ImgPricing: Everyone Can Earn Proper Rewards by Simply Taking Photos

Qinya Li, Fan Wu, Guihai Chen
Shanghai Key Laboratory of Scalable Computing and Systems, Shanghai Jiao Tong University, China
Email: qinyali@sjtu.edu.cn; fwu@cs.sjtu.edu.cn; gchen@cs.sjtu.edu.cn

***Abstract*—A high-quality and large-scale image collection is a fundamental demand in the 3D reconstruction. Crowdsourcing can help us collect lots of diversified images. However, it is not easy to attract people to do this task due to their self-interest. Moreover, the collected images are quality-varying. Those low-quality images may disturb the performance of reconstruction. To avoid low-quality images and lead participants to collect high-quality data, we take images quality into account when allocating rewards. The rewards of participants should be proportionable with their contribution. In this paper, we propose a pricing mechanism, called ImgPricing, to determine the reward of participants in 3D reconstruction system. We model the process of image collection as a cooperative game, and regard each participant's contribution and corresponding image quality as critical factors when allocating rewards. ImgPricing differs from traditional pricing schemes, such as Shapley value, as it introduces the image sequence as an indispensable factor. Finally, we implement our design on the Android platform and evaluate its performance. We use some metrics, such as computational efficiency, fairness and anti-interference, to evaluate ImgPricing and compare with other traditional schemes. Our analyses show ImgPricing is superior to others in terms of computational efficiency and fairness.**

I. Introduction

Crowdsourcing, as a new cooperative mode, has received extensive attention. It has gradually become an effective and acceptable method to handle tasks, especially the difficult tasks. It has been applied to various fields, such as floorplan reconstruction [1], [2], indoor navigation [3], POI labelling addition [4], and environmental monitoring [5].

In this paper, we try to apply crowdsourcing to another field: 3D reconstruction, which is a popular topic in the computer vision and aims to reconstruct 3D geometry models from large numbers of images. In recent decades, researchers have achieved some great success [6], [7], [8]. An accurate 3D model has a wide range of applications. For example, Google Earth tries to offer 3D models about cities and landscapes. City-scale 3D models are one of indispensable parts of the urban planning for government organizations. However, 3D applications need many high-quality images. The general collection method is to download from the photo-sharing web sites, such as Flickr.com. But, it seems to be not enough, especially when some items leave few images on the Internet.

This work was supported in part by the State Key Development Program for Basic Research of China (973 project 2014CB340303), in part by China NSF grant 61672348, 61672353, and 61472252, and in part by Shanghai Science and Technology fund 15220721300. The opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies or the government.

F. Wu is the corresponding author.

The popularity of crowdsourcing inspires us to collect images by crowds. However, people usually are self-interested, and will not spend time on the non-profit businesses. To encourage people to participate tasks, we consider to design a pricing mechanism, and provide a monetary incentive to attract participants. There already exist some works on incentive mechanisms [9], [10]. But, they only focused on motivating participants to collect high-quality data, and ignored the specific reward allocation. In this paper, we study the specific pricing mechanism that aims to return a proper reward to participants. A proper reward allocation mechanism is the premise to attract participants. Besides, to encourage participants to collect high-quality images, we take image quality into account when designing pricing mechanisms. Participants' rewards should be proportional to their contributions and the image quality. To our best knowledge, it is the first study to incorporate crowdsourcing into 3D reconstruction, and build the bridge between the image quality and the rewards of participants.

Pricing mechanism design in 3D reconstruction has many challenges. To start with, it is not easy to assess the image quality. 3D reconstruction concerns about the final 3D models instead of the quality of a single image. We cannot simply assess images through their resolution and covered contents. Each image has close relationship and mutual influence with others. Due to this connection, an image with a lower resolution may be as valuable as others with a higher resolution. Secondly, how to handle similar images. Participants will not avoid repetitive images voluntarily. Furthermore, since image valuation is unknowable, we cannot simply say those similar images are redundant and unnecessary. Perhaps, multiple similar images with a low resolution may complement each other, and provide high quality contents. Thirdly, a proper reward is a great motivation for participants. However, the reward allocation needs to consider both contributions and satisfactions of participants, which are both difficult to quantize. A pricing mechanism with fairness, rationality can inspire the enthusiasm of participants. So how to make pricing mechanism achieve fairness and efficiency is challenging.

To address these challenges, we present a crowdsourcing-based 3D reconstruction system including two modules: (1)a client module: participants interact with the server through it. (2)a server module: 3D reconstruction and pricing mechanism are run here. Here, we propose a novel pricing mechanism ImgPricing. It introduces a key factor: image order when determining rewards, which is neglected in traditional strate-

978–1–5386–2542–2/18/$31.00 © 2018 IEEE

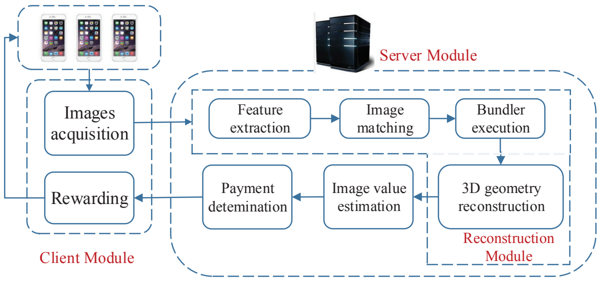


Fig. 1. System Architecture

gies. The image order has an impact on received payments, which can encourage participants to get information as soon as possible. With the increase of images, the server knows more about the object. It becomes more difficult for participants to collect those missing information, and the potential value of subsequent images may decline. Lastly, we experimentally evaluate ImgPricing and compare it with traditional pricing mechanisms from many aspects, like computational efficiency, fairness and anti-interference. The results show that ImgPricing has a better performance on fairness and effectiveness than others. In summary, our contributions are listed as follows:

- We quantify the image quality in the 3D reconstruction, and take the image quality into account when determining the contributions of participants.
- We propose a novel pricing mechanism ImgPricing, which regards the image order as a critical factor in the pricing mechanism design. And, it works better than traditional strategies on the fairness and effectiveness.

The rest paper is organized as follows. In Section II, we present the overview of our system, and review some reconstruction knowledge. In Section III and IV, we describe the system design and main techniques in details. In Section V, we show some evaluation results. In Section VI, we review some related works, and in Section VII, we give the conclusion.

II. Preliminaries

A. System architecture

Figure 1 describes the framework of our system, which is mainly composed of two modules: client module and server module. The former consists of a large number of participants with smart devices. They are paid to collect images. The latter is responsible for 3D reconstruction and pricing mechanism operation. The server releases n tasks to participants. Each task has its own information, including the location, deadline and so on. Participants can choose any task to do, and upload images before deadline. The collected images are stored in the database. Once the deadline passes, the server starts to reconstruct 3D geometries, execute pricing mechanism, and return participants rewards. Then the system ends.

B. 3D reconstruction

Bundler: Bundler is a structure from motion (SFM) system, which aims to reconstruct 3D geometries from unordered images. It was applied in [7] for the first time. It consists of three steps: image features extraction, image matching, and 3D reconstruction. The details are listed as follows:

1) Using SIFT detector (Scale Invariant Feature Transform) to extract feature points of images.
2) Using ANN library (Approximate Nearest Neighbors) to match images, and using multiview constrains to reduce mismatches. Matching pairs can form a track, which is the set of 2D features corresponding to the same 3D point.
3) Using SFM to recover the camera pose and scene geometry of each track. The goal of SFM is to minimize the sum of squared projection error:

$$g(X,R,T)=\sum_{i=1}^{m}\sum_{j=1}^{n}w_{ij}\left\|P(X_i,R_j,t_j)-p_{i,j}\right\|^2, \qquad (1)$$

where w_{ij} is an indicator to show whether track i is visible in I_j. If yes, $w_{i,j}=1$, if not, $w_{i,j}=0$. $p_{i,j}$ is the true point. $P(X_i,R_j,t_j)$ is the predicted point in I_j.

III. Client module design

Here, we mainly focus on the design details of the client and server modules. We try to study two major problems: (1) How to lead participants to collect high-quality and comprehensive data. (2) How to estimate the contributions of participants and the image valuation, and design a proper pricing mechanism.

The client module is the interactive platform between participants and the server. Participants receive a series of tasks $T=\{t_1,t_2,\cdots,t_n\}$, and choose any task to do by themselves. Each task $t_i=\{L_i,B_i,D_i,U_i\}$ contains the object L_i, budget B_i, deadline D_i and the current upload information U_i. U_i records the images that the server owns currently. Besides, U_i can provide some information for participants. For example, which part of the object is with too many images and which part is missing. It can lead participants to collect those missing information. After all, those scarce images carry important information, and can bring more payments to their owns. The server uses U_i to guide participants to collect more complete information, and avoid losing the information in corners.

In details, we divide the object into m parts, and give each part an initial budget $b_j=\frac{B_i}{m}$. The budget b_j varies with the image addition. We define $U_i=\{(u_1,b_1),(u_2,b_2),\cdots,(u_m,b_m)\}$, where u_j is one part of the object and b_j is the corresponding budget.

$$b_j=\frac{B_i}{m}-\gamma(N_j-N_{min})+\mu(N_{max}-N_j), \qquad (2)$$

where N_j is the image number of part j, N_{min} and N_{max} are the minimum and maximum number of images in m parts. γ, μ are the coefficients to adjust the extent of the variation of the budget. If the images in part j increase, b_j decreases, otherwise b_j increases. In brief, the image number of part j is roughly reflected in b_j. From b_j, participants can be aware that whether there are too many images, and then they can adjust their image collection strategies to maximize their profits.

Moreover, there exists another problem: how to determine which part the new image belongs to. We propose a modified

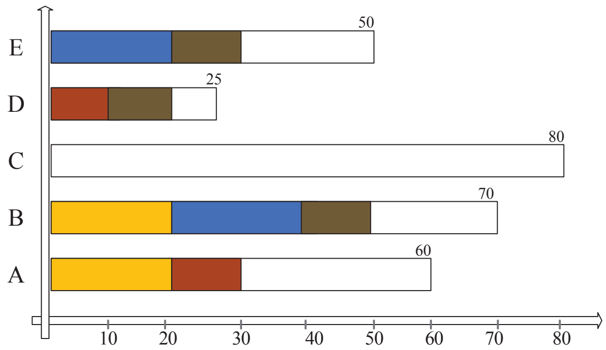


Fig. 2. An example of weight value method

clustering algorithm to solve this problem. Initially, due to the lack of the information, we can only make a rough judgement. With the increase of images, the clustering result will become better. Due to the space limit, we omit the pseudo-code here, and the main idea of our algorithm is described as follows:

(1) Let $A = \{a_1, a_2, \cdots, a_m\}$ denote m empty arrays. When image i comes, to match i with other images in A. If there exists an array a_j that has a high matching with i, putting i into a_j. Otherwise, putting i into an empty array.
(2) Repeat the step (1) until m arrays are all not empty.
(3) Using k-means to cluster all images in A into new arrays A, which aims to correct the previous rough matching.
(4) When a new image p comes, to match p with images in A, and put p into the best matching array. If p cannot find such a matching array, using k-means to cluster all images in A and p into a new A.
(5) Repeat (4) until the deadline or no image comes. Then, the algorithm ends.

IV. Server module design

In this part, we study the problem about the image redundancy, and pricing mechanism design.

A. Image redundancy

As we mentioned above, 3D reconstruction needs a large size of images. It is inevitable that there exist some images covering the similar contents. Too many redundant images may cause extra time consumption, and affect the reconstruction result. How to deal with these redundant images. Our basic idea is to convert those similar images into one big image, which contains the whole contents of them. We consider it is inappropriate to simply remove these redundant images. Perhaps, two similar images with a low resolution can complement each other, and provide a clearer information for the reconstruction. Thus, we preprocess these similar images.

We hope to improve the accuracy of information and reduce the redundancy while guaranteeing the integrity of data. Due to the space limit, we only describe some specific steps:

(i) In the client module, we divide the collected images into m parts. For each part, we use k-means to cluster images.

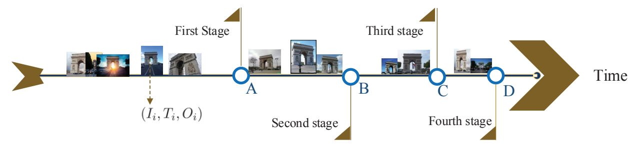


Fig. 3. the principle sketch of ImgPricing

(ii) After the clustering, to stitch the images in the same cluster. Image stitching can retain the original data and reduce image redundancy.
(iii) Lastly, each part consists of many preprocessed images.

B. Pricing mechanism design

In this part, we present a simple pricing strategy: weight value method, and a novel pricing mechanism: ImgPricing.

1) Weight value-based method: As we know, 3D point is recovered from the 2D features in images. An image consists of many features, which can be detected by SIFT. However, not all these features can be used. And, the image value is embodied in those useful 2D features. So the basic idea is that the contributions and rewards of images depend on their valid features. Assume image I_i has m useful features, and the image value is defined as $v_i = \sum_{j=1}^{m} \frac{r}{n_j}$, where n_j is the number of images including the jth feature. r is the reward equally shared with the whole useful features. The image value is the summation of the value of its useful features.

Figure 2 shows an example of this method. Horizontal axis represents the feature number of images, and vertical area is the images set. The chart shows the detected feature points in each image. Colored parts represent the useful features while blank parts represent the useless features. For example, image A has 60 features and only 30 features are useful. Moreover, the parts with similar color mean that the features generated from the same 3D points. For instance, A and B have 20 similar features. As shown, the reconstruction totally uses 60 features. Each useful feature has an equal reward $\frac{1}{60}$. Then, the images containing this feature share the reward equally. Here, the reward of image A is calculated as $\frac{1}{6} \times \frac{1}{2} + \frac{1}{3} \times \frac{1}{2} = 0.25$.

2) ImgPricing:approximate bitcoin method: ImgPricing is inspired by the principle of bitcoin. We model the process of image collection as a bitcoin mining. The sequence of images has an important impact on the reward distribution.

In early period, since the server owns little information about the object, any image seems to be important and valuable. They may be identified as contributors of great worth. And, high contributions bring high payments, which can motivate participants to collect images as soon as possible, and guarantee to satisfy the basic requirements of 3D reconstruction. After all, too few images are not sufficient to construct 3D geometries. With the increase of images, the server has already got much information, and the unknown information becomes less. Hence, it becomes more difficult for participants to collect those useful information and get a high payment, which is caused by the more rigid requirements

Algorithm 1: ImgPricing(N,λ,θ,m)

input : Image set N, two coefficients λ and θ, the number of time nodes m
output: The reward of each image r_i

1 Set evaluation function as $f = \lambda f_1 - \theta f_2$;
2 Sorting images according to T_i and O_i, $\{I_1, I_2, \cdots, I_n\}$;
3 Calculating the weight of each image $\{w_1, w_2, \cdots, w_n\}$;
4 Get the consequence of final reconstruction, $f(N)$;
5 **for** $i = \frac{1}{2}; i \geq \frac{1}{2^m}; i = i/2$ **do**
6   the image set S_i is uploaded in the $\frac{1}{i}$th phase, and the reconstruction results: $f_1(S_i)$ and $f_2(S_i)$;
7   **if** $0.95 f(N) \leq (\lambda f_1(S_i) - \theta f_2(S_i)) \leq 1.05 f(N)$ **then**
8   S_i is the whole images in the $\frac{1}{i}$th phase, and the last image l_i in S_i is the breakpoint;
9   **else**
10   continue to search the S_i;
11 **for** $i = \frac{1}{2}; i \geq \frac{1}{2^m}; i = i/2$ **do**
12   **for** $j \leftarrow l_{i-1} + 1$ *to* l_i **do**
13         $r_i = \frac{w_i}{\sum_{k=l_{i-1}+1}^{l_i} w_k} \times R \times i$;
14 output the reward of each image r_i.

of the server in the later stage. Therefore, a rational participant is likely to fulfill the task early.

Figure 3 shows the general framework of ImgPricing. The horizontal axis denotes the time-line. Images are uploaded in different time slots. Each image is denoted as (I_i, T_i, O_i), where I_i is the image information, T_i is the uploaded time, and O_i is the image order. In Figure 3, the points A, B, C, D are the time nodes (or say breakpoints) of the end of each phase. We annotate 4 phases in this graph. Actually, we can divide it into n phases. Each phase shares the half of remaining rewards. For example, the images uploaded before A share a half of rewards. The images in B share a quarter of rewards.

Based on the above principle, our first problem is how to determine the time nodes in the image collection process. Actually, this process is changeable and unfixed. How many images will be uploaded and the image order cannot be known in advance. Hence, it is difficult to select time nodes. Here, we choose the reconstruction result as the main basis to select time nodes. Time nodes need to meet the function: $f(S_\kappa) = \frac{1}{2^\kappa} f(N)$, where f is the reconstruction result, N is the whole image set, κ denotes the κth phase, and S_κ is the image set in the κth phase. However, through extensive experiments, we find it is hard to find the breakpoint that satisfies this function exactly. So we use $\approx$ to extend the search area. If there exists an image set S_j with the deviation between $f(S_j)$ and $\frac{1}{2^\kappa} f(N)$ no more than $\pm 5\%$, we consider $S_j = S_\kappa$. Therefore, the judgement function can be formulated as:

$$0.95 \times \frac{1}{2^\kappa} f(N) \leq f(S_k) \leq 1.05 \times \frac{1}{2^\kappa} f(N) \tag{3}$$

Additionally, another problem is how to define $f(S_\kappa)$. f is the reconstruction result, and it is reflected by two aspects: *more feature num* and *lower projection error*. The former makes f clearer, and the latter makes f more accurate. Thus, a good reconstruction result is with a high feature num and a low projection error. Thus, we formulate it as:

$$f(S_\kappa) = \lambda f_1(S_\kappa) - \theta f_2(S_\kappa), \tag{4}$$

where λ and θ are the coefficients. f_1 and f_2 are the effect of feature num and projection error respectively. Since the projection error makes a reverse effect, which means that a lower projection error makes f be higher. Thus, we use a minus sign. We use λ and θ to regulate the proportion of the impact of feature num and projection error on f. Through our experiments, we find that $\lambda = 0.8$ and $\theta = 0.2$ are suitable. Actually, the effect of feature num performs more clearly than projection error's on the result. We can easily realize the feature addition on the result, while projection error will not give such an obvious and intuitive result.

Algorithm 1 is the pseudo-code of ImgPricing. We locate the time nodes in line 5-10, and allocate rewards according to their weights in line 11-13, which is defined in Section IV-B1. The time complexity is $O(mn)$.

V. Performance Evaluation

A. Desirable Properties

- **Fairness**: It is an important property. Everyone hopes to get a proper and fair reward. Here, we use Jain's index as a primary metric. The original of Jain's index is defined as $J = \frac{(\sum_{i=1}^n x_i)^2}{n \sum_{i=1}^n x_i^2}$. However, the image order is a crucial factor in our system, and the image value is related with its order. Hence, the input of Jain's index is modified to $\frac{x_i}{w_i} \times o_i$, where x_i is the reward of participant i, w_i is the corresponding weight, and o_i is the image order. We have Equation 5 and rename it as the order-related index.

$$J = \frac{(\sum_{i=1}^n \frac{x_i}{w_i} \times o_i)^2}{n \sum_{i=1}^n (\frac{x_i}{w_i} \times o_i)^2}, \tag{5}$$

  In addition, to ensure the completeness of experimental results, we also use some other fairness index:
  - *Standard deviation*: It reflects the deviation degree among rewards.

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \frac{\sum_{i=1}^n x_i}{n})^2}. \tag{6}$$

  - *QoE fairness*: It's original definition is to quantify fairness by considering the quality of experience perceived by end users. We use it to quantify the satisfaction of participants. A high QoE means a great experience and a high satisfaction.

$$F = 1 - \frac{2\sigma}{H - L} \tag{7}$$

    where σ is the standard deviation, H is the upper reward and L is the lower reward.

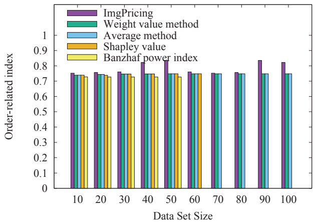


Fig. 4. Order-related index comparison

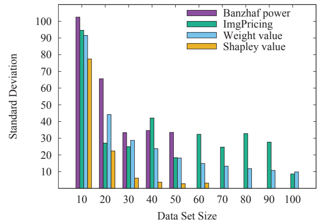


Fig. 5. Standard deviation comparison

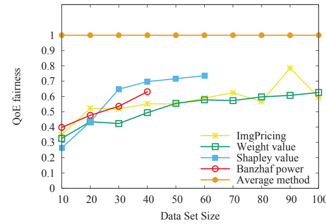


Fig. 6. QoE comparison

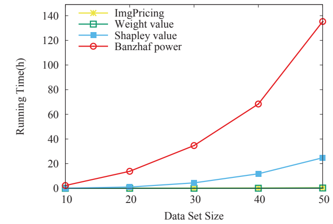


Fig. 7. Running time comparison

- **Computational efficiency**: To ensure the effectiveness of implementation, the reward allocation mechanism should be executed in polynomial time.
- **Anti-interference performance**: The reward allocation should not be influenced by bad or noisy images.

B. Experimental setup

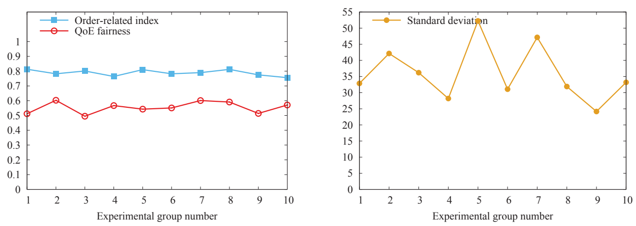
We install our system on Google Nexus 7 and recruit 10 participants to take part in image collection tasks. Each participant carries a Nexus 7. They will receive a series of tasks, and they can choose any task based on their interest. Once they decide, the server will send them the current details about this task. Moreover, participants will be told the reward allocation strategies in advance.

C. Pricing mechanism performance

We compare ImgPricing with other four strategies: average method, weight value method, Shapley value [11] and Banzhaf power index [12].

1) Fairness comparison: We evaluate the fairness performance according to the metrics mentioned above. Due to the time complexity of strategies, we only do the experiments with less than 60 images in Shapley value method and less than 50 images in Banzhaf power index.

- *The order-related index:* Figure 4 shows the results of the order-related index comparison. From the histogram, we conclude that ImgPricing has a higher order-related index, and it works better than others. ImgPricing has a slight fluctuation, while others keep almost unchanged. The main reason is that order-related index is related with $\frac{x_i}{w_i} \times o_i$. Here, $\frac{x_i}{w_i}$ is unchanged in each phase, because the image's reward is proportionable to its weighting in each phase. In traditional methods, we regard the whole collection process as a single phase, while ImgPricing divides it into n phases. So, the former can be simplified to $\frac{(\sum_{i=1}^{n} o_i)^2}{n\sum_{i=1}^{n} o_i^2}$. o_i is always from 1 to n. Thus, these traditional methods keep invariable. In ImgPricing, $\frac{x_i}{w_i}$ is different in each phase, which causes the fluctuation.
- *Standard deviation:* Figure 5 shows the results of standard deviation comparison. Because the results of average method are 0, we omit it here. A low standard deviation means a concentrated and even distribution, while a high one shows a decentralized and uneven distribution. As shown, the standard deviation almost decreases with the increase of images in these methods except ImgPricing.



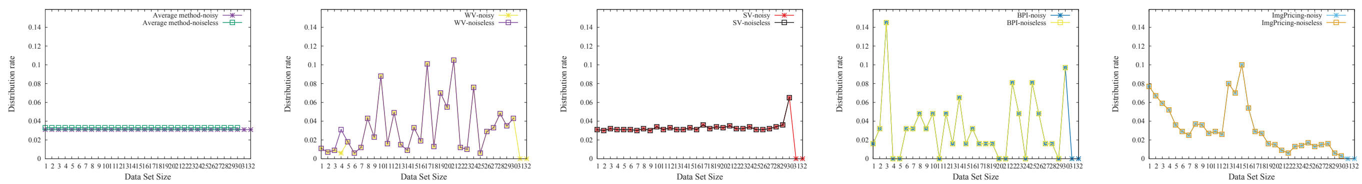
(a) (b)

Fig. 9. Source performance comparison

ImgPricing fluctuates all the time. It is because that ImgPricing distributes rewards depending on their value, and the image value is various and irregular. Thus, ImgPricing cannot present an equal or close allocation result. But it doesn't mean it is unfair or wrong.
- *QoE fairness:* Figure 6 shows the results of QoE fairness comparison. The average method performs a highest QoE, which is due to the even distribution. Participants usually have no knowledge about how much their images are worth. Thus, they may prefer to get the same reward as other. Conversely, due to the different ideas of other methods, participants make the different contributions on the reconstruction, and they receive different rewards. Hence, these four methods do not have a high QoE value.

2) Computational efficiency: Figure 7 shows the time consumption results of pricing mechanisms. The average method is too fast and negligible, so we omit it here. We observe that ImgPricing and weight value method are with a slow growth, which is reasonable and acceptable. However, Shapley value and Banzhaf power index are with an exponential growth, which is unacceptable especially with the increase of images.

3) Anti-interference performance: In this part, we evaluate the resistance of strategies to interference. We consider a good strategy can recognize the noise and distribute nothing to them. Thus, we put two noisy images in the experimental data. Figure 8 shows the reward distribution results. Due to the space limit, we only show the results of 30 images. We compare the distribution results in noisy and noiseless situation. The last two number denote two noisy images. In Figure 8, except average method, two noisy images have no impact on other images, and they are paid 0. Hence, we consider these strategies, except average method, are anti-interference ability.



(a) Avg method (b) WV method (c) SV method (d) BPI method (e) ImgPricing

Fig. 8. The reward distribution results of 30 images

4) The performance of different data sets comparison: In this part, we evaluate the results of different data sets on different fairness metrics. It is necessary that a good pricing mechanism maintains stable in different data sets. We prepare 10 data sets, and each one has 30 images. Figure 9 shows the results of 10 data sets. We put the results of order-related index and QoE fairness together due to their same interval [0,1]. The x-axis denotes 10 data sets, and the y-axis denotes the results. Figure 9(b) is the results of standard deviation. In Figure 9(a), these data sets almost perform similarly, and approximately fluctuate within a narrow range. Figure 9(b) shows a relatively high fluctuation, which is due to the diversity of rewards. In general, ImgPricing works stably in different data sets.

VI. Related work

Pricing mechanism: Pricing mechanism design is a popular topic in crowdsourcing. Many researchers have studied incentive mechanism and pricing mechanism for many years, and achieved much success. For example, Han *et al.*[13] proposed LIME to incentive multi-minded user participation. Singer *et al.*[14] proposed a framework to address the pricing and task allocation problems in an online crowdsourcing market. Singla *et al.*[15] proposed a no-regret posted price mechanism using the regret minimization in online learning. Luo *et al.*[16] proposed an incentive mechanism for crowdsourcing, which is based on all-pay auction. Our paper differs from above works. We aim to design a concrete pricing mechanism to attract participants. We take the data quality and participants' monetary incentive into account when determining rewards.

3D reconstruction: 3D reconstruction has been studied for long time in computer vision. Many researchers have proposed abundant methods to reconstruct 3D geometry from unorganized images. Snavely *et al.*[7] presented a photo explorer system to browse a large number of unorganized photos. Agarwal *et al.*[6] achieved the possibility of reconstructing a city-scale architecture in less than 24 hours. Furukawa *et al.*[8] proposed a new multiview stereo method to handle unordered images and use it in a city-scale reconstruction.

Unlike above 3D reconstruction studies, we focus on another problem in this field: how to get many high-quality images. A large size of images are the foundation of the reconstruction. Generally, people collect image from the Internet, which is not enough. Hence, we consider to collect images by crowds. Additionally, our work differs from the traditional crowdsourcing problems. We aim to design a concrete monetary incentive to motivate participants to join the crowdsourcing tasks.

VII. Conclusion

In this paper, we have presented a crowdsourcing-based 3D reconstruction system, which aimed to achieve a 3D model via collected image, and design a proper pricing mechanism to motivate participants to collect high-quality data. We proposed a novel pricing mechanism called ImgPricing, which took image quality and its order into account when determining rewards. Lastly, we have compared ImgPricing with other mechanism from multiple aspects. Through extensive experiments, the results have showed that ImgPricing have a better performance than others both in the fairness and effectiveness.

References

[1] R. Gao, B. Zhou, F. Ye, and Y. Wang, "Knitter: Fast, resilient single-user indoor floor plan construction," in *Proc. of INFOCOM*. IEEE, 2017, pp. 1–9.
[2] R. Gao, M. Zhao, T. Ye, F. Ye, Y. Wang, K. Bian, T. Wang, and X. Li, "Jigsaw: Indoor floor plan reconstruction via mobile crowdsensing," in *Proc. of MobiCom*. ACM, 2014, pp. 249–260.
[3] Y. Zheng, G. Shen, L. Li, C. Zhao, M. Li, and F. Zhao, "Travi-navi: Self-deployable indoor navigation system," *IEEE/ACM Transactions on Networking*, vol. 25, no. 5, pp. 2655–2669, 2017.
[4] H. Hu, Y. Zheng, Z. Bao, G. Li, J. Feng, and R. Cheng, "Crowdsourced poi labelling: Location-aware result inference and task assignment," in *Proc. of ICDE*. IEEE, 2016, pp. 61–72.
[5] "Green City Streets," http://www.greencitystreets.com.
[6] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski, "Building rome in a day," in *Proc. of ICCV*. IEEE, 2009, pp. 72–79.
[7] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: exploring photo collections in 3d," in *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3. ACM, 2006, pp. 835–846.
[8] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski, "Towards internet-scale multi-view stereo," in *Proc. of CVPR*. IEEE, 2010, pp. 1434–1441.
[9] D. Peng, F. Wu, and G. Chen, "Pay as how well you do: A quality based incentive mechanism for crowdsensing," in *Proc. of MobiHoc*. ACM, 2015, pp. 177–186.
[10] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing," in *Proc. of MobiCom*. ACM, 2012, pp. 173–184.
[11] L. S. Shapley, "A value for n-person games," *Contributions to the Theory of Games*, pp. 31–40, 1953.
[12] Y. Bachrach, E. Markakis, A. D. Procaccia, J. S. Rosenschein, and A. Saberi, "Approximating power indices," in *Proc. of AAMAS-Volume 2*. International Foundation for Autonomous Agents and Multiagent Systems, 2008, pp. 943–950.
[13] K. Han, Y. He, H. Tan, S. Tang, H. Huang, and J. Luo, "Online pricing for mobile crowdsourcing with multi-minded users," in *Proc. of MobiHoc*. ACM, 2017, p. 18.
[14] Y. Singer and M. Mittal, "Pricing mechanisms for crowdsourcing markets," in *Proc. of WWW*. ACM, 2013, pp. 1157–1166.
[15] A. Singla and A. Krause, "Truthful incentives in crowdsourcing tasks using regret minimization mechanisms," in *Proc. of WWW*. ACM, 2013, pp. 1167–1178.
[16] T. Luo, H.-P. Tan, and L. Xia, "Profit-maximizing incentive for participatory sensing," in *Proc. of INFOCOM*. IEEE, 2014, pp. 127–135.