Efficient Viewpoint Selection for Urban Texture Documentation

Houtan Shirani-Mehr, Farnoush Banaei-Kashani, and Cyrus Shahabi *

University of Southern California, Los Angeles, CA 90089, USA
{hshirani,banaeika,shahabi}@usc.edu

Abstract. We envision participatory texture documentation (PTD) as a process in which a group of participants (dedicated individuals and/or general public) with camera-equipped mobile phones participate in collaborative/social collection of the urban texture information. PTD enables inexpensive, scalable and high resolution urban texture documentation. PTD is implemented in two steps. In the first step, minimum number of points in the urban environment are selected from which collection of maximum urban texture is possible. This step is called *viewpoint selection*. In the next step, the selected viewpoints are assigned to users (based on their preferences and constraints) for texture collection. This step is termed *viewpoint assignment*. In this paper, we focus on the viewpoint selection problem. We prove that this problem is NP-hard, and accordingly, propose a scalable (and efficient) heuristic with approximation guarantee for viewpoint selection. We study, profile and verify our proposed solution by extensive experiments.

1 Introduction

The advent of earth visualization tools (e.g., Google Earth™, Microsoft Virtual Earth™) has inspired and enabled numerous applications. Some of these tools already include *texture* in their representation of the urban environment. The urban texture consists of the set of images/photos collected from the real environment, to be mapped on the façade of the 3D model of the environment (e.g., building and vegetation models) for photo-realistic 3D representation. Currently, urban texture is collected via aerial and/or ground photography (e.g., Google Street View). As a result, texture collection/documentation is 1) expensive, 2) unscalable (in terms of the required resources), and 3) with low temporal and/or spatial resolution (i.e., texture cannot be collected frequently and widely enough).

To address these limitations, we propose leveraging the popularity of camera-equipped mobile devices (such as cell phones and PDAs) for inexpensive and scalable urban texture documentation with high spatiotemporal resolution. With such *participatory texture documentation*, termed PTD hereafter, a group of participants

* This research has been funded in part by NSF grants IIS-0238560 (PECASE), IIS-0534761, CNS-0831505 (CyberTrust), and the NSF Center for Embedded Networked Sensing (CCR-0120778). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

(dedicated individuals and/or general public) with camera-equipped mobile phones participate in collaborative/social collection of the urban texture information[1]. By enabling low-cost, scalable, accurate, and real-time texture documentation, PTD facilitates various applications such as eyewitness news broadcast, urban behavior analysis, and real-estate monitoring as well as critical emergency-response and disaster management applications (e.g., in case of earthquake, hurricane, and wildfire).

PTD can be implemented as a two-step process. In the first step, called *viewpoint selection*, a set of points in the urban environment is selected for texture collection. We call such points as *viewpoints*. Collection of the maximum possible urban texture should be doable by taking images at the set of selected viewpoints collectively. Because of the participatory nature of PTD, available resources (users' participation time) are usually limited and, therefore it is critical to minimize the number of selected viewpoints. In the second step, termed *viewpoint assignment*, considering the constraints and preferences of the participants, the selected viewpoints are assigned to different individuals for texture collection.

In this paper, we focus on the first step, viewpoint selection. We formally define the viewpoint selection problem and prove that it is an NP-hard problem by *reduction from* minimum set cover problem [1]. Therefore, optimal solutions for the problem are rendered unscalable as the extent of the urban environment grows large; hence, we propose an efficient heuristic, termed GVS, with approximation guarantee to select the viewpoints. GVS solves a given instance of viewpoint selection problem by *reduction to* an instance of the minimum set cover problem. Based on our experimental results, as compared to the naïve approach which selects the environment points by imposing a grid with the cell size of $c \times c$, GVS reduces the number of selected viewpoints by 83% on average over different values of c.

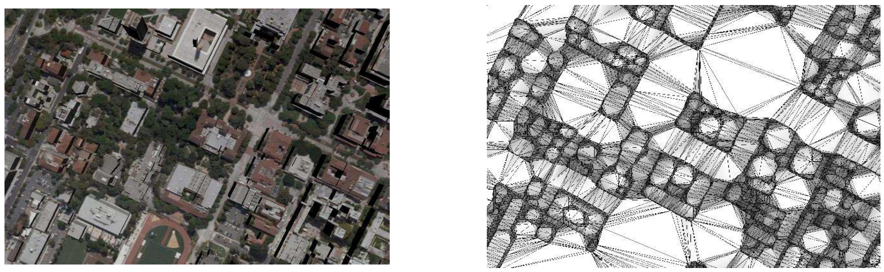
The rest of this paper is organized as follows. In Section 2, we formally define the viewpoint selection problem. We study the complexity of the problem in Section 3. Section 4 discusses our approach to solve the viewpoint selection problem. Section 5 presents the results of our extensive empirical analysis of the proposed solution. Finally, we discuss the related work in Section 6, and conclude in Section 7.

2 Problem Definition

An urban environment consists of various 3D elements such as buildings, trees and the terrain (Figure 1(a)).The environment can be modeled by any object-level model, i.e., a model in which the environment is represented by a set of objects. Here, without loss of generality, we assume a TIN (Triangulated Irregular Network) [2] model is used to represent all 3D elements in an urban environment (Figure 1(b)). The texture of the environment is the set of images mapped on the triangles of the TIN model. Correspondingly, participatory texture documentation (PTD) is defined as the process of collecting and mapping the texture onto the TIN model of the urban environment.

With PTD, participants/users move in the urban environment and make stops at selected viewpoints to take images. We assume users' movements are limited to an urban road network and, therefore, the viewpoints must be selected from the road network.

[1] A PTD system is currently under development at Information Laboratory in University of Southern California (see http://infolab.usc.edu/projects/GeoSIM).

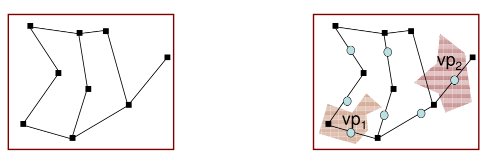


(a) (b)

Fig. 1. Representing an urban environment for texture documentation: (a) The environment and (b) its representation in TIN model.

The road network can be modeled by a weighted graph G, which we call the *road network graph* hereafter. We assume a user walks or drives on G and she can stop and take images at only certain points on G, denoted by P. Users are presumed to carry similar camera-equipped cell phones. A user takes a panoramic image at an assigned viewpoint v on the road network and therefore collects the maximum texture one can collect at v.

Given the aforementioned assumptions, we define the viewpoint selection problem as follows. Consider an environment E represented by TIN model with T as the set of TIN triangles, and a road network graph G in E. Assume the set of points on G at which a user can make stop and take images is denoted by P. We define $T' \subseteq T$ as the TIN triangles whose texture can be collected from the points in P (note that viewing the texture of all the triangles in T might not be possible from P, because P is only a limited subset of all possible viewpoints in E). Accordingly, we call a set $S \subseteq P$ a *texture covering* set, if collection of texture for *all* the triangles in T' is possible by taking images at the points in S. The viewpoint selection problem is defined as the process of finding a texture covering set V with minimum size. Figure 2 illustrates this process. Figure 2(a) shows the original road network graph G. In Figure 2(b), the selected viewpoints on G are shown as circles. For two viewpoints vp_1 and vp_2, the areas whose texture can be collected at either of them are highlighted.



(a) (b)

Fig. 2. Viewpoint selection: (a) Original road network graph (vertices are shown as squares); (b) Road network after viewpoint selection (viewpoints are shown as circles).

3 Complexity Result

In this section we prove that viewpoint selection problem is NP-hard by reduction from *minimum set cover* problem [1]. We first state the minimum set cover problem.

Definition 1. *Let $S = \{s_1, s_2, \ldots, s_m\}$ be a collection of finite sets, s_i's , whose elements are drawn from a universal set U. Let $F = \bigcup_{i=1}^{m} s_i$ where $F \subseteq U$. Minimum set cover finds a set C with minimum cardinality where $C \subseteq S$ and $\bigcup_{s \in C} \ s = F$.*

For example, assume $U = \{1, 2, 3, 4, 5, 6\}$ and $S = \{s_1, s_2, s_3\}$, where s_1=$\{1, 2\}$, s_2=$\{2\}$ and s_3=$\{1, 3\}$. Thus, $F = \{1, 2, 3\}$ and the minimum set cover is $C = \{s_1, s_3\}$. Minimum set cover problem is NP-hard. The following theorem proves that viewpoint selection problem is also NP-hard.

Theorem 1. *The viewpoint selection problem is NP-hard.*
Proof. We prove the theorem by providing a polynomial time reduction from minimum set cover problem. Towards that end, we prove that given an instance of the minimum set cover problem, denoted by SCI, there exists an instance of the viewpoint selection problem, denoted by VSI, such that the solution to SCI can be converted to the solution of VSI in polynomial time. Consider a given SCI having U as the universal set, $S = \{s_1, s_2, \ldots, s_m\}$ where $s_i \subseteq U$, and let $F = \bigcup_{i=1}^{m} s_i$. To solve SCI, we select a set $C \subseteq S$, with minimum cardinality, to cover all the elements in F. Correspondingly, to solve a VSI, we look for a $V \subseteq P$, with minimum cardinality, such that collection of texture for all the triangles in T' is possible from the points in V. Therefore, we propose the following mapping from SCI components to VSI components to reduce SCI to VSI. Suppose the universal set U corresponds to the set of triangles T. Each $s_i \in S$ is mapped to a point $p_i \in P$ as selection of sets in SCI corresponds to selecting the viewpoints in VSI. Finally, we map F to T'. The intuition behind the last mapping is that with SCI we want to cover each element $t \in F$ and accordingly we aim to cover the triangles of T' in VSI with texture. We next explain each mapping in detail.

For mapping S to P, we assume there exist a point p_i in E corresponding to $s_i \in S$. A road network may exist to connect the points in P. Next, we assume a triangle $tr_j \in T'$ exists corresponding to $t_j \in F$. tr_j is visible to $p_i \in P$ if and only if $t_j \in s_i$ and is invisible to the other points because of existence of obstacles in E. It is easy to observe that if the answer to VSI is the set V, the answer to SCI will be the set $C = \{s_i | p_i \in V\}$. This completes the proof.□

Based on the above theorem, viewpoint selection problem is NP-hard, which makes the optimal algorithms impractical. In the next section, we provide an efficient heuristic with approximation guarantee for the viewpoint selection problem.

4 Efficient Viewpoint Selection

Our proposed heuristic algorithm, termed GVS, is based on reduction to (not reduction from) the minimum set cover problem. Such a reduction enables us to adapt the existing algorithms for minimum set cover problem to solve viewpoint selection problem. Here, we first explain our proposed reduction, and thereafter describe our GVS algorithm.

4.1 Reduction

For each point $p \in P$, we define the *visibility* set $vs(p)$ as the set of triangles in T' that are visible from p; i.e., $vs(p) = \{tr \in T' | V(p, tr) = 1\}$, where $V(p, tr) = 1$ if tr is visible from p. Without loss of generality, we assume a triangle tr is visible from

p if every point of tr is visible from p. For example with $vs(p) = \{tr_1, tr_2, tr_3\}$ the triangles tr_1, tr_2 and tr_3 are visible from p. Therefore, a user standing at p can collect texture for these triangles. We construct the corresponding instance of the minimum set cover problem as follows. We define $F = \bigcup_{p \in P} vs(p)$ and $S = \{s = vs(p) | p \in P\}$. The universal set U can be any set such that $F \subseteq U$. For example, for an instance of viewpoint selection problem in which $P = \{p_1, p_2\}$, $vs(p_1) = \{tr_1, tr_2\}$, and $vs(p_2) = \{tr_3\}$, the corresponding set cover instance has $F = \{tr_1, tr_2, tr_3\}$, $S = \{vs(p_1), vs(p_2)\}$ and $U = F$. If the answer to the constructed minimum set cover instance is C, then the answer to the original viewpoint selection can be derived as V, where $p_i \in V$ if and only if $vs(p_i) \in C$.

Consequently, we can use any of the heuristics for minimum set cover to solve viewpoint selection problem. To develop our viewpoint selection algorithm (see Section 4.2), we are inspired by the greedy minimum set cover algorithm from [3], which has a linear running time of $O(\sum_{i=1}^{m} s_i)$ where $s_i \in S$. The greedy minimum set cover algorithm works by iteratively selecting the set $s_i \in S$ that covers the greatest number of remaining uncovered elements of F. This algorithm is guaranteed to result in a sub-optimal answer with an approximation guarantee of $\ln(n) + 1$, where n is the cardinality of the set $s_j \in S$ with the largest number of elements.

4.2 Greedy Viewpoint Selection (GVS)

Before presenting our viewpoint selection algorithm, we define our terminology. *Texture score* is a measure which represents the amount of texture that can be collected from viewpoints. A user standing at a point p can collect texture for all the triangles visible from p. Correspondingly, the texture score of p, denoted by $TS(p)$, is the number of triangles visible from p; i.e., $TS(p) = |vs(p)|$. Similarly, the texture score of a set of points S, $TS(S)$, is defined as the number of triangles visible to *any* point in S; i.e., $TS(S) = |vs(S')|, S' = \bigcup_{p \in S} vs(p)$.

The pseudocode of our viewpoint selection algorithm, termed Greedy Viewpoint Selection (GVS), is presented in Algorithm 1. The algorithm takes as input the triangles in $T' \subseteq T$ and the set of road network points P. We explain the logic of the algorithm as follows. First, GVS computes the visibility set of each point $p_i \in P$ (lines $3 - 9$). Thereafter, with a greedy approach the viewpoint p with maximum texture score is iteratively selected among all points in P, removed from P, and added to the result set V (lines $11 - 15$). Note that once p is added to V the corresponding triangles visible to p are excluded from T' (because they are covered); consequently, the texture score of V (and P) is updated at each iteration. The iteration correctly terminates when V becomes a texture covering set, i.e., $TS(V) = collectableTexture$.

To prove the correctness of GVS, note that all the triangles in T' can be texture mapped by at least one point in P. During each iteration, a point is selected and all the triangles visible to it are excluded. Therefore, the number of remaining triangles which cannot be texture mapped from the already selected viewpoints decrease, and correspondingly $TS(V)$ increases until the iteration terminates. The returned set V is the approximate answer to the viewpoint selection problem as it may have larger cardinality than the optimal viewpoint selection answer. Assuming that the optimal answer is V_{opt}, the size of the answer returned by GVS satisfies the following inequality:

$$\frac{|V|}{|V_{opt}|} \leq \ln(TS_{max}) + 1,$$

where TS_{max} is the texture score of the point with the largest texture score in P. The above inequality guarantees that the size of V is at most $\ln(TS_{max})+1$ times larger than the size of the optimal answer. This bound follows from the approximation guarantee of greedy minimum set cover algorithm.

Algorithm 1 $GVS(T', P)$

```
V = ∅;
vs = ∅;
for i = 1 to |P| do
    for j = 1 to |T'| do
        if V(p_i, t_j) = 1 then
            vs(p_i) = vs(p_i) ∪ t_j;    {Visibility set of each point is set}
        end if
    end for
end for
collectableTexture = |T'|
while TS(V) < collectableTexture do
    p = MaxTS(P);          {MaxTS(P) returns p ∈ P with maximum texture score}
    V = V ∪ p;             {p is added to the answer set}
    T' = T' − vs(p);       {T' is updated}
end while
return V;
```

5 Experiments

In this section, we evaluate our proposed solution by extensive empirical analysis. We first describe our experimental setting and then present the experimental results.

5.1 Experimental Methodology

We conducted our experiments using two real-world datasets. The first dataset, LA dataset, is the elevation data of Los Angeles area, from USGS (http://data.geocomm.com), covering a $10km \times 10km$ area. The second dataset, USC dataset, is the elevation data of University of Southern California campus covering a $1.5km \times 1.5km$ area. The road network data for LA dataset is acquired from NAVTEQ (http://www.navteq.com). For USC dataset, we assume all points with elevation zero comprise the road network; hence, the road network includes both roads and sidewalks.

To generate the TIN model for the urban environments covered by the datasets, first a set of ground positions with 3D coordinates among the total of approximately $180,000$ ($120,000$) points are sampled from LA (USC) dataset, and subsequently Delaunay Triangulation [4] is used to generate the TIN model. We can change the number of triangles and consequently the resolution of the texture to be mapped by changing the number of sampled ground points. Increasing (decreasing) the number of samples will increase (decrease) the number of triangles and hence the resolution of the texture to be mapped. The number of samples and that of the corresponding TIN triangles for the two datasets are shown in Figure 3. We selected fewer number of triangles to represent USC dataset in TIN model as it covers a smaller area as compared to LA dataset.

Moreover, to quantize the road network space, we impose a grid on the road network and use the intersection of road network segments and grid cells as the collection of road network points. This approach enables us to emulate various viewpoint selection restrictions by changing the granularity of the imposed grid. We assume user can stand at any of the resulted road network points. In our experiments we imposed grids with different granularity. The number of resulted points P by imposing different grids is shown in Figure 4 for each dataset. We denote the set P generated by imposing a grid with the cell size of $c \times c$ meters as P_c. Finally, with all of our experiments we assume a point further than 400 meters to a point p is invisible from p.

Number of Samples	2000	4000	6000	8000	$10,000$	$20,000$	$50,000$
Number of Triangles	4000	8000	$12,000$	$16,000$	$20,000$	$40,000$	$100,000$

(a) LA

Number of Samples	2000	3000	4000	5000	6000	7000
Number of Triangles	4000	6000	8000	$10,000$	$12,000$	$14,000$

(b) USC

Fig. 3. Number of samples vs. number of triangles for different datasets.

Grid	P_{20}	P_{40}	P_{60}	P_{80}	P_{100}
|P|	46303	11569	5152	2963	1825

(a) LA

Grid	$P_{2.5}$	P_{5}	$P_{7.5}$	P_{10}	$P_{12.5}$
|P|	57056	14317	6362	3637	2295

(b) USC

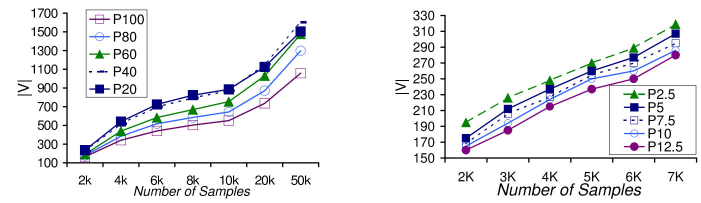
Fig. 4. Different values of $|P|$ in our experiments.

Our experimental system is implemented in Java, and runs on a typical Intel 2.66GHz PC with 3.25GB RAM. The operating system is Windows XP SP2. For each setting, we tested the algorithm by running it 10 times to compute the average values. Next, we present our experimental results.

5.2 Experimental Results

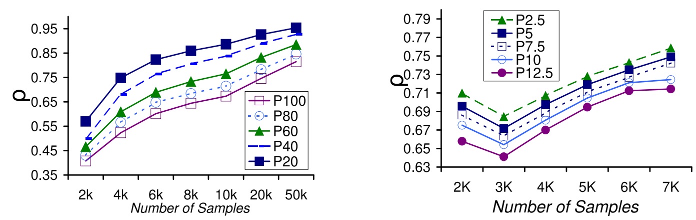
Based on our experiments, the optimal algorithm for viewpoint selection problem takes more than a day for $|P|$=25 (i.e., when P includes 25 randomly picked points from the road network imposed on LA dataset) and also its running time increases exponentially which makes it impractical. We observed that for $|P| < 25$, $\frac{|V|}{|V_{opt}|} \geq 95\%$, where $|V|$ ($|V_{opt}|$)is the number of viewpoints calculated by GVS (optimal) algorithm. Therefore, GVS clearly outperforms the optimal algorithm in efficiency and scalability, while providing almost optimal result. In this section, we study the effect of different parameters on the performance of GVS algorithm.

5.2.1 Size of Selected Viewpoints With this experiment, we evaluate the effect of resolution on the number of selected viewpoints, i.e., $|V|$. The result is shown in Figure 5 where each curve is generated for a specific P_c. Using GVS, on average $|V|$ is 83% (90%) less than $|P|$ for LA (USC) dataset over all the cases. This reduction significantly improves the scalability of any viewpoint assignment algorithm. As expected, as the number of samples grows $|V|$ increases, because the number of triangles which must be texture mapped increases. Similarly, increasing the number of road network points results in larger $|V|$, because more triangles become visible to the road network viewpoints.



(a) LA (b) USC

Fig. 5. $|V|$ vs. number of samples.



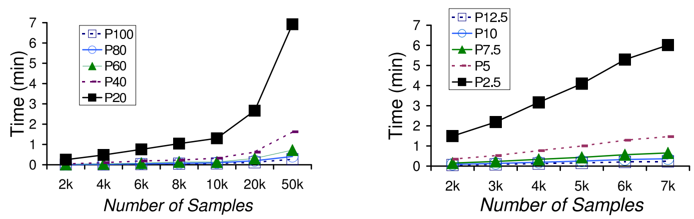
(a) LA (b) USC

Fig. 6. ρ vs. number of samples.

5.2.2 Collected Texture Viewing the texture for all the triangles in T (the collection of all the environment triangles) might not be possible from the points in P (see Section 2). Here, We measure the percentage of texture which can be collected from a set of selected viewpoints. This percentage is denoted by $\rho = \frac{TS(V)}{|T|}$. For example, $\rho = 50\%$ states that only 50% of the environment triangles can be texture mapped by taking images at the viewpoints. Figure 6 illustrates how ρ varies for different number of samples and road network points. For both datasets, increasing the number of samples increases the value of ρ since smaller triangles will be introduced and the chance that a triangle is visible to a point increases. The increase rate of ρ for USC dataset is slower as a large portion of this dataset is the ground points with the same elevation of zero for which increase in the resolution by raising the number of samples does not have much effect.

For a fixed number of triangles, increasing the number of road network points raises the value of ρ as more texture can be collected from more road network points. The maximum value of ρ for LA dataset is $\rho = 96\%$ which is obtained with 50000 samples and P_{20}. For the same number of samples if we choose P_{40}, ρ drops to 93%. This means that having much smaller P ($\approx 75\%$ decrease in the number of road network points) we can collect almost the same amount of texture. The reason is that for the finest grid , i.e., P_{20}, the distance between a point and its neighbor is very small and hence their visibility sets are almost the same. Therefore, the amount of texture which can be collected is not much more than the coarser grid of P_{40}. Based on our experiments and for a specific number of samples, the difference between the value of ρ for different number of road network points is at most 30% for LA dataset. This difference is at most 7% for USC dataset and less than 5% for 7000 samples.

5.2.3 Running Time We also measure the average running time of GVS algorithm. Our results are shown in Figure 7. As expected the running time of the algorithm increases when the number of samples or the number of road network points increases. Although the maximum value of running time for LA dataset is less than 7 minutes in the worst case (i.e., with 50000 samples and P_{20}) but as mentioned earlier we can collect almost the same amount of texture by having P_{40}. In this case, the running time is reduced to less than 3 minutes. Similarly, for USC dataset the running time is reduced to less than 2 minutes when using $P_{7.5}$ instead of $P_{2.5}$ which results in collection of only 5% fewer texture.



(a) LA (b) USC

Fig. 7. Running time vs. number of samples.

6 Related Work

Texture mapping based on the images acquired by cameras is extensively studied in the literature [5, 6]. For example, in [5] a system is developed to generate texture of building exteriors from mosaics of close-range photographs acquired with commodity digital cameras. Although with our work we also use the images taken by users to generate texture information, our focus is on selecting the minimum number of points in the environment from which gathering the complete texture of the environment is possible.

A second body of relevant work is the literature on the sensor deployment and sensing coverage with sensor networks. In [7, 8], the coverage problem is formulated as a decision problem to determine whether every point in the service area of the sensor network is covered by at least k sensors. On the other hand, with sensor deployment the goal is to maximize the coverage by proper sensor placement, somewhat similar to our viewpoint selection problem. However, most of the proposed approaches for sensor deployment assume simple sensing models with circular (omnidirectional or unidirectional) coverage for sensors [9–11]. While these models properly approximate the coverage of the sensors with typical sensing modalities (e.g., sound and temperature sensors), visibility coverage is more complex; hence, rendering these approaches inapplicable for visual sensor deployment. The most relevant work to our work is on visual sensor deployment [12, 13]. The art gallery problem is a classic work in this category [12]. Our work extends this category by studying the coverage at the object level, where objects can be modeled by TIN model (we emphasize that the choice of TIN model in our problem setting is arbitrary and other object models can be used equivalently to represent the urban environment). Moreover, we consider spatial restrictions in viewpoint selection (e.g., selected points must be on a road network). Most importantly, while the previous work has focused on solving the visual sensor deployment problem in a continuous space, with GVS we assume a discrete space for viewpoint selection to model restrictions on where users can take images.

7 Conclusion and Future Work

In this paper, we introduced and studied the problem of viewpoint selection for participatory texture documentation. We studied the complexity of the problem, and since the problem is NP-hard we proposed GVS as an efficient heuristic solution with approximation guarantee. Moreover, we showed the efficiency of GVS empirically by extensive experiments. As part of our future work, in short term we intend to extend GVS to consider the visual "quality" of the texture in addition to texture coverage. The visual quality of the texture can be affected by distance and/or view angle. This will also require a fuzzy setting for our texture-score measure. In long term, we plan to address the viewpoint assignment problem.

References

1. Chakravarty, S., Shekhawat, A.: Parallel and serial heuristics for the minimum set cover problem. J. Supercomput. **5**(4) (1992) 331–345
2. Fowler, R.J., Little, J.J.: Automatic extraction of irregular network digital terrain models. In: SIGGRAPH '79, New York, NY, USA, ACM (1979) 199–207
3. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, Second Edition. McGraw-Hill Science/Engineering/Math (July 2001)
4. Chen, G.H., Yu, M.S., Liu, L.T.: Two algorithms for constructing a binary tree from its traversals. Inf. Process. Lett. **28**(6) (1988) 297–299
5. Tsai, F., Lin, H.C.: Polygon-based texture mapping for cyber city 3d building models. Int. J. Geogr. Inf. Sci. **21**(9) (2007) 965–981
6. Guillou, E., Meneveaux, D., Maisel, E., Bouatouch, K.: Using vanishing points for camera calibration and coarse 3d reconstruction from a single image. The Visual Computer **16**(7) (2000) 396–410
7. Huang, C.F., Tseng, Y.C.: The coverage problem in a wireless sensor network. In: WSNA 2003
8. Meguerdichian, S., Koushanfar, F., Potkonjak, M., Srivastava, M.B.: Coverage problems in wireless ad-hoc sensor networks. In: INFOCOM. (2001) 1380–1387
9. Wu, C.H., Lee, K.C., Chung, Y.C.: A delaunay triangulation based method for wireless sensor network deployment. Comput. Commun. **30**(14-15) (2007) 2744–2752
10. Dhillon, S.S., Chakrabarty, K.: Sensor placement for effective coverage and surveillance in distributed sensor networks. Volume 3. (2003) 1609–1614 vol.3
11. Guestrin, C., Krause, A., Singh, A.P.: Near-optimal sensor placements in gaussian processes. In: ICML 2005
12. Lee, D.T., Lin, A.K.: Computational complexity of art gallery problems. IEEE Trans. Inf. Theor. **32**(2) (1986) 276–282
13. Hörster, E., Lienhart, R.: On the optimal placement of multiple visual sensors. In: VSSN 2006