Cheaper by the Dozen: Group Annotation of 3D Data

Aleksey Boyko
Princeton University
aboyko@cs.princeton.edu

Thomas Funkhouser
Princeton University
funk@cs.princeton.edu

ABSTRACT

This paper proposes a group annotation approach to interactive semantic labeling of data and demonstrates the idea in a system for labeling objects in 3D LiDAR scans of a city. In this approach, the system selects a group of objects, predicts a semantic label for it, and highlights it in an interactive display. In response, the user either confirms the predicted label, provides a different label, or indicates that no single label can be assigned to all objects in the group. This sequence of interactions repeats until a label has been confirmed for every object in the data set. The main advantage of this approach is that it provides faster interactive labeling rates than alternative approaches, especially in cases where all labels must be explicitly confirmed by a person. The main challenge is to provide an algorithm that selects groups with many objects all of the same label type arranged in patterns that are quick to recognize, which requires models for predicting object labels and for estimating times for people to recognize objects in groups. We address these challenges by defining an objective function that models the estimated time required to process all unlabeled objects and approximation algorithms to minimize it. Results of user studies suggest that group annotation can be used to label objects in LiDAR scans of cities significantly faster than one-by-one annotation with active learning.

ACM Classification Keywords

H.5.2. Information Interfaces and Presentation (e.g. HCI): User Interfaces; I.2.10. Artificial Intelligence: Vision and Scene Understanding; I.5.5 Pattern recognition: Applications

INTRODUCTION

There has recently been an explosion in the amount of 3D data collected in urban environments, as several companies (e.g., Google and Navteq) and government agencies (e.g., U.S. Geological Survey) are continuously collecting LiDAR data using scanners mounted on cars and/or airplanes flying overhead. For example, one such data set combining both types of data from Ottawa, Canada is shown in Figure 1.

While this LiDAR data provides immediate opportunities for visualization applications, its true value cannot be realized until semantic objects in the data have been segmented and labeled (colored points in Figure 1). If a semantically labeled 3D model of a city could be acquired, then applications such as urban planning, augmented reality maps, virtual tourism, and emergency response planning would be greatly enhanced. For example, identifying locations of stop signs, traffic lights, and street signs would augment electronic maps and help guide navigation of self-driving cars, and labeling fire hydrants, electrical power boxes, and fire escapes would help emergency response planning and disaster simulation.

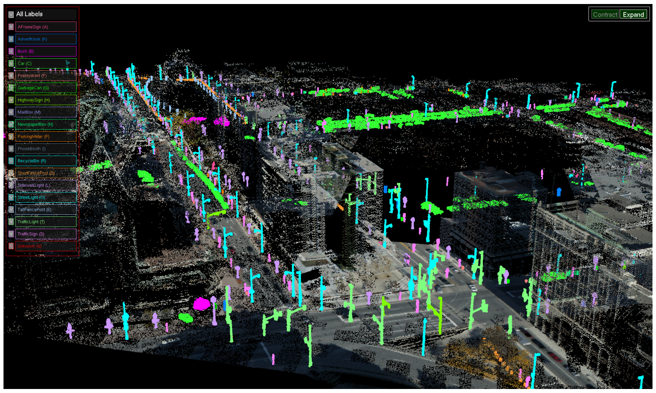


Figure 1: Our goal is to label small objects in a 3D point cloud. This image shows a section of Ottawa, with the semantic labels of 1,224 small objects represented by colors.

With these applications in mind, a number of researchers have begun to develop systems aimed at automatic segmentation and labeling of 3D LiDAR point clouds. In particular, great progress has been made over the last several years on recognition of roads, buildings, trees and other large urban structures. Unfortunately, recognition of small objects (cars, signs, fire hydrants, etc.) is more difficult, and thus the accuracy of current labeling algorithms are relatively poor for those objects. For example, state-of-the-art labeling algorithms based on supervised learning [11, 19, 20, 33] achieve only 42%–82% accuracies depending on the number of categories and their generality, which is far lower than the 95+% that is required for consumer mapping, augmented reality, and urban simulation applications. Therefore, a person has to check and fix the predicted label for *every* object before a data set can be deployed to users (a practice common even for large urban structures at most companies that annotate electronic maps).

The goal of our project is to develop an interactive system for high-throughput and high-accuracy labeling of small objects in LiDAR scans of cities. This goal is a bit different than any previous system, and thus raises several interesting research opportunities. First, our goal is to produce a label

UIST 2014, October 5–8, 2014, Honolulu, HI, USA.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3069-5/14/10 ...$15.00.
http://dx.doi.org/10.1145/2642918.2647418

for every object in a specific data set with as much computer assistance as possible, which is different than previous work in computer vision focusing on crowd-sourced creation of object recognition benchmarks where computer assistance is specifically avoided to ensure unbiased ground truth labels (e.g., [7, 39]). Second, our goal is to minimize the total time required to provide and/or confirm a label for every object in a data set, which is different than most previous work on interactive machine learning where the goal is mainly to maximize the accuracy of a classifier with the fewest training examples (e.g.,[28]). Finally, our goal is to label all objects residing in a single 3D environment (a city), and thus it is possible for our system to show multiple objects to a user in a single view and ask him/her to label them with a single command (e.g., if they all require the same label), which is different than almost all previous interactive labeling and classification systems where objects must be displayed and/or labeled separately.

This paper describes a system called IGRA (Intelligent Grouping for Rapid Annotation), which integrates ideas from human-computer interaction, active machine learning, and perceptual psychology to provide an interactive interface for labeling objects in LiDAR scans of cities quickly and accurately. The system starts from a point cloud acquired from LiDAR scans of a city and executes a number of preprocessing steps to segment the points into objects and compute geometric features for each object. It then executes an interactive labeling program that iteratively highlights groups of objects in the 3D point cloud and asks the user to perform one of three commands for each group: 1) confirm the predicted label for all objects in the group, 2) select a label for all objects in the group, or 3) decline to label the group, in which case a subset of the group will be highlighted in the next iteration. Each of these commands can be executed with a single key press (e.g., hitting the space bar confirms the predicted label for a group), and each command provides labels for multiple objects, and thus the system is very high-throughput. Moreover, the label for every object is explicitly confirmed by the user in at least one group, and thus the resulting set of labels is very high-accuracy.

The main research contribution of the paper is the introduction of an interactive labeling approach in which users are iteratively asked to label groups of objects. To investigate this approach, we developed: 1) an active learning algorithm to construct groups of objects that aims to maximize the expected labeling throughput (confirmed labels per unit time), 2) a perceptual model based on Gestalt principles to estimate the amount of time required for a user to recognize the label of a group or determine there is not one such label, and 3) an interactive system that incorporates visualization of 3D point clouds with interactive labeling in an interface that can be learned by novices in a few minutes. Experiments with this system suggest that our group annotation approach can label all small objects in a large city 1.7 times faster than a traditional one-by-one labeling approach.

RELATED WORK

Interactive annotation of multimedia data is an important problem with a long history of prior work.

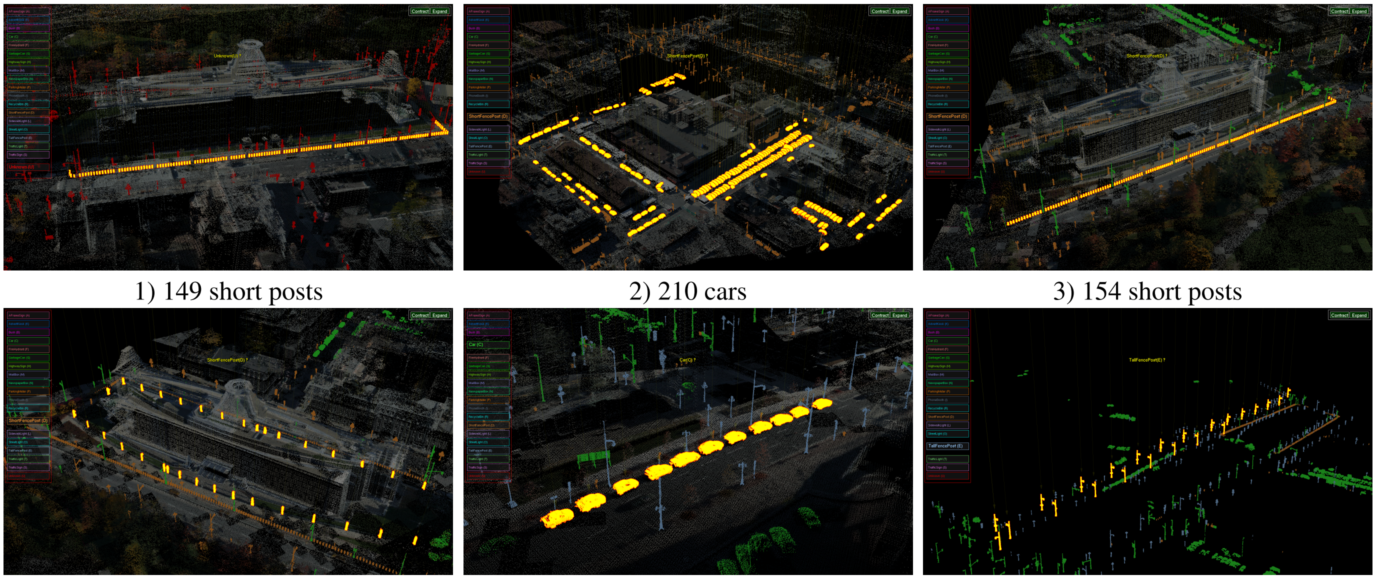
Manual Annotation. There has been much work over the last decade on user-interfaces for manual annotation of visual data. For example, interactive methods have been proposed for labeling images by object category [7], segmenting and labeling scenes in images [26], and labeling 3D point clouds of indoor [30] and outdoor scenes [10]. These methods are directed mainly at producing ground-truth data sets for object recognition benchmarks and thus purposely limit the amount of computer-assistance provided to people when choosing object labels so as not to bias the results. In contrast, our paper is targeted at applications where the goal is to label a given 3D data set with as much computer assistance as possible.

Computer-Assisted Annotation. Other researchers have investigated interactive tools to annotate 3D data sets with algorithmic assistance [18]. For example, Nan et al. [21] and van de Hengel et al. [14] have proposed interactive tools for users to annotate primitives in 3D data and specify spatial relationships between them that allow an automatic algorithm to propagate annotations based on detected regular patterns. Other systems produce annotation hypotheses automatically and allow users to fix errors and/or refine them interactively [29, 38]. For example, Shao et al. propose a system of this type for segmentation and labeling of indoor RGBD scenes. While these systems share the same goal as ours, they require a much more taxing type of user interaction: the user must search for annotations to create/revise and then execute direct manipulation commands to apply them, which are difficult operations in complex 3D data sets. As a result, their interfaces require far greater skill and time than our approach.

Active Learning. Many systems provide assistance not only for specifying annotations, but also for finding which examples to annotate. For example, active learning systems use a utility function to choose examples for users to label when training a classifier [28]. This approach has been widely used to select frames to be annotated in images and videos [17], and it has been used to specify segmentations in 3D images interactively [32]. While these methods are related to ours, their goals and operations are quite different: they usually ask users about examples one-by-one with the goal of training a classifier with the highest accuracy in the fewest interactions. In contrast, we ask users about groups of objects for labeling a specific data set in the least amount of time.

Cost-Sensitive Active Learning. Others have made the observation that not all examples take the same amount of time for users to annotate, and thus the "cost" of an annotation should be considered when selecting examples for users to label (e.g., for training classifiers of images [34]). Other research have considered a value of information (*VOI*) framework that chooses the samples by balancing the risk of mislabeling a sample and the cost of annotation [23]. Our system leverages these ideas by introducing a model to estimate the accuracy and time for a user to annotate a group of objects in 3D, which is used to minimize the total time required for an interactive labeling session.

Multiple Instance Learning. Other systems have considered labeling data with multiple objects in a single query [2]. For example, in multiple-instance learning (MIL), labels are as-



1) 149 short posts 2) 210 cars 3) 154 short posts

4) 33 tall posts 5) 11 cars 6) 23 street lights

Figure 2: Screenshots from an initial sequence of labeling operations in IGRA. The screenshots shows the objects (highlighted in yellow) labeled by a user in the first 6 group annotations, with the number of objects in each group listed below.

sociated with *bags*, which may contain multiple objects, and labels associated with a bag usually indicate that at least one object in the bag has the given label [8]. This MIL approach is best suited for data that inherently has multiple instances in a single data item that cannot be separated (e.g., multiple objects appearing in the same image). Since objects in 3D point clouds are relatively simple to separate with standard segmentation algorithms, we can easily construct arbitrary groups of objects to be labeled in our system.

OVERVIEW

This paper describes an interactive system for high-throughput and high-accuracy labeling of small objects in LiDAR scans of cities using group annotation. The input to the system is a set of segmented *objects*, each represented by a set of 3D points within a LiDAR scan of a city, and the output is a manually specified semantic label for each object.

At startup, the user is presented with an interface for viewing 3D point clouds, with which they can rotate, zoom, and pan a virtual camera and/or adjust how points and objects are displayed (e.g., turn on/off the display of points that are not parts of objects, turn on/off the display of current label names, etc.), much like many other 3D visualization systems.

The key new aspect of the system is that it iteratively selects groups of objects, shows them to the user (drawn in bright yellow from a spinning centered view as shown in Figure 2), predicts a single label for all objects in the group (shown in text above the group), and then asks the user to execute one of three actions: 1) confirm the predicted label (hit the space bar), 2) specify a new label (select a label from a menu on the left of the screen or hit an alphanumeric character associated with the label), or 3) ask the system to contract or expand the group (left-arrow or right-arrow key, respectively). If the user confirms or specifies a label for the group (actions 1 or 2), then all objects within the group are assigned that label and subsequently displayed in the color associated with the label. Otherwise, a new group is shown to the user. The system iterates this group labeling interaction until labels have been confirmed or specified for all objects in the data set (Figure 1).

This user interface was chosen for a number of reasons. First, it leverages a person's ability to recognize a semantic label for a group of related objects more quickly than a sequence of individual objects – e.g., it is possible to recognize that all the objects highlighted in first image of Figure 2 are short posts of a fence without carefully inspecting each one individually. Second, it automatically finds (groups of) objects that need labels and moves the virtual camera to view them automatically – e.g., avoiding the need for expensive visual search and interactive camera control. Third, it provides a rapid way for users to confirm the labels for all objects in the data set – e.g., if the labels predicted for a sequence of groups are correct, then the user must only hit the space bar to confirm them.

As an example, consider the sequence of screenshots in Figure 2 of the first six groups annotated by a novice user in an actual session in our user study. Each screenshot shows a group of objects (in yellow) labeled by a user with a single interaction (key click). From these screenshots, we can see that it is often possible to recognize the label for a large group of objects even without inspecting any single object closely – the pattern of objects helps us identify the appropriate label. We also observe that large numbers of objects can be labeled with just a few simple commands. The sequence shown in Figure 2 required a total of 7 key clicks (the fifth group was contracted before it was labeled) and provided confirmed labels for 578 objects in a total elapsed time of 137 seconds.

METHODS

The main research challenge in implementing our system is to develop effective methods for selecting groups to be annotated by a user at each step of an interactive labeling session.

Ideally, selected groups should contain very large number of objects, all of which require the same label, most of which have not been previously labeled, and all of which are arranged in patterns quickly recognizable by a person – then, large groups of objects could be labeled quickly and easily with our interface. However, building groups with these properties is non-trivial. Since the labels of objects are not known in advance, the system must predict them and model the probability that a group contains objects of the same type. Since different groups of objects require different amounts of time for a user to recognize their labels, the system must employ a model of human perception to estimate the cost of asking the user about a group. These considerations must be encoded into an objective function that models the benefit (time-savings) of labeling objects in a group.

Given such an objective function, we must develop an algorithm to search for the group with maximal benefit at every step of the labeling process. Our overall goal is to construct a sequence of groups that minimize the total time required by a user to label and/or confirm every object in the data set. Of course, achieving this goal is NP-Hard: choosing just one group is an instance of the optimal subset selection problem, which is NP-hard (e.g., [37]), let alone optimally choosing the entire sequence. In our interactive system, we select groups that approximately optimize this goal given current estimates of the object labels and annotation times.

Our solutions for these two main issues (defining an objective function and implementing a group selection algorithm) are discussed in the following sections.

OBJECTIVE FUNCTION

Our objective function, $B(G, L)$, estimates the expected benefit (time savings) of asking a user to select a label from a set L for a group of objects G (rather than annotating the objects one-by-one). Specifically, we define our objective function as:

$$B(G, L) = P_{\text{Label}}(G) T_{\text{1x1}}(G, L) - T_{\text{Group}}(G, L) \qquad (1)$$

where $T_{\text{Group}}(G, L)$ is the expected time it will take a user to provide a response for a group G; $P_{\text{Label}}(G)$ is the expected probability that the user will provide a label for the group G (rather than contract or expand it); and $T_{\text{1x1}}(G, L)$ is the expected time it would take a user to label all objects in G one-by-one (without group annotation).

This benefit formulation reflects the fact that the total time for the annotation session is reduced by $T_{\text{1x1}}(G, L)$ time if the user provides a label for the group, which occurs with probability $P_{\text{Label}}(G)$. $T_{\text{Group}}(G, L)$ time is added to the session for the user to process the group, regardless of whether a label is provided or not. So, intuitively, $B(G, L)$ is higher if groups are larger ($T_{\text{1x1}}(G, L)$ is higher), more likely to contain objects of the same label ($P_{\text{Label}}(G)$ is higher), and faster for a person to recognize the label ($T_{\text{Group}}(G, L)$ is lower).

While this formulation is nice theoretically, it requires estimating three terms ($P_{\text{Label}}(G)$, $T_{\text{Group}}(G, L)$, and $T_{\text{1x1}}(G, L)$), all of which depend on unknowns (object labels, user behavior, etc.). The following paragraphs provide details for how we estimate values for those terms.

Estimating Probability that the User Provides a Label. Our first challenge is to estimate $P_{\text{Label}}(G)$, the probability that the user will provide a label for a given group G – i.e., confirm the predicted label or provide a new one explicitly. This can occur only if all objects in G belong to the same category (require the same label). Thus, we compute $P_{\text{Label}}(G)$ by estimating the joint probability that all objects $o_i \in G$ belong to the same category.

The challenge in estimating the joint probability is that we don't know which label will be assigned. In fact, it is possible that no instances of the correct label for a group have been previously entered by the user, or in fact that no labels have been entered at all, and therefore it is not always possible to simply train a classifier to estimate the probability of assigning any given label based on previous training data.

Our approach is to develop an estimator for the probability $P_{\text{Label}}(o_i, o_j)$ that any two objects o_i and o_j have the same label, and then combine those probabilities to estimate the joint probability of a single label for all objects in the group. To do this, we compute the product of $|G|-1$ pairwise probabilities, where the pairs are chosen to be the highest probability ones that span all objects in G. That is, if MST_{Label} is the minimum spanning tree of a fully-connected graph, where nodes represent objects and edges represent $1 - P_{\text{Label}}(o_i, o_j)$, then:

$$P_{\text{Label}}(G) = \prod_{(o_i, o_j) \in MST_{\text{Label}}} P_{\text{Label}}(o_i, o_j) \qquad (2)$$

This method for combining the pairwise probabilities is based on two assumptions. First, the pairwise probabilities, $P_{\text{Label}}(o_i, o_j)$, are more reliable for high probability pairs than for any other (i.e., long-range distances in the feature space should not be trusted), and thus combining probabilities for pairs of objects connected in the MST is more reliable than using other pairs at all. Second, the pairwise probabilities for pairs of objects in the MST are independent, and thus they can be multiplied to estimate the joint probability.

While many ways could have been used to estimate $P_{\text{Label}}(G)$, this formulation was chosen because it leverages the latent structure of the data, even when labels have not yet been assigned for nearby examples (which is a common case in our system). This is a form of transductive learning [4, 9]. Like semi-supervised learning methods based on diffusion, we form groups of objects that are all likely to have the same label using transitivity through closely related objects.

Estimating Probability that a Pair Has Same Label. Motivated by this formulation for $P_{\text{Label}}(G)$, we must estimate $P_{\text{Label}}(o_i, o_j)$, the probability that any two objects o_i and o_j have the same label. To do this, we start by computing a feature vector for each object representing properties of its shape, such as height, width, etc.[1] Then, we model the dissimilarity of two objects based on the distance $D(o_i, o_j)$ between their feature vectors.

Instead of simply using the Euclidean distance in the feature space, we base our affinities on a density-sensitive distance metric using the formulation from [5] and algorithms from [40]. Density-sensitive distances capture cluster information about objects in the feature space and thus provide a better model for predicting which objects are in the same category. The way they do it is through analyzing the connectivity graph in shape space and computing distance as a weighted length of edges on the path between two objects in a fashion that reflects the changes in lengths of edges (variations of densities) along the way.

We follow the general density-based distance approach, but with several augmentations appropriate for our setting. First, to normalize for local densities, we shift distances down so that closest k nearest neighbors of every object are 0. This introduces asymmetry, and so we restore symmetry for every pair of objects o_i, o_j by taking the largest density-sensitive distance between $o_i \rightarrow o_j$ and $o_j \rightarrow o_i$.

Second, to account for objects that have already been assigned a label by the user (e.g., in a prior interaction within the same session), we define $D(o_i, o_j) = \infty$ if the nearest labeled neighbors of o_i and o_j are assigned different labels. Otherwise, we set it to be the minimum of $D(o_i, o_j)$ and $max(D(o_i, o_j^{NN}), D(o_j, o_i^{NN}))$, where o_i^{NN} and o_j^{NN} are the previously labeled objects (with the same label) that are closest in feature space to o_i and o_j, respectively. This adjustment creates zero-distance "wormholes" between separated clusters of objects that share the same label, a method derived from previous work on constrained clustering (e.g. [36]).

Third, we convert each distance $D(o_i, o_j)$ into an affinity $A_{\text{Label}}(o_i, o_j)$ by transforming into the range $[0, 1]$ by:

$$A_{\text{Label}}(o_i, o_j) = \frac{1}{1 + D(o_i, o_j)}$$

Finally, we estimate the probability that two objects will be assigned the same label by:

$$P_{\text{Label}}(o_i, o_j) = A_{\text{Label}}(o_i, o_j) C(o_i, o_j)$$

where $C(o_i, o_j)$ is a penalty term to account for pairs of objects that have appeared in groups of objects contracted by the user in previous interactions – i.e., if two objects were part of a group that the user declined to label in the past, they probably should not be grouped again. Initially, $C(o_i, o_j)$ for every pair of objects is one. Then, every time a group is contracted by the user all $C(o_i, o_j)$ where $o_{i,j} \in G$ are multiplied by $1 - 1/|G|$. This factor provides a soft penalty for showing poor groups of objects repeatedly.

Estimating Time for a User to Provide a Response. The next problem we must address is to develop a model for $T_{\text{Group}}(G, L)$, which estimates how long it will take a user to provide an annotation response when shown a selected group G for a given set of candidate labels L.

Such a model is difficult to estimate accurately, since it depends on the spatial reasoning aptitudes of individual users and complex factors related to perception of groups. However, we can apply basic principles of perceptual psychology to form a simple, approximate model that is adequate for our system. Our model is a sum of three terms:

$$T_{\text{Group}}(G, L) = T_{\text{Id}}(G, L) + T_{\text{Verify}}(G) + T_{\text{Cmd}}(L) \qquad (3)$$

where $T_{\text{Id}}(G, L)$ is the time required to identify a label for the group, $T_{\text{Verify}}(G)$ is the time to verify that all objects in the group require the same label, and $T_{\text{Cmd}}(L)$ is the time to convey the response to the system. Note that the second term is zero if there is only one object in G, and so

$$T_{\text{1x1}}(G, L) = (T_{\text{Id}}(G, L) + T_{\text{Cmd}}(L))|G|$$

Predicting the Time to Recognize a Label for a Group. The human visual system is capable of rapidly grasping the gist of images representing scenes. After exposure of as little 100ms [27], people can answer specific questions about what they saw [31]. The process of recognizing sets of similar objects is also fast [6] and robust [13]. Since a group of objects can be represented cognitively with summary statistics [3], recognition times for salient groups can be extremely rapid. Since items perceived as a group can "pop-out" from clutter [16, 25], they can often be recognized as a whole more quickly and accurately than as individuals [1, 24].

Accordingly, we model the time $T_{\text{Id}}(G, L)$ for a person to identify the label for a group as a function that is independent of the size of the group, but dependent on the numbers of labels to choose from. Specifically, we estimate it as the choice reaction time (CRT) of choosing among $|L|$ labels using Hick's law [15]:

$$T_{id}(G, L) = a_{\text{Id}} H_{eq}(|L|)$$

where $H_{eq}(n) = \log_2(n + 1)$ is the information-theoretic entropy of a decision among n equiprobable options, and a_{Id} is a processing speed constant factor.

Predicting the Time to Search for an Outlier. Although recognizing the label for a group of similar objects is extremely fast, the process of checking whether there is an outlier within a group can be much slower. In our system, we model the time $T_{\text{Verify}}(G)$ for a person to verify that the labels of all objects in the group are the same category using a sequence of binary decisions, where each binary decision determines whether two objects within G are in the same category or not. Under the assumption that a person considers best grouped perceptually pairs of objects available when making such a decision, we estimate the total time as a sum of binary decisions made for pairs of objects connected in a minimum spanning tree (MST_{Gest}):

$$T_{\text{Verify}}(G) = \Sigma_{(o_i, o_j) \in MST_{\text{Gest}}} T_{\text{Verify}}(o_i, o_j, G) \qquad (4)$$

[1] In our current implementation, we compute the following six shape properties for each feature vector: (1-3) the 5% trimmed spread, median, and median absolute deviation of Z-coordinates (Z is up), (4) the 5% trimmed maximum distance of a point from the centroid in the XY plane, and (5-6) the variances of points distribution in the two principle axis directions in the XY-plane.

where $T_{\text{Verify}}(o_i, o_j, G)$ is the time it takes a person to decide whether two objects within G have the same label, and MST_{Gest} is a minimum spanning tree constructed based on a cognitive measure of affinities between objects, A_{Gest}, which is defined below.

Predicting the Time to Check if Two Objects Have the Same Label. We model $T_{\text{Verify}}(o_i, o_j, G)$, the time it takes for a person to make a decision about whether two objects have the same label or not, as a choice reaction time (CRT) using Hick's law:

$$T_{\text{Verify}}(o_i, o_j, G) = a_{\text{Verify}}(o_i, o_j, G) H(P_{\text{Label}}(o_i, o_j))$$

where $a_{\text{Verify}}(o_i, o_j, G)$ is a task complexity function that depends on geometric properties of the object pair and $H(p) = p \log_2(\frac{1}{p} + 1) + (1 - p) \log_2(\frac{1}{1-p} + 1)$ is the information-theoretic entropy of a binary decision, where p is the probability of each outcome, in our case modeled as the affinity between the two object shapes ($p = P_{\text{Label}}(o_i, o_j)$).

When estimating the task complexity function, $a_{\text{Verify}}(o_i, o_j, G)$, we expect people to recognize the match between labels of two objects more quickly if they are closer to one another and/or are arranged in a regular pattern governed by Gestalt rules [35]. We combine these factors as follows:

$$a_{\text{Verify}}(o_i, o_j, G) = T_{\text{Verify}}(A_{\text{Gest}}(o_i, o_j, G) + (1 - A_{\text{Gest}}) d(o_i, o_j))$$

where T_{Verify} is the fastest possible time required to recognize whether two objects have the same labels, $A_{\text{Gest}}(o_i, o_j, G)$ is a measure [0-1] of how much the pair of objects participates in a group with strong Gestalt principles, and $d(o_i, o_j)$ is the distance between the objects in relative units ($d(o_i, o_j) = \frac{|o_i - o_j|_2}{max(|o_i|_\infty, |o_j|_\infty)}$). Intuitively, the value of $a_{\text{Verify}}(o_i, o_j, G)$ is equal to T_{Verify} for objects connected by strong Gestalt cues (when $A_{\text{Gest}}(o_i, o_j, G) = 1$), and otherwise is larger than T_{Verify} when Gestalt cues are weaker and/or the distance between objects is larger.

Estimating the Effects of Gestalt Cues. To estimate the affinity of two objects based on Gestalt cues, we detect regular patterns in the arrangements of objects in a group, and we model the proximities/sizes of objects:

$$A_{\text{Gest}}(o_i, o_j, G) = R(o_i, o_j, G) S(o_i, o_j) \qquad (5)$$

where $R(o_i, o_j, G)$ is a value [0-1] representing how much o_i and o_j participate in a regular pattern within G, and $S(o_i, o_j)$ represents a measure of the proximity of two objects relative to their sizes.

When estimating $R(o_i, o_j, G)$, we consider objects as forming a regular pattern if they are situated along a line and share similar spacing between them. Hence the smallest size of the group that can exhibit such traits is three, and we set $R(o_i, o_j) = R(o_i) = 1$ for all pairs/objects in groups of less than three. To evaluate the regularity for any three objects $o_{i,j,k}$ we compute the translation between every two, replicate it and compute the distance to the third. Then we take the smallest of these distances $d_r(o_i, o_j, o_k)$ and compute regularity score as $R(o_i, o_j, o_k) = \exp(-d_r(o_i, o_j, o_k)/min(|o_i|_\infty, |o_k|_\infty, |o_k|_\infty))$. To define regularity available for two objects o_i and o_j within a group $R(o_i, o_j, G) = max_{o_k \in G}(R(o_i, o_j, o_k))$.

When estimating $S(o_i, o_j)$, we account for the effect that similarities of objects are harder to recognize when the objects are smaller on the screen. Since the virtual camera is positioned so that the entire group is within view, objects that are further apart relative to their sizes in world space appear smaller on the screen after perspective projection. So, we model the effects of object proximity and size as $S(o_i, o_j) = \frac{1}{1 + max(0, d(o_i, o_j) - c)}$.

Estimating Time for User to Enter Label. Finally, we model the time $T_{\text{Cmd}}(L)$ for a person to provide a label to the system as another choice among $|L|$ labels using Hick's law [15]:

$$T_{\text{Cmd}}(L) = a_{\text{Cmd}} H_{eq}(|L|)$$

where a_{Cmd} is a processing speed constant factor.

In all, our model predicts that groups can be labeled more quickly if they have objects with more similar shapes, stronger regular patterns, closer proximities in 3D, larger sizes on the screen, and fewer potential labels.

GROUP SELECTION ALGORITHM

Given this objective function, our main algorithmic task is to select the best group of objects to present as a query to the user at each step of the interactive labeling process.

Finding the best group is a very difficult problem. Selecting an optimal subset is intractible even with simple objective functions [37], and ours is far from simple since the benefit of every group depends on a non-linear function of all objects in the group (e.g., to estimate Gestault cues).

To limit the search space, we first construct a hierarchical clustering tree $\mathfrak{T}$ of all objects in the data set $\mathfrak{D}$, with each node representing a group of objects in the leaf nodes of its subtree. Then, having constructed $\mathfrak{T}$, we search for the best group among its non-root nodes by evaluating $B(G, L)$. This limits the search space to a set of candidates that is linear in $\|\mathfrak{D}\|$. Additionally, it produces a hierarchical nesting of groups that naturally lends itself to implementation of the group contraction and expansion commands supported by our user interface.

Even with this reduced search space, building $\mathfrak{T}$ is non-trivial. In particular, hierarchical clustering based on $B(G, L)$ can lead to groups with poor regular patterns, since decisions made greedily in the early stages of the algorithm are based mainly on shape similarity and spatial proximity between objects (e.g., when just two objects are merged into a group) and therefore are likely to form small groups that cannot later be merged into large ones with good regular patterns. To avoid this, instead of joining groups according to $B(G, L)$, we join groups in the best-first order of $A_{\text{Gest}}(o_i, o_j, \mathfrak{D})$ - a version of $A_{\text{Gest}}(o_i, o_j, G)$ from eq. (5) that represents the best possible Gestalt score of a pair o_i, o_j with any third object from the entire $\mathfrak{D}$, not only from G. Specifically, our algorithm traverses pairs of objects $o_i \in G_i, o_j \in G_j, G_i \cap G_j = \emptyset$

in a descending order of $A_{\text{Gest}}(o_i, o_j, \mathfrak{D})$ until $B(G_{ij}, L) \geq \max(B(G_i, L), B(G_j, L))$; once such pair $(o_j, o_j)_k$ is discovered groups G_i and G_j are merged into G_{ij}. This enforces the priority of joining pairs of objects that are globally in stronger patterns.

Constructing $\mathfrak{T}$ in this way still requires evaluating $B(G, L)$ for $O(\|\mathfrak{D}\|^2)$ candidate groups. It is thus desirable to avoid having to recompute MST_{Label} (in eq. (2)) and MST_{Gest}(in eq. (4)) in every evaluation. Notice that when a decision to join groups G_i and G_j into G_{ij} is made, it is mandated by a pair of objects (o_j, o_j) in respective groups that appear in a globally strong pattern according to $A_{\text{Gest}}(o_i, o_j, \mathfrak{D})$. A set of all such edges $\{(o_j, o_j)_k\}$ that contributed to the appearance of a group G on $\mathfrak{T}$ thus forms a spanning tree $ST^{\sim}(G)$ over all objects in G with edges, chosen with the objective of the largest $A_{\text{Gest}}(o_i, o_j, \mathfrak{D})$ and non-decreasing benefit. $ST^{\sim}(G)$ is not exactly MST_{Gest} due to $A_{\text{Gest}}(o_i, o_j, \mathfrak{D}) \neq A_{\text{Gest}}(o_i, o_j, G)$, however $ST^{\sim}(G)$ is readily available without additional computations and allows for a recursive formulation of $T_{\text{Verify}}(G_{ij})$ from eq. (4)

$$T^{\sim}_{\text{Verify}}(G_{ij}) = T_{\text{Verify}}(o_i, o_j, \mathfrak{D}) + T^{\sim}_{\text{Verify}}(G_i) + T^{\sim}_{\text{Verify}}(G_j)$$

which reuses values previously computed for G_i and G_j. For the same reason of immediate availability we use $ST^{\sim}(G)$ instead of MST_{Label}, however, $ST^{\sim}(G)$ is built not directly considering $P_{\text{Label}}(o_i, o_j)$ at all. Using a product on $ST^{\sim}(G)$ as in eq. (2) is thus likely to over-constrain construction of $\mathfrak{T}$ and build only very small groups due to many elements of the product being much smaller than what they would have been if the actual MST_{Label} was used. To adjust, we replace the product with the minimum, which only estimates the group's density in the feature space instead of performing a likelihood computation, in the following fashion:

$$P^{\sim}_{\text{Label}}(G_{ij}) = \min(P_{\text{Label}}(o_i, o_j), P^{\sim}_{\text{Label}}(G_i), P^{\sim}_{\text{Label}}(G_j)).$$

which is also a recursive formulation allowing for reusing of values computed for subgroups during the tree construction.

These two approximations were made in the interest of constructing candidate groups (nodes of $\mathfrak{T}$) with large-scale regular patterns at interactive rates. They do not guarantee optimality of the candidate groups according to $B(G, L)$. However, we evaluate $B(G, L)$ for each group in the tree $\mathfrak{T}$ and choose the best one to show the user. Empirically, we observe that this approach produces groups with good $B(G, L)$ at interactive rates ($\sim$1 second per group).

RESULTS

We have performed a series of experiments to test how well group annotation works for labeling small objects in 3D points clouds with the IGRA system.

Data Set. For our experiments, we tested the proposed methods on a point cloud captured with 4 terrestrial (car-mounted) and 1 aerial (airplane-mounted) LiDAR scanners within a 6 km^2 section of downtown Ottawa, Canada [22] (see Figure 1).

The data set contains approximately 1 billion points, each of which is represented by a 3D position, RGB color, and scalar intensity, though the colors and intensities of terrestrial points are very noisy and thus not used by our system. It was chosen because it is has been used for evaluation by previous systems for recognition of small objects in LiDAR scans of cities. In particular it was first used by Golovinskiy et al. [12], who reported approximately 58% precision and 65% recall using fully automatic algorithms to recognize 1,063 objects amongst 17 semantic categories in a $0.3km^2$ "evaluation area." We executed our tests on similar set of 18 object categories (bush, fire hydrant, mailbox, newspaper box, parking meter, advertising kiosk, garbage can, recycle bin, phone booth, traffic sign, highway sign, A-frame sign, sidewalk light, street light, traffic ligth, short fence post, tall fence post, and car) in the same evaluation area of Ottawa, using a slightly expanded "ground-truth set" of 1,224 objects.

Data Segmentation. We segmented the LiDAR points into objects using the algorithms described in [11, 12]. Specifically, plane extraction algorithms were used to remove the points associated with major planar structures in the environment (ground and buildings), and then hierarchical clustering and graph cut algorithms were used to detect the locations of potential objects and cluster points into objects. The results of these automatic segmentations were improved interactively with a simple tool that allows a user to specify points inside and/or outside any object and then resolves the object segmentation boundary with a graph cut. These improved segmentations were used in our tests – the user only had to provide a label for each object.

We chose to create segmentations as a pre-process in our tests for several reasons. First, most small objects in a LiDAR scan of a city can be segmented automatically with high accuracy (90% in [12]), and so the segmentations in our tests are quite representative of those created with state-of-the-art algorithms. Second, segmentations can often be refined automatically after objects have been labeled – e.g., using an algorithm that aligns objects of the same type to form a consensus of the correct segmentation. So, precise pre-segmentations may not be necessary. Of course, there are complicated interplays between the accuracy of the segmentations with the effectiveness of the shape-based classifiers, the robustness of the consensus algorithms, the ability of people to recognize objects, the quality of the data, and so on. We defer investigation of these interplays to future work in order to focus on the main idea of the paper: group annotation. Finally, the proposed group annotation approach is not specific to labeling LiDAR data. It is a general idea for labeling any kind of data that can be shown in groups (perhaps cells in stained microscopic images, tumors in medical images, etc.). The segmentation challenges for each of those types of data is different. Hence, we avoid mixing our evaluation of the group annotation metaphor with the specific challenges of segmenting a particular data set.

User Study Design. To test whether group annotation is helpful for labeling objects in this data set, we ran a user study with 10 participants ranging in age from 20 to 40, including 4 women and 6 men.

We split the participants randomly into two sets of five. We asked one set of participants (GA) to label objects in the ground truth set using the group annotation interface described in this paper. We asked the other set (IA) to label the same objects with the same interface but with a classical least confidence active learning algorithm which suggests objects one-by-one. The order of assigning participants to one of these two conditions was random, and no participant was aware of which condition was the control.

Each participant was given written instructions and photographic examples of the types of objects to be labeled (copies are in the supplemental materials). Then, he/she executed a short interactive tutorial by labeling 163 objects covering all 18 object categories within a completely different area of Ottawa. During this tutorial, the participant was given immediate feedback if any label was assigned incorrectly and could proceed only after providing the correct label for every object.

Once the tutorial was completed, the participant proceeded to provide or confirm the label for all 1,224 objects in the evaluation area without any feedback or guidance using either the one-by-one or group annotation interface. Once done labeling all objects, he/she completed an exit questionnaire.

Experimental Results. During each user session, we logged all interactions and recorded times required to complete the task of providing/confirming labels for all objects in the evaluation set. Our hypothesis is that labeling all objects with the group annotation interface is faster.

Figure 3 shows a comparison of the results averaged across all five participants using each of the two interfaces. The horizontal axis represents the F-measure of the object labels predicted by the system with a nearest-neighbor classifier from the set of labels confirmed by the user, and the vertical axis indicates the earliest wall-clock time (in seconds) since the start of the test session for the system to achieve that F-measure. At the beginning, the system starts with no objects labeled, and so the curve starts at the origin. As it proceeds, the participant provides or confirms labels for more and more examples, which are fed into the training set, and so the nearest-neighbor classifier predicts larger numbers of objects correctly, and the F-measure rises. At the end (right side of the plot), every object has been confirmed by the participant explicitly, and so the F-measure is very close to 1.0 (deviations from 1.0 represent labeling mistakes by the participants).

We can make several interesting observations from this plot. First and foremost, they suggest that the group annotation interface (GA) provides faster times to task completion than the one-by-one interface (IA) – i.e., the group annotation curve is lower in the plot. In particular, the average time to completion is 2281+/-561 seconds with the group annotation interface and 3855+/-837 seconds with one-by-one labeling, a difference that is statistically significant at the 1% confidence level according to an independent two-sample two-tailed t-test. This result confirms our main hypothesis.

Second, the results suggest that the average number of labeling mistakes made by users of the group annotation interface (GA) is approximately the same as made with the one-by-one interface (IA). This can be seen by the fact that the f-measure of the rightmost point in both curves is almost the same in both plots ($95\% +/-1\%$ versus $95\% +/-2\%$).

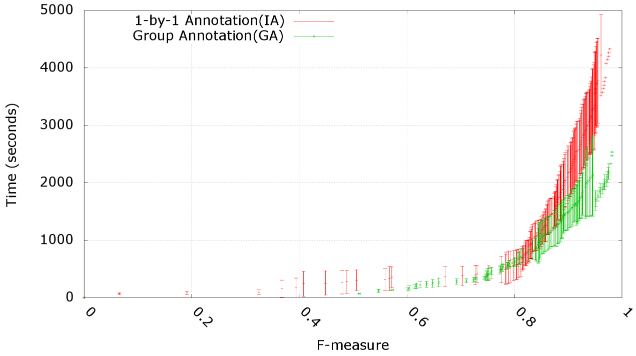


Figure 3: Comparison of average time to achieve F measure when labeling with group annotation (green) and 1-by-1 annotation (red) interfaces.

Finally, we see that it is possible for novice users to label 1,224 objects with 95% accuracy in approximately 40 minutes with group annotation – that's approximately one object every 2 seconds after a few minutes of training. This is much faster than alternative 3D labeling interfaces tested during pilot studies that require the user to interactively find and select mislabeled objects because the computer automatically moves the camera and selects objects, the operations that are very expensive for people to do.

Further Analysis. It is possible to analyze the activity logs and exit surveys to investigate further how people used the group annotation interface. For example, we can ask questions like: "how many groups were contracted by a user?," "what types of groups were contracted?," "were mistakes made more commonly in large groups of objects?," etc.

We first ask "how does the size of a group affect the probability that a person will provide a label for it?" Figure 4 addresses this question by showing a breakdown of what type of command users provided when presented groups of different sizes. The horizontal axis lists of the sizes of groups shown to the user. The vertical axis represents average numbers of objects in groups that were labeled/confirmed (green), contracted (red), or expanded (blue) for each group size. The plot shows that users labeled objects in groups as large as a few hundred objects at a time. For smaller groups the results show comparable frequencies of both labeling and contraction, and closer to the lower sizes of groups users prefer labeling to contraction. According to exit surveys, the main reasons for group contraction was impurity of the group, only 1 user mentioned that it was due to inability to see the entire group in the field of view. In the left-most bar, we see that ~20% of objects were labeled on their own, which is not surprising, since some object categories in this data set (e.g., mailboxes, garbage cans, fire hydrants, etc.) have instances scattered throughout the city, and thus are not conveniently labeled in a large group.

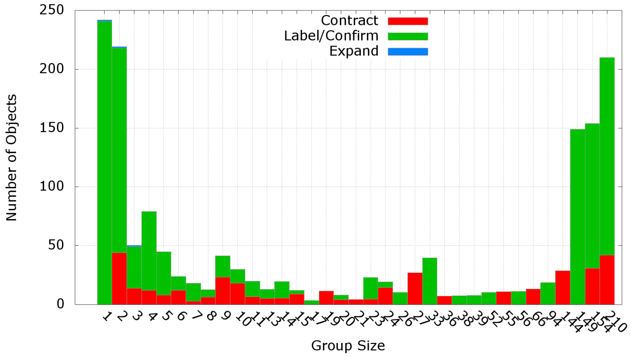


Figure 4: Average number of objects labeled (green), contracted (red), or expanded (blue) in groups of different sizes.

The second question we ask is "how does the size of a group affect the probability that a person will make a mistake when labeling it?" Figure 5 addresses this question by showing a breakdown of correct versus incorrect labels provided by users for groups of different sizes. Again, the horizontal axis lists of the sizes of groups shown to the user. The vertical axis represents average numbers of objects in groups that were labeled correctly (green) or incorrectly (red) for each group size. The plot indicates that the fraction of mistakes is larger for smaller groups. We have two explanations for this observation. First, some objects are very difficult to recognize – so the user probably contracts groups containing them before providing a label, and then sometimes still makes a mistake. Second, large groups appearing in regular patterns are easier to recognize than individual objects – the pattern provides context that aids recognition.

Overall, we conclude that the group annotation interface helps users label objects in less time and with as much accuracy as alternative interfaces based on one-by-one labeling.

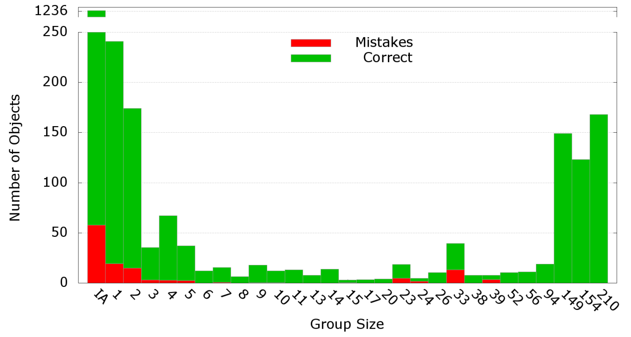


Figure 5: Average number of objects labeled correctly (green) or incorrectly (red) in groups of different sizes.

CONCLUSION AND FUTURE WORK

This paper investigates a group annotation approach to labeling 3D data. Besides the introduction of this idea, our research contributions include an active learning algorithm to construct groups of objects that minimize the expected time to label remaining objects, a perceptual model based on Gestalt principles to estimate the amount of time required for a user to recognize the label for a group of objects, and design of an interactive system that incorporates visualization of 3D point clouds with interactive labeling commands into an interface that can be learned by novices in a few minutes.

This paper provides an initial investigation and thus has many limitations. So far, we have focused only on pre-segmented data. This choice is appropriate for LiDAR scans of cities because automatic segmentation algorithms can achieve 90+% precision and recall for this type of data [12]. However, for other types of data (e.g., Kinect scans of interior environments), integration of segmentation into the labeling process provides an exta challenge that must be addressed. Second, we consider models of human recognition based only on shape similarities and spatial patterns, but of course other factors (e.g., color) are important as well and should be considered in further studies.

An interesting question for future work is to investigate "what other types of group annotation are most helpful?" We ask a user to provide a single label for *all* objects in a group or to indicate that none is possible. Previous work has asked users to provide a single label for *any* object in a group [8] or to explicitly select outliers before providing a label [7]. Different alternatives provide different levels of information and require different amounts of time for a user. We believe that the method proposed in this paper is one interesting point in this design space, but investigating other alternatives is an important topic for future work.

ACKNOWLEDGMENTS

We are grateful to Rebecca Fiebrink, David Mimno, Sean Gerrish, and Xiaofeng Ren for discussion and guidance. We thank Neptec, John Gilmore, and Wright State University for the 3D LiDAR dataset. We acknowledge NSF (IIS-1251217, CCF-0937139), Google, and Intel (ISTC-VC) for funding.

REFERENCES

1. Alvarez, G. A. Representing multiple objects as an ensemble enhances visual cognition. *Trends in cognitive sciences* (2011).

2. Amores, J. Multiple instance classification: Review, taxonomy and comparative study. *Artificial Intelligence 201* (2013), 81 – 105.

3. Ariely, D. Seeing sets: Representation by statistical properties. *Psychological Science 12*, 2 (2001), 157–162.

4. Chapelle, O., Vapnik, V., and Weston, J. Transductive inference for estimating values of functions. In *NIPS* (1999), 421–427.

5. Chapelle, O., and Zien, A. Semi-supervised classification by low density separation. In *Proc. 10th International Workshop on Artificial Intelligence and Statistics* (2005).

6. Chong, S. C., and Treisman, A. Representation of statistical properties. *Vision Research 43*, 4 (2003), 393–404.

7. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *Proc. CVPR* (2009), 248–255.

8. Dietterich, T. G., Lathrop, R. H., and Lozano-Pérez, T. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence 89*, 1-2 (1997), 31–71.

9. Gammerman, A., Vovk, V., and Vapnik, V. Learning by transduction. In *Fourteenth conference on Uncertainty in artificial intelligence* (1998).

10. Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)* (2013).

11. Golovinskiy, A., and Funkhouser, T. Min-cut based segmentation of point clouds. In *IEEE Workshop on Search in 3D and Video (S3DV) at ICCV* (2009).

12. Golovinskiy, A., Kim, V. G., and Funkhouser, T. Shape-based recognition of 3D point clouds in urban environments. *Proc. ICCV* (2009).

13. Haberman, J., and Whitney, D. The visual system discounts emotional deviants when extracting average expression. *Attention, Perception, & Psychophysics 72*, 7 (2010), 1825–1838.

14. Hengel, A. v. d., Dick, A., Thormaehlen, T., Torr, P., and Ward, B. Building models of regular scenes from structure and motion. In *Proc. BMVC* (2006).

15. Hick, W. E. On the rate of gain of information. *Quarterly Journal of Experimental Psychology 4*, 1 (1952), 11–26.

16. Itti, L., and Koch, C. Computational modelling of visual attention. *Nature Reviews Neuroscience 2*, 3 (2001), 194–203.

17. Kapoor, A., Grauman, K., Urtasun, R., and Darrell, T. Active learning with gaussian processes for object categorization. In *Proc. ICCV* (2007).

18. Kim, Y., Mitra, N., Huang, Q., and Guibas, L. J. Guided real-time scanning of indoor environments. *Computer Graphics Forum (Pacific Graphics) 32*, 7 (2013).

19. Lin, H., Gao, J., Zhou, Y., Lu, G., Ye, M., Zhang, C., Liu, L., and Yang, R. Semantic decomposition and reconstruction of residential scenes from lidar data. *ACM Trans. Graph. 32*, 4 (2013).

20. Musialski, P., Wonka, P., Aliaga, D. G., Wimmer, M., van Gool, L., and Purgathofer, W. A survey of urban reconstruction. In *EUROGRAPHICS 2012 State of the Art Reports*, Eurographics Association (2012), 1–28.

21. Nan, L., Sharf, A., Zhang, H., Cohen-Or, D., and Chen, B. Smartboxes for interactive urban reconstruction. In *Proc. SIGGRAPH*, ACM (2010), 93:1–93:10.

22. Neptec. Wright state 100, 2009. `http://www.ws-arc.org/applied-research-corporation/ottawa-data-files.html`. (Accessed April, 2014).

23. P. Welinder, P. P. Online crowdsourcing: rating annotators and obtaining cost-effective labels. In *Proc. CVPR* (2010).

24. Pelli, D., Majaj, N., Christian, C., Kim, E., and Palomares, M. Grouping in object recognition: the role of a Gestalt law in letter identification. *Cognitive Neuropsychology 26*, 1 (February 2009), 36–49.

25. Rosenholtz, R. A simple saliency model predicts a number of motion popout phenomena. *Vision Research 39*, 19 (1999), 3157 – 3163.

26. Russell, B. C., Torralba, A., Murphy, K. P., and Freeman, W. T. Labelme: A database and web-based tool for image annotation. *IJCV 77*, 1-3 (2008), 157–173.

27. Sanocki, T., and Epstein, W. Priming spatial layout of scenes. *Psychological Science 8*, 5 (1997), 374–378.

28. Settles, B. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

29. Shao, T., Xu, W., Zhou, K., Wang, J., Li, D., and Guo, B. An interactive approach to semantic modeling of indoor scenes with an rgbd camera. *ACM Trans. Graph. 31*, 6 (2012), 136:1–136:11.

30. Silberman, N., Kohli, P., Hoiem, D., and Fergus, R. Indoor segmentation and support inference from RGBD images. In *European Conference on Computer Vision* (2012).

31. Thorpe, S., Fize, D., and Marlot, C. Speed of processing in the human visual system. *Nature 381* (1996), 520.

32. Top, A., Hamarneh, G., and Abugharbieh, R. Active learning for interactive 3D image segmentation. *Med Image Comput Comput Assist Interv 14*, 3 (2011), 603–610.

33. Velizhev, A., Shapovalov, R., and Schindler, K. Implicit shape models for object detection in 3D point clouds. *ISPRS Annals I-3*, 2012 (2012), 179–184.

34. Vijayanarasimhan, S., Jain, P., and Grauman, K. Far-sighted active learning on a budget for image and video recognition. In *Proc. CVPR* (2010), 3035–3042.

35. Wagemans, J., Elder, J. H., Kubovy, M., Palmer, S. E., Peterson, M. A., Singh, M., and von der Heydt, R. A century of gestalt psychology in visual perception: I. perceptual grouping and figure–ground organization. *Psychological bulletin 138*, 6 (2012), 1172.

36. Wang, Y., Asafi, S., van Kaick, O., Zhang, H., Cohen-Or, D., and Chen, B. Active co-analysis of a set of shapes. *ACM Trans. Graph. 31*, 6 (2012), 165:1–165:10.

37. Welch, W. J. Algorithmic complexity: Three NP-hard problems in computational statistics. *Journal of Statistical Computation and Simulation 15* (1982), 17–25.

38. Wong, Y.-S., Chu, H.-K., and Mitra, N. J. Smartannotator: An interactive tool for annotating rgbd indoor images. *CoRR abs/1403.5718* (2014). http://arxiv.org/pdf/1403.5718.pdf.

39. Xiao, J., Hays, J., Ehinger, K., Oliva, A., and Torralba, A. Sun database: Large-scale scene recognition from abbey to zoo. In *Proc. CVPR* (2010).

40. Zhu, Q., and Yang, P. Density sensitive based spectral clustering. In *Computer and Information Sciences*, vol. 62. 2010, 139–142.