong and Venkatesh [15] review many of these summarization strategies, and Valdes and Martinez[16] have presented a taxonomy for classifying these different techniques.

In our previous work we have explored individually, the idea of online summaries [4], and offline summaries [5]; however this work develops these ideas further and explores the relation between these techniques.

III. The Offline Problem

We are interested in generating our summaries online, however, we first look at the offline version of the summarization problem. Although an offline summary is much more costly to compute compared to an online algorithm, and might not be feasible in many scenarios, we use these offline summaries to measure the goodness of our online summaries.

Let each observation be a point in a high dimensional Euclidean space. We then pose the summarization problem as a sampling problem, where we would like to identify observations belonging to the summary set, which minimizes the maximum distance of any observation to its closest observation in the summary.

A. Cost Function

Given the set of all observations $\mathbf{Z} = \{Z_i\}$, and the number of desired samples k in the summary set, we would like to find the set $\mathbf{S} \subseteq \mathbf{Z}, |\mathbf{S}| = k$, such that the

$$\mathrm{C}ost(\mathbf{S}) = \max_i \min_j d(Z_i, S_j) \tag{1}$$

is minimized. This is like finding centers of k balls of smallest (but equal) sizes, which cover all the points in $\mathbf{Z}$. This is essentially the k-centers problem.

B. Approximate Solution

If the distance function obeys the triangle inequality, then not only is the k-center problem NP-hard, but Huse and Nemhauser [9] showed that α-approximation of this problem is also NP-hard for $\alpha < 2$ (i.e. for any approximate that guarantees an approximate this good).

Consider the greedy strategy presented in Algorithm 1, which we refer to as the *Extremum Summary* algorithm. We initialize the summary with an arbitrary observation, then in each iteration, we choose an observation which is farthest away from the observations in the current summary, and add it to the summary set. This algorithm has an approximation ratio of 2, and hence is likely the best we can do unless P=NP [7].

$\mathbf{S} \leftarrow \{Z_{\text{random}}\}$
$\mathbf{Z} \leftarrow \mathbf{Z} \setminus Z_{\text{random}}$
repeat
  $m \leftarrow \text{argmax}_i \ \ \min_j d(Z_i, S_j)$
  $\mathbf{S} \leftarrow \mathbf{S} \cup \{Z_m\}$
  $\mathbf{Z} \leftarrow \mathbf{Z} \setminus Z_m$
until $|\mathbf{S}| \geq k$
return S

Algorithm 1: EXTREMUMSUMMARY $(\mathbf{Z}, k)$. Computes a summary as a subset of input samples $\mathbf{Z}$, by greedily picking the samples farthest away from samples in the current summary.

C. k-Center Summaries Favor Outliers

Figure 1 highlights the characteristic difference in summaries generated by the extremum summary algorithm, and the k-Medoids algorithm. The dataset consists of 200 points generated randomly around a circle in $\mathbb{R}^2$. The summaries generated by the two algorithms are shown in the first row. Since there are no outliers in the dataset, the summaries seem similar.

In the second row of Figure 1, we add 8 extra samples from a different distribution, which are all outliers in the context of the other points. Adding these outliers highlights the differences between the two strategies. We see that extremum summary favors picking the outliers, whereas the k-medoids summary ignores these outliers completely.

In the last row of Figure 1, we reduce the summary size and see the differences exaggerated even more. The extremum summary is almost entirely made up of the outliers, whereas the k-medoid summary is still only representative of the mean.

Although a k-medoids summary might be useful when we want to model the mean properties of an environment, if however we are interested in identifying the range of what was observed, then an extremum summary is more useful since its objective function ensures that each observations is close to at least one of the summary samples.

IV. ONLINE SUMMARIES

Consider the task of deciding whether to include an incoming observation into the summary or not, immediately after its arrival. Broder et. al. [1] named the strategy of picking samples above the mean or median score of the previous picks as "Lake Wobegon" hiring strategies [1]. Variation of such a strategy have supposedly been used by companies like Google and GE to hire a continuous stream of employees. We take inspiration from this idea to pick our summary samples.

A. Picking Above the Mean

Given the current summary $\mathbf{S} = \{S_i\}$, we define the score of an observation Z_t, observed at time t as:

$$\text{Score}(Z_t) = \min_i d(Z_t, S_i). \tag{2}$$

[1]Named after the fictional town "Lake Wobegon", where according to the Garrison Keillor "all the women are strong, the men are good looking, and all the children are above average."[1]

Similarly, we can define the picking threshold γ as the mean score of the samples currently in the summary:

$$\gamma = \frac{1}{|\mathbf{S}|} \sum_i \min_{j, j \neq i} d(S_i, S_j) \tag{3}$$

Now for each incoming observation, we compute its score given the current summary, and then if the score is above the current picking threshold γ, we add it to the current summary. Algorithm 2 summarizes one iteration of this algorithm.

$\textit{threshold} \leftarrow \frac{1}{|\mathbf{S}|} \sum_i \min_{j, j \neq i} d(S_i, S_j)$
if $\min_j d(Z_t, S_j) > \textit{threshold}$ **then**
  $\mathbf{S} \leftarrow \mathbf{S} \cup \{Z_t\}$
end
return S

Algorithm 2: ONLINESUMMARYUPDATE $(\mathbf{S}, Z_t)$. Updates the summary $\mathbf{S}$ by picking the incoming observation Z_t if its score is above the mean score of observations already in the summary.

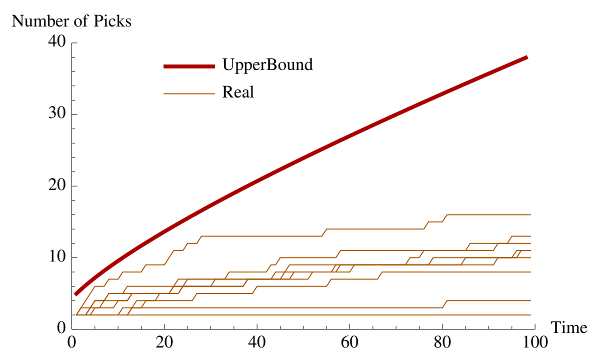


Fig. 2. Upper Bound on Picking Rate. The plot shows number of picks by Algorithm 2 over time for 10 random starting locations in the street view dataset. The upper bound defined in Equation 11 is shown in thick red.

B. Analysis of Picking Rate

To simplify our exposition and analysis without loss of generality, we assume that all observations lie on a high dimensional grid, and two consecutive observations only differ by one step on this grid, which corresponds to a distance of 1 unit.

Let the threshold after k picks be γ_k. Then, the score of the observation which leads to the next pick must be $\gamma_k + 1$. Using this, we can define the threshold after k picks recursively as:

$$\gamma_k = \frac{(k-1) * \gamma_{k-1} + \gamma_{k-1} + 1}{k} \tag{4}$$

$$= \gamma_{k-1} + \frac{1}{k} \tag{5}$$

$$\approx \log k. \tag{6}$$

Hence, the threshold grows as $\Theta(\log k)$ with the number of picks, and as a result the picking rate should slow down with time.

Given a threshold γ_k, it will take us at least γ_k time to pick a new sample, since in each time step the observation changes only by 1 unit. Hence, the expected time for the next pick is greater than γ_k.

Let T_k be the expected time for k picks. We then know that

$$T_k \geq T_{k-1} + \gamma_k \tag{7}$$
$$= \sum_{i=1}^{k} \gamma_i \tag{8}$$
$$\approx \sum_{i=1}^{k} \log k = \log k!. \tag{9}$$

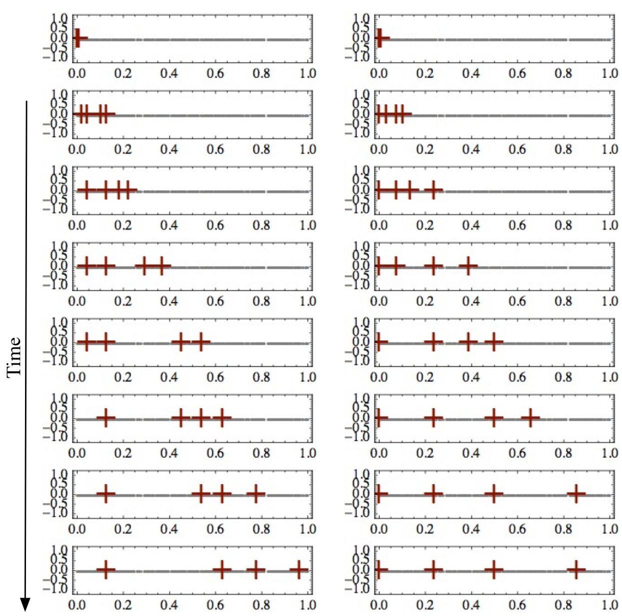
Using Sterling's approximation, we get,

$$T_k \geq k \log k - k \tag{10}$$

If we solve the above equation for equality, we can get an upper bound on the number of picks in a given time:

$$k(T) \leq \frac{T}{\text{ProductLog}\left(\frac{T}{e}\right)}. \tag{11}$$

Figure 2 shows a plot of this upper bound along with plots of number of picks over time for the street view dataset described in Section VI-B, with 10 random starting times.



(a) Strategy 1 (b) Strategy 2

Fig. 3. Discarding Strategies. There are 400 points in the dataset, distributed randomly over (0,1), and presented sequentially to the algorithm in monotonically increasing order of x axis values. The red '+' sign corresponds to the location of samples in the current summary. The successive state of the summary as time progresses is shown by successive rows. With strategy 1, we see that after a few time steps, the summary splits into two groups, and stays split for rest of the duration. This is because the minimum score sample to be discarded is always the 1st sample in the right group. Strategy 2 does not have this problem, and clearly has lower cost as defined by Equation 1. See text for more details.

C. Finite Summaries

Given infinite time, Algorithm 2 will give us a summary of infinite size. To make the summary size tractable, we can trim the summary when its size exceeds a desired size by removing a sample. We would like to do this in a way which ensure that the cost of the summary stays low.

```
threshold ← (1/|S|) Σ_i min_{j,j≠i} d(S_i, S_j)
if min_j d(Z_t, S_j) > threshold then
    S ← S ∪ {Z_t}
end
if |S| > k then
    TrimSummary(S)
end
return S
```

Algorithm 3: k-OnlineSummaryUpdate $(\mathbf{S}, Z_t, k)$. Updates the summary $\mathbf{S}$ of size k by picking the incoming observation Z_t if its score is above the mean score of observations already in the summary. If the summary size exceeds k, then we trim the summary.

Two simple strategies for identifying the sample to be discarded are the following:

Strategy 1: We can discard sample S_i if it has the lowest score defined as:

$$\text{Score}(S_i) = \min_{j,j \neq i} d(S_i, S_j). \tag{12}$$

This identifies sample with the smallest nearest neighbour distance.

Although this strategy seems reasonable, it fails in the simplest of cases as demonstrated in Figure 3(a). There are 400 points in the dataset, distributed randomly over (0,1), and presented sequentially to the algorithm in monotonically increasing order of x axis values. The red '+' sign corresponds to the location of samples in the current summary. The successive state of the summary as time progresses is shown by successive rows. We see that after a few time steps, the summary splits into two groups, and stays split for rest of the duration. This is because the minimum score sample to be discarded is always the 1st sample in the right group. Hence, this is not a good strategy.

Strategy 2: Summary trimming can also be modeled as summarization problem. The current summary is of size $k + 1$, from which we would like to select k representative samples. We can hence run the extremum summary algorithm, with $\mathbf{S}$ initialized using the last selected sample. This technique is immune to problem faced by strategy 1 described above.

Figure 3(b) shows progression of the summary on the same dataset, using strategy 2. Using this strategy clearly produces lower cost summaries as defined by Equation 1, and hence is the recommended strategy.

V. Distance Functions

Although the techniques described in this paper are generic enough to be used with almost any kind of sensor measurements, we keep our focus only on images, and gps readings.

Defining a distance function over GPS readings is trivial. One can simply take the Euclidean distance between the latitude/longitude measurements, which should work at most places on earth except the poles. It is however not so easy to define a distance function for images.

A. Visual Bag of Words

Coming up with a distance function over images is a non-trivial task, with an application dependent answer. For our experiments we use the *bag-of-words* representation of an image [14], where each image is represented by a histogram of frequency counts of visual words appearing in the image. This technique is known to work well in image matching and location recognition. Visual words are typically generated by clustering SIFT [12] features extracted from all images which we need to compare.

To compute the distance between two histograms h_1, h_2, we use KL divergence. KL divergence is not symmetric and hence is not a true distance metric, however, it can be made into a distance metric if we define the distance as

$$d(h_1, h_2) = d_{KL}(h_1||h_2) + d_{KL}(h_2||h1), \qquad (13)$$

where the function $d_{KL}(.||.)$ computes the KL divergence between the two distributions.

B. Combining Multiple Descriptions

Computing summaries by combining multiple descriptors requires that we must come up with a normalization strategy so that different descriptors can be given equal weight. For offline summarization, this can be done trivially by normalizing the distance matrix to ensure all distances are between (0,1]. However it is not possible to do this normalization in the online case since the maximum distance between two samples is not known. Instead, we propose to use the maximum distance between two samples in the summary as the normalization constant. This ensures that sample scores for all descriptor are of similar scale. Finally, we use of L_∞ norm to combine these different distances.

VI. Experiments

A. Location Based Summaries

Consider the task of collecting k samples such that no visited location is very far from one of the locations from which a sample was taken. We would like to do this even when the path taken by the robot is not known in advance. These samples for example could be rocks collected by a planetary rover as it is exploring a cave like environment, and we would like to have a good representation of the kind of rocks present in that area. Also due to energy, physical or logistics constraints, we might not able to come back to a previously visited location.

In such a scenario one can use Algorithm 3 to maintain a summary computed using location information. Each pick will corresponds to collecting the rock sample at that location, and removing a sample from the summary will correspond to dropping the rock. Figure 5 shows an example of such a summary. The subsampled dataset contains 1255 geo-tagged images taken by a car as it goes around a city

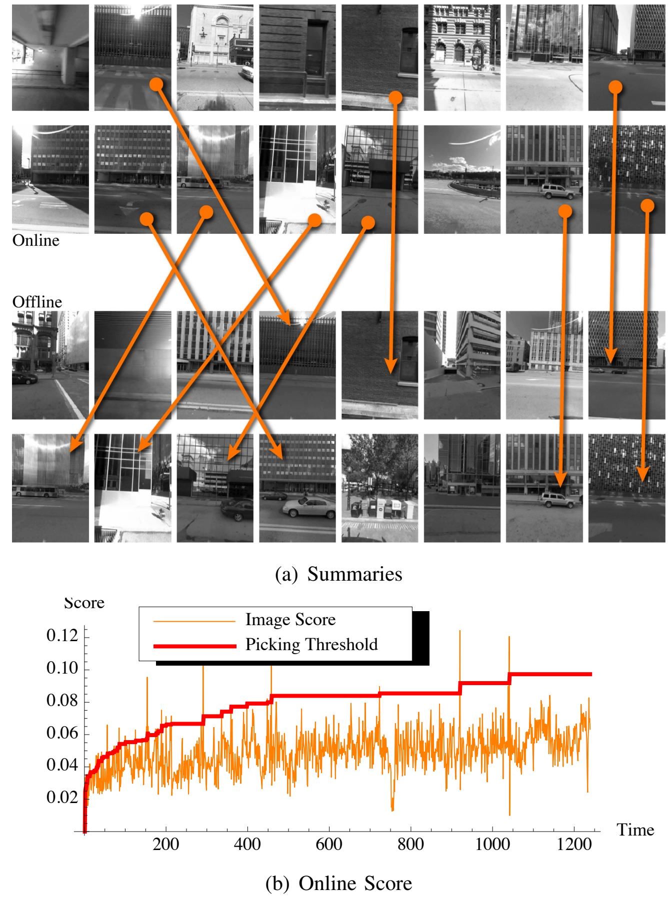
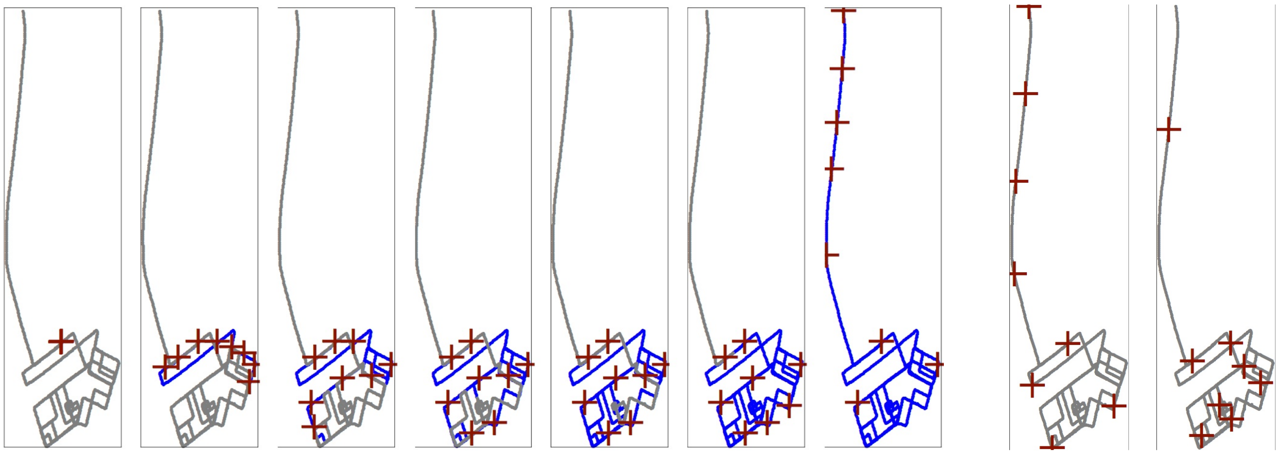


Fig. 4. Summarizing Images in the Street View Dataset. The dataset contains 1255 geo-tagged images taken by a car as it goes around a city centre. The bottom summary in (a) shows images selected by the offline extremum summary algorithm, and the top summary in (a) shows images selected by the online summarization algorithm described in Algorithm 3. Large number of matches indicated by the arrows suggest the effectiveness of the online algorithm in picking the same outliers as the offline algorithm, even in high dimensional spaces.

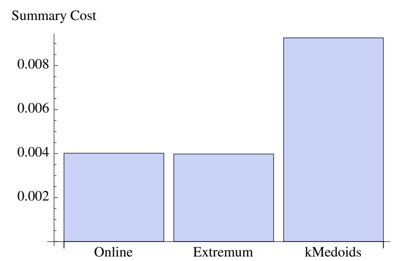
centre. The points in the dataset are shown using grey dots, and the points in the summary are shown using a red '+' sign. Figure 5(b) shows the extremum summary of these points. We see that the points are well distributed, and every point in the dataset is close to at least one of the summary point, whereas in the summary generated by the k-medoids algorithm shown in Figure 5(c), the points in the long tail of the dataset are very far from the closest summary point.

Figure 5(a) shows the evolution of the online summary generated using Algorithm 3. Points in blue correspond to portion of the path already traversed at that time. We start with all 8 summary points as the first 8 observed points. This is represented by the first column of Figure 5(a). Column 7 shows the final summary, which is similar to the extremum summary.

Figure 5(d) shows quantitatively the difference in performance of the different algorithms. The summary cost of the three summaries, was computed using Equation 1. We see that the online summary performs similar to the offline summary, and both are much better than the k-medoids



(a) Online (b) Extremum (c) kMedoids



(d) Summary Cost

Fig. 5. Location Based Summaries. The street view dataset contains 1255 geo-tagged images taken by a car as it goes around a city centre. The points in the dataset are shown using grey dots, and the points in the summary are shown using a red '+' sign. The summaries shown are generated using only the GPS data. (b) shows the extremum summary of these points. We see that the points are well distributed, and every point in the dataset is close to a summary point, whereas in the summary generated by k-medoids algorithm shown in (c), the points in the long tail of the dataset are much farther from the closest summary point. Figure (a) shows the evolution of the online summary generated using Algorithm 3. Points in blue correspond to portion of the path covered at that time. We start with all 8 summary points as the first 8 observed points. This is represented by the first column of (a). In the subsequent columns we see the evolution of the summary points as time progresses. Column 7 shows the final summary, which is similar to the extremum summary. (d) shows the summary cost of the three summaries, computed using Equation 1. Online summary performs similar to the offline summary, and both are much better than the k-medoids summary.

summary. Both the online and the offline extremum summary have a cost of about 0.004, whereas the k-medoids summary has a much higher cost of about 0.009.

B. Visual Summaries

Figure 4 shows the summaries generated for the same street view dataset described in the previous section, however, instead of summarizing over GPS readings, we generate summaries using the image data. Each image was described using a bag of words histogram. We used a static vocabulary of size 1000, computed by clustering SURF[8] features extracted from all the images.

The bottom summary in Figure 4(a) shows the images selected by the extremum summary algorithm. Many of the images selected by the extremum summary algorithm correspond to images of building with different repetitive patterns, occupying large portions of the camera's field of view. Such images would be represented by histograms with a sharp peaks at different locations when using the bag-of-words representation of an image. As a result, the KL divergence between these images is very high, and hence they are favored by the extremum summary algorithm.

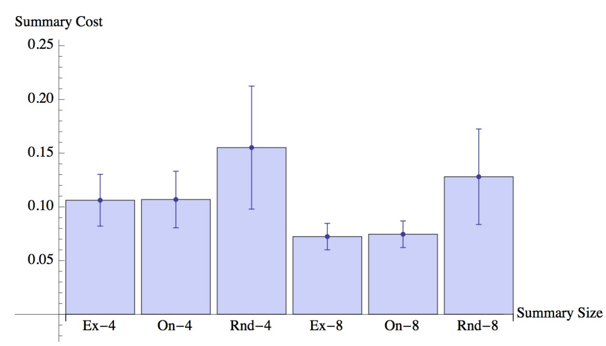


Fig. 6. Mean Summary Cost of Street View Dataset. We took 20 random subsequences of length 100 from the street view dataset, and then computed the summary cost as defined in Equation 1. Mean cost of both the offline(Ex) and the online(On) summaries is shown for summaries of size 4 and 8. Error bars corresponds to 1 stddev. For comparison, we also show performance of a random algorithm(Rnd), which just chooses the summary samples randomly. We see that performance of the online algorithm is similar to the offline algorithm, and both perform much better than the random algorithm. Increasing the summary size decreases the cost, which is expected.

The top summary in Figure 4(a) shows images selected by the online summarization algorithm described in Algorithm 3. We see that 9 out of 16 images in the online summary also exists in the offline summary. This indicates that the online algorithm is able to identify many of the same summary samples as the offline algorithm, even in the extremely high dimensional spaces corresponding to the bag of words histogram descriptions.

To quantify the measure of goodness of the online summaries, we took 20 random subsequences of length 100 from the street view dataset, and then computed the summary cost as defined in Equation 1 for each of them. Mean cost of both; the offline, and the online summaries for these sequences is shown in Figure 6. For comparison, we also show performance of a random algorithm, which randomly chooses its summary samples. We see that for a summary size of 4, performance of the online algorithm is similar to the offline algorithm, and both have an average cost of about 0.106, whereas the random algorithm has a much higher mean cost of about 0.155. For a summary size of 8, we see that the costs are lower for all the algorithms, which is expected since more summary points imply less distance between a sample and the closest summary point. Mean cost of the offline extremum summary for k=8 is about 0.068, and the online algorithm is slightly higher at about 0.071, and random is almost twice at about 0.137.

One of the fundamental difference between the offline and online algorithm is also the memory and computation cost. Offline algorithm requires the entire dataset to be in the memory, and hence as new data is added, it becomes slower linearly with time. The online algorithm does not have this problem, and has a constant memory footprint, proportional to the size of the desired summary, and a constant computation cost as new samples arrive.

VII. Conclusion

We have presented a novel online algorithm which approximates the solution of the k-centers problem. We use this algorithm to produce summaries which minimize the maximum distance of an observation from the closest summary sample. These summaries ensure that no observation is far from at least of one the observation samples in the summary, and hence are useful for applications such as surveillance, exploration and sample collection by a planetary rover

We have shown that the proposed online algorithm matches the performance of the near ideal but impractical-in-practice offline solution, when dealing with real world data consisting of images and location data.

In future we hope to explore other other variants of the online problem, such as one where we are allowed some amount of back-tracking, and see how that improves the quality of the summaries. Also we would like to explore other applications which might result from applying the proposed algorithm to other kinds of sensor data.

Acknowledgment

We would like to thank Google Inc. for generously allowing us to use their street view data.

References

[1] Andrei Z Broder, Adam Kirsch, Ravi Kumar, Michael Mitzenmacher, Eli Upfal, and Sergei Vassilvitskii. The hiring problem and Lake Wobegon strategies. In *SODA '08: Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1184–1193, Philadelphia, PA, USA, 2008. Society for Industrial and Applied Mathematics.

[2] M Cummins and P Newman. FAB-MAP: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6):647, 2008.

[3] M R Garey and D S Johnson. *Computers and intractability. A guide to the theory of NP-completeness. A Series of Books in the Mathematical Sciences.* WH Freeman and Company, San Francisco, Calif, 1979.

[4] Yogesh Girdhar and Gregory Dudek. ONSUM: A System for Generating Online Navigation Summaries. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems,* {*IROS*}, pages 746–751, October 2010.

[5] Yogesh Girdhar and Gregory Dudek. Offline Navigation Summaries. In *IEEE International Conference on Robotics and Automation (ICRA)*, May 2011.

[6] Yihong Gong and Xin Liu. Video Summarization Using Singular Value Decomposition. In *Proc. of CVPR*, pages 174–180, 2000.

[7] Teofilo F Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.

[8] Luc Van Gool Herbert Bay Tinne Tuytelaars. SURF: Speeded Up Robust Features. *ECCV*, 2006.

[9] Wen-Lian Hsu and George L Nemhauser. Easy and hard bottleneck location problems. *Discrete Applied Mathematics*, 1(3):209–215, 1979.

[10] Laurent Itti and Pierre Baldi. Bayesian surprise attracts human attention. *Vision Research*, 49(10):1295–1306, 2009.

[11] K Konolige, J Bowman, J D Chen, P Mihelich, M Calonder, V Lepetit, and P Fua. View-based maps. *RSS'09*, 2009.

[12] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), 2004.

[13] A Ranganathan and F Dellaert. Bayesian surprise and landmark detection. In *Proceedings of the 2009 IEEE international conference on Robotics and Automation*, pages 1240–1246. Institute of Electrical and Electronics Engineers Inc., The, 2009.

[14] Josef Sivic and Andrew Zisserman. Video Google: Efficient Visual Search of Videos. In Jean Ponce, Martial Hebert, Cordelia Schmid, and Andrew Zisserman, editors, *Toward Category-Level Object Recognition*, volume 4170 of *Lecture Notes in Computer Science*, pages 127–144. Springer Berlin / Heidelberg, 2006.

[15] Ba Tu Truong and Svetha Venkatesh. Video abstraction. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 3(1):3–es, February 2007.

[16] Víctor Valdés and José M. Martínez. A framework for video abstraction systems analysis and modelling from an operational point of viewhttp://www.springerlink.com/content/j350425v67p66712/fulltext.pdf. *Multimedia Tools and Applications*, 49(1):7–35, October 2009.