Collaborative 3D Modeling by the Crowd

Ryohei Suzuki* Takeo Igarashi†

The University of Tokyo

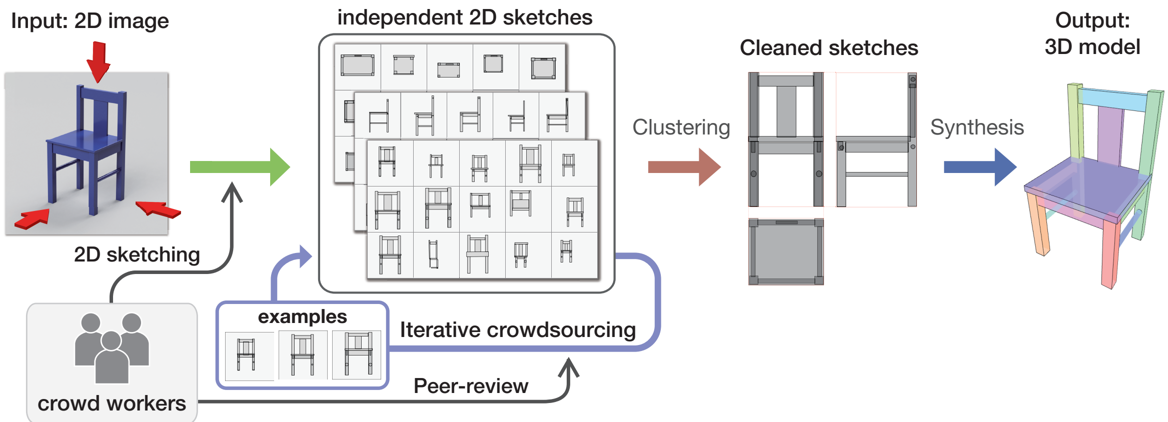

Figure 1: Overview of the proposed method. Each crowd worker draws a sketch representing an orthographic projection of a reference image. The proposed crowdsourcing workflow iteratively improves the submission quality by leveraging peer-reviewing. The gathered sketches are integrated into a clean multi-view projection by the system to generate a final 3D model.

Abstract

We propose a collaborative 3D modeling system that deconstructs the complex 3D modeling process into a collection of simple tasks to be executed by nonprofessional crowd workers. Given a 2D image showing a target object, each crowd worker is directed to draw a simple sketch representing an orthographic view of the object, using their visual cognition and real-world knowledge. The system then synthesizes a 3D model by integrating the geometrical information obtained from a collection of gathered sketches. We show a set of algorithms that generates clean line drawings and a 3D model from a collection of incomplete sketches containing a considerable amount of errors and inconsistencies. We also discuss a crowdsourcing workflow that iteratively improves the quality of submitted sketches. It introduces competition between workers using extra rewards based on peer-reviewing as well as an example-sharing mechanism to help workers understand the task requirements and quality standards. The proposed system can produce decent-quality 3D geometries of various objects within a few hours.

Index Terms: H.5.m [Information Interfaces and Presentation (e.g., HCI)]: Miscellaneous; I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques

1 Introduction

While recent advances in digital fabrication (e.g., 3D printing) have significantly expanded the interests of 3D object design among the consumer, modeling with professional authoring software is a difficult task for novice users. On the other hand, it is relatively easy for such users to prepare simple drawings or pictures that depict desired 3D geometries. Thus, we envision that a system or service that can generate 3D models from such simple drawings or pictures would be highly valuable.

In this paper, we present a 3D modeling system where crowd workers collaboratively work to produce a 3D model from a reference image. We deconstruct the complex modeling process into a set of simple tasks that can be independently processed in a few minutes by many nonprofessional workers, and we exploit their visuospatial functions to interpret the spatial arrangement of a still image. The system gathers many simple sketches of the target object in orthographic views from workers and then synthesizes the 3D shape by integrating the geometrical information obtained from the sketches.

Unlike typical automated methods for single-image 3D reconstruction, this system can process various types of reference images, ranging from pictures to scribbles, as long as they can be interpreted uniformly by humans. Since humans can infer the existence of hidden elements in an image using their real-world knowledge, the system can reproduce parts of the target object that are not explicitly present in the reference image. Compared to simply outsourcing a modeling job to a single professional, our method can provide rapid results with stable quality. This is because our method does not require any single person with specialized skills, and a vast amount of nonprofessional crowd workers from all over the world can contribute to the task.

Our system recruits crowd workers using a microtask marketplace such as Amazon Mechanical Turk (MTurk) and then allows them to execute the sketch task on our web-based system. Recruited workers are given instructions for sketching a reference image in an orthographic view using a simple vector-drawing interface that provides the basic functionalities of creating and transforming 2D primitives, such as rectangles and ellipses. A collection of sketches gathered from the workers is integrated into a single drawing for each orthographic view, and then the associations between 2D primitives in each angle are inferred to enumerate combinations corresponding to the 3D parts. Finally, 3D primitives, such as cuboids, cylinders, and ellipsoids, are generated from the combinations, and the 3D model of the target object is obtained. This algorithm can synthesize

*e-mail: ryoheis@acm.org
†e-mail: takeo@acm.org

shapes that are composed of simple primitives, such as chairs, tables, buildings, as well as simplified versions of complex objects.

In order to improve the quality of submitted sketches, we designed a crowdsourcing workflow for boosting workers' engagement. We provide the sketch task as a competition where the workers who submitted high-ranked sketches could obtain extra monetary rewards. Additional crowd workers are recruited to evaluate the quality of submissions (peer-reviewing). High-ranked sketches selected via the peer-review process are shared in the subsequent competition as examples to help the workers to understand the instructions and quality expectations. This iterative strategy promotes both collaboration and competition between workers, which researchers have highlighted as important factors for successful creative crowdsourcing [23].

We confirmed the feasibility of our method by showing the synthesis results for multiple inputs, including pictures and illustrations. We compared our method against outsourcing to a single professional worker by using an agent service. The result shows that our method costs more but can shorten turnaround time and communication overhead. We also observed that the submission qualities improved during the iterative crowdsourcing process. These results support our claim that human computation can be used for highly creative purposes, such as 3D modeling, and can serve as a rapid and widely available computational resource for users who need low-polygon 3D models.

The contribution of this research is summarized as follows. First, we present a practical approach using microtask crowdsourcing for creating a 3D model from a single reference image. Secondly, on a more abstract level, we show the possibility of applying human computation to a highly complex and creative undertaking through task deconstruction. Lastly, we demonstrate a design for an iterative crowdsourcing workflow that motivates both competition and collaboration among workers to improve the results.

2 Related Work

2.1 Human Computation for Creative Purposes

Ahn [27] first proposed the idea of human computation as "a paradigm for utilizing human processing power to solve problems that computers cannot yet solve." Human computation has been applied to problems that require complex cognitive functions, such as semantic image annotation [28], language understanding [2], and 3D manipulation of objects [6]. A systematic survey of human computation methods was given by Quinn and Bederson [24].

Several works have aimed at applying human computation to creative purposes. Soylent is a word processor that provides proofreading and editing functions by allowing crowd workers to review the texts [2]. Human computation of visual perception was used to yield depth annotation of still images [11] and aesthetically pleasing shading of line drawings [12]. Higher-level visual cognition was used to analyze the aesthetic preference distribution in the parameter space of digital content manipulation, such as photo color correction, to provide visualization and optimization [18]. While these preceding works are limited to the *analysis* and *refinement* of existing data by human computation, our research is unique in that it applies human computation to generate a complete 3D model from scratch.

2.2 Content Creation by Crowdsourced Workflow

Many crowdsourcing systems for producing attractive content have been developed; for example, narrative creation is a representative area of crowdsourced content creation. The *A Million Penguins* project [22] used a wiki community to write stories via the collaboration of many visitors. Since most creative activities (e.g., article writing) require diverse working efforts, such as data collection and organization, frameworks for supporting such complex and interdependent tasks with microtask crowdsourcing have been proposed [16, 19]. To our knowledge, we have realized 3D content generation with *microtask* crowdsourcing for the first time by substituting sketch gathering from nonprofessional workers for the complex 3D modeling process.

Crowdsourced systems generally must contend with the workers' weak engagement. In order to address this problem, a variety of studies have examined quality control mechanisms and dedicated workflow to achieve the maximum potential of the workers. Crowd vs. Crowd [23] is a team-based crowdsourcing workflow that was designed based on the idea that collaboration and competition between workers are both important to achieve better outcomes in the creative process. Our proposed workflow is also aimed at promoting both competition and collaboration to improve submission quality. The contribution of our approach lies in achieving the promotion of human computation involving only microtasks–sketching and reviewing–while Crowd vs. Crowd requires the long-term participation of workers in the design process.

Yu and Nickerson [29] developed a collaboration platform for designing furniture using a human-based genetic algorithm [17]. They outsourced the design evaluation to human judgment and also had workers make new designs by combining two designs that had been highly rated in a previous generation. Though their system produced highly unusual designs, practicability of the designs was not improved across generations. Our system has a similar iterative method utilizing previous submissions as examples but ensures quality improvement across iterations by providing clear criteria for evaluation and motivates workers by introducing competition.

2.3 3D Modeling from 2D Image Input

Sketch-based interfaces have been studied to make the 3D modeling task easy and quick for people without professional skills [15, 30]. Rivers et al. designed a modeling method using silhouettes viewed from three angles as the inputs specifying 3D shapes [25]. Our method uses a similar strategy to utilize a 2D interface for 3D modeling but instead outsources the silhouette drawing process to crowd workers. While their method requires the user to explicitly specify the relations between elements in each view, our system integrates many incomplete and inconsistent sketches into a set of clean orthographic drawings and infers the correspondence between them. 3-Sweep [4] extracts models composed of generalized cylinders from still images employing automatic edge detection and human assistance to annotate geometry using interactive interfaces. Since our method uses a simple and classical 2D drawing interface, it does not require the user (worker) to learn special 3D editing operations, such as a sweep.

3D reconstruction from a single or a limited number of pictures has been a fundamental problem of computer vision. Geometric constraints such as vanishing points have been used as clues for inferring the spatial relationship between planes and edges in a picture [13]. Machine-learning based [7] and model-driven [14] methods have been reported to produce reasonable results for some types of inputs, but a general method to generate 3D models from still images has not been proposed. Utilization of human perception for 3D modeling has been highlighted in recent years. Gingold et al. proposed a depth estimation system employing human computation that can produce plausible results for images with limited geometrical clues [11]. Our method further focuses on exploiting the cognitive functions and real-world knowledge of human workers by providing a more challenging task. It allows our system to correctly reconstruct parts of objects that are not explicitly present in the input images.

3 System Overview

Our system is provided as an application that the user (customer) and workers can access via a web browser. The basic workflow for using the system is as follows. First, a user uploads a reference image to the system, which then generates the necessary tasks and registers job offerings on a microtask marketplace. In this step, the user overlays the image with arrows indicating the front/side/top directions of the object. He/she is also required to input the number of parts that compose the object for ensure the complexity of sketch which workers will make. Twenty workers who accept the job are guided to a web page that provides the interface for executing the tasks. They receive rewards after finishing their submissions. The 3D model synthesis is processed when a certain number of sketch submissions are collected, and the resulting model is then provided to the user. If the user is satisfied with the quality of the result, the workflow is finished. Otherwise, the system starts the next session of iterative crowdsourcing to gather more sketches to improve the quality. The system manages the jobs in the microtask marketplace, including monetary transactions.

3.1 Sketch Task

Each worker engaging in the sketch task is given an image of a 3D object and an arrow indicating one of the orthographic viewing directions. They are required to draw the shapes of *all* the parts of the object in the specified view, including the partially or completely hidden parts in the original image. In order to make the task processable within a few minutes as a *microtask*, we assign only one sketch per worker. If we requested that each worker draw three views, it would be difficult to recruit a sufficient number of workers. Instructions for the task are provided via verbal description and some sketch examples for simple objects, such as LEGO blocks. As described in the following sections, workers in the second or later stages of iteration can also see as examples the high-ranked sketches of the same reference image submitted in the previous stage.

Each worker performs the sketch task using a 2D vector drawing interface (Figure 2). They can create and transform primitives (rectangles and ellipses) via mouse operation in a conventional manner. A sketch is required to have at least a specified number of primitives, decided by the user. Every worker who submitted a sketch is given \$.36 as a basic reward, irrespective of the quality of the submission. Workers are informed that they could get an extra reward of \$.18 if their sketches are evaluated in the top 20% among all submissions in the same stage. The evaluation criteria are described as follows: "the sketch contains all the parts of the object"; "the arrangement of the elements is correct"; and "the sketch does not contain any element that does not exist in the target object." We set the basic reward equivalent to the minimum wage in the United States (\$7.25/hr) assuming the required time for finishing the task is three minutes, observing Dynamo payment guidelines for research on MTurk [1].

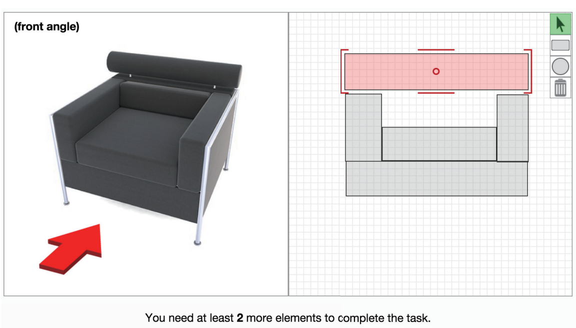

Figure 2: Sketch task interface. Workers can create and transform rectangles and ellipses via mouse operation. The red arrow indicates the viewing direction from which the sketch should be drawn.

3.2 Peer-Review Task

Human computation is generally erroneous and generates highly noisy output because of the weak engagement of workers and the intrinsic inconsistency in human work. Therefore, providing a quality control mechanism is essential for obtaining meaningful results. In our system, the quality of submitted sketches is evaluated by other workers using a crowdsourced peer-review process. Reviewers are recruited from a microtask marketplace in the same way as the sketch task. They are shown the instructions for the sketch task and are asked to score the submitted sketches based on how sketches meet the criteria using a 7-level scale (1: poor; 7: excellent) as shown in Figure 3. Each sketch is scored by 20 reviewers, and each reviewer scores 20 sketches submitted in a single stage. The final score for a sketch is calculated as the average of the scores given by the reviewers. All reviewers uniformly receive \$.24 as the reward.

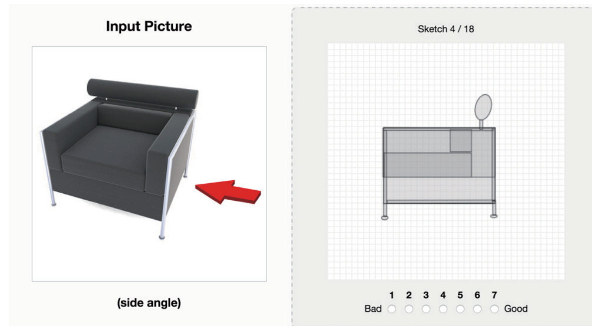

Figure 3: The review interface where the workers score each sketch from 1 (poor) to 7 (excellent). Peer-reviewers are given the same reference image and sketch instructions as the sketching workers.

4 Iterative Crowdsourcing

In this section, an iterative crowdsourcing workflow for quality improvement is described. In the early stage of the research, we conducted a preliminary study of crowdsourced sketching, which uses the same sketch interface described above, with some simple input images and real crowd workers recruited from CrowdFlower. We did not introduce competition and extra rewarding in the study, and therefore the submitted sketches were not evaluated and shared with other workers. We paid \$.05 for each submission regardless of the quality. In the study, we found two problems with the gathered sketches that resulted in wasting resources.

The first problem was the existence of many invalid sketches. Fewer than 50% of the submissions in the study were drawn correctly. Though some of the invalid submissions were completely meaningless, a considerable proportion of the rest seem to have resulted from misunderstanding of the task instructions. For example, about 25% of the submissions were carefully drawn but from the wrong view. Reducing the number of cheating workers is a fundamental problem of crowdsourcing, and there have been analyses of such workers and methods for excluding their influence [9, 10]. However, in this work, we focus on helping workers to interpret the task correctly, to reduce the amount of invalid submissions.

The second problem was the existence of incomplete sketches; most of the valid sketches still had missing parts and a considerable mismatch of element arrangements. Even with a large collection of valid sketches, a fine model cannot be synthesized if most of the sketches lack certain parts. This problem might be chiefly the result of the lack of motivation among workers to draw better sketches rather than merely meeting the minimum requirements. In order to improve the completeness of the submitted sketches, it would be wise to introduce additional incentives for workers to draw sketches accurately. Provision of the expected quality standard is also desirable to allow workers a sense of the required effort.

4.1 Utilizing Example-Sharing and Competition

Our goal in designing the workflow was twofold: to help the workers understand the task instructions for reducing invalid submissions and raising the minimum standard of sketch quality and to encourage workers who can draw valid sketches to make more effort. We introduced the following concepts for achieving these goals.

For the first goal, we provided workers with other workers' good sketches for the same input. By referring to the examples, elementary mistakes, such as confusion of viewing direction, can be avoided, and more detailed task requirements (e.g., drawing occluded parts) also become more comprehensible. Even some workers reluctant to put effort into the tasks can produce adequate submissions by merely following the style of the examples. In summary, we promoted implicit collaboration between workers through introducing an *example-sharing* mechanism.

For the second goal, we introduced competition between workers for extra rewards to incentivize them to create higher-quality sketches. Submissions were scored based on their completeness through a peer-review process, and then the authors of the high-ranked submissions received a monetary bonus. As shown in the literature, competition improves participants' engagement with a job, when appropriate rewarding is provided [3,23]. The example-sharing mechanism described above is also beneficial when combined with competition, as it enables workers to grasp the expected quality required for receiving the bonus.

4.2 Iterative Competition Workflow

In order to introduce both the sharing of good examples and competition between workers in an integrated manner, we designed the iterative workflow as follows (Figure 4):

1. Gather the initial 20 sketches from crowd workers without showing examples (first stage of the competition).
2. Score the submitted sketches by crowdsourced peer reviewing.
3. Pay extra rewards to the submitters ranked in the top 20%.
4. Stop if the user (customer) was satisfied with the synthesized model from current submissions.
5. Gather an additional 20 sketches while showing the high-ranked sketches in the previous stage.
6. Go to step 2.

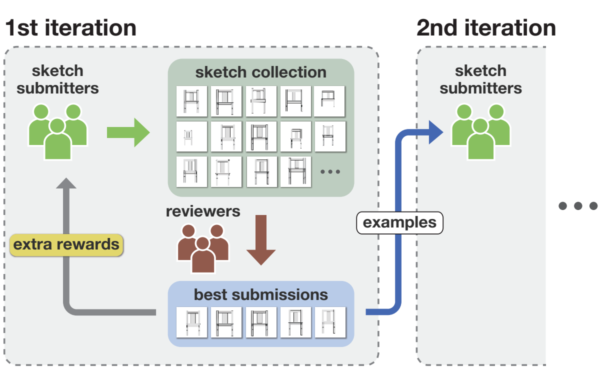

Figure 4: Overview of the iterative crowdsourcing workflow.

Essentially, different individuals are involved in each stage, and the workers in the review task are recruited independently from the sketch task in the same stage. This workflow gradually improves the average submission quality stage-by-stage. This refinement process can be considered a variation on the genetic algorithm [17], where both selection (peer-review) and genetic operations (sketching by referring to previous winners) are conducted by human workers.

5 3D Model Generation

In this section, a 3-step process to synthesize a 3D model from the gathered sketches is described. In the first step, we extract the valid sketches from the collection utilizing a clustering-based filtering strategy. The second step is integration of sketches into a clean projection (consensus drawing) for each angle. We classify the primitives in the sketches representing the same part of the target object, and then calculate the average shape of them for each part. The third step is inference of the relationship between 2D primitives in the consensus drawings of the three views. We make a collection of triplets of primitives where each one corresponds to a part in the target object. Finally, we generate a 3D primitive, such as a cuboid or a cylinder, for each triplet and then the final result is obtained.

5.1 Valid Sketch Selection

Invalid sketches exhibit significant diversity in their appearance, while valid sketches are similar to each other. We designed a valid sketch selection method using the clustering based on their similarity; the largest cluster is expected to contain only the valid ones.

Dissimilarity Measure A dissimilarity measure that specifies the distance between two sketches should be defined to handle the difference of valid and invalid sketches appropriately. We use modified Hausdorff distance (MHD) [8] for this purpose, because it has desired properties for comparing diverse sketches, such as robustness against parallel shifts. We first normalize the sketches by resizing the axis-aligned bounding box of them into a fixed size squares to cancel the variation in drawing size, which is followed by rasterizing them into 128×128 binary edge images. Then we calculate the MHD between them to obtain a distance matrix of all sketches. Table 1 shows the MHD values between the example sketches in Figure 5.

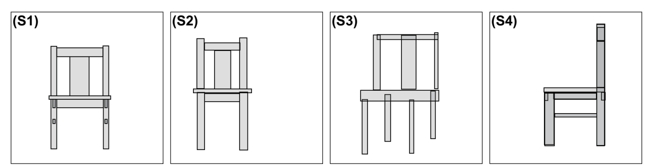

Figure 5: Example sketches submitted by crowd workers.

Table 1: MHD between sketches in Figure 5. The values correctly reflect the similarity between S1 and S2, fewer similarity between S1/S2 and S3, and the dissimilarity between S4 and the others.

	S1	S2	S3	S4
S1	0	2.637	9.025	19.50
S2		0	7.319	17.54
S3			0	15.58
S4				0

Sketch Clustering After calculating the MHD between the sketches, we cluster them based on the distance matrix. We adopted Medoidshifts [26] for the clustering method. For our situation, this algorithm has advantages over other well-known clustering methods, such as k-means and hierarchical clustering: it can be applied to a general set where only the distance matrix is available and the mean is not defined, and it does not require the number of clusters beforehand. Medoidshifts requires a constant value h specifying the bandwidth of kernel function; we empirically chose 3.5. A clustering result of several example sketches using MHD and Medoidshifts is shown in Figure 6. The largest cluster (cluster 1) contains only the sketches correctly drawn from the same view. The sketch in cluster 6 can be seen as valid, but was excluded because its shape was quite different from other sketches in cluster 1.

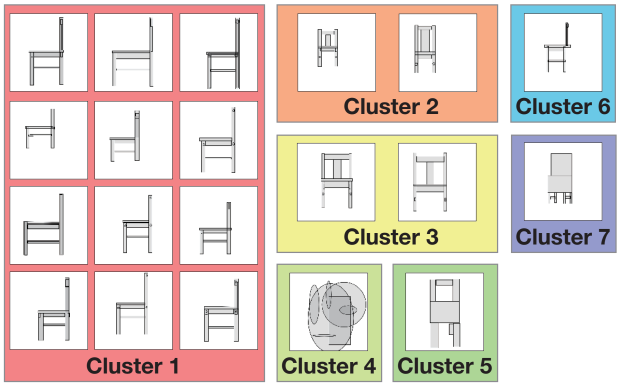

Figure 6: Clustering result of 20 example sketches submitted from crowd workers for the side view of a chair.

5.2 Primitive-Wise Consensus Making

Valid sketches extracted by clustering still have errors and fluctuations in the shapes and the arrangement of primitives. However, as shown in the previous studies [12], an *average* of a large number of inputs would give a plausible approximation. We calculate the consensus of a set of valid sketches by the following strategy.

First, we make clusters of 2D primitives from all the sketches, each of which corresponds to a single part of the target object. We then simply average the transformation variables (translation, scaling, and rotation) of primitives in each cluster. By applying this "primitive-wise consensus making" to all the clusters of primitives in a collection, we obtain a clean drawing. MHD and Medoidshifts are used as in the previous step to classify all the primitives in the sketches. Rectangles and ellipses are clustered separately, and clusters with fewer than four elements are discarded to suppress the noise. Figure 7 shows an example result of consensus making for 12 sketches. We can see that the fluctuation in the arrangement of primitives in the collection is successfully cancelled in the result.

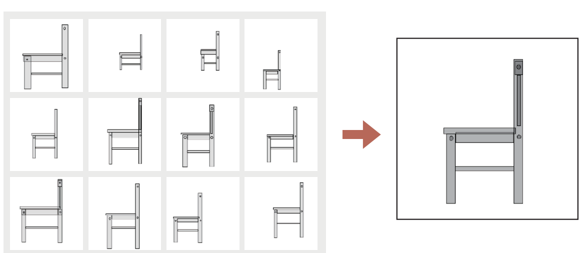

Figure 7: Example of a consensus drawing made from multiple valid sketches for the side view of a chair.

5.3 Triplet Extraction and 3D Model Generation

After obtaining consensus drawings for the three orthographic views, we infer the relationship between the elements of them, because the three drawings are created independently from the submissions of different workers. The goal is extracting triplets each of which consists of 2D primitives from each orthographic view and corresponds to a single part in the target object. We designed a simple algorithm that extracts a triplet with the cost value under a threshold.

The system first scales the three drawings to ensure the consistency between the bounding boxes of the drawings. We define the aspect ratios of the front/side/top sketches by h_{front}/w_{front}, w_{side}/h_{side}, and w_{top}/h_{top}, where w_{front} is the bounding box width of the front view and h_{side} is the height of the side view, for example. We uniformly multiply the ratios to make the product of them equal to 1, and then decide the new dimensions of the bounding boxes by following the new ratios and matching the lengths of shared edges between drawings. The cost of a triplet is defined as the sum of the costs of the three pairs of primitives it contains. The cost of a pair is defined as the square-sum of the distance between the segment endpoints of the bounding boxes on the shared axis of two views. For example, if a primitive in the front view has its segment (0, 50) and a primitive in the side view has a segment of (10, 40) along the vertical axis, the cost between them is calculated as $(0-10)^2+(50-40)^2 = 200$. We calculate the scores of all the possible triplets made from the elements in three views, and then accept those with costs lower than a threshold. We empirically chose 1,000 pixels^2 as the threshold value when canvas width is 400 pixels, but some complex input images require a greater value to reconstruct all the parts in the target object. Currently, the system asks the user to decide the value by observing the synthesis results.

Each accepted triplet is used to generate a 3D primitive whose projection to each view matches the elements of the triplet. This process is similar to that used by [25], but we currently accept limited kinds of 2D primitive combinations. For example, a triplet made from three axis-aligned rectangles is converted to a cuboid. The list of the supported combinations is shown in Table 2. We also support cuboids and ellipsoids rotated around one axis (Figure 8). 3D rotation about an arbitrary axis is not supported, because it causes complex orthographic projections that cannot be drawn using our sketch interface. More elaborate techniques for 3D solid reconstruction from 2D projection line drawings have been studied [5]. We could increase the kinds of supported 3D primitives using these existing methods, but we do not focus on doing so in this paper.

Table 2: The list of supported 3D primitives.

view 1	view 2	view 3	3D primitive
rectangle	rectangle	rectangle	cuboid
rectangle	rectangle	ellipse	cylinder
ellipse	ellipse	ellipse	ellipsoid

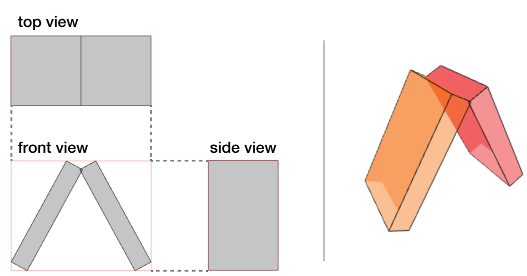

Figure 8: Left: an example of orthographic views containing objects rotated around an axis. Right: synthesis result for the example.

6 Results

We tested the presented system using six example reference images containing simple objects (Figure 9). Authors played the role of a customer, and real crowd workers were recruited to process the tasks. The results are shown in Figure 10, and the statistics for the required resources are shown in Table 3. While there are several missing parts and mismatches, we can see that the synthesized models contain the major elements of the target objects in the right arrangement. Some of the detailed parts, such as the leg covers of the sofa and the horizontal pillar of the chair, were also finely reproduced. Interestingly, the model of the chair contains the back apron (Figure 11), which was not explicitly present in the reference image. This might be the result of inference by the workers based on their common knowledge about ordinary chairs. It shows the unique capability of our synthesis method to reflect real-world knowledge, which could not be achieved by purely geometry-based approaches.

Table 3: Statistics for required resources for synthesis.

input	time	cost	# iterations	# sketches
a (chair 1)	3h 34m	$228.9	5	300
b (sofa)	3h 03m	$228.9	5	300
c (chair 2)	1h 52m	$137.34	3	180
d (table)	1h 10m	$91.56	2	120
e (camera)	1h 27m	$91.56	2	120
f (bottle)	0h 45m	$45.78	1	60

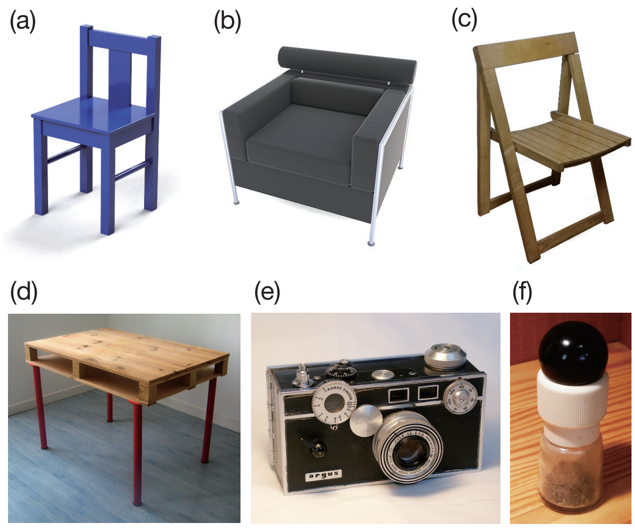

Figure 9: Example input images used for the evaluation. (d) Image courtesy of Flickr user “pierre vedel” (CC-BY-SA-2.0). (e) Image courtesy of English Wikipedia user “Camerafiend” (CC-BY-SA-3.0).

6.1 Costs for 3D Model Synthesis

Time Consumption The average required time to complete the collection of 20 sketches from crowd workers in a single stage was 61 minutes, while collecting submissions from 20 reviewers required 38 minutes ($N = 54$). Most of the required time was consumed by a small proportion of slow workers. In the experiments, we proceeded to the next stage when 15 of 20 sketches or reviews were submitted in order to reduce the relevant time consumption. Fifteen submissions were made within an average of 23 minutes and 12 minutes for sketching and reviewing, respectively. Our synthesis results showed that the required number of iterations varies between one and five depending on the complexity of the input image and the desired output quality. In total, the required time for completing synthesis for an input ranged between 45 minutes and 3.5 hours.

Task Difficulty We hypothesized that the sketching and reviewing tasks are easy enough to finish within several minutes for workers without professional skills. The median of the required time for completing a sketch was 8.0 minutes and that for reviewing was 3.8

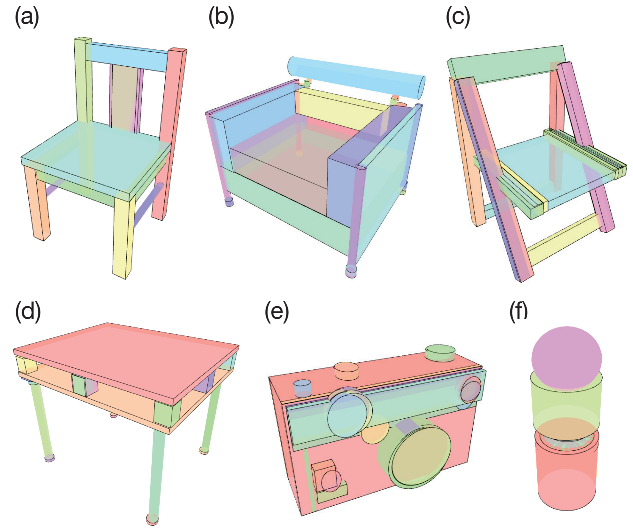

Figure 10: 3D model synthesis results for example images (Figure 9).

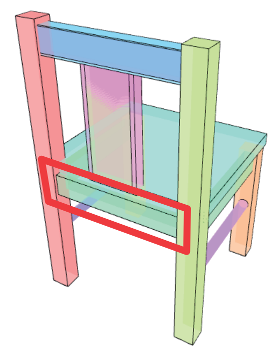

Figure 11: Back view of the synthesized chair model. The apron in the red lines was not explicitly present in the reference image (Figure 9-a).

minutes ($N = 1080$). We collected a survey from crowd workers about their impressions of the tasks using the standard functionality of the CrowdFlower platform. The sketch task was rated 4.7 (5 is the maximum) in terms of overall job satisfaction; the clarity of the instructions was rated 4.5; the ease of the job was rated 4.1; and the payment was rated 4.1. The review task was rated 4.6 in terms of overall job satisfaction; the clarity of the instructions was rated 4.5; the ease of the job was rated 4.1; and the payment was rated 4.3. These values indicate that the crowd workers considered these tasks reasonable to do as microtasks.

Monetary Cost The money required to pay the crowd workers is directly calculated by the number of submissions and bonuses given. We recruited 20 workers for sketching and another 20 workers for reviewing per stage and additionally gave bonuses to four workers (20%) who draw high-rated sketches. Basic rewards for the sketch and review tasks were $.36 and $.24 per submission, respectively, while the bonus was $.18 per worker. By summing these up, $.36 * 20 + $.24 * 20 + $.18 * 4 = $12.72 was required to complete a single stage of the competition. Additionally, the CrowdFlower platform imposes a 20% transaction fee for each job, so the total cost for a single stage of competition was $15.26 per view. Since one to five stages of competition are needed to generate a plausible result, the total cost for synthesizing a single 3D model ranges between $45.78 and $228.9.

Comparison with Outsourcing to a Professional Worker In order to evaluate the advantages of our method compared to recruitment of professionals, we also outsourced 3D modeling of the sofa image (Figure 9-b) to a single professional via a crowdsourcing service (Lancers). The result (Figure 12) was more detailed and precise compared to our method (Figure 10-b). In addition, it cost only \$45, which is much less than our method (\$228.9). On the other hand, it took a whole day to get the result, so the turnaround time was much longer than our method (three hours). It was also necessary to exchange a total of ten email messages to negotiate the details of the work and fee. Additionally, reliable and cheap professional workers are not always available in crowdsourcing platforms, while our system is ready to use at any time, because it relies on the vast pool of nonprofessional workers.

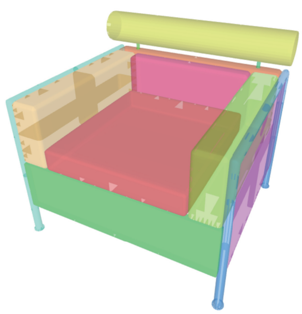

Figure 12: Modeling result by a professional worker.

6.2 Effects of Iterative Competition

Since our iterative crowdsourcing workflow deeply depends on peer reviewing, the reliability of such reviewing is crucial for quality assurance. According to our experiments, a certain proportion of reviewers indeed misunderstood the reviewing criteria, though the average scores correctly reflected the quality of sketches in most cases. Evaluation of the effectiveness of the workflow in terms of quality improvement is given below.

Valid Sketch Ratio Figure 13 shows the transition of valid sketch ratios over five stages for the input images of a chair and a sofa. The validity for each sketch was manually judged by the authors in this evaluation. The ratios increased after showing the first examples in stage 2 for all the tasks and continued increasing in the majority of the tasks. Workers were especially likely to misunderstand the arrow indication for the top view; indeed, the valid sketch ratios for the top views were only around 40% for both input images at stage 1. However, the ratios improved to around 70% in the following stages of the iteration. The ratios for most views seemed saturated at around 80%, perhaps because of the existence of malicious workers or the intrinsic tendency of human workers to make mistakes. We consider the current achievement to be sufficient to gather an adequate number of valid sketches needed for 3D model synthesis.

Parts Coverage Figure 14 shows the parts coverage in the valid sketches at each stage; if all the visible parts in the input image were reproduced by any worker, the coverage became 100%. The ratios were also manually judged by the authors. Significant improvement is observed in the front view of the chair and the side view of the sofa. In the first stage, submitted sketches for these views covered only about 70% of the parts, but in the following stages, the workers gradually added the remaining parts and achieved about 100% coverage at stage 5. The other views already had a high coverage at the first stage. Some views, such as the side view of the chair, decreased the coverage between two successive stages. This was caused by the omission of a small part that cannot be clearly seen in the example sketches that were shown to the workers in a miniaturized size. Since the synthesis process uses the sketches from all the stages for a view, such an omission might not immediately harm the synthesis quality.

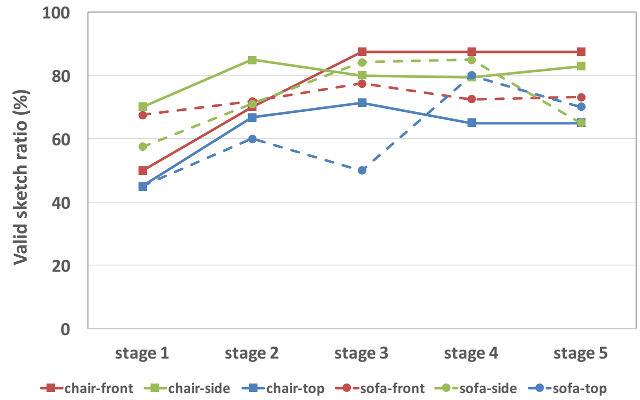

Figure 13: The transition of the valid sketch ratio over five stages for the chair and sofa input images.

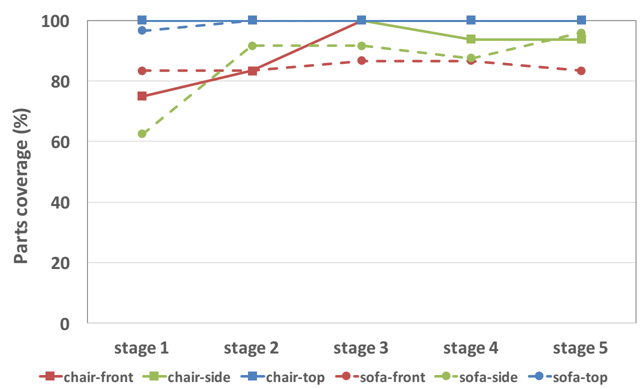

Figure 14: The transition of parts coverage in valid sketches over five stages for the chair and sofa input images.

In total, the proposed workflow was shown to be effective for improving the average quality of submitted sketches. Figure 15 shows the improvement of synthesis results through the stages. In the workflow, workers implicitly collaborated by shepherding the understanding of the tasks through examples and also competed to draw a better sketch to win the bonus.

6.3 Applying to Non-Photograph Input

Since humans can recognize the scene composition depicted in illustrations as well as that of photographs, our system is capable of generating a 3D model from illustrations without modification of the algorithms. Figure 16 shows a hand-drawn illustration of a drawer as a reference image and the 3D model of the drawer synthesized by only a single stage of competition. We can see that all the parts in the reference illustration were correctly reproduced in the model. This result suggests that our method can produce a 3D model regardless of the type of the reference image, as long as the crowd workers can understand the spatial structure of the object.

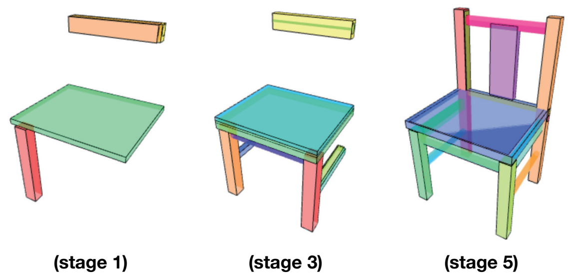

Figure 15: The transition of synthesis results of the chair from 60 sketches gathered in stage 1, 3, and 5, respectively.

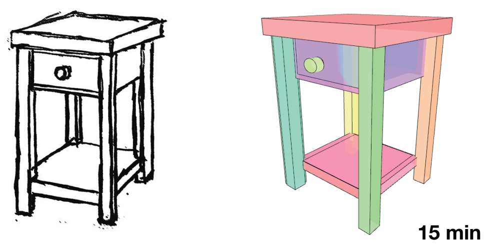

Figure 16: An example illustration input of a drawer (Left), and the synthesized model of the illustration (Right).

7 Discussion

7.1 Potential Users

The proposed method offers a rapid and widely available means for creating simple 3D models from a variety of reference images, ranging from pictures to scribbles. It enables consumer users who do not have 3D modeling skills to create their original models for 3D printing of, for example, toothbrush holders, phone stands, game tokens, and furnitures for a doll house. The number of professional workers who provide 3D modeling service is limited. In contrast, our method can serve virtually infinite number of customers at constant cost and quality employing the vast pool of microtask workers.

It could also benefit those users who have to generate a large collection of simple 3D models from image assets. Such potential users include architects who want to generate a collection of background 3D assets to be arranged in their architectural designs and indie game developers who need a large collection of low-polygon models of game items. Our system can also be used as a part of another system, such as a 3D game for children where the players can create their own items or characters by submitting a hand drawing. Embedding the crowdsourcing task in games and paying the fees as in-game currencies might be benefitial to reduce the monetary cost.

7.2 Advantages Over Simplified 3D Modeling Software

A possible criticism is that the someone can easily and quickly create simple 3D models manually by using simplified modeling software (e.g., SketchUp). However, a critical problem is that it takes time for a first-time user to learn 3D modeling, even if it is simple, because it inevitably requires the user to perform 3D rotation tasks (both view changes and object rotation), which in our experience is inherently difficult for many people. Getting the right view takes an unexpectedly long time, and sometimes users just give up.

Our system significantly reduces this learning requirement and difficulty. A user (customer) merely needs to provide an image depicting the target object and the number of parts, which requires almost zero cost. A crowd worker needs to complete 2D operations without using a 3D view or 3D rotation. This significantly reduces the necessary commands/modes (making learning easy) as well as the number of required operations in the actual task. Empirically, it can take 30 minutes to one hour to create these models from scratch, including the time for learning simple modeling systems. This demands significant engagement, and it is nearly impossible to assign such a heavyweight task to microtask crowd workers. On the other hand, in our system, each crowd worker spends roughly eight minutes, which is acceptable for a microtask.

7.3 Novelty and Applicability of the Iterative Workflow

Iterative processing and human voting or peer reviewing have been traditionally used in the literature as basic elements of crowdsourcing algorithms [20, 21]. However, the proposed iterative crowdsourcing workflow is novel in that it employs a unique integration method of such components for efficiently promoting collaboration and competition among workers. Our study showed the effectiveness of the method for gradually improving both the ratio of valid submissions and their completeness.

One major advantage of the workflow is that it is thoroughly automated and does not require supervision by the customer for quality control, except for finishing the iteration process based on satisfaction with the output. Since the workflow is simply composed of peer reviewing and iterative competition that includes extra rewarding and example sharing, it could be applied to a wide range of microtask crowdsourcing systems. It would help workers dealing with tasks that are simple but likely to cause misunderstanding, such as natural language processing and interpreting visual instructions.

7.4 Limitations and Future Work

We mainly focused on the mechanism of the system in this paper, and more work is needed to improve the usability of the method from a user's point of view. The current system requires several manual operations by the customer, such as specifying the number of parts composing the object and the threshold for triplet extraction. The former is needed just for preventing reluctant workers from submitting an invalid sketch composed of too small number of elements, hence we could simplify the way of specification to reduce the complexity of the usage. For example, the customer could select the number of parts from "1 – 5", "6 – 10", and "more than 10". Automation of such specifications employing crowdsouring is also a subject of our future work.

Since this paper is mainly aimed at showing the potential of microtask crowdsourcing as applied to collaborative 3D modeling, we focused on building the basic crowdsourcing workflow for 3D synthesis and confirming its feasibility. The proposed system thus only supports a limited number of 3D primitive types (cuboids, cylinders, and ellipsoids) that are required for showing the very basic synthesis results. Other primitives, such as freeform surfaces, and advanced operations, such as rotation around an arbitrary axis, revolution, extrusion, and CSG operations, are not supported.

The 3D model generation algorithm also has difficulty in processing many overlapping parts viewed from one direction. Figure 17 shows a possible case where the current algorithm cannot generate a correct geometry from orthographic projections. In fact, the result of modeling a camera (Figure 10-e) has multiple dials overlapping each other. These misconfigurations are caused by the confusion in correspondence between an ellipse in the front view and multiple rectangles in the side view. As a future work, human computation could be used to select a correct configuration from candidates generated from workers' ambiguous submissions.

The nature of microtask crowdsourcing is that the workers work only for several minutes, which restricts the completeness of sketches and the maximum complexity of synthesized models. As shown in the results section, the iterative competition clearly contributes to improving the parts coverage of complex shapes. However, if the customer wants to synthesize the model of an object

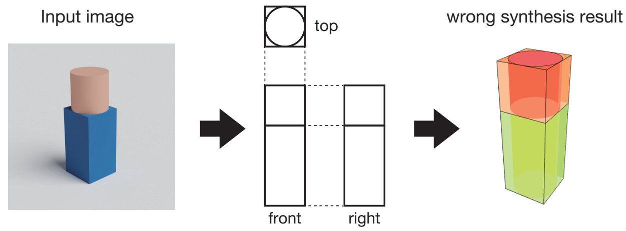

Figure 17: Left: input image. Center: orthographic projections for the image. Right: reconstruction from the projections.

composed of many (e.g., several tens of) parts, we should consider introducing a divide-and-conquer strategy to split the image into several smaller sections that can be handled by crowd workers.

We expect that microtask crowdsourcing can be applied to 3D modeling procedures with a greater variety of steps by further task decomposition. For example, we could ask crowd workers to decide the degree of corner rounding by simple 2D operations. Geometric constraints on 3D shapes, such as symmetries, alignments, and distribution of elements, could be inferred with human visual perception. Synthesis accuracy could be improved through crowdsourced supervision to point out the incorrectness of intermediate synthesis results. Future work on such attempts will reveal the potential of microtask crowdsourcing for creative tasks in greater depth.

8 Conclusions

In this paper, we proposed a novel collaborative modeling system for synthesizing a 3D shape from a reference image. We deconstructed the complex 3D modeling process into a collection of easy sketching and reviewing tasks that can be handled by nonprofessional crowd workers. We discussed both the algorithms for generating 3D geometry from a set of incomplete sketches and a crowdsourcing workflow that boosts the engagement of workers to improve the submission quality. The plausible synthesis results for several input images showed the feasibility of the proposed system as well as the possibility of applying human computation for creative purposes such as 3D modeling. The proposed workflow utilizing peer reviewing, iterative competition, and example sharing was also confirmed to be effective through experiments, and it may be applicable to a wide range of microtask crowdsourcing situations.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 26240027.

References

[1] Guidelines for academic requesters - wearedynamo wiki. `http://wiki.wearedynamo.org/index.php?title=Guidelines_for_Academic_Requesters`.
[2] M. S. Bernstein, G. Little, R. C. Miller, B. Hartmann, M. S. Ackerman, D. R. Karger, D. Crowell, and K. Panovich. Soylent: A word processor with a crowd inside. In *Proc. UIST*, pp. 313–322. ACM, NY, USA, 2010.
[3] K. J. Boudreau, N. Lacetera, and K. R. Lakhani. Incentives and problem uncertainty in innovation contests: An empirical analysis. *Manage. Sci.*, 57(5):843–863, apr 2011.
[4] T. Chen, Z. Zhu, A. Shamir, S.-M. Hu, and D. Cohen-Or. 3-sweep: Extracting editable objects from a single photo. *ACM Trans. Graph.*, 32(6):195:1–195:10, nov 2013.
[5] P. Company, A. Piquer, M. Contero, and F. Naya. A survey on geometrical reconstruction as a core technology to sketch-based modeling. *Comput. Graph.*, 29(6):892–904, dec 2005.
[6] S. Cooper, F. Khatib, A. Treuille, J. Barbero, J. Lee, M. Beenen, A. Leaver-Fay, D. Baker, Z. Popovic, and F. players. Predicting protein structures with a multiplayer online game. *Nature*, 466(7307):756–760, aug 2010.
[7] E. Delage, H. Lee, and A. Ng. A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image. In *Proc. CVPR*, pp. 2418–2428. IEEE, 2006.
[8] M.-P. Dubuisson and A. Jain. A modified hausdorff distance for object matching. In *Proc. ICPR*, pp. 566–568. IEEE, 1994.
[9] C. Eickhoff and A. P. Vries. Increasing cheat robustness of crowdsourcing tasks. *Inform. Retrieval*, 16(2):121–137, feb 2013.
[10] U. Gadiraju, R. Kawase, S. Dietze, and G. Demartini. Understanding malicious behavior in crowdsourcing platforms: The case of online surveys. In *Proc. CHI*, pp. 1631–1640. ACM, NY, USA, 2015.
[11] Y. Gingold, A. Shamir, and D. Cohen-Or. Micro perceptual human computation for visual tasks. *ACM Trans. Graph.*, 31(5):119:1–119:12, sep 2012.
[12] Y. Gingold, E. Vouga, E. Grinspun, and H. Hirsh. Diamonds from the rough: Improving drawing, painting, and singing via crowdsourcing. In *Proc. HCOMP*, 2012.
[13] E. Guillou, D. Meneveaux, E. Maisel, and K. Bouatouch. Using vanishing points for camera calibration and coarse 3d reconstruction from a single image. *The Visual Computer*, 16(7):396–410, 2000.
[14] Q. Huang, H. Wang, and V. Koltun. Single-view reconstruction via joint analysis of image and shape collections. *ACM Trans. Graph.*, 34(4):87:1–87:10, aug 2015.
[15] T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: A sketching interface for 3d freeform design. In *Proc. SIGGRAPH*, pp. 409–416. ACM Press/Addison-Wesley Publishing Co., NY, USA, 1999.
[16] A. Kittur, B. Smus, S. Khamkar, and R. E. Kraut. Crowdforge: Crowdsourcing complex work. In *Proc. UIST*, pp. 43–52. ACM, NY, USA, 2011.
[17] A. Kosorukoff. Human based genetic algorithm. In *Proc. SMC*, pp. 3464–3469. IEEE, 2001.
[18] Y. Koyama, D. Sakamoto, and T. Igarashi. Crowd-powered parameter analysis for visual design exploration. In *Proc. UIST*, pp. 65–74. ACM, NY, USA, 2014.
[19] A. Kulkarni, M. Can, and B. Hartmann. Collaboratively crowdsourcing workflows with turkomatic. In *Proc. CSCW*, pp. 1003–1012. ACM, NY, USA, 2012.
[20] G. Little, L. B. Chilton, M. Goldman, and R. C. Miller. Exploring iterative and parallel human computation processes. In *Proc. HCOMP*, pp. 68–76. ACM, NY, USA, 2010.
[21] A. Mao, A. D. Procaccia, and Y. Chen. Better human computation through principled voting. In *Proc. AAAI*, pp. 1142–1148. AAAI Press, 2013.
[22] B. Mason and S. Thomas. A million penguins research report. *Institute of Creative Technologies, De Montfort University, Leicester, United Kingdom*, 2008.
[23] C. H. Park, K. Son, J. H. Lee, and S.-H. Bae. Crowd vs. crowd: Large-scale cooperative design through open team competition. In *Proc. CSCW*, pp. 1275–1284. ACM, NY, USA, 2013.
[24] A. J. Quinn and B. B. Bederson. Human computation: A survey and taxonomy of a growing field. In *Proc. CHI*, pp. 1403–1412. ACM, NY, USA, 2011.
[25] A. Rivers, F. Durand, and T. Igarashi. 3d modeling with silhouettes. *ACM Trans. Graph.*, 29(4):109:1–109:8, jul 2010.
[26] Y. Sheikh, E. Khan, and T. Kanade. Mode-seeking by medoidshifts. In *Proc. ICCV*, pp. 1–8. IEEE, 2007.
[27] L. von Ahn. *Human Computation*. PhD thesis, Carnegie Mellon University, 2005.
[28] L. von Ahn and L. Dabbish. Labeling images with a computer game. In *Proc. CHI*, pp. 319–326. ACM, NY, USA, 2004.
[29] L. Yu and J. V. Nickerson. Cooks or cobblers?: Crowd creativity through combination. In *Proc. CHI*, pp. 1393–1402. ACM, NY, USA, 2011.
[30] R. C. Zeleznik, K. P. Herndon, and J. F. Hughes. Sketch: An interface for sketching 3d scenes. In *Proc. SIGGRAPH*, pp. 163–170. ACM, NY, USA, 1996.