C-Reference: Improving 2D to 3D Object Pose Estimation Accuracy via Crowdsourced Joint Object Estimation

JEAN Y. SONG, University of Michigan, USA
JOHN JOON YOUNG CHUNG, University of Michigan, USA
DAVID F. FOUHEY, University of Michigan, USA
WALTER S. LASECKI, University of Michigan, USA

Converting widely-available 2D images and videos, captured using an RGB camera, to 3D can help accelerate the training of machine learning systems in spatial reasoning domains ranging from in-home assistive robots to augmented reality to autonomous vehicles. However, automating this task is challenging because it requires not only accurately estimating object location and orientation, but also requires knowing currently unknown camera properties (e.g., focal length). A scalable way to combat this problem is to leverage people's spatial understanding of scenes by crowdsourcing visual annotations of 3D object properties. Unfortunately, getting people to directly estimate 3D properties reliably is difficult due to the limitations of image resolution, human motor accuracy, and people's 3D perception (i.e., humans do not "see" depth like a laser range finder). In this paper, we propose a crowd-machine hybrid approach that jointly uses crowds' approximate measurements of multiple in-scene objects to estimate the 3D state of a single target object. Our approach can generate accurate estimates of the target object by combining heterogeneous knowledge from multiple contributors regarding various different objects that share a spatial relationship with the target object. We evaluate our joint object estimation approach with 363 crowd workers and show that our method can reduce errors in the target object's 3D location estimation by over 40%, while requiring only 35% as much human time. Our work introduces a novel way to enable groups of people with different perspectives and knowledge to achieve more accurate collective performance on challenging visual annotation tasks.

CCS Concepts: • **Information systems → Crowdsourcing**; • **Human-centered computing → Human computer interaction (HCI)**; • **Computing methodologies** → *Computer vision.*

Additional Key Words and Phrases: Crowdsourcing; Human Computation; Answer Aggregation; 3D Pose Estimation; Computer Vision; Optimization; Soft Constraints

ACM Reference Format:
Jean Y. Song, John Joon Young Chung, David F. Fouhey, and Walter S. Lasecki. 2020. C-Reference: Improving 2D to 3D Object Pose Estimation Accuracy via Crowdsourced Joint Object Estimation. *Proc. ACM Hum.-Comput. Interact.* 4, CSCW1, Article 51 (May 2020), 28 pages. https://doi.org/10.1145/3392858

1 INTRODUCTION

Extracting precise 3D spatial information from the abundant collection of existing 2D datasets to create high quality 3D training data is a grand challenge for computer vision researchers [4,

Authors' addresses: Jean Y. Song, University of Michigan, Ann Arbor, MI, USA, jyskwon@umich.edu; John Joon Young Chung, University of Michigan, Ann Arbor, MI, USA, jjyc@umich.edu; David F. Fouhey, University of Michigan, Ann Arbor, MI, USA, fouhey@umich.edu; Walter S. Lasecki, University of Michigan, Ann Arbor, MI, USA, wlasecki@umich.edu.

2573-0142/2020/5-ART51 $15.00
https://doi.org/10.1145/3392858

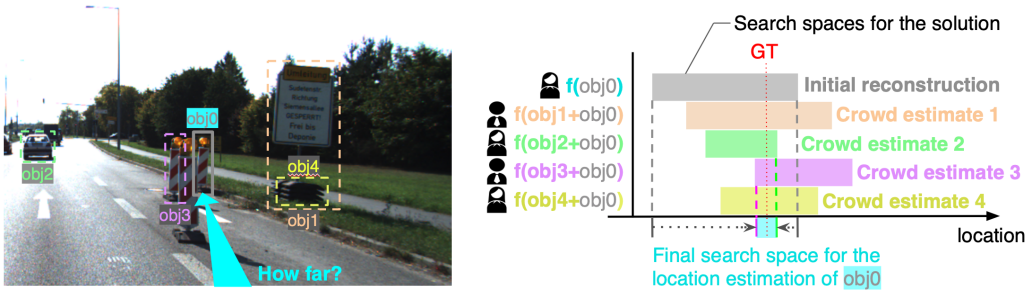


Fig. 1. This paper introduces an approach to crowd-powered estimation of the 3D location of a target object (here, `obj0`) by jointly leveraging approximate spatial relationships among other in-scene objects (`obj1`–`obj4`). Our approach lets crowd workers provide approximate measurements of familiar objects to improve collective performance via our novel annotation aggregation technique, which uses the spatial dependencies between objects as soft constraints that help guide an optimizer to a more accurate 3D location estimate.

5, 32] since it can expedite the successful deployment of real-world applications, such as self-driving vehicles [48, 50], interactive assistive robots [25, 60], or augmented and virtual reality systems [44, 51]. This conversion of 2D to 3D typically involves collecting manual annotations, where people provide the computers with the necessary information to bridge the gap between 2D and 3D, e.g., pixel-level indications of where the edges of an object are. To collect these annotations at scale, crowdsourcing is often used since it conveniently enables prompt and flexible worker recruitment [1, 10].

We consider the 2D to 3D object pose estimation task as a cooperative crowd-machine task where a computational process of parameter estimation is performed based on the manual annotations from a group of people with diverse background and knowledge. We chose to focus on the problem of searching for a solution through an iterative optimization process because the 3D pose estimation cannot be computed deterministically due to the many unknown parameters and the annotation noise [2, 36, 40, 41]. Manual annotations are usually noisy and limited by the finite resolution of an image, which results in low precision annotations due to both limited subpixel-level information and restrictions in humans' perceptual abilities and motor skills [58].

Our key insight in estimating the pose of a particular *target* object in a 2D RGB scene is that the joint use of annotations on multiple in-scene objects enables a more accurate solution while providing a means of leveraging more diverse knowledge (of different objects) from groups of people. Our approach converts crowd workers' approximations about the *reference* objects (that are near the target object) into soft constraints for an optimization algorithm that estimates the pose of the target object. As shown in Figure 1, the crowd-generated soft constraints for the optimizer penalize unlikely solutions and improve the chances of finding a more accurate solution. By relaxing the accuracy requirements for each individual person, our approach also allows crowd workers with diverse levels of knowledge to contribute to the system.

To explore the design space of the crowd worker interface, in Section 4, we conducted an experiment using different question formats to elicit measurement approximations of in-scene objects. We found that annotation accuracy does not vary hugely with respect to the question format given to the workers. Next, to understand the potential performance improvement using the proposed annotation and aggregation approach, in Section 5, we conducted a characterization study with a controlled experiment using a synthesized virtual dataset with absolute ground

truth. Based on the controlled study results, in Section 6, we developed C-Reference, a crowd-powered system that reconstructs the 3D location of a target object based on manual annotations on target and reference objects. To demonstrate the effectiveness of the system, we recruited 339 crowd workers from Amazon Mechanical Turk to annotate 90 total objects across 15 realistic computer-generated images of indoor and outdoor scenes. The end-to-end experimental results, from annotation collection to 3D location estimation, show that our approach significantly reduces the average 3D location estimation error by 40% with only 35% as much human time compared to using single-object annotations.

The crowd-machine hybrid method we presented in this paper could be used in other computer-supported cooperative work (CSCW) settings by enabling human annotators to quickly provide approximations of useful values and by allowing computers to perform precise and complex optimization tasks using these annotations. A necessary precondition in applying our approach is the availability of connecting annotations on different and diverse objects so that they can inform one another. For example, in tasks such as word sentiment annotation, one can imagine collecting annotations on diverse different, but connected, nearby words, then computationally aggregating them to estimate characteristics of a target word of interest.

This paper makes the following contributions:

- We introduce a crowd-machine cooperative approach that strategically aggregates heterogeneous information from multiple different objects with a shared spatial relationship to more accurately estimate the pose of the target object of interest.
- We present a characterization study to demonstrate the effectiveness of our proposed approach via a controlled experiment with a large synthetic dataset.
- We create C-Reference, a system that implements our proposed multi-object aggregation method to more accurately estimate 3D poses of objects from 2D images.
- We report experimental results and analysis from a study using C-Reference that demonstrates our proposed approach can more efficiently and accurately estimate the 3D location of a target object compared to using single-object annotations.

2 RELATED WORK

This research draws upon prior work on designing crowdsourcing workflows to elicit diverse responses, creating aggregation techniques for combining crowdsourced annotations, and combating challenges in 2D to 3D object pose estimation.

2.1 Eliciting Diverse Responses from the Crowd

Crowdsourcing is a powerful method for obtaining human-labeled datasets that artificial intelligence systems using machine learning algorithms need to function. Usually, diverse responses, which are treated as a random variable, are combined using an aggregation method to estimate the single best label (e.g., by computing the arithmetic mean of the responses). While conventional approaches perceive the diversity in crowd responses as errors that should be canceled out by accumulating more responses, recent studies started to look at the diversity as a special property to be strategically leveraged.

In paraphrasing tasks, for example, diverse responses are encouraged because novel paraphrases are expected, and they can be elicited by priming the annotators with different example paraphrases [27]. Similarly, in text summarization tasks, more accurate results were achieved when asking crowd workers to write multiple summaries covering different aspects of text compared to one summary that includes all the key elements [26]. In entity annotation tasks, it was shown

that identifying diverse, but valid, crowd worker interpretation provides insight into sources of disagreement [29]. Recent work shows that having each crowd worker suggest diverse answers is beneficial in an emotion annotation task because response diversity enables the efficient construction of a large collective answer distribution [7]. Another effective diversity elicitation approach has been demonstrated in crowd-powered GUI testing, where diverse navigation paths increase the test coverage [6].

Other works systematically elicit diverse responses from the crowd to obtain an accurate single aggregated artifact. For example, using multiple tools for the same image segmentation task to elicit different responses shows that the aggregation quality is better than using a single homogeneous tool [56, 57]. Other research similarly finds that dynamically switching workflows for the same task yields diverse responses that can be aggregated into a single valid answer for NLP training [39]. While these works focus on leveraging diverse responses to reduce error biases induced by the tools or the workflows, other works leverage diverse responses as a means to compensate the uncertainty of data [7, 58].

The common thread behind these research efforts is that they leverage diverse responses to increase collective information, which can reduce aggregate noise or compensate for biases when combined appropriately. Our work contributes to this line of research by eliciting diversity in knowledge and perspectives from the crowd in order to provide rough, but effective, relevant values to a computational optimizer.

2.2 Aggregation Techniques for Combining Crowdsourced Annotations

Aggregation is a core aspect of crowdsourcing that ensures the quality of task results [31, 42]. Typically, diverse responses from multiple crowd workers performing the same task are aggregated to obtain high quality annotations [38, 55]. Many aggregation techniques have been studied, ranging from the majority vote [35, 55] to individually weighted data aggregation methods, such as the expectation maximization (EM) algorithm [11, 24, 34]. Recent studies in crowdsourcing have explored methods for aggregating heterogeneous sets of annotations to achieve even better performance than when homogeneous sets of annotations are aggregated. We define a heterogeneous set of annotations as a set of annotations that are *not* generated from an identical task setting (e.g., generated from different interface or different input data). In image segmentation tasks, Song et al. [56, 57] introduced an EM-based aggregation technique, which aggregates a heterogeneous set of responses generated by using different tools on the same task to achieve higher accuracy than any homogeneous set of responses. In the task of reconstructing 3D pose from 2D videos, Song et al. [58] used particle filtering-based method to aggregate heterogeneous annotations from different video frames to assure higher annotation accuracy.

While the idea of aggregating heterogeneous annotations of a single target object has been explored, to the best of our knowledge, there is no experimental study on the effectiveness of combining annotations for multiple heterogeneous objects. Our joint object annotation aggregation introduces a new idea of aggregating annotations of multiple *different* objects by using the shared spatial relationship between them. The proposed techniques enable creating soft constraints for an optimizer, which helps find a better solution by reducing the uncertainty of the 3D pose estimation.

2.3 Challenges in 2D to 3D Object Pose Estimation Problem

Despite the great progress in computer vision on problems such as object category detection and object 2D bounding box detection, estimating the 3D properties of objects from a single RGB image is still a challenging problem [13, 23, 52, 64]. There have been breakthrough approaches using RGB-D sensor, which leverage the depth information from the additional channel to estimate the 3D pose

of objects [10, 22, 66]. However, these methods do not provide solutions when depth information is missing, e.g., estimating 3D properties of objects shot by ordinary monocular cameras.

Estimating 3D state from depth-less 2D images is especially challenging because the projection operation from the 3D world to the 2D image removes a dimension of information and, as a result, an infinite number of 3D worlds can correspond to a single 2D image [19]. As a data-driven approach, deep learning based on convolutional neural networks (CNNs) is often used, which requires a large number of training datasets of the target object [14, 21]. To overcome this limitation, non-data-driven approaches have been introduced which use explicit visual cues in images, such as object boundaries, occlusion, linear perspective, parallel line pairs, or aerial perspective [9, 45]. As it is difficult to get these cues with machine computation only, they are usually manually annotated with approaches such as keypoint annotations [62], bounding box annotations [18], or dimension line annotations [58]. Also, an interesting novel approach, LabelAR [33] has been introduced, which uses augmented reality for fast in-situ collection of images and 3D object labels for acquiring training datasets. Since this method requires an AR-enabled camera to perform annotation tasks, it does not scale to the general problem of converting plain 2D image to 3D.

Manual annotation tasks often require annotators to "estimate" or "guess" contextual information based on their perception, e.g., if an object is occluded, the worker needs to guess the hidden part to draw a bounding box or a dimension line as accurate as possible. While this estimation process could be challenging due to limited perception or knowledge, the annotation task could be challenging as well, due to factors such as limited human motor skill and restricted resolution of the images. This affects the 2D to 3D conversion performance, because the preciseness needed is often on the level of sub-pixels. As demonstrated in Popup [58], even a single pixel noise in annotation can lead to a few meters of error when converted into 3D.

Even though they lack 3D information, 2D images contain rich texture information, which can be leveraged to infer 3D spatial information (e.g., looking at other objects in the image as a reference). While recent studies in computer vision have explored the benefit of leveraging the texture information [18, 43, 53, 68, 71], crowdsourcing techniques for annotation task mostly focused on looking at a single target object to be annotated, e.g., providing tools to help annotate the target object more precisely [1, 49]. In this paper, we propose a novel annotation aggregation method that allows annotators to estimate approximate measurements of reference objects around a target object to help improve the performance of 3D state estimation of a target object.

3 EVALUATION METHOD

To add clarity for the remainder of the paper, we begin by describing the dataset that we use for all of our empirical results, as well as our metrics of success.

3.1 Dataset

To evaluate our approach, we need a 2D image dataset with corresponding 3D ground truth answers. However, we found that existing open datasets contain errors from the sensors used to capture the scenes, and from mistakes made during the manual annotation of 3D bounding boxes [3]. For example, Figure 2 shows the ground truth bounding box of the KITTI dataset [17], which has non-negligible errors in their ground truth 3D bounding box annotations. We manually hand coded 140 ground truth annotations, randomly selected from the dataset, where 26.4% of the annotations were categorized as "accurate", 42.2% were categorized as "inaccurate", and 31.4% were categorized as "cannot tell". In fact, other large-scale datasets such as Pascal3D+ [70], ObjectNet3D [69], and SUN RGB-D [59] have also been identified as having insufficient accuracy in 3D pose annotations [67].

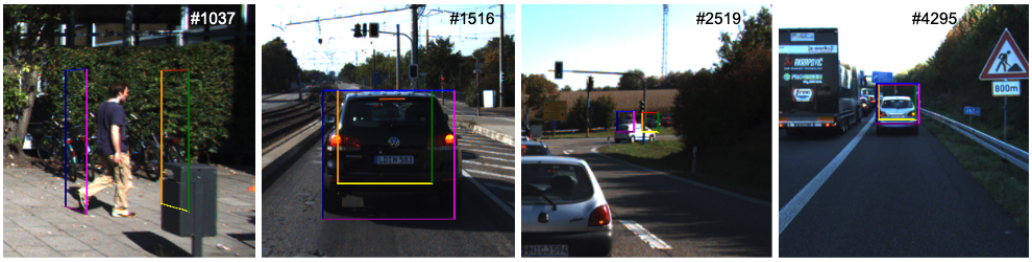


Fig. 2. We hand-coded 140 images of ground truth 3D bounding box annotations, which were randomly sampled from the well-known KITTI open dataset [17]. The errors of the ground truth annotations were non-negligible, where 42.2% were categorized as inaccurate. This figure shows four examples of the incorrect annotations.

Therefore, instead of testing on existing datasets, we created a synthetic 3D dataset using the Unity 3D game engine, which let us generate absolutely correct ground truths for the 3D object pose values. We created 15 unique 2D scenes (10 indoor and five outdoor scenes) with absolute 3D ground truths of all objects' position values, dimension values, and the 2D pixel values. The resolution of each image we synthesized was 2880×1800 pixels. For each image, we selected one *target object* at random to be estimated. Next, we arbitrarily choose five *reference objects* for each test image. We assume that no information (size, type, etc) is known by the requester or crowd workers for these reference objects. To demonstrate the robustness of our approach, we include objects that are small, heavily occluded (more than 50% occluded), or have a limited view angle from the camera's perspective. Examples of these challenging-to-annotate objects are shown in Figure 3. We report the performance of the object pose estimations in Section 7. We observed that the quality of estimation results are approximately uniform across object types. To show the scope of our synthesized dataset, we included the characteristics of the scenes and objects in Table 1.

Number of scenes	10 indoor scenes and 5 outdoor scenes
Type of scenes	**4 types of indoor scenes:** bedroom, living room, kitchen, bathroom **1 type of outdoor scene:** outdoor street
Type of objects	**19 types of indoor objects:** bookcase, dining table, coffee table, office desk, chair, couch, sink, fridge, mirror, stand, acoustic system, bed, etc **9 types of outdoor objects:** vehicle, hydrant, traffic sign, bush, bench, chair, trash basket, etc
Objects' range of distance	**2.25** to **45.48** meters
Objects' range of height	**0.24** to **3.82** meters
Objects' range of length	**0.025** to **4.63** meters
Objects' range of width	**0.06** to **5.24** meters

Table 1. Summary of our synthesized dataset

3.2 Metrics

To assess the quality of intermediate and final output of our system, we use percentage error to represent deviation from the ground truth value. We used percentage error instead of absolute measurement error because the same absolute measurement error can mean different severity of errors with respect to the target object, e.g., 0.5 meter measurement error for a chair's length can be a severe error, but the same absolute measurement error would trivial for the length of a train. If the output is a range, we also measure precision.

Fig. 3. Examples of challenging objects in our synthesized image dataset.

Percentage Error:

If the output is a scalar value, we compute the percentage error as follows:

$$\text{err}_s = \frac{|\tilde{m} - \text{GT}|}{\text{GT}} \times 100 \quad \text{(Eq. 1)}$$

where $\tilde{m}$ is the estimate value, GT is the ground truth value, and $|\cdot|$ indicates absolute value.

When the output value is a range with scalar valued bounds, we use a slightly modified formulation as follows:

$$\text{err} = \frac{|(\tilde{m}_\text{L} + \tilde{m}_\text{U})/2 - \text{GT}|}{\text{GT}} \times 100 \quad \text{(Eq. 2)}$$

where $\tilde{m}_\text{L}$ is the lower bound of the output range and $\tilde{m}_\text{U}$ is the upper bound of it, which measures the percentage error; this measure assumes that the value is at the center of the given range.

If the output is a vector value, we compute the percentage error as follows:

$$\text{err}_v = \frac{||\bar{\boldsymbol{m}} - \mathbf{GT}||_2}{||\mathbf{GT}||_2} \times 100 \quad \text{(Eq. 3)}$$

where $\bar{\boldsymbol{m}}$ denotes the estimated 3D location vector, $\mathbf{GT}$ denotes the ground truth 3D location vector, and $||\cdot||_2$ denotes Euclidean distance.

Precision:

When the output value is in range, the precision of the range is computed as follows:

$$\text{precision} = \frac{(\tilde{m}_\text{U} - \tilde{m}_\text{L})}{(\tilde{m}_\text{L} + \tilde{m}_\text{U})/2} \times 100 \quad \text{(Eq. 4)}$$

where $\tilde{m}_\text{L}$ and $\tilde{m}_\text{U}$ are lower and upper bounds of the range, respectively.

Note that for both percentage error and precision, a lower value means better performance.

4 WORKER INTERFACES FOR OBJECT MEASUREMENT APPROXIMATION

This section explores different formatting conditions for the approximate object measurement annotations. There are multiple ways to design an annotation interface, which we explore along three dimensions: annotation selection, annotation directness, and annotation granularity. The findings from this section shows that regardless of the design of the annotation interface, crowd workers' annotation accuracy is similar. Each interface asks annotators to draw the corresponding length lines on the reference objects, as shown in Figure 4(c). We explore diverse formatting conditions of measurement estimates to understand trade-offs in crowd worker performance (accuracy and precision) when generating different measurement approximations.

4.1 Formatting Conditions of Measurement Estimate Annotation

Providing various options for the formats of measurement estimate annotation helps facilitate the use of varied knowledge of the size and dimensions of different objects from crowd workers. While measurement estimates of reference objects can be asked and can be answered in a wide range of formats, we explored three different dimensions in designing the input format for the measurement estimates: selection (length of an object, width of an object, or distance of an object from the camera), directness (direct measurement of an object or relative measurement compared to another object), and granularity (single valued answer or range valued answer). We chose these three conditions because they are orthogonal and thus can be combined (e.g., annotating the *length* of an object via a *range* that is *relative* to a different object's length).

4.1.1 Selection of Measures. While our annotation aggregation method can make use of any line drawn on the ground and its corresponding actual measurement value, it is hard for humans to estimate the actual measurement value of a line if there is no visual reference. Therefore, we asked workers to annotate lines that have visual object reference points in the scene. We asked them to annotate the length and width of in-scene objects that can be estimated based on prior exposure to and knowledge about everyday objects (e.g., the length of a table is usually greater than that of a chair). Our approach projects these annotations onto a share reference plane (any reference plane could be used without loss of generality), which in this study, was chosen to be the ground plane of the scene. Among the three dimension of an object, width, length, and height, we did not asked to annotate height because it would be orthogonal to our selected reference plane. We did not include height measurements of objects since they cannot be drawn on the ground. Each object's distance from the camera can be inferred based on the scene in a given image, e.g., if there are three cars in a row, one can tell approximately how far the last car is from the camera. Example measurement estimates are:

- *Length*: The object is about 165 inches long.
- *Width*: The object is about 50 inches wide.
- *Distance*: The object is about 35 feet away.

Note that because the camera location is not visible from the image, instead of asking the distance from the camera, we designed the interface to ask crowd workers to consider the distance "from the bottom of the image". In the instructions, we provided example GIF images that demonstrate how to draw length lines to help workers understand the task.

4.1.2 Directness of Measures. Depending on the context, sometimes it can be easier to estimate a relative measurement than the direct measurement of an object. This is especially true when the object is not familiar to the annotator, because people naturally use prior knowledge of other objects to infer the properties of a new object [20, 61]. Therefore, we implemented both an interface to input direct measurements as well as relative measurements. However, if we let crowd workers make the inference based on any object in the task image, it becomes hard for the computer to aggregate the annotations, because the computer does not know the true measurements of the other objects. Therefore, we restricted this comparison to be done only with the target object to be reconstructed, which we already know the exact true size of. Example measurement estimates are:

- *Direct*: The object's length is about 80 inches.
- *Relative*: The object is about 10% longer than the target object.

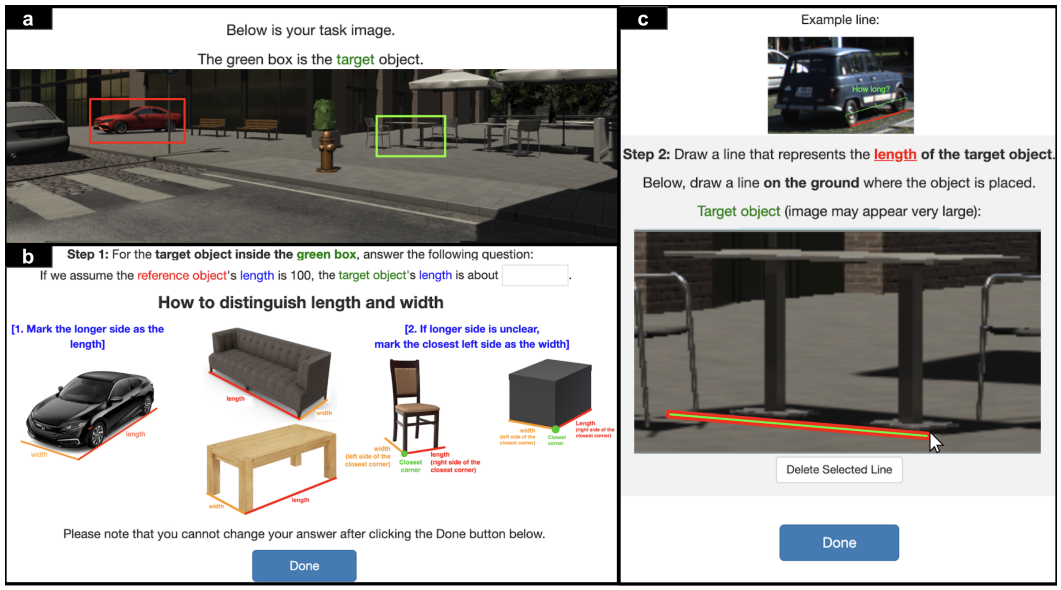
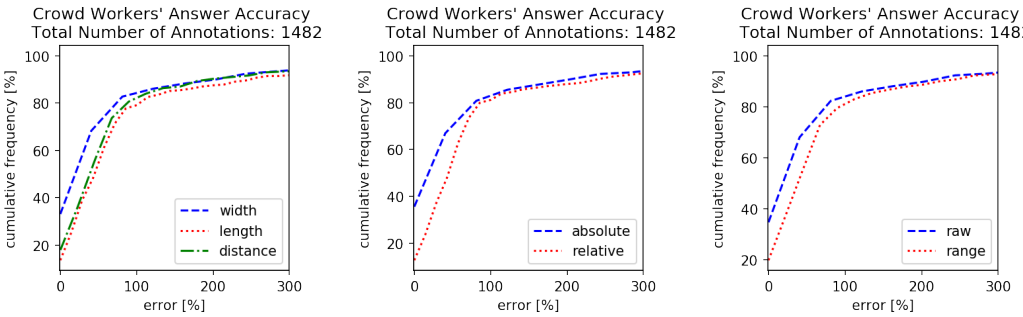


Fig. 4. The interactive worker UI is comprised of three steps in which workers approximate object measurements and annotate dimension lines. (a) The instructions and task image step: the reference object to be annotated is marked with a green box. If the *relative* condition is assigned, the UI also provides an indication of the target object (red box). However, for the workers, we reversed the name of the objects, since the reference object for the optimizer is the target to manually annotate for the workers. (b) The measurement-approximation step: each worker sees different instructions based on the condition they are assigned. (c) Length line annotation step: crowd workers were instructed to draw the line that represents the measurement they provided in the second step.

4.1.3 Granularity of Measures. Unless the annotator knows the exact make and model of an object, it is infeasible to precisely identify the length or width of it. Similarly, for the distance measurement, it is hard to tell the exact distance of an object from a single image. Therefore, we designed two elicitation approaches, a single valued estimate and a ranged valued estimate. For the range valued approach, the annotators can freely choose the granularity of their answer. Our proposed joint object annotation aggregation method can handle multiple granularities because the penalty function (Eq. 6) is designed to accept lower and upper bounds. Example measurement estimates are:

- *Single*: The target is about 32 feet away.
- *Ranged*: The target object is about 30 feet to 40 feet away.

Even though there are 12 possible combinations of measurement estimation formats (3 selection $\times$ 2 directness $\times$ 2 granularity), we used 10 formats in the study because without at least one absolute measurement for the *distance* selection, the system of equation becomes under-constrained with no absolute solution. Therefore, we excluded the two combinations $distance \times relative \times single$ and $distance \times relative \times ranged$ from both the interface design and the data collection.



(a) Different selection (b) Different directness (c) Different granularity

Fig. 5. Cumulative frequency of annotation is plot with respect to percentage error of the annotation. No significant difference was observed within each dimension.

	length	**width**	**distance**
direct × single	61.98/120.43(180.05)	60.65/64.21(57.77)	51.02/60.09(61.84)
direct × range	67.31/167.56(270.63)	61.94/184.46(698.51)	49.88/55.15(49.19)
relative × singe	65.24/238.02(658.27)	65.02/109.38(151.22)	-
relative × range	58.51/291.16(1209.50)	68.01/104.58(129.45)	-

Table 2. Median/Average(Standard Deviation) percentage error computed as in Eq. 1 and Eq. 2 to evaluate worker answers for different measure input formatting conditions.

4.2 Task Interface

Our task interface presents crowd workers with step-by-step instructions and web annotation tools. After reading through the instructions, crowd workers can proceed to the task. Then they are shown an image with reference objects to be annotated, which are indicated with a green box. For the *relative* condition, the target object to be compared is also indicated with a red box. To avoid confusion, we note that the terminologies were reversed for the crowd workers, the UI calls target object as reference object and vice versa, because the reference object for the computer (optimizer) is actually a target object for the workers to manually annotate. After checking the task image, the next step is to provide estimated measurement values, as in Figure 4(b). The workers are allowed to choose a unit of measurement from the following four options: meters, feet, inches, and yards. The last step is to mark the corresponding line on the selected reference object, as in Figure 4(c). Since the length and width of an object can be ambiguous in certain cases, e.g., when an object has no apparent longer side, we set a rule to distinguish between length and width. The rule is explained in the instructions with various example objects as in Figure 4(b). For distance estimate annotation, these instructions were hidden. The instructions in Step 2 included examples of corresponding lines, and we reminded workers that the line should be drawn on the ground in the image where the objects are positioned.

4.3 Evaluating the Impact of Measure Input Format

To evaluate the impact of measurement input format on workers' annotation accuracy, we recruited 300 crowd workers. We asked the workers to annotate the 15 task images, 10 indoor and five

	length	width	distance
direct × range	25.00/28.98(17.32)	28.57/31.09(21.48)	28.57/36.66(24.08)
relative × range	18.18/25.82(27.38)	22.22/30.62(29.64)	-

Table 3. Median/Average(Standard Deviation) precision computed as in Eq. 4 to evaluate worker answers for different measure input formatting conditions.

outdoor images, using the 10 different measurement formatting conditions. Images were grouped into fives to distinguish indoor and outdoor images. The order of the images within a group and the objects within an image were randomized to avoid learning effects. Each worker annotated one object per image using a single measurement format that was given. Participants were limited to workers from the US who had a task acceptance rate ≥ 95%. Each worker could only participate once, and was paid $1.35 per task, yielding an average wage of $9/hr—above the 2019 U.S. federal minimum wage.

4.4 Results

A total of 1500 annotations were collected across the 10 formatting conditions, but 18 annotations were dropped due to task submission errors. Figure 5 shows the cumulative frequency of the percentage error for each element within each dimension. The trend is similar across conditions: a steep increase until 100 percent error, and then slows down.

The median and average percentage error of crowd workers' responses for the 10 measurement formatting conditions are shown in Table 2. To compare the performance of the 10 measurement conditions, we ran $\binom{10}{2} = 45$ (10 choose 2) pairwise comparisons using a Mann-Whitney U test because the worker responses were skewed (non-normal). With Bonferroni correction, we considered the comparison result significantly different if the p-value was below .05/45 = .0011. From the 45 comparisons, the pairs with significant difference were the following four:

- ✓ **distance×direct×range** outperformed **length×direct×range** ($U = 8554.0$, $n_1 = 150$, $n_2 = 150$, and $p < .0005$)
- ✓ **distance×direct×single** outperformed **length×direct×range** ($U = 8658.0$, $n_1 = 150$, $n_2 = 150$, and $p < .0005$)
- ✓ **distance×direct×range** outperformed **width×relative×range** ($U = 8658.0$, $n_1 = 150$, $n_2 = 149$, and $p < .001$)
- ✓ **distance×direct×single** outperformed **width×relative×single** ($U = 8658.0$, $n_1 = 150$, $n_2 = 145$, and $p < .0001$)

The results show that overall crowd workers performance were similar across different formatting conditions, but direct distance estimations significantly outperformed some of the other conditions.

The median and average precision of crowd workers' responses for the five measurement conditions are shown in Table 3. All *single* estimation answers were ignored because precision is always 0 for a single value. To compare the performance of the five measurement conditions, we ran $\binom{5}{2} = 10$ (5 choose 2) pairwise comparisons using a Mann-Whitney U test because the worker responses were skewed (non-normal). With Bonferroni correction, we considered the comparison

	length	width	distance
absolute × raw	47.31/67.24(68.02)	44.77/56.05(43.48)	36.36/46.77(33.76)
absolute × range	45.97/62.63(53.54)	50.95/67.78(49.17)	37.48/58.21(72.93)
relative × raw	39.25/54.31(40.28)	46.45/71.61(117.76)	-
relative × range	53.78/85.37(139.65)	53.30/73.97(64.30)	-

Table 4. Median/Average (Standard Deviation) task time for different measure input formatting conditions.

result significantly different if the p-value was below .05/10 = .005. From the 10 comparisons, the pairs with significant difference were the following five:

- ✓ **length**×**relative**×**range** outperformed **length**×**direct**×**range** ($U = 8296.5$, $n_1 = 150$, $n_2 = 150$, and $p < .0001$)
- ✓ **length**×**direct**×**range** outperformed **distance**×**direct**×**range** ($U = 9007.0$, $n_1 = 150$, $n_2 = 150$, and $p < .005$)
- ✓ **length**×**relative**×**range** outperformed **width**×**direct**×**range** ($U = 8658.5$, $n_1 = 150$, $n_2 = 153$, and $p < .005$)
- ✓ **length**×**relative***times***range** outperformed **distance**×**direct**×**range** ($U = 6562.0$, $n_1 = 150$, $n_2 = 150$, and $p < .0001$)
- ✓ **width**×**relative**×**range** outperformed **distance**×**direct**×**range** ($U = 8399.5$, $n_1 = 149$, $n_2 = 150$, and $p < .0005$)

The results show that workers tend to provide a narrower range when asked to annotate the *relative* measurement conditions. We also report the task time difference in Table 4, which did not significantly differ across formatting conditions. Overall, the average accuracy and precision of worker annotations were similar across different formatting conditions, even though there were some cases with significant performance differences. While images may contain various different objects in varying contexts, providing workers with as many different ways for them to provide estimates as possible will allow to cover the diverse cases of use, maximizing the benefit of diverse knowledge among workers.

5 C-REFERENCE: JOINT OBJECT 3D LOCATION ESTIMATION

In this section, we introduce our proposed joint object estimation method, which estimates the 3D location of a target object using diverse sets of 2D annotations from other objects in the scene. Our approach transforms the approximate size or distance measurement annotations of multiple objects to soft constraints that are then used by an optimizer, making it possible to use multiple levels of measurement granularity. This enables our C-Reference system to leverage heterogeneous information in a way that collectively generates more accurate system output than using a single object.

5.1 Naive Iterative Optimization for Estimating the 3D Location of a Target Object

For 3D location estimation, we build on the method from Popup [58], which estimates the 3D pose of a target object using three "dimension line" (length, width, and height of an object) annotations drawn on 2D images. Dimension lines provide richer information compared to other annotation

methods. Specifically, they can be used to determine both 3D location and orientation information of an object, while keypoint annotation [62] can only provide orientation information and 2D bounding boxes [18] can only provide location information.

The three dimension lines, as shown inside the white box in Figure 6, create four corners (c_1, c_2, c_3, and c_4), which is used to convert the problem of 3D location estimation to a perspective-n-point problem [16, 37], where the intrinsic camera parameters are unknown and the manual annotations are noisy. As in Popup [58], we assume that the target object as well as the dimensions of it are known, and the objective is to determine the orientation and location of the target object in 3-space.

5.1.1 *Designing the Baseline Cost Function.*

We estimate five unknown variables, x, y, z, θ, and μ, using the four corners of dimension line annotations of a target object. Here, x, y, z are the 3D location of the target object (x and z along the ground plane, where z is the depth from the camera, and y being the normal (perpendicular) direction from the ground plane), θ is the yaw-orientation of the object, and μ is the camera focal length. The corners of dimension line annotations are input to an iterative optimizer, which we implemented based on the L-BFGS-B algorithm [58, 72]. While any iterative computational optimization method can be used for our application [30] (e.g., Newton's method or Nelder-Mead method), we chose L-BFGS-B within `scipy.optimize.minimize` library [28] because it enables us to set bounded constraints and is known to be memory-efficient [15]. We applied a basin-hopping technique [65] to iterate multiple times with random initialization, only accepting a new solution when its cost is minimal along all the candidate solutions visited during the iteration. The objective function to be minimized is designed as follows:

$$\text{cost}(s,\mu) = \sum_{i=1}^{4} \left(-\log(f(||\boldsymbol{t}_i - C_\mu(\boldsymbol{x}_s)_i||_2; 0, \sigma^2)) + \left| ||\boldsymbol{t}_i||_2 - ||C_\mu(\boldsymbol{x}_s)_i||_2 \right| \right) \quad \text{(Eq. 5)}$$

where the optimizer finds $\bar{s}, \bar{\mu} = \underset{\{s \in S, \mu\}}{\text{argmin}}(\text{cost}(s,\mu))$. Here $\bar{s}$ denotes the estimated 3D pose, $\bar{\mu}$ denotes the estimated camera focal length, s denotes one of the 3D pose candidates, S denotes all valid 3D pose candidates, and μ denotes one of camera focal length candidates. In Eq. 5, $f(w; 0, \sigma)$ denotes the probability density function of a normally distributed random variable w with a mean of zero and standard deviation σ, $\boldsymbol{t}_i$ denotes one of the four corners from a set of dimension lines, i denotes the index of each corner, $C_\mu(\cdot)$ denotes the camera projection matrix with focal length μ, and $\boldsymbol{x}_s$ denotes the 3D bounding box of the target object for pose s based on x, y, z. Lastly, $||\cdot||_2$ denotes the Euclidean distance and $|\cdot|$ denotes absolute value. Note that we did not use the particle filter-based method proposed in Popup [58] because it is not applicable to static images.

5.1.2 *Limitations of Using the Baseline Cost Function.*

While the optimizer is designed to find the minimum cost of the objective function, there are a few factors that can cause poor estimation results. First, annotation noise in the dimension lines can affect the estimation result. Even a small discrepancy (e.g., smaller than five-pixel error in 2D) in the annotation accuracy can lead to a significantly amplified error (e.g., larger than 20-meter error in 3D) in the estimation result depending on the camera parameters [58]. Unfortunately, collecting super-precise visual annotations on 2D images is challenging because of factors such as limited human motor precision, limited pixel resolution, and limited human visual perception. Second, while the optimizer can find the global optimum in cases where input annotations have zero noise and the initial values are set at the global optimum, in other cases it fails to find the global optimum, instead finding local optima as a solution, making the performance highly dependent on the initial values chosen. To resolve the local minimum problem, more information such as additional constraints for the search area can be added to the optimization process to help avoid searching near infeasible

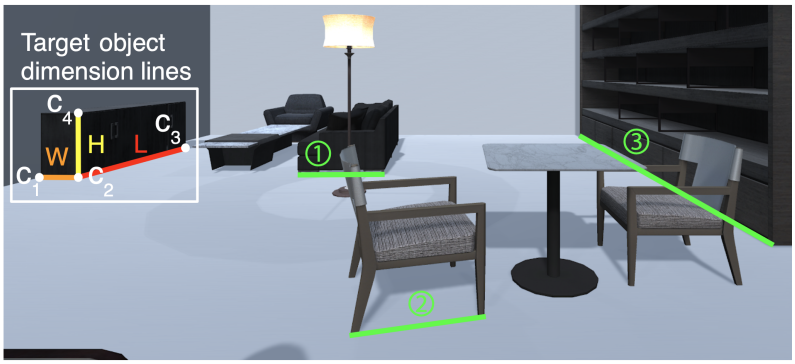


Fig. 6. A test image with known ground truth of objects. Inside the white bounding box is the target object (a cupboard) to be estimated. Three colored line on the object represents the ground truth dimension lines, length (L), width (W), and height (H). Green lines (①,②, and ③) are the reference object annotations.

solutions. In the next section, we introduce a novel annotation aggregation approach that helps the optimizer overcome these limitations by generating additional soft constraints from even rough and less precise annotations.

5.2 Proposed Joint Object Annotation Aggregation

To achieve better estimation results that overcome limitations from pixels and local optima, we introduce an approach that guides a better search area using soft constraints that are generated from diverse crowd annotations on multiple objects. The key insight is that having multiple objects will allow crowd workers to use their diverse knowledge about familiar objects and not have to rely on a single source of information when annotating. To combine the heterogeneous annotations from different objects/sources, we first introduce a penalty function that is generated by merging multiple soft constraints. The advantage of converting the annotations into soft constraints is that the accuracy requirement for usable annotations can be relaxed, allowing even rough approximations to contribute to the system performance. Next, we introduce a novel aggregation method that uses the shared spatial relationship among the objects to unify and transform the annotations into a useful input value for the penalty function.

5.2.1 Designing a Penalty Function.

We introduce a penalty function that creates a soft constraint for the optimizer using approximate search bounds. The soft constraints penalize the optimizer for selecting an unlikely solution and encourages finding a better solution, ideally near the ground truth. We design a penalty function based on a weighted sigmoid function which penalizes x if x is below l or above u as follows:

$$P(x) = S(l - x) + S(x - u) \quad \text{(Eq. 6)}$$

where $S(x)$ is the weighted sigmoid function with a weight $x + 1$,

$$S(x) = max\left(\frac{x+1}{(1+e^{-ax})}, M\right) \quad \text{(Eq. 7)}$$

and l is a lower bound, u is an upper bound, a is the (curve) sharpness parameter, and M is a threshold to prevent the penalty function from dominating the objective function. While the two parameters a and M can be arbitrarily tuned, the additional information of l and u is best chosen to be near the ground truth so that the penalty function can narrow down the search area for the optimizer. In the next section, we design and introduce a joint object annotation aggregation method, which aggregates approximate annotations from different reference objects—all other objects in the scene except the target object—to obtain reasonable values for both l and u.

5.2.2 *Annotation Aggregation Method.*

As input, the proposed annotation aggregation approach uses (i) 2D line annotations of the size of reference objects and distance of those objects from the camera and (ii) measurement values for those lines as in Figure 4. The goal of this annotation aggregation is to approximate the position of the target object relative to the camera position (x and z optimized in Eq. 5), and to use it as the search bound for the soft constraint (l and u in Eq. 6). With this goal, our proposed aggregation approach utilizes the spatial relationship between the reference objects and the target object, which is possible because they share the same ground plane and vanishing points, to transform the 3D properties of the reference objects into the pose estimation of the target object. This setting enables us to make use of the 3D affine measurements even with a single perspective view of a 2D scene, given only minimal geometric information (e.g., vanishing points and horizontal line) from the image [8].

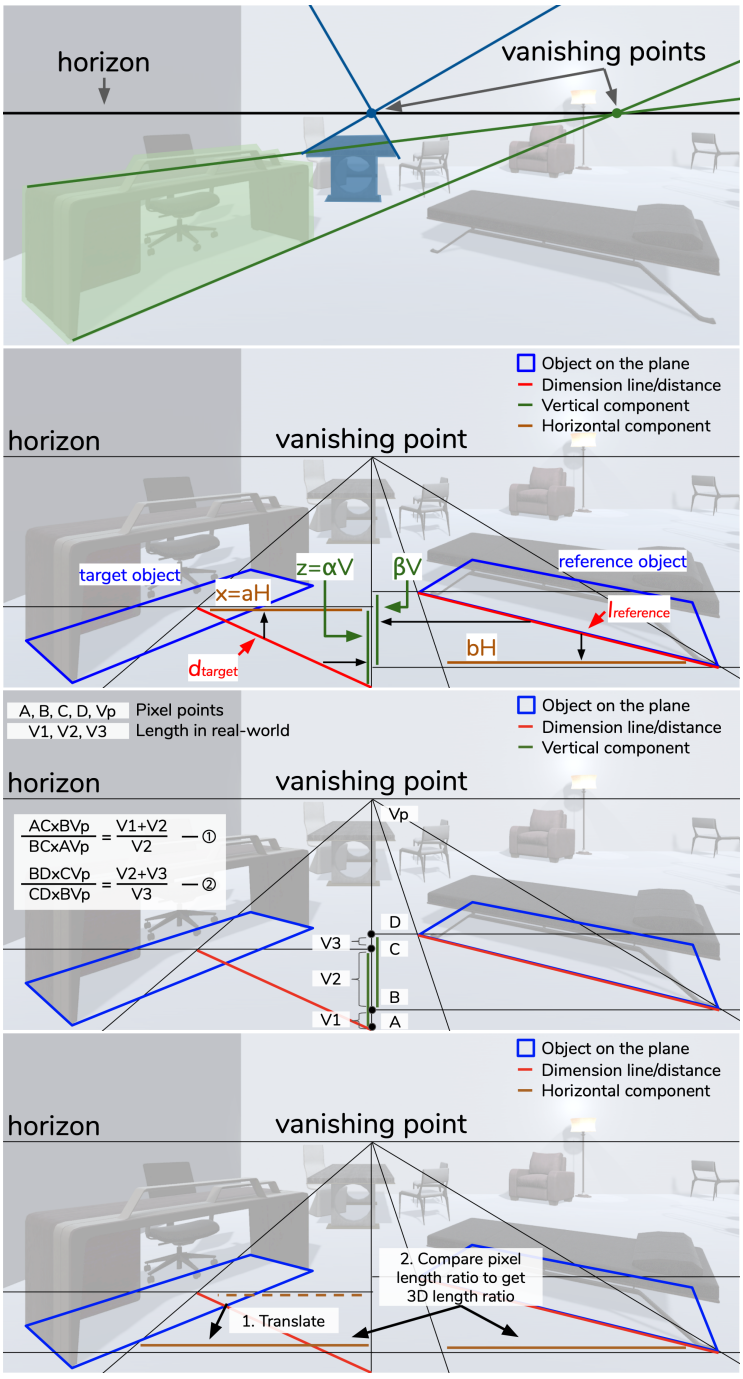
Specifically, our goal is to find the position of the target object relative to the camera, which can be represented as x and z that are optimized in the cost function (Eq. 5). To get these values, we use the following four pieces of information: (i) the line between the target object and the camera drawn on the 2D task image (d_{target} in Figure 7(b)), (ii) two lines from two reference objects (one from each object) in the 2D task image, which represent the length, width, or the distance from the camera ($l_{reference}$ in Figure 7(b)), (iii) the approximation of the reference lines' actual measurements, and (iv) the horizon in the 2D image, obtained from the connection of vanishing points (Figure 7(a)). The center of the target object's bottom, d_{target}, is computed from the target object's dimension lines. The horizontal and vertical component of d_{target} is x and z, respectively. The horizon can be estimated using off-the-shelf computer vision algorithms [12, 63] or it can be obtained manually. As shown in Figure 7(a), we assume that both the reference objects and target object share the same ground plane and vanishing points. This is reasonable because objects are place on the same plane for many of use cases such as reconstructing outdoor driving scenes.

To get approximate values of x and z, we solve equations for length relations between d_{target} and $l_{reference}$ in real-world 3D space for all $l_{reference}$. For $l_{reference}$, we first decomposed x and z into vertical and horizontal components as in Figure 7(b), since horizontal and vertical components have different characteristics when they are projected from 3D to 2D. After decomposition, for each of horizontal and vertical components, we calculate the length ratio between the components from d_{target} and that from $l_{reference}$. The length of each component is denoted by multiplying the ratio and the unit lengths (V and H) for vertical component and horizontal component, respectively. Using the length ratio, we can set the following equation with one reference object line annotation:

$$(d_{target})^2 = (aH)^2 + (\alpha V)^2 \quad \text{(Eq. 8)}$$

$$(l_{reference})^2 = (bH)^2 + (\beta V)^2 \quad \text{(Eq. 9)}$$

Here, a and b are constants calculated from the ratio of the horizontal component length of d_{target} and $l_{reference}$, and α and β are constants calculated from the ratio of the vertical component length of those two. As stated before, V and H are unit lengths for vertical and horizontal components.



(a) Horizon line and vanishing point obtained using the scene features.

(b) Aggregation Step 1: decompose the virtual line to the target object's center (d_{target}) and the annotation line of the reference object ($l_{reference}$) into vertical (along the vanishing point) and horizontal (parallel to the horizon) components.

(c) Aggregation Step 2: find the four points along the vertical line from camera center to a vanishing point to conduct the cross-ratio based computation, using either ① or ②.

(d) Aggregation Step 3: use the cross-ratio based computations (① and ② in (b)) to map the decomposed components' length in real-world with the crowd workers' measurement estimation annotations.

Fig. 7. Step-by-step aggregation of reference object annotations using cross-ratio and vanishing points.

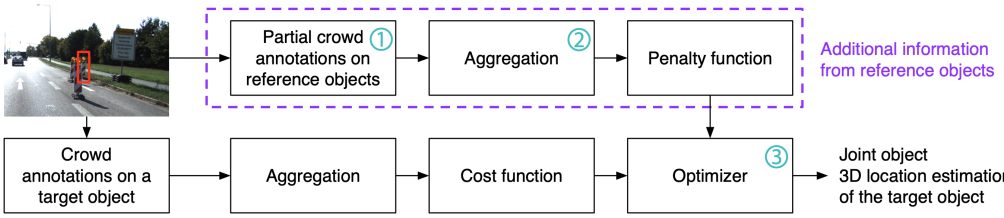


Fig. 8. Overview of the pipeline of our prototype application, C-Reference, which estimates the 3D location of a target object using a novel joint object estimation approach. The additional information from the joint object annotations (①) is aggregated (②) and transformed into a soft penalty function (③), allowing diverse granularity of approximate annotations to contribute to improving the system performance.

While a, b, α, and β can be calculated from the 2D task image, d_{target}, V, and H are unknown variables. With two reference object annotations, we can set one equation for the target object (Eq. 8) and two equations for reference annotations (Eq. 9), which share the unknown variables. Then we can solve the equation problem for these unknown variables. With solved V and H, we can find x and z, which are αV and aH, respectively.

To obtain the ratio of vertical components between two lines, we used a cross-ratio, which is the ratio between four points on the same straight line as in Figure 7(c). Because a cross-ratio has a projective invariant property, where the ratio in projected pixels and lengths in the real-world 3D space are the same (since cross-ratios are invariant to perspective projection), we can use it to compute the length ratio of lines in the real-world 3D space. For instance, in ① of Figure 7(c), the 2D line $\overline{AC}$ corresponds to 3D real-world length $V1 + V2$, and $\overline{BC}$ to $V2$. As the 2D line $\overline{BV_p}$ and $\overline{AV_p}$ have the length of infinity in 3D real-world space, we can consider these lines have equal length in real-world, and the cross-ratio equation for point A, B, C, and V_p would be as ①. The same projective invariant property also applies to ②, and with these two equations, we can calculate 3D real-world length ratio between $V1$, $V2$, and $V3$, which enables us to calculate α and β.

For the ratio of horizontal components between two lines, we first translate the decomposed horizontal component so the sub-components are on the same straight line parallel to the horizon as in Figure 7(d). The reason is because lengths of horizontal components can be compared as the real-world 3D length only when they are at the same vertical distance from the camera. After the translation, we compute the length ratio of the two horizontal components with the ratio of the pixel length. This ratio is same as the ratio of the 3D real-world length of line aH and bH, which enables us to calculate a and b. If the annotations contain parallel line (either correctly or incorrectly), then the system of equations from Eq. 8 and Eq. 9 will have no solutions. In this case, we do not input this in the penalty function to avoid unrealistic penalty function.

5.3 C-Reference

Based on the proposed joint object annotation aggregation method, we implemented C-Reference, a crowd-powered 3D location estimation system that leverages approximate annotations from the reference objects to more accurately estimate the 3D location of a target object. Figure 8 shows the system pipeline of C-Reference.

The penalty function we designed (Eq. 6) is integrated into the objective function (Eq. 5) as follows:

$$\text{cost}' = \text{cost} + \sum_{i,j \in R} P_{ij}(\boldsymbol{x}_s) \quad \text{(Eq. 10)}$$

where cost is the objective function in Eq. 5, R is a set of reference object annotations, and

$$P_{ij}(\boldsymbol{x}_s) = S(d_{ij,l} - \boldsymbol{x}_s) + S(\boldsymbol{x}_s - d_{ij,u}) \quad \text{(Eq. 11)}$$

where $S(\cdot)$ is the weighted sigmoid function described in Eq. 7, d_{ij} is d_{target} approximated from reference object annotations i and j. Finally, l and u are the lower and upper bounds of the approximation of d_{ij}. Because the penalty function can accept lower and upper bounds, it is possible to leverage the approximate measurement of objects at any level of granularity.

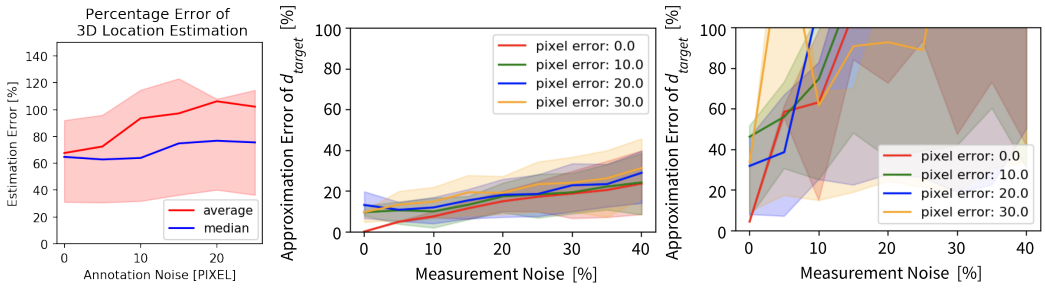
6 CONTROLLED STUDY OF SIMULATED ANNOTATION ERROR

To verify the feasibility of our aggregation method, we conducted two controlled studies with simulated data points that allows us to obtain absolute ground truths. This way, we were able to systematically control the annotation and measurement errors and investigate the system performance with respect to the level of input error. We first investigate the performance of module ③ in Figure 8 *without* the additional information from the penalty function. Next, we investigate the performance of our joint object aggregation method (module ② in Figure 8) which generates the input to the penalty function.

6.1 Parameter Settings

In Eq. 5, σ was set to 100, which was heuristically selected based on the feasible solution region. The basin-hopping iteration (random restart) number was set to 100, which was chosen by manual parameter sweeping in a preliminary study. We picked a large number of iterations to improve the overall performance for every condition including the baseline. There is a trade-off between accuracy and speed when setting this parameter, which means that a larger number of iterations will improve accuracy along with computational cost. The optimizer stopping criteria was set to 10^{-8}, which we observed was reasonable for the types of scenes explore in this paper (i.e., scenes containing objects in the order of meters in size and distance from the camera). The optimizer will converge when the following condition meets for the stopping criteria: $(s^k - s^{k+1})max\{|s^k|, |s^{k+1}|, 1\} \leq$ stopping criteria.

If the number is too large, there is a risk that the optimizer stops before converging. The optimizer bounds were set as $-10 \leq x \leq 10$, $-10 \leq y \leq 10$, $1 \leq z \leq 100$, $-\pi \leq \theta < \pi$, and $100 \leq \mu \leq 2000$. The bounds for the location parameters were selected based on a feasible search area. For example, we bound the y value, which defines the distance between the bottom of the object and the ground, to $-5 < y < 5$. This is because the images that we will be looking at contain objects that are expected to be placed approximately at ground level. The bound for the unknown orientation parameter was selected to ensure the uniqueness of the orientation solution (since 90° could be any of $\pi/2 + 2 * i$ for any integer i). The focal length bound was selected to include typical focal lengths of commercially manufactured cameras. For the penalty function in Eq. 7, the two parameters were set as $a = 8$ and $M = 50$. These were selected based on the desired sharpness of the sigmoid function and the desired size of the penalty that we want to give to the values outside the feasible solution region. For example, if M is too large, the penalty function could dominate the objective function, which is not desired. Lastly, we manually obtained the horizontal line using parallel lines on the ground plane.



(a) Using target object alone. (b) Using line 1 and 3. (c) Using line 1 and 2.

Fig. 9. Results of the controlled studies. (a) Performance characterization of the optimizer *without* our annotation-derived penalty function. The result shows error characterization result of 748 data points where each point was generated with zero to 25 pixels noise (five pixels interval) in random direction for each corner, c_1, c_2, c_3, and c_4. Shaded area is the interquartile range. The result shows an average of 70% error in 3D location estimation even with zero input noise. The error reduces to zero when the initial input values are set as the ground truth. (b) Performance characterization of our proposed joint object annotation aggregation method. The aggregation results approximates d_{target} in Eq. 8. The result shows the average error of aggregating line ① and line ③ in Figure 6. While the approximation error of d_{target} consistently increases according to both pixel and measurement noises, the error can be reduced to zero if no noise is added to the annotations. (c) The result shows the average error of aggregating line ① and line ② in Figure 6. Because the two lines are parallel, the approximation error exponentially increases according to reference object annotation noise. Shaded areas indicate the interquartile range.

6.2 Performance Characterization of the Optimizer without the Penalty Function

We ran a controlled study with 748 virtual data points with varying annotation noise on each data point. The amplitude of the noise varied from zero pixel to 25 pixels in random directions, which we generated with an interval of five pixels. The annotation noise was added to the corners of each dimension line of the target object, L, W, and H in Figure 6. Then, each data point was input to the optimizer (Eq. 5) to compute the 3D location estimation of the target object. After the estimation was finished, we computed the percentage error of the 3D location estimation of each data point as in Section 3, Eq. 3.

Figure 9(a) shows the controlled study results to estimate the 3D location of a target object. To simplify the problem, we gave the center point location of the target object (which is at the bottom-middle of the object) as the input value to the system. The results show that even with zero noise added to the input values, the estimation result had an average of 70% error due to the optimizer converging to a local minimum. This motivated us to add a penalty function to the original cost function, that can penalize implausible solutions and increase the chance to find a better solution [54]. We also observed that the average error increased with the size of the annotation noise. In summary, the controlled study results show that the baseline optimization is vulnerable to both annotation noise and the selection of the search area.

6.3 Performance Characterization of the Joint Object Annotation Aggregation

The design of our penalty function and annotation aggregation algorithm is robust to annotation noise for two reasons; (i) the threshold parameter M in Eq. 7 prevents the penalty function from creating large basins that can dominate and confuse the objective function, and (ii) unreasonable annotations can be filtered out if needed based on the expected size of the approximation value

d_{target} in Eq. 8 (e.g., one can set a threshold for this filter such as 10,000 meter, which is an unreasonable value for an indoor scene).

Beyond the mathematical observation, we ran a similar controlled study as in Section 6.2 to better understand the performance of our proposed joint object annotation aggregation method. We generated 748 virtual data points with varying noise from zero to 30 pixels in random directions, generated in 10 pixels interval. The image resolution was 2880×1800 pixels. The noise was added to the target object dimension lines and three reference object annotation lines, ①, ②, and ③, as shown in Figure 6. We also added the measurement noise to these lines, from 0 percentage noise to 40 percentage noise compared to the ground truth, generated in 10 percentage interval.

Figure 9(b) and (c) show two example results from the controlled study. Figure 9(b) shows the average result using line ① and line ③ in Figure 6 to approximate d_{target} value. It is observed that the approximation error increases according to both the pixel noise and the measurement noise. Unlike the error characterization results when using the target object annotations alone (Figure 9(a)), our joint object annotation aggregation method shows that the location estimation error can be reduced to zero when there is no input noise. Even with 40% measurement noise and 30 pixel annotation line noise, the error in approximating d_{target} was below 30% on average. Figure 9(c) shows the average result using line ① and line ② in Figure 6 to approximate d_{target} value. A characteristic of these two lines is that they are parallel in 3D. These parallel lines mean that the system of equation will not have a unique solution. The synthesized noise added to these lines creates large errors due to the lines intersecting at some point where the distance to the intersection could be very large due to the lines being near parallel. In practice, the impact of these parallel lines can be minimized because multiple combinations of annotations can be created from a set of reference objects, or one can set a threshold to filter out unreasonable combinations.

7 SYSTEM EVALUATION

The conceptual contribution of our crowdsourcing approach is in enabling aggregation of annotations on different objects to leverage the contextually relevant heterogeneous information to compute more accurate target information with the crowd. For the evaluation of this approach, we look at the 2D to 3D object pose estimation problem, where we assumed that the target object's dimension values (length, width, and height) were known but were not mapped to the image, requiring dimension line annotations for the mapping, which is the same scenario as in [58]. To validate our system, C-Reference, we conducted a user study and ran three sets of analyses: (i) an investigation on the effect of the number of additional reference object annotations on the accuracy of 3D location estimation, (ii) an investigation on the performance gain compared to a baseline method that uses the same number of total annotations, but only from the target object dimension line annotations, and (iii) an investigation on the performance of the penalty function. While the first study allows us to understand the trade-off between the cost and performance of our approach, the second study allows us to understand the benefit of using our proposed approach. The last study allows us to understand the impact of the design of the penalty function on the overall system performance.

7.1 Experimental Setup

For target object annotations, we recruited 39 crowd workers to annotate 15 images of target objects. Each image contained one target object whose 3D location is to be estimated. As mentioned in Section 3.1, we included challenging objects such as occluded objects, objects with a limited view angle, or small objects to make the task more complex (Figure 3). The image order was randomized similar to Section 4.3. A total of 13 annotations were collected for each target object. Different

workers were recruited to annotate the reference objects. The eligibility and reward setting were the same as in Section 4.3. For reference object annotations, we used the same set of annotations, which was collected in Section 4. We note that we updated two of the parameters from Section 6.1, the basin-hopping iteration number as 200, and the L-BFGS-B bounds as $-10 \leq x \leq 10$, $-5 \leq y \leq 5$, $1 \leq z \leq 50$, $-\pi \leq \theta < \pi$, and $100 \leq \mu \leq 3000$. We updated these parameters because the expected scene characteristics are different from the controlled simulation study, e.g., the image size, average distance of the objects, etc.

7.1.1 *Analysis 1: Effect of the number of reference object annotations being aggregated.*

To understand the effect of the number of additional reference object annotations, we started without reference object annotations, only aggregating the target object dimension line annotations. For target object dimension line annotations, five annotations were randomly chosen from the 13 total annotations. To avoid selection bias, we ran this 50 times, drawing a random group from a total of $\binom{13}{5} = 1287$ (13 choose 5) possible cases each time. The median, average, and standard deviation of the percentage error of the 50 samples were computed for each target object. When combining multiple target object dimension lines, we tested taking both the average and the median. We used the median throughout the study because the median is more robust to outliers. .

For reference object annotations, we randomly selected one annotation from each reference object, and combined the selected annotations using our joint object annotation aggregation technique. We increased the number of reference objects from two to five as shown in Figure 10. We skipped the single annotation condition because joint estimation requires multiple reference annotations. Whenever selecting a new annotation, we selected it from an entire pool of the 10 measurement types. The reason for this was to collect diverse combinations of the types, and investigate the effect of the combination of types on the final 3D location estimation accuracy. We ran this 50 times, the same as the target object annotations, drawing a random annotation from a total of $\binom{20}{1} = 20$ (20 choose 1) possible cases for each object at each time.

7.1.2 *Analysis 2: Performance comparison with a baseline.*

To investigate the benefit of using reference object annotations, we fix the number of total annotations to 10 (for both the target annotation only baseline, and the target plus reference annotation C-Reference for a fair cost comparison) and compare C-Reference with a baseline as follows:

- Baseline (10 target object dimension line annotations): estimates a target object's 3D location without using reference object annotations, only uses the target object's dimension line annotations. Ten target object dimension line annotations were randomly chosen from the 13 total annotations, total 50 times—drawing a random group from a total of $\binom{13}{10} = 286$ possible cases each time. Same as in Results for Analysis 1, we used the median when combining multiple target annotation dimension lines to avoid the effect of outliers.
- C-Reference (five target object dimension line annotations and five reference object annotations): estimates target object's 3D location using both target object annotations and reference object annotations. For target object dimension line annotations, we randomly chose five from the 13 total annotations, total 50 times—drawing a random group from a total of $\binom{13}{5} = 1287$ possible cases each time. Same as in Baseline, we used the median when combining multiple target annotation dimension lines. For reference object annotations, we randomly sampled five annotations with the same drawing scheme we used in Results for Analysis 1.

For performance evaluation of the 3D location estimation for both conditions, we used the percentage error as in Eq. 1. The average, median, and standard deviation were computed.

7.1.3 Analysis 3: Handling reference object annotations with large noise.

To understand the performance of the penalty function, we conducted an in-depth analysis of C-Reference's output from Results for Analysis 1. The goal is to determine whether the penalty function was able to automatically handle poor quality annotations. We used the estimation results from aggregating two reference object annotations and divided the results into two groups. One group had no input value for the penalty function, because no single pair of reference object annotations generated a valid d_{target} value (there was no solution to the corresponding system of equations). The second group had input values for the penalty function. We call these groups *skipped* and *non-skipped*, respectively. The *skipped* condition can be thought of as the penalty function being turned off, because there is no input or output from the function.

7.2 Results

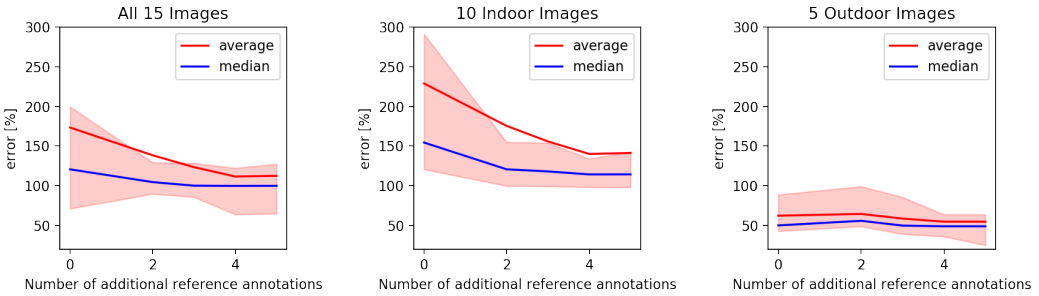
In this section, we investigate the performance of C-Reference, which implements our proposed joint-object aggregation method to elicit and leverage the knowledge diversity of crowd workers. We report the results of the three analysis studies below.

7.2.1 Results for Analysis 1: Adding more reference annotations consistently improved performance until hitting a saturation point.

As shown in Figure 10, the percentage error of the 3D location estimation of the target object consistently decreased as more reference annotations were added. Across all 15 target objects, the maximum improvement was obtained when four reference annotations were aggregated, which was the saturation point. To compare the performance of zero reference annotations with four reference annotations, we used a Mann-Whitney U test pairwise comparison because both of the results were skewed (non-normal). The result showed that there was a 36% performance improvement from four added annotations, which was a significant improvement ($U = 209955.0$, $n_1 = 750$, $n_2 = 750$, and $p < .0001$). To understand the performance better, we separated images to indoor and outdoor images and computed the percentage error of 3D location estimation of the target objects. While indoor images had 39% error reduction, outdoor images only had 13% error reduction. The average percentage error of indoor and outdoor images at four reference annotations were 139.97% and 54.57%, respectively. The average ground truth distance of target objects in indoor images was 4.95 meters and in outdoor images was 29.29 meters, which indicates that longer distances may reduce both the percentage error of estimation results and the performance gain from adding reference annotations.

7.2.2 Results for Analysis 2: C-Reference not only significantly improved accuracy but also required less annotation time compared to the baseline.

We compare the performance of C-Reference with the baseline to understand the effect when the number of total annotations is the same for the two conditions. We used a Mann-Whitney U test pairwise comparison because both of the results were skewed (non-normal). The result showed that there was a significant 40.4% performance improvement from four added annotations ($U = 201574.0$, $n_1 = 750$, $n_2 = 750$, and $p < .0001$). While C-Reference consistently improved the performance as more annotations were added, the baseline did not improve the performance according to the added dimension line annotations on the target object. We also computed the average task time for both annotation tasks: target object annotation and reference object annotation. The average task time for target object annotation was 183.5 seconds, while that for reference object annotation was 64.4 seconds, which is only 35% of the task time of the target object annotation task. Therefore, collecting more reference object annotations instead of more target object annotations could save on average 65% time per added annotation.



(a) 36% error reduction (b) 39% error reduction (c) 13% error reduction

Fig. 10. Percentage error comparison on different number of reference object annotations that are aggregated. Adding more reference object annotations decreased the percentage errors increasingly. (a) shows the result of all 15 images. There was maximum 36% error reduction from adding four reference object annotation, compared to adding no reference object annotations. (b) shows the result of all 10 indoor images. There was maximum error reduction of 39% when adding four reference object annotations. (c) shows the result of all five outdoor images. Maximum error reduction was 13% when four annotations were combined. More gain was observed with indoor images.

7.2.3 Results for Analysis 3: Penalty function was effective in penalizing infeasible solutions, and was robust to noisy input annotations.

We analyze the penalty function performance by comparing the 3D location estimation result of the two groups: the *skipped* annotation group and the *non-skipped* annotation group. Then we further divided the estimation results into six subgroups based on the selection pairs to investigate if the observed pattern is consistent across aggregation pairs. As shown in Figure 11, we observed that the *non-skipped* group always performed better than the *skipped* group regardless of the selection pair. That is, when the penalty function is considered as turned off (the *skipped* group), the performance is worse than when it is turned on, for any selection pair. This result along with the results from Results for Analysis 2 demonstrates the benefit of our proposed joint object annotation aggregation approach in improving the accuracy of 3D location estimation. For example, one may collect additional annotations until the set of annotations is *non-skipped* to improve the accuracy of the 3D location estimation.

8 DISCUSSION

The insight gained from this research can be applied to other tasks in which human annotators can quickly provide approximations of relevant values. A necessary precondition to applying our approach is the existence of underlying relationships between the annotations of different objects. In our experiment these relationships are derived from the objects being contained in the same scene, and thus having the same camera parameters, camera position, etc.

The experimental results using C-Reference demonstrate that our proposed annotation aggregation method can effectively create soft constraints from information on multiple different in-scene objects. In this section, we discuss 1) the generalizability and limitations and 2) the potential benefits of combining machine optimization with crowd-generated constraints to create synergistic effect of performance improvement.

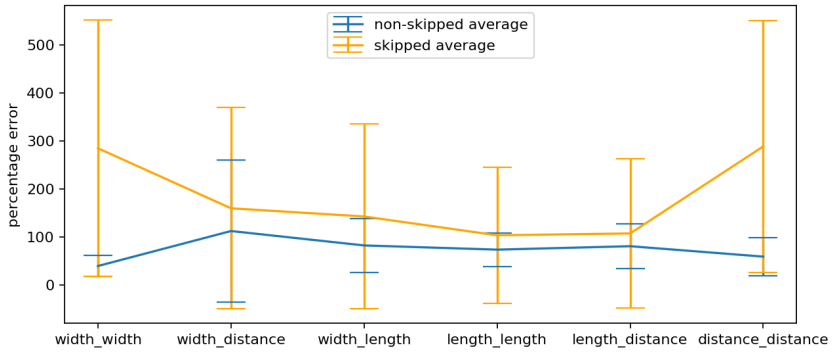


Fig. 11. Performance comparison between *skipped* and *non-skipped* groups when two reference object annotations are aggregated. Here, we further divided the groups into six aggregation pairs. The average percent error of *non-skipped* group was always lower than the *skipped* group, indicating the penalty function is beneficial.

8.1 Generalizability and Limitations

Our specific annotation and aggregation methods work on any real world image that is taken with any kind of RGB camera. That is, our method can be used to estimate the location of objects in the images even when no depth information is available. However, there are conditions that should be met for our method to be useful. The objects should be on the same plane, the ground truth dimensions of the target object should be known a priori, and the horizon should be obtainable.

Overall, we believe that the idea of leveraging knowledge diversity could be generalized to other applications than just location estimation from RGB images. For example, the annotations on different but connected words could be aggregated in forms such as graphs and vectors, or could be projected to a target word. The benefit of enabling annotations on diverse objects in the same image/corpus is that we could leverage the diverse knowledge among workers.

As an aggregation technique, we introduced the approach of turning a range or single-valued estimations into a soft constraint via weighted sigmoid functions. Unlike conventional voting aggregation, which requires the majority of people to agree on one value, or averaging, which assumes that one error offsets the other, the soft constraint allows us to aggregate crowd answers with a more flexible assumption on the patterns of noise. For example, in our system, we could not assume a specific error pattern because crowd workers will have different knowledge and generate different errors. Thus, conventional aggregation approaches that assume specific error patterns would not work for our problem. However, our approach of combining soft constraints and optimizations addresses our problem since it is more flexible to diverse error patterns.

An alternative to generating soft constraints based on workers' direct approximates is combining the soft constraints with other answer elicitation approaches, such as confidence rating [46, 47]. For instance, if a worker estimates a value with high confidence, we can transform this into the soft constraint with a small range with the center value being the estimated value. On the other hand, if the worker estimates a value with low confidence, we can turn it into a soft constraint with a wider range. As we applied a maximum penalty function to prevent overshooting penalty with wrong estimations, we would also be able to apply such techniques to generate soft constraints with confidence. One design decision that should be made for turning confidence to soft constraints would be how to model the mapping between confidence and range.

8.2 Combination of Machine Optimization and Crowd-generated Constraints

The approach of reinforcing machine optimization with crowd inputs is applicable to a set of tasks with a certain assumptions. The machine should be able to conduct the search over the solution space, as crowd inputs only provide guidance as to which part of the solution space is worth searching. For such optimization tasks, crowd workers would be leveraged in other ways, such as expanding the search space to the extent that machines cannot know yet. The synergistic combination of machine optimization with crowd-generated constraints would be more valuable for cases where there are multiple unknown correlated variables that need to be estimated. If there is only one variable to be found, deterministic approaches such as averaging or voting would be sufficient based on the detection of the error patterns.

9 CONCLUSION

This paper presents the design, implementation and assessment of C-Reference, a system that collects heterogeneous annotations of various different objects and jointly aggregates such annotations to accurately estimate the 3D location of a target object in a 2D image. Through a controlled characterization study, we investigated the potential performance benefit of our joint object annotation aggregation method that harnesses C-Reference. Based on the study results, we implemented a system, C-Reference, that collects diverse object measurement approximations from crowd workers. Using our proposed method, the annotations from crowd workers could be transformed into a soft constraint for an optimizer, penalizing infeasible solutions and encouraging the optimizer to find a better solution based on the collective annotations. Our results show that C-Reference's improves the quality of 3D location estimation of objects by more than 40% while requiring only 35% as much human time. More broadly, our work proposes a new approach to leveraging contextually relevant information from different objects to more accurately compute target object estimation.

10 ACKNOWLEDGMENTS

We thank crowd workers who participated in our study. We also thank ourlab mates and reviewers for their constructive feedback on this work. This research was supported in part by Toyota Research Institute (TRI) and by a DARPA Young Faculty Award but this article solely reflects the opinions and conclusions of its authors and not of any other entity.

REFERENCES

[1] Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. 2013. OpenSurfaces: A richly annotated catalog of surface appearance. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 111.

[2] Xun Cao, Alan C Bovik, Yao Wang, and Qionghai Dai. 2011. Converting 2D video to 3D: An efficient path to a 3D experience. *IEEE MultiMedia* 18, 4 (2011), 12–17.

[3] Liang-Chieh Chen, Sanja Fidler, Alan L Yuille, and Raquel Urtasun. 2014. Beat the mturkers: Automatic image labeling from weak 3d supervision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3198–3205.

[4] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. 2016. Single-Image Depth Perception in the Wild. In *Advances in Neural Information Processing Systems 29*. Curran Associates, Inc., 730–738.

[5] Weifeng Chen, Shengyi Qian, and Jia Deng. 2019. Learning single-image depth from videos using quality assessment networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5604–5613.

[6] Yan Chen, Mauli Pandey, Jean Y. Song, Walter S. Lasecki, and Steve Oney. 2020. Improving Crowd-Supported GUI Testing with Structural Guidance. In *Proceedings of the SIGCHI conference on human factors in computing systems*.

[7] John J.Y. Chung, Jean Y. Song, Sindhu Kutty, Sungsoo Ray Hong, Juho Kim, and Walter S. Lasecki. 2019. Efficient Elicitation Approaches to Estimate Collective Crowd Answers. In *Proceedings of the ACM conference on Computer-Supported Collaborative Work (CSCW '19)*. ACM, New York, NY, USA.

[8] Antonio Criminisi, Ian Reid, and Andrew Zisserman. 2000. Single view metrology. *International Journal of Computer Vision* 40, 2 (2000), 123–148.

[9] J. E. Cutting and P. M. Vishton. 1995. Perceiving layout and knowing distances: The interaction, relative potency, and contextual use of different information about depth. In *Perception of space and motion*. 69–117.

[10] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. 2017. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5828–5839.

[11] Alexander Philip Dawid and Allan M Skene. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied statistics* (1979), 20–28.

[12] Patrick Denis, James H Elder, and Francisco J Estrada. 2008. Efficient edge-based methods for estimating manhattan frames in urban imagery. In *European conference on computer vision*. Springer, 197–210.

[13] David Eigen and Rob Fergus. 2014. Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture. *CoRR* abs/1411.4734 (2014).

[14] Haoqiang Fan, Hao Su, and Leonidas J Guibas. 2017. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 605–613.

[15] Yun Fei, Guodong Rong, Bin Wang, and Wenping Wang. 2014. Parallel L-BFGS-B algorithm on gpu. *Computers & graphics* 40, 1–9.

[16] Martin A Fischler and Robert C Bolles. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24, 6 (1981), 381–395.

[17] Andreas Geiger, Philip Lenz, and Raquel Urtasun. 2012. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

[18] Andreas Geiger, Christian Wojek, and Raquel Urtasun. 2011. Joint 3d estimation of objects and scene layout. In *Advances in Neural Information Processing Systems*. 1467–1475.

[19] R. I. Hartley and A. Zisserman. 2004. *Multiple View Geometry in Computer Vision* (second ed.). Cambridge University Press, ISBN: 0521540518.

[20] Evan Heit. 1994. Models of the effects of prior knowledge on category learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 20, 6 (1994), 1264.

[21] Tomas Hodan, Rigas Kouskouridas, Tae-Kyun Kim, Federico Tombari, Kostas Bekris, Bertram Drost, Thibault Groueix, Krzysztof Walas, Vincent Lepetit, Ales Leonardis, et al. 2018. A Summary of the 4th International Workshop on Recovering 6D Object Pose. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 0–0.

[22] Tomas Hodan, Frank Michel, Eric Brachmann, Wadim Kehl, Anders GlentBuch, Dirk Kraft, Bertram Drost, Joel Vidal, Stephan Ihrke, Xenophon Zabulis, et al. 2018. Bop: Benchmark for 6d object pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 19–34.

[23] D. Hoiem, A.A. Efros, and M. Hebert. 2005. Geometric Context from a Single Image. In *ICCV*.

[24] Panagiotis G Ipeirotis, Foster Provost, and Jing Wang. 2010. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation*. ACM, 64–67.

[25] Stephen James and Edward Johns. 2016. 3d simulation for robot arm control with deep q-learning. *arXiv preprint arXiv:1609.03759* (2016).

[26] Youxuan Jiang, Catherine Finegan-Dollak, Jonathan K. Kummerfeld, and Walter Lasecki. 2018. Effective Crowdsourcing for a New Type of Summarization Task. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 628–633.

[27] Youxuan Jiang, Jonathan K. Kummerfeld, and Walter S. Lasecki. 2017. Understanding Task Design Trade-offs in Crowdsourced Paraphrase Collection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Vancouver, Canada, 103–109.

[28] Oliphant T. Peterson P. Jones, E. (2001, accessed 2 January 2020). SciPy: open source scientific tools for Python. http://www.scipy.org

[29] Sanjay Kairam and Jeffrey Heer. [n.d.]. Parting crowds: Characterizing divergent interpretations in crowdsourced annotation tasks *(CSCW '16)*.

[30] CT Kelley. 1999. *Iterative Methods for Optimization*. SIAM Publications, Philadelphia.

[31] Aniket Kittur, Jeffrey V Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matt Lease, and John Horton. 2013. The future of crowd work. In *Proceedings of the 2013 conference on Computer supported cooperative work*. ACM, 1301–1318.

[32] Janusz Konrad, Meng Wang, and Prakash Ishwar. 2012. 2d-to-3d image conversion by learning depth from examples. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 16–22.

[33] Michael Laielli, James Smith, Giscard Biamby, Trevor Darrell, and Bjoern Hartmann. 2019. LabelAR: A Spatial Guidance Interface for Fast Computer Vision Image Collection. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 987–998.

[34] Walter S. Lasecki, Mitchell Gordon, Danai Koutra, Malte F. Jung, Steven P. Dow, and Jeffrey P. Bigham. 2014. Glance: Rapidly coding behavioral video with the crowd. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. ACM, 551–562.

[35] Walter S Lasecki, Rachel Wesley, Jeffrey Nichols, Anand Kulkarni, James F Allen, and Jeffrey P Bigham. 2013. Chorus: a crowd-powered conversational assistant. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, 151–162.

[36] Dawei Leng and Weidong Sun. 2009. Finding all the solutions of PnP problem. In *2009 IEEE International Workshop on Imaging Systems and Techniques*. IEEE, 348–352.

[37] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. 2009. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision* 81, 2 (2009), 155.

[38] Hongwei Li, Bo Zhao, and Ariel Fuxman. 2014. The wisdom of minority: Discovering and targeting the right group of workers for crowdsourcing. In *Proceedings of the 23rd international conference on World wide web*. ACM, 165–176.

[39] Christopher H. Lin, Mausam Mausam, and Daniel S. Weld. 2012. Dynamically Switching Between Synergistic Workflows for Crowdsourcing. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI'12)*. AAAI Press, 87–93.

[40] David G. Lowe. 1991. Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 5 (1991), 441–450.

[41] C-P Lu, Gregory D Hager, and Eric Mjolsness. 2000. Fast and globally convergent pose estimation from video images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 6 (2000), 610–622.

[42] An T Nguyen, Matthew Lease, and Byron C Wallace. 2019. Explainable modeling of annotations in crowdsourcing.. In *IUI*. 575–579.

[43] Shubham Tulsiani Abhinav Gupta Nilesh Kulkarni, Ishan Misra. 2019. 3D-RelNet: Joint Object and Relational Network for 3D Prediction. In *ICCV*.

[44] Sergio Orts-Escolano, Christoph Rhemann, Sean Fanello, Wayne Chang, Adarsh Kowdle, Yury Degtyarev, David Kim, Philip L Davidson, Sameh Khamis, Mingsong Dou, et al. 2016. Holoportation: Virtual 3d teleportation in real-time. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, 741–754.

[45] Robert P O'Shea, Donovan G Govan, and Robert Sekuler. 1997. Blur and contrast as pictorial depth cues. *Perception* 26, 5 (1997), 599–612.

[46] Satoshi Oyama, Yukino Baba, Yuko Sakurai, and Hisashi Kashima. 2013. Accurate Integration of Crowdsourced Labels Using Workers' Self-reported Confidence Scores. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI '13)*. AAAI Press, 2554–2560.

[47] Satoshi Oyama, Yukino Baba, Yuko Sakurai, and Hisashi Kashima. 2013. EM-based inference of true labels using confidence judgments. In *First AAAI Conference on Human Computation and Crowdsourcing*.

[48] Xinlei Pan, Yurong You, Ziyan Wang, and Cewu Lu. 2017. Virtual to real reinforcement learning for autonomous driving. *arXiv preprint arXiv:1704.03952* (2017).

[49] Dim P Papadopoulos, Jasper RR Uijlings, Frank Keller, and Vittorio Ferrari. 2017. Extreme clicking for efficient object annotation. In *Proceedings of the IEEE International Conference on Computer Vision*. 4930–4939.

[50] Ramya Ramakrishnan, Ece Kamar, Besmira Nushi, Debadeepta Dey, Julie Shah, and Eric Horvitz. 2019. Overcoming Blind Spots in the Real World: Leveraging Complementary Abilities for Joint Execution. (2019).

[51] Aditya Sankar and Steve M Seitz. 2017. Interactive Room Capture on 3D-Aware Mobile Devices. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. ACM, 415–426.

[52] Ashutosh Saxena, Jamie Schulte, and Andrew Ng. 2007. Depth Estimation using Monocular and Stereo Cues. In *IJCAI*.

[53] Alexander G. Schwing, Sanja Fidler, Marc Pollefeys, and Raquel Urtasun. 2013. Box in the Box: Joint 3D Layout and Object Reasoning from Single Images. In *The IEEE International Conference on Computer Vision (ICCV)*.

[54] Alice Smith, Alice E Smith, David W Coit, Thomas Baeck, David Fogel, and Zbigniew Michalewicz. 1997. Penalty functions. (1997).

[55] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng. 2008. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics, 254–263.

[56] Jean Y. Song, Raymond Fok, Juho Kim, and Walter S. Lasecki. 2019. FourEyes: Leveraging Tool Diversity as a Means to Improve Aggregate Accuracy in Crowdsourcing. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 10, 1 (2019), 3.

[57] Jean Y. Song, Raymond Fok, Alan Lundgard, Fan Yang, Juho Kim, and Walter S. Lasecki. 2018. Two Tools Are Better Than One: Tool Diversity As a Means of Improving Aggregate Crowd Performance. In *23rd International Conference on Intelligent User Interfaces (IUI '18)*. ACM, New York, NY, USA, 559–570.

[58] Jean Y. Song, Stephan J. Lemmer, Michael Xieyang Liu, Shiyan Yan, Juho Kim, Jason J. Corso, and Walter S. Lasecki. 2019. Popup: reconstructing 3D video using particle filtering to aggregate crowd responses. In *Proceedings of the 24th*

International Conference on Intelligent User Interfaces. ACM, 558–569.
[59] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. 2015. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 567–576.
[60] Alexander Sorokin, Dmitry Berenson, Siddhartha S Srinivasa, and Martial Hebert. 2010. People helping robots helping people: Crowdsourcing for grasping novel objects. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2117–2122.
[61] Robert J Sternberg and Karin Sternberg. 2016. *Cognitive psychology*. Nelson Education.
[62] Ryan Szeto and Jason J Corso. 2017. Click Here: Human-Localized Keypoints as Guidance for Viewpoint Estimation. In *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 1604–1613.
[63] Jean-Philippe Tardif. 2009. Non-iterative approach for fast and accurate vanishing point detection. In *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 1250–1257.
[64] Shubham Tulsiani, Saurabh Gupta, David Fouhey, Alexei A. Efros, and Jitendra Malik. 2017. Factoring Shape, Pose, and Layout from the 2D Image of a 3D Scene. *arXiv* (2017).
[65] David J Wales and Jonathan PK Doye. 1997. Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms. *The Journal of Physical Chemistry A* 101, 28 (1997), 5111–5116.
[66] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. 2019. Densefusion: 6d object pose estimation by iterative dense fusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3343–3352.
[67] Yaming Wang, Xiao Tan, Yi Yang, Ziyu Li, Xiao Liu, Feng Zhou, and Larry S Davis. 2018. Improving Annotation for 3D Pose Dataset of Fine-Grained Object Categories. *arXiv preprint arXiv:1810.09263* (2018).
[68] Yu Xiang, Wongun Choi, Yuanqing Lin, and Silvio Savarese. 2015. Data-driven 3d voxel patterns for object category recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1903–1911.
[69] Yu Xiang, Wonhui Kim, Wei Chen, Jingwei Ji, Christopher Choy, Hao Su, Roozbeh Mottaghi, Leonidas Guibas, and Silvio Savarese. 2016. Objectnet3d: A large scale database for 3d object recognition. In *European Conference on Computer Vision*. Springer, 160–176.
[70] Y. Xiang, R. Mottaghi, and S. Savarese. 2014. Beyond PASCAL: A benchmark for 3D object detection in the wild. In *IEEE Winter Conference on Applications of Computer Vision*. 75–82.
[71] Muhammad Zeeshan Zia, Michael Stark, and Konrad Schindler. 2014. Are cars just 3d boxes?-jointly estimating the 3d shape of multiple objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3678–3685.
[72] Ciyou Zhu, Richard H Byrd, Peihuang Lu, and Jorge Nocedal. 1997. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)* 23, 4 (1997), 550–560.

Received October 2019; revised January 2020; accepted March 2020