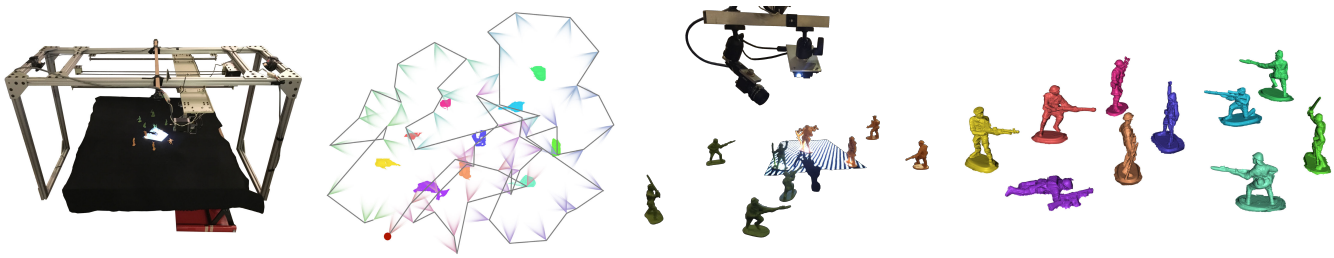
Automated View and Path Planning for Scalable Multi-Object 3D Scanning

Xinyi Fan
Princeton University

Linguang Zhang
Princeton University

Benedict Brown
University of Pennsylvania

Szymon Rusinkiewicz
Princeton University



(a) System Layout (b) View and Path Planning (c) Structured-light Scanning (d) Acquired 3D Models

Figure 1: *Our scanning system (a) automatically performs 3D scanning of multiple objects. Based on a silhouette-carved rough model, it plans views and a path to automatically scan all objects (b), positioning a structured-light 3D scanner to capture the necessary views (c). We are able to capture dozens of objects at once (d).*

Abstract

Demand for high-volume 3D scanning of real objects is rapidly growing in a wide range of applications, including online retailing, quality-control for manufacturing, stop motion capture for 3D animation, and archaeological documentation and reconstruction. Although mature technologies exist for high-fidelity 3D model acquisition, deploying them at scale continues to require non-trivial manual labor. We describe a system that allows non-expert users to scan large numbers of physical objects within a reasonable amount of time, and with greater ease. Our system uses novel view- and path-planning algorithms to control a structured-light scanner mounted on a calibrated motorized positioning system. We demonstrate the ability of our prototype to safely, robustly, and automatically acquire 3D models for large collections of small objects.

Keywords: 3D acquisition, view planning

Concepts: •**Hardware** → **Scanners;** •**Computing methodologies** → **Graphics input devices;** *3D imaging; Mesh models;*

1 Introduction

3D scanning is becoming a common and even expected mode of documentation for a variety of purposes. Naturally, 3D models represent the shape and surface characteristics of the tangible world more completely than 2D images. This benefits applications ranging from industrial inspection and online retailing to museum archive digitization and archaeological documentation. Fully realizing the potential of 3D scanning, however, will require scanning large numbers of objects with high quality and at reasonable cost. While scan quality and speed are continuously being improved by state-of-the-art scanning systems [Levoy et al. 2000; Rusinkiewicz et al. 2002; Brown et al. 2008; Yan et al. 2014], their dependence on manual labor remains a major bottleneck to scalability.

We argue that the key to making 3D scanning at scale practical is to reduce the manual effort required per object. This is in contrast to the design goals of many existing 3D scanning systems, which achieve high data quality within a reasonable amount of time but require the user to *plan* which views of the object are to be taken and possibly *position* the scanner and object relative to each other. Even if the set of views is fixed and the object is moved e.g. using a turntable, the user still interacts with the system every few seconds or minutes by positioning a new object, starting the scan sequence, and occasionally rotating the object to uncover parts that could not be seen. Streamlining the 3D scanning process therefore requires reducing the *number* of user interactions, not just their length.

We present a system for automatically scanning *multiple* 3D objects at a time. In our system, the user places several to several-dozen objects in the working volume, and the system automatically acquires their rough shapes and positions. The system then plans an optimal set of views to scan the objects at high quality, as well as the exact path along which the scan head should move. Using a 3-degree-of-freedom positioning system, our scanner automatically performs the 3D scans, restricting the necessary user interaction to placing the objects initially, then flipping them over halfway through scanning if necessary. The latter interaction could be avoided by placing the objects on a sheet of glass and using a second scan head to scan from below; we run preliminary experiment to explore the feasibility of this further refinement in this paper. We also explore scanning of larger objects by adding a manually-adjusted fourth degree of freedom.

The main benefit of our system, therefore, is that it allows the entire scanning process (which might take minutes or hours) to happen with no human interaction. In contrast, scanning the same number of objects one-by-one with an existing system might require a similar total scanning time, but with human interaction required every few seconds or minutes. Our system achieves additional benefits as well, including:

SA '16 Technical Papers, December 05 - 08, 2016, Macao
ISBN: 978-1-4503-4514-9/16/12 $15.00
DOI: http://dx.doi.org/10.1145/2980179.2980225

- **Reduction of the number of views** required to achieve a given surface quality, relative to spacing the views equally or using a greedy Next-Best-View (NBV) strategy.
- **Utilizing an optimal path** to reduce scan time.
- **Safety** for the objects being scanned, since the scanner is moved while the objects remain stationary. Furthermore, our positioning system keeps the scanner away from the objects by design, eliminating the possibility of collision.

This paper describes the design of our system, focusing on the novel view- and path-planning algorithms that enable automatic 3D acquisition. We also describe the design of our structured-light scanner and positioning system, which are optimized for acquisition of relatively small objects, such as fragments of archaeological artifacts. We demonstrate automated acquisition of two dozen objects at a time, though we believe that the system trivially scales to even larger working volumes.

2 Related Work

3D scanning. Various work has digitized objects at fine resolution using 3D scanning techniques [Bernardini and Rushmeier 2002]. Laser stripe triangulation has been widely used in previous work to acquire the 3D shape of archaeological artifacts of different size. Levoy et al. [2000] built a system which employs laser triangulation rangefinders to digitize large statues by Michelangelo. Brown et al. [2008] proposed a 3D model acquisition system for large numbers of fresco fragments with a laser scanner. In both works, laser scanners provide sub-millimeter resolution, but the data quality is proportional to scanning time, since slower sweeping of the laser stripe across the surface leads to higher-density acquisition.

Structured light triangulation accelerates the scanning process by projecting a set of temporally-coded patterns onto the object and returning a full range image at a time, as opposed to a single stripe of 3D data from a laser scanner. Various structured-light based acquisition systems have been designed to obtain high-quality 3D geometry data. Bernardini et al. [2002] used a lower-resolution structured light system coupled with photometric stereo to digitize Michelangelo's Florentine Pietà. Structured light scanning systems can be fast enough to achieve real-time 3D model acquisition [Rusinkiewicz et al. 2002; Weise et al. 2009]. Data resolution can be enhanced by optimizing pattern design [Salvi et al. 2004], and by combining fine details obtained from normal maps [Berkiten et al. 2014].

View planning. A variety of research has addressed the problem of view planning for 3D reconstruction and inspection [Scott et al. 2003]. Existing approaches can be categorized as model-based or non-model-based. Model-based methods can be divided into subcategories according to different techniques for model representation, including visibility matrices [Tarbox and Gottschlich 1995; Scott et al. 2001], aspect graphs [Tarbox and Gottschlich 1995; Bowyer and Dyer 1990], and "art gallery" floor plans [Urrutia 2000]. Solutions can be found in the research field of set theory, graph theory, and computational geometry. Similar work on optimal camera placement from the field of distributed sensor networks [Gonzalez-Barbosa et al. 2009; Zhao et al. 2013] can also be adapted to solve view planning for 3D acquisition. However, none of these methods incorporate explicit quality goals for the reconstructed object model, nor do they consider view-overlap constraints for registration. Recent work by Xu et al. [2015] proposed an automatic object-in-scene scanning system, but it requires physically moving the objects using the robot. This is incompatible with our goal of safe, contact-less scanning of valuable objects. Incorporating general robotic systems into 3D acquisition [Chen et al. 2011; Kriegel et al. 2013] is an interesting topic, but it would be difficult to guarantee safety for the objects being scanned.

Non-model-based view planning is also known as the Next-Best-View (NBV) problem. It seeks to find the viewpoint that provides the greatest expected reduction in uncertainty about the object being scanned [Scott et al. 2003]. Most of these methods assume no a priori knowledge about the object, and plan each view iteratively based on the acquired data. Wu et al. [2014] presented a Poisson-guided autonomous scanning method and demonstrated high-quality reconstruction. However, their method is quality-driven and does not aim to minimize the number of scans needed to cover the object's surface; it will therefore require a large number of views to scan multiple objects, with correspondingly long acquisition times.

Another related technique is based on viewpoint entropy [Vázquez et al. 2001]. It is generally concerned with selecting informative views, which is a little different from our need of complete coverage. However, among the choices for views that give full coverage, those that contain maximum entropy in the overlapping areas tend to align and merge most robustly.

Research on view and path planning can also be found in robotics [Wang et al. 2007; Cheng et al. 2008; Englot and Hover 2010], where the trajectory of agents is designed based on simplified, abstract models that capture environment features. Inspired by this, and considering our goal of reducing acquisition time, we design our system to first perform scene exploration to acquire a simplified model of the objects, then formulate the view planning as an optimization problem that optimizes viewing quality at every point on the model.

Positioning systems. Calibrated actuators are usually incorporated in 3D scanning systems, since multiple views are always necessary in order to obtain complete surface models. For example, Levoy et al. [2000] made use of a multi-degree-of-freedom gantry to achieve horizontal, vertical, and tilting motion for the scanner, making it possible to scan relatively hard-to-reach regions of the surface. However, the large working volume was specially designed for scanning single large objects like statues, and is not optimized for our scenario with multiple small objects. Brown et al. [2008] adopted a turntable to obtain scans from different views, but the working volume is limited by the size of the spinning plane.

3 System Overview

Capturing from multiple view points is necessary for a scanner to acquire a complete and high-fidelity 3D model of an object. The choice of views directly influences the overall scan quality since it determines whether there is full coverage of the object and whether good data can be captured for every part of every object (scan quality is generally affected by the object's distance from the scanner and angle of incidence). Most existing motorized scanning systems uniformly sample view space, often by rotating a turntable. Objects with deep concavities or other irregularities often need very dense sampling in this scenario, even if large parts of the object are convex and can be covered by few scans.

Our system, in contrast, optimizes the number and position of views using a low-resolution overview model acquired with a set of webcams. Because we move the (small) scan head rather than the (large) table of (potentially fragile or unsteady) objects, we can support a wider range of motion and obtain more optimal views. Our scanner supports automatic motion with three degrees of freedom — two directions of horizontal translation as well as rotation around the vertical axis — as a reasonable compromise between en-

gineering complexity and flexibility. It works well for scanning collections of small objects. For larger objects that need scans from different heights, we can manually raise or lower the scanning platform between sets of scans. In any case, the view planner handles arbitrary degrees of freedom if the scanning stage provides them.

3.1 System Design for Scanning Multiple Objects

Figure 1a illustrates the physical layout of our scanning system. The objects are placed on a flat, stationary platform, with the scanner mounted overhead at a fixed, 45° tilt. The scanner has three degrees of freedom of motion, which allows it to translate in the x and y directions (parallel to the platform) and rotate about the z axis. The tilt and height are both fixed, although they can be adjusted manually to accommodate objects of different sizes.

Automatic scanning systems typically move either the object or the scan head in order to obtain multiple views. Moving the object is more common, because a motorized turntable works well for single objects, is relatively easy to build, and does not take much space. Alternatively, robot- or vehicle-mounted scanners work well for navigation applications and for scanning buildings and large outdoor scenes.

Scanning many closely spaced, small objects at once falls into neither of these categories. Full coverage requires a denser set of views than a turntable can provide. The scanning stage would need, at a minimum, to move forward, back, and side-to-side, as well as rotating around its axis. To prevent objects from tipping over, breaking, or crumbling, vibrations would need to be damped. Because the scan head is small and can tolerate vibration, we believe that moving the scan head is a simpler and cheaper option to engineer.

A free-moving robot would run into a different problem in our scenario: it is an alternative way to move the scan head rather than the objects, but it would still need to navigate *between* the objects. Unless the objects are spaced far apart, this is a physical impossibility. Nevertheless, our view planning algorithm is heavily influenced by approaches from robotic navigation. (Of course, the robot could be an autonomous aerial vehicle that flies over the objects. That would provide more degrees of motion freedom than our gantry, but guaranteeing it will not crash into the objects would be more difficult.)

3.2 Pipeline

The design of our automatic acquisition system follows the workflow shown in Figure 1. The system starts with a scene exploration process, which examines the shapes and poses of the objects to be scanned. This information is passed to the view planning algorithm, which outputs an optimized set of scanner poses. Following an optimized path, the scanner is then brought to these desired poses by a calibrated positioning system and stops at each to perform a scan. Standard registration and integration algorithms are applied to the captured data to generate high-fidelity 3D models for the objects.

Scene exploration. Our system starts by finding the rough geometry of all the objects in the scene, then generating a set of candidate scanner views that will be a superset of the final selected views. The availability of cheap sensors makes it possible to quickly acquire sufficient information about the scene to enable view planning. Specifically, with the objects placed on the scanning platform, we use a set of fixed calibrated webcams to capture the layout of the objects from the top, then perform silhouette carving [Laurentini 1994] to obtain approximate object models for the view plan.

View planning. Based on the rough models produced by the scene exploration stage, we plan an optimal set of scanner poses (also referred as *views*). These adaptively cover the accessible part of the objects, while also ensuring a fair amount of overlap between adjacent views. Our view planning selects the best views by optimizing a view-quality-based objective function. Details of the view planning are presented in Section 5, which discusses several alternative approaches for optimizing the same objective.

Path planning and positioning. With the set of best views computed by view planning, we use a calibrated motion system to position the scanner at the desired poses. Due to stability concerns we assume that the scanner moves at a moderate speed, and hence in a scaled-up scenario with a large number of objects, the total travel time will be non-trivial. This motivates us to equip the positioning system with a path-planning component, which computes an approximate shortest path to traverse all the desired scanner poses. We discuss the path planning and calibration of our positioning system in Section 6.

Scanner setup. Once we have positioned the scan head, we acquire 3D data using a standard structured-light technique adapted from the work of Taylor [2012]. As illustrated in Figure 1c, the scanner consists of a compact camera (a 3.2-megapixel Point-Grey Flea3) and projector (a 0.4-megapixel TI DLP LightCrafter), mounted at an angle of approximately 20° to each other. Both devices are compact enough to be attached to the positioning system, and the center of the rig is attached to the rotational axis of the positioning system.

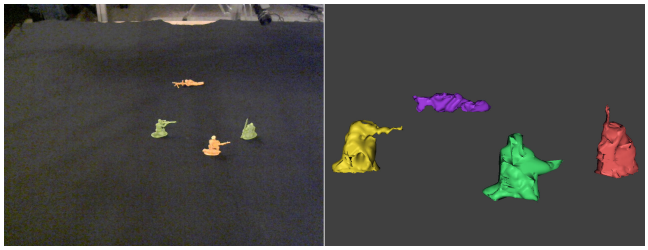
We use a combination of Gray code and phase-shift patterns for scanning. The LightCrafter is photometrically linear, so no special calibration is required to use phase-shift patterns. To make the most of the projector's resolution, we design the projected patterns to align with the orientation of the projector's mirror array. The camera can be synchronized to the projector either by using the sync signal as a trigger or by setting its exposure time to a multiple of the projector's refresh rate. We use the latter approach.

Registration and integration. We adopt standard techniques to register and integrate the scanned data into a complete 3D surface model. We perform ICP [Rusinkiewicz and Levoy 2001] to align multiple scans of a single object, with the initial poses provided by the calibrated positioning system. The aligned meshes are merged into a single complete model using VRIP [Curless and Levoy 1996] and screened Poisson surface reconstruction [Kazhdan and Hoppe 2013].

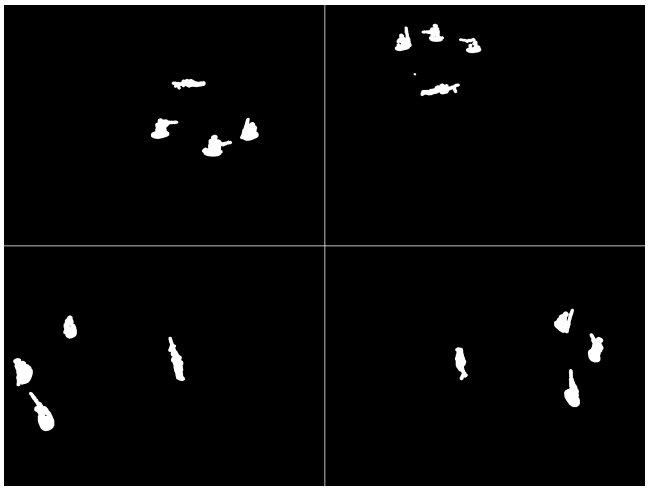
4 Scene Exploration

We perform a scene exploration step to obtain an approximate model of the scene with proxy geometry for all objects to be scanned. Our system generates a set of candidate scanner views based on this rough geometry, and passes both the rough geometry and candidate views to the view planner.

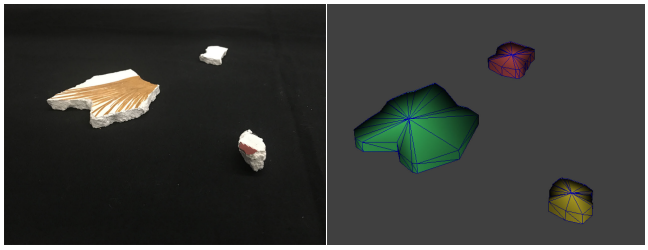
Approximate object models. The objects are placed on the scanning platform, which is covered in black cloth for ease of object segmentation. Four static, calibrated webcams positioned around the platform capture images of the scene from above. The webcam poses are calibrated using the patterns in Figure 6a, which will be described in Section 6. We run background subtraction to segment the objects from the captured images — Figure 2b shows the object masks. Silhouette carving is performed on these masks, and the carved volume surfaces are triangulated into meshes, where the



(a) A scene with four toy soldiers (left) and the approximate models obtained via silhouette carving (right).



(b) Four binary masks of the four toy soldiers scene obtained from the webcams for silhouette carving.



(c) A scene with three flat objects (left) and the approximate models obtained via extruding the 2D contours, where mesh triangle edges are shown to better present the model shape (right).

Figure 2: *A scene exploration step is performed to acquire approximate models for the objects, and such models will be employed as input to the following view planning.*

(user-specified) n largest connected components are detected as the approximate models, as illustrated in Figure 2a. We use a $200 \times 200 \times 200$ voxel grid and have not observed any view planning problems from missing data, but it is possible to expose the grid size as a parameter to handle objects with finer detail such as thin protrusions.

For flat objects we simplify the carving process by extracting 2D contours and extruding them upwards by a user-specified height to approximate the 3D shapes, as shown in Figure 2c. In our experiments, view planning has never been sensitive to variations in the thickness to which we extrude the contours.

We believe that depth sensors may also provide a solution for obtaining rough 3D models, and in some situations they may work better than silhouette carving. For small objects, however, we observe that the resolution of currently available depth sensors, such as Kinect, is inadequate to improve upon the models produced by silhouette carving.

Candidate scanner views. Our view planning approach (discussed below) is based upon selecting a subset of candidate views that provide sufficient coverage of the 3D surface of an object. To generate these candidate views, we first fit an elliptical cylinder to the approximate object model, then dilate the ellipse by several different amounts corresponding approximately to the scanner's "standoff" (i.e., the distance between the camera position and points ranging from the front to the back of the scanner's working volume). The candidate scanner positions are obtained by uniformly sampling angles on the ellipses. At each potential scanner position, we consider a number of scanner orientations centered around the direction facing the middle of the object. The view planner selects a small subset of these candidate views that provides both complete coverage of the object and enough overlap between views to support scan registration.

In our current setup, the user-adjustable parameters for candidate view sampling are the radii of the ellipses around which we select views, the different heights of the scanner (constant for small objects), the angular density of views around each object, and the maximum angular deviation of the scanner from each object center. The angular deviation can be increased when the object shape is extreme, e.g. the long thin geometry as shown in Figure 15. Table 1 shows the parameters used in our experiments.

Table 1: *Default setting for the user-adjustable candidate view sampling parameters.*

parameter	default setting
ellipse radii	10 to 20 cm plus the object bounding box diagonal radius
scanner heights	1.5 to 3 multiples of the object bounding box height
angular density	10°
angular deviation	$\pm 20^\circ$

5 View Planning

The goal of view planning is to automatically find a suitable subset of the candidate views, from which the scanner can acquire the complete surface of an object with high quality. We first define a per-point view quality score, then integrate it over multiple views and many surface samples to form an objective function that measures the quality of a complete set of views. Finally, we explore ways of optimizing this objective, including a greedy method, simulated annealing, and integer programming for an approximate version of the objective. We are not aware of previous work that enables systematic comparison across algorithms for optimizing the same view-quality objective function.

5.1 View Quality Metric

Given an approximate object model provided by scene exploration, we begin by defining a view quality function that measures how well a 3D point on the object surface p is "seen" by a single scanner view v:

$$f(p, v) = h(p, v) \cdot g(p, v), \tag{1}$$

where $h(p, v)$ is a visibility term and $g(p, v)$ is a geometry term. Note that a scanner view v is defined by all its constituent optical devices, and those devices can be either sensors (e.g., cameras) or lighting devices (e.g., projectors). Our prototype adopts a one-camera, one-projector structured-light configuration, but the met-

ric developed in this paper applies generally to any multi-camera, multi-projector setup.

Visibility term. A point is *visible* to a device view if the point is within the field of view of that device and the point is not occluded by other parts of the surface model. We define the visibility term as a binary function

$$h(p,v) = \begin{cases} 1 & \text{if } p \text{ is } \textit{visible} \text{ to all device views at } v, \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

We obtain each device's field of view by calibration, and check occlusion by performing efficient ray-mesh intersection.

Geometry term. For points that are visible to a scanner view, we define the geometry term to quantify how well each point is "seen" from the view:

$$g(p,v) = \max\{0, \min\{\vec{c}_v^{(1)} \cdot \vec{n}_p, \vec{c}_v^{(2)} \cdot \vec{n}_p, ..., \vec{c}_v^{(K)} \cdot \vec{n}_p\}\}, \quad (3)$$

where $\vec{n}_p$ is the surface normal vector at point p, K is the total number of optical devices in the scanner setup, and $\vec{c}_v^{(k)}$ is the viewing vector from p to the center of projection of device k. The dot products are clamped at 0 because the surface becomes invisible when the angle between the two vectors is less than 90°. The geometry term ensures that all the optical devices "see" the point frontally.

Notice that our geometry term does not take into account the distance between the object and the sensor in the geometry term, because the working volume of our scanner is small enough relative to its distance from the camera center that it made no difference. However, for setups where the working volume spans a large depth, a distance term can be re-incorporated to encourage the point to be seen somewhere close to the in-focus plane of the scanner.

Integration. Given a candidate scanner view set V and a surface sample set P, we integrate the per-point, per-device view quality scores $f(p,v)$ over V and P to form an objective function that measures the quality of any subset of scanner views from V. For some selected set of scanner views $V^* \subset V$, we therefore define the best view for each point as

$$\beta_1(p) = \arg\max_{v \in V^*} f(p,v). \quad (4)$$

The basic objective function that we want to maximize can be written as

$$F(P,V^*) = \frac{1}{|P|} \sum_{p \in P} f(p, \beta_1(p)) - \gamma \cdot |V^*|, \quad (5)$$

where γ represents the cost of introducing one more view. We select the value of γ such that with one more view added, the summation of view quality in the objective function should increase by γ in expectation.

Note that for each point p we do not take the summation of its view qualities over all views, but instead over only the best one. This will ensure that each point has at least one "good" view, as opposed to a larger number of views with mediocre quality.

5.2 Overlap-Aware Objective

The basic objective function in Equation 5 encourages full "good view" coverage over all the points. It does not, however, necessarily guarantee *overlap* between scans from adjacent views, which is essential to the subsequent registration step. We therefore propose heuristics to improve the objective function so that it addresses view overlap.

Second-best views. In order to acquire more accurate data and encourage view overlap for registration, we would like each point to be "seen" by the scanner from at least *two* views as opposed to only *one* as indicated in Equation 5; and hence for some selected set of views V^* we define the second-best views for each point as

$$\beta_2(p) = \arg\max_{v \in V^* \setminus \{\beta_1(p)\}} f(p,v). \quad (6)$$

The objective function is then re-written as

$$F_2(P,V^*) = \frac{1}{|P|} \sum_{p \in P} f_2(p, \boldsymbol{\beta}(p)) - \gamma \cdot |V^*|, \quad (7)$$

where

$$f_2(p, \boldsymbol{\beta}(p)) = (1-\epsilon) \cdot f(p, \beta_1(p)) + \epsilon \cdot f(p, \beta_2(p)). \quad (8)$$

In this equation, $\epsilon \in [0,1]$ is a user specified weight that defines how much we rely on the quality of the second-best views. Now we ensure that each point has at least two "good" views.

Neighborhood view quality aggregation. To encourage overlap around sharp corners, we measure the view quality of a point more conservatively by evaluating the view quality of all points in its neighborhood. For any point $p \in P$ with its small neighborhood $\mathcal{N}(p) \subset P$, and a given view v, the neighborhood aggregated view quality is defined as

$$f_N(p,v) = (1-\tau) \cdot \min_{p' \in \mathcal{N}(p)} f(p',v) + \tau \cdot \frac{1}{|\mathcal{N}(p)|} \sum_{p' \in \mathcal{N}(p)} f(p',v). \quad (9)$$

Plugging this into Equations 7 and 8, we obtain a new objective

$$F_{2,N}(P,V^*) = \frac{1}{|P|} \sum_{p \in P} f_{2,N}(p, \boldsymbol{\beta}(p)) - \gamma \cdot |V^*|, \quad (10)$$

where

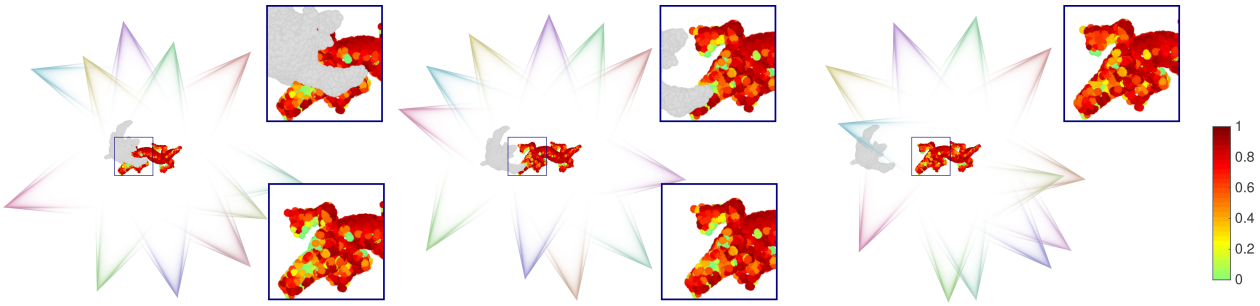
$$f_{2,N}(p, \boldsymbol{\beta}(p)) = (1-\epsilon) \cdot f_N(p, \beta_1(p)) + \epsilon \cdot f_N(p, \beta_2(p)). \quad (11)$$

Figure 3 shows an example that visualizes view quality with different objective functions. With only a single best view considered and no neighborhood heuristic (Equation 5), all the points have very good view quality, but the view assignment is not addressing overlaps. When the second best view is also considered (Equation 7), the planning tends to add slightly more views and encourages overlap between some pairs of adjacent views. When neighborhood aggregation is introduced to the objective function, with the same view cost but only a single best view considered (Equation 10 with $\epsilon = 0$), it sacrifices per-point view quality in favor of encouraging overlap where it is often difficult to achieve manually, such as around sharp corners. The hybrid objective with both neighborhood aggregation and second-best view (Equation 10 with $\epsilon = 0.5$) also gives reasonable results, but usually uses the most views because it is the most conservative. We adopt the single-best-view objective function with neighborhood aggregation (Equation 10 with $\epsilon = 0$) in the following experiments.

(a) *single best view (Eq. 5)* ***(b)*** *two best views (Eq. 7)* ***(c)*** *single best view + neighborhood (Eq. 10 with $\epsilon = 0$)* ***(d)*** *two best views + neighborhood (Eq. 10 with $\epsilon = 0.5$)*

Figure 3: *View assignment quality visualization with different objective functions. Each scanner view is represented by a camera-projector pair of frustums connected with a dotted line. The object model is represented by surface samples with the view quality value encoded according to the jet color map, as shown in (d). Red means good view quality and green means poor.*

(a) *Models of "dragon fighting armadillo" as the armadillo moves from "close" (left) to "near" (middle), and then "far" (right) from the dragon. The models are synthesized at similar level of quality with the approximate models acquired from the scene exploration.*



(b) *Visualization of the surface sample view quality based on the corresponding selected views for the three different scenes: "close" (left), "near" (middle), and "far" (right). Red represents good view quality and green poor. The scanner views are simplified by only visualizing corresponding camera views. Zoomed-in views of the dragon head are shown to illustrate the view quality change. For the "close" and "near" scenes, we show closeups of the dragon head both with and without the armadillo occlusion.*

Figure 4: *Given a scene of "dragon fighting armadillo" with increasing distance between the two objects (a), we visualize the surface sample view quality based on the corresponding selected views (b). The close-up views show that, as the armadillo moves from "close" (left) to "near" (middle), and then "far" (right) from the dragon, the view quality of the head of the dragon improves.*

Multi-object objective. Our view quality metric easily generalizes to the multi-object case. We sum up the objective function over all objects, modifying the visibility term by checking occlusion from all object surfaces in the scene to avoid inter-object occlusion. Figure 4 shows an experiment evaluating occlusion detection performance in three different cases. As shown at right, when there is plenty of space between the two objects, the view rendered in light blue is selected to cover most of the region of the dragon head. However, when the armadillo is moved closer, as shown in the middle and left, the originally desired view can no longer see the dragon's head well due to occlusion, and therefore the view planner has to select alternate views from further back. As illustrated in the zoomed-in area, the view-quality visualization provides feedback to the user in these cases: a significant amount of green area suggests that flipping or rearranging the objects will be necessary to acquire a complete model. In most cases, however, inter-object occlusion detection ensures proper view selection to avoid occlusion.

5.3 Optimizing the View Planning Objective Function

We explore several different approaches to solving the view planning problem. In practice, the positioning system that moves the scanner to each selected view is limited by the precision of its motors and gears; it is therefore reasonable to examine the space of scanner candidate views as a discrete space. Given that each point needs to be seen at least once or twice, depending on the choice of objective function, maximizing the objective reduces to the classic NP-complete set-cover and multicover problems [Karp 1972]. Therefore, we explore a number of ways of approximating the problem in order to find solutions with practical computation time.

Sequential greedy optimization. An intuitive way of optimizing our objective function is using the classic greedy approach. In fact, there are inapproximability results [Feige 1998] showing that

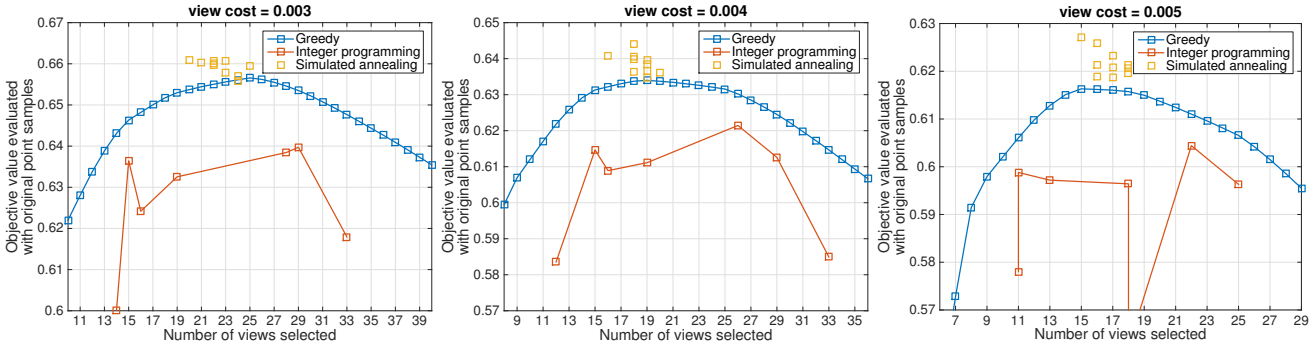


Figure 5: *The optimal objective value achieved by different approaches with varying view cost, with higher value indicating better overall view quality.*

the sequential greedy approach is the best possible polynomial-time approximation algorithm for set cover. In our scenario, we begin with $V^* = \emptyset$ and iteratively add the view that yields the largest increase in the objective function. The number-of-views penalty term γ in our objective ensures that, at some point, no new view can be found that leads to an increase in the objective function, terminating the algorithm and controlling the number of views we select.

Simulated annealing. The greedy approach is simple to implement and very efficient, but due to its deterministic nature the objective will not improve once a local optimum is achieved. Simulated annealing [Kirkpatrick et al. 1983] is a probabilistic method for approximating the global optimum of an objective function that may possess many local optima, at a cost of relatively long running time.

Algorithm 1 details our implementation of simulated annealing for optimizing the objective in Equation 10. The algorithm is initialized with random views, and at each iteration updates a state vector $\vec{X} = [X_1, X_2, ..., X_{|V|}]$ consisting of indicator variables representing whether a candidate view is selected or not, such that $V^* = \{v \,|\, X_v = 1, v \in V\}$. While a basic implementation might simply enable or disable a single view at each iteration, we take advantage of the structure of the candidate view space V to improve efficiency. Specifically, with probability one-half we swap some view v for a neighboring view $v' \in \mathcal{N}(v)$, instead of simply switching a view on or off. The energy function $E(\vec{X})$ guiding whether a state transition is accepted is set equal to the objective function $F(P, V^*)$ with V^* defined by $\vec{X}$, and the annealing temperature T decreases exponentially.

As shown below, we find that simulated annealing, if given a slow-enough annealing schedule and enough iterations, typically outperforms the greedy approach. Moreover, it automatically decides the exact number of views needed in the optimal solution based on the view cost parameter γ.

Integer programming. Another way to approximate the view planning optimization is to formulate it as a binary integer programming problem. In this case, the objective function needs to be quantized based on a view quality threshold η, and thus given a point p and a view v, measuring the view quality becomes simply checking whether it is "good enough", namely above η. Specifically, we define a set of indicator variables W_{pv}, which are 1 if $f(p, v) > \eta$ and 0 otherwise. The objective function is then approximated by

$$\sum_{p \in P} \min\{W_{pv} \cdot X_v,\ |\boldsymbol{\beta}(p)|\} - \gamma \sum_{v \in V} X_v. \qquad (12)$$

Algorithm 1 Simulated Annealing for View Planning

Input: random initialization $\vec{X}_0$
  repeat
    draw Pr from uniform (0, 1) distribution
    if $Pr <$ threshold **then**
      randomly select view $v \in V^*$
      randomly select view $v' \in \mathcal{N}(v)$
      $\vec{X}'_t \leftarrow \vec{X}_t$ with X_v and $X_{v'}$ swapped
    else
      randomly select view $v \in V$
      $\vec{X}'_t \leftarrow \vec{X}_t$ with X_v flipped
    end if
    $T_t = \alpha^t$, $\Delta E = E(\vec{X}'_t) - E(\vec{X}_t)$
    if $\Delta E > 0$ **then**
      $\vec{X}_{t+1} \leftarrow \vec{X}'_t$
    else
      with probability $\exp(\Delta E \cdot T_t)$, $\vec{X}_{t+1} \leftarrow \vec{X}_t$
      with probability $1 - \exp(\Delta E \cdot T_t)$, $\vec{X}_{t+1} \leftarrow \vec{X}'_t$
    end if
    $t \leftarrow t + 1$
  until convergence

where $|\boldsymbol{\beta}(p)|$ is the number of best views considered for each point. A branch-and-bound method [Gurobi Optimization 2015] is applied to solve this integer program exactly.

5.4 Evaluation

We evaluate the performance of the three approaches on the same dataset by comparing the optimal objective values they obtain, as shown in Figure 5. In each figure, the blue curve shows the evolution of the objective value against the number of views selected by the sequential greedy algorithm, with the ultimate result of the greedy algorithm being the highest point. The red curve shows the objective value achieved by integer programming, with different values of the view quality quantization threshold η. The scattered orange squares are results from 10 different runs of simulated annealing, using different random seeds.

Varying View Cost. The three plots in Figure 5 show results for different values of the view cost multiplier γ. We leave the choice up to the user, to select γ to be the desired increase in the average view quality, as one additional view is added. As γ increases, the required benefit of adding a view increases, and hence the optimal number of views decreases.

Algorithm Comparison. The figures show that simulated annealing achieves better objective values compared to the greedy approach and integer programming. With the threshold η properly chosen, the integer programming can perform as well as the greedy approach, but is less predictable, since the number of views and ultimate quality do not vary monotonically with η. While simulated annealing does require more computation (a few minutes per object), we generally prefer it for our system. If this computation time is unacceptable, the greedy algorithm usually picks a near-optimal number of views, though the views themselves may be sub-optimal. We also provide a clustering strategy to help improve efficiency, which will be discussed in Section 8.

6 Path Planning and Positioning

We propose a novel positioning system that is designed to support efficient 3D acquisition of multiple objects. Motion of the system is calibrated so that the scanner is able to arrive at desired poses based on the view planning results. Because the motors' travel time is not trivial compared to the entire acquisition process, we compute a path that minimizes the total time to traverse all the scanner views, which is especially beneficial to acquisition at scale.

6.1 Path Planning

Once we have obtained a set of desired scanner positions for each object within the working volume, planning the optimal path among them is naturally formulated as the Traveling Salesman Problem (TSP) [Lawler et al. 1985]. Between any pair of views, we compute a motion cost corresponding to the time taken by the positioning system to move between those views, taking into account that motion along multiple axes can happen simultaneously. We solve the TSP on a complete graph, where each node in the graph corresponds to a scanner pose (x, y, θ_z). For any pair of nodes (x_i, y_i, θ_{zi}) and (x_j, y_j, θ_{zj}), there is an edge between them, and the distance is defined as the travel time

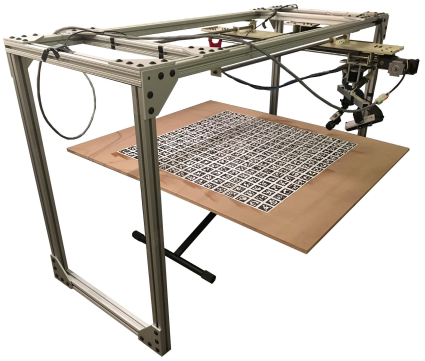
$$\max\left\{\frac{|x_i - x_j|}{v_x}, \frac{|y_i - y_j|}{v_y}, \frac{\min\left\{|\theta_{zi} - \theta_{zj}|, 2\pi - |\theta_{zi} - \theta_{zj}|\right\}}{v_{\theta_z}}\right\},$$

where v_x, v_y, and v_{θ_z} respectively represent the motor speed along the linear axes and around the rotational axis. We use the algorithm of Christofides [1976] to obtain a $\frac{3}{2}$-approximated optimal path.

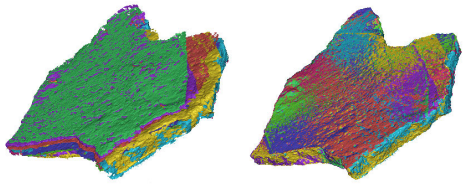
6.2 Motion Calibration

Motion ability. As shown in Figure 1a, the positioning system consists of 2 linear axes orthogonal to each other, and a rotational axis orthogonal to the plane they define. The scanner is attached to the rotational axis. The system is driven by three stepper motors: two for translation with a step size of 0.05 mm and one for rotation with a step size of 0.9°. Speed and acceleration of the motors is controlled by an Arduino-based micro-controller.

Global coordinates. The scene exploration, view planning, and scanning need to happen in a unified coordinate system, so that the positioning system is able to accurately position the scanner to reach the poses specified by view planning, and scanned data from different views can be registered and integrated into a complete model. A global coordinate system is defined by employing the AprilTags fiducial system [Olson 2011], where each tag is a unique 2D bar code. Our calibration pattern uses 256 tags arranged in a 16×16 2D array and glued onto the scanning platform, as shown in Figure 6a. This pattern is used to calibrate all the sensors employed in our acquisition system, including the four fixed RGB cameras used for scene exploration and the camera in the structured light rig for scanning. Camera extrinsics are estimated by taking a picture of the calibration pattern and detecting the unique tags with their poses known in the global coordinates.



(a) *The positioning system, with a calibration target on the scanning platform.*



(b) *The initial scan poses provided by the scanner calibration (left), together with the final result of registration (right).*

Figure 6: *Calibration of our scan system. The overall working volume is approximately 1 square meter, while the scanned object is about* $8 \times 8 \times 1$*cm.*

Transform fitting. The motor controller receives commands in the form of (x^s, y^s, θ_z^s) triplets, but these need not correspond to the global coordinate system defined by the AprilTags. To calibrate the motor coordinates, we employ an interpolation-based strategy. During the calibration phase, the positioning system moves the scanner to a set of sparse samples in (x^s, y^s, θ_z^s) space, and at each stop the scanner captures an image of the calibration pattern to compute its corresponding pose in the global coordinates. A quadratic model is fit to the sampled data, to interpolate the transform from any desired pose in global coordinates to a motor command triplet. The reason for a quadratic, rather than linear, model is to account for any flex of the linear rails along which the axes move. After calibration, the positioning system achieves 0.5 cm accuracy over approximately a 1 m $\times$ 1 m area. Figure 6b shows the accuracy provided by our initial calibration, and the good final alignment achieved with automatic registration beginning with those poses.

7 Results

We have conducted experiments evaluating the view- and path-planning components of our system, as well as the system as a whole. In each case, we present results for scanning time and quality, comparing our system to possible simpler implementations. We also demonstrate that our system is capable of scanning a variety of 3D objects with different geometry. The structured-light scanner in our system can achieve a 0.1 mm resolution.

7.1 View Planning Evaluation

To demonstrate that our view plan improves the *combination* of scan time and quality, we compare the acquisition results based on

our view planning algorithm to those from a naive strategy commonly adopted by previous work, namely placing views uniformly around an object's centroid, at a fixed radius.

Efficiency. We demonstrate the improvement in efficiency due to introducing view planning into the acquisition pipeline on two test scenes (on both sides) with four objects, each object having different shape and size. We run our view planning algorithm to obtain the view schedules for all the objects. Then we compare to a naive strategy with a fixed number of views, spaced equally around the centroid of each object, with the number of views set equal to either the fewest or most views associated with any object in our view-planning result from each scene.

Table 2: *Comparison of total time for our view planning vs. naive strategies employing a fixed number of views per object.*

	min fixed	max fixed	adaptive
total number of scans	44	64	52
planning time (min)	2×10^{-5}	2×10^{-5}	3.20
total scan time (min)	27.28	40.30	32.63
total travel time[1](min)	7.97	11.27	8.83
total time (min)	35.25	51.57	44.66
avg. time per object (min)	8.06	12.89	11.17

[1] Note that in Table 2 the total travel time includes a five second pause per scan, for vibration damping before capture, while in Table 3 the total travel time refers to the amount of time the positioning system spent on moving only.

Table 2 summarizes the acquisition time for each of these scenarios. Note that we achieve an improvement over the naive plan with the maximum number of views, with no penalty (or even improvement) in the quality of the acquired data. Of course, our view planning strategy is not as fast as the naive method with the minimum number of views, but it is less prone to missing areas of the surface or ending up with low data quality.

Figure 7 shows the final reconstructed models of the objects on both sides from the acquisition with the adaptive view plan.

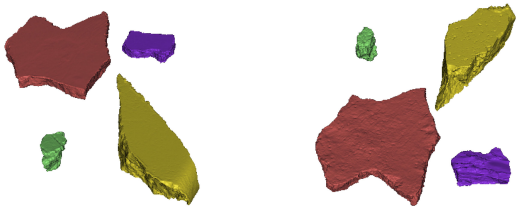


Figure 7: *Front (left) and back (right) side of the reconstructed models of four objects scanned simultaneously with adaptive view planning. Models with the same color correspond to each other.*

Coverage. Unlike the naive approach, which equally distributes a fixed number of views around an object, the simulated annealing based view plan adaptively selects the number of views for each object. Therefore, an acquisition with view planning usually yields better coverage, especially for non-convex objects. We show a comparison on the scans of an object acquired in the last 4-object experiment. Figure 8 shows a closeup to the aligned raw scans from naive methods and our adaptive view plan (white regions indicate missing data). The scans acquired from the naive method with five views are missing data from both the tip of a sharp corner and a deep concavity. Even with the same number of views equally spaced around the object, the naive method with nine views is still missing some data at the tip of the sharp corner. With the neighborhood aggregation improvement added to our objective function, the view plan optimization places views around sharp corners to increase *meaningful overlap*.

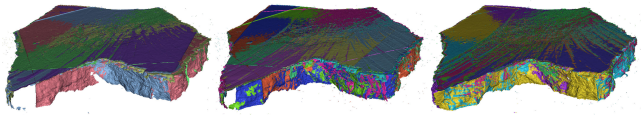
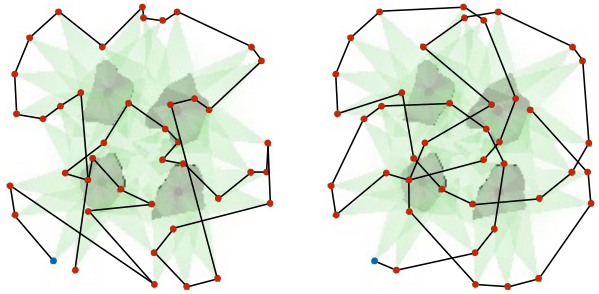


Figure 8: *We compare the scans obtained using our view planning (right) to those acquired with a naive method employing five (left) or nine (middle) views, equally spaced around the centroid of the object. Our view planning result also selects nine views in this case.*



(a) *TSP-based path planning* ***(b)*** *Naive path planning*

Figure 9: *Comparison of our TSP-based path planning (a) and naive path planning (b). The former reduces motion time by approximately* 15%.

7.2 Path Planning Evaluation

Given a set of views for multiple objects produced by the view planning stage, we compare our TSP-based path planning strategy to a naive algorithm. For the latter, we use the views selected for each object in sequence, always beginning the scanning of an object from the nearest view to the last one in the previous object. Figure 9 shows the results of the two different strategies for a simple scene with four objects.

Table 3 compares the travel distances and times for the two strategies. Notice that the TSP-based strategy achieves an improvement in travel time of 15%, even with as few as four objects. For more objects, we have observed even greater savings in travel time, with small increases in computation time.

Table 3: *Comparison of total distances and times for our TSP-based path planning vs. a naive path planning strategy.*

	our path plan	naive path plan
number of scans	44	44
planning time (μs)	579	19
total translation distance (m)	5.30	6.05
total rotation distance (deg)	2330	2690
total travel time (s)	228	268

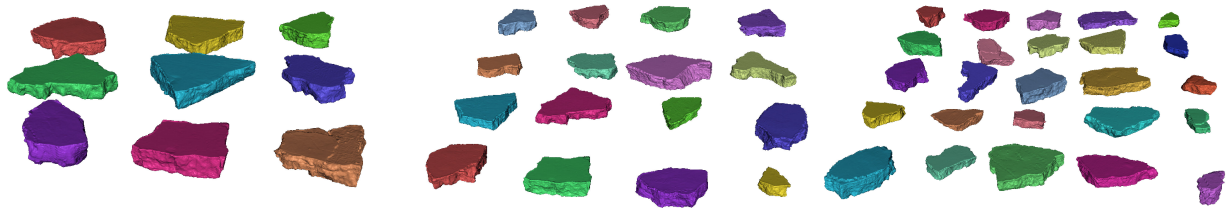


Figure 10: *Reconstructed models of increasingly-large sets of fresco fragments, as used in our scalability experiment. The batch size is, from left to right, 9, 16, and 25.*

(a) *Approximate object models.*

(b) *Reconstructed models.*

(c) *Close-up visualization.*

Figure 11: *We scan a scene with ten toy soldiers based on views planned from approximate, silhouette-carved models (a), and reconstruct high resolution 3D models (b). Each toy soldier is about* 5 *cm tall, and our system reconstructs sub-millimeter geometric detail (c).*

7.3 System-Level Evaluation

We compare the performance of our system to another acquisition system which employs the same structured light scanner, but uses a turntable as a simple positioning system. The turntable system adopts the naive view planning strategy described above, which uniformly samples a fixed number of views around the table center. We assume that the turntable system chooses the average number of views for the objects planned by our algorithm as its fixed number of views. Therefore the scanning time per view of both systems should be approximately the same.

Figure 12 presents comparative statistics from scanning batches of fragments using both systems, illustrating how the total human interaction time and the idle time between interactions scale up with increasing number of objects scanned. In this set of experiments we use fresco fragments as test objects, which are an important category of objects in archaeological digitization applications.

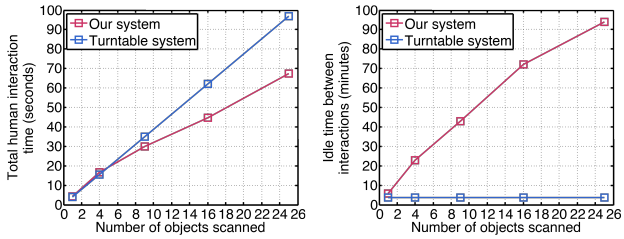


Figure 12: *The two plots show the total amount of time a human interacted with the scanning system (left) and the total amount of idle time when the human did not need to attend the system between two adjacent interactions (right) on scanning batches of fresco fragments using both our system and the turntable system. Our system required less interaction time overall and afforded far larger gaps between interactions during which the operator was free to do other work undisturbed.*

Scalability. The interaction time for the turntable system is linear in the number of objects, because for each single object the average operating time stays the same. On the other hand, the total interaction time for our system grows (slightly) sub-linearly, indicating that our system makes large scale acquisition tasks more efficient.

More important is the comparison of (human-operator) idle time. This shows a significant advantage of our system over the turntable system in that the user is free from tending to our system for long periods of time. This is essential for practicality in a scaled-up scanning scenario. In the case of fresco fragments, the idle time is actually half of the entire acquisition time, because each fragment is flipped once during the acquisition to obtain data from both sides. This leads to only two interactions with the system, while the turntable system requires constant flipping and replacing of objects. Notice that the sub-linear scalability of the idle time is mostly due to the travel time, as a result of our path planning.

Quality. Figure 10 shows the reconstructed models for arrangements of 9, 16, and 25 objects scanned using our acquisition system. Based on manual inspection of the resulting 3D models, our system achieves at least comparable surface coverage over the objects being scanned, as compared to a turntable-based system.

7.4 Object Variety Evaluation

We demonstrate the capability of our system to scan a variety of 3D objects in addition to the fresco fragments.

Multiple small 3D objects. Figure 1 and Figure 11 together show an example of scanning a scene with ten toy soldiers. Each soldier is represented in the same color in Figures 1b, 11a and 11b. The entire scene is scanned from the views visualized in Figure 1b, and the scanner travels along the path planning result, where the color changes in camera view visualization correspond to the order along the path.

As demonstrated in Figure 11a, the approximate models acquired from our silhouette carving-based scene exploration capture sufficient meaningful detail, as opposed to the models obtained from consumer depth sensors such as the Kinect, which tend to have more random noise and miss detail. Our structured-light scanner achieves a resolution of 0.1 millimeter and captures an abundance of detail on the toy soldiers, as illustrated by the closeup visualization in Figure 11c. We note the importance of capturing structures such as the long barrel of the weapon in the exploration phase: this is used in the subsequent view-planning stage to place the necessary scanner positions to capture this tricky area. The reconstructed results suggest the potential of our system to be used for large-scale capture of stop-motion animation.

Large object. Figure 13 shows a reconstructed model acquired from our scanning system compared to the real figurine. The angel figurine is about 20 centimeters tall and has complicated self-occlusion. Scanning it from a single height would yield a large amount of missing data. Thus, we augment our prototype system with a platform that we can manually raise and lower. We restrict the number of heights (to three for this experiment), and calibrate them, allowing all of the scan positions to still be planned using the same view-planning algorithm. The final model is reconstructed by combining all of the scans, using a pipeline essentially identical to that for the single-height case.

Figure 13: *A 20 cm tall angel figurine (left) is scanned and reconstructed (right) using our system, with the object platform adjusted to three heights.*

Figure 14: *Two differently sized soldiers (left) are scanned together and reconstructed (right) with our system.*

Objects with different scales. Our system is capable of simultaneously scanning objects with different scales thanks to the adaptive view planning. Figure 14 shows two soldiers at different sizes and their reconstructed scanned models. As introduced in Section 4, candidate views are sampled based on an elliptical cylinder fit to each approximate object model. In this case, the candidate views for the larger soldier span a much wider range compared to those for the smaller soldier. This allows greater flexibility in the scale of objects being scanned at the same time, compared to a turntable scanning system in which the candidate views are always sampled on a circle with fixed radius.

Long, thin object. Figure 15 demonstrates that our system is able to handle extreme geometry such as the flower with its long, thin stem. Due to the fact that we compute candidate views based on an elliptical cylinder fit to the approximate object model, it is easy for our system to focus on areas such as the stem and the backs of the flower's petals. A turntable scanning system that places views uniformly around the object at a fixed radius is likely to yield very poor surface coverage of this flower.

Figure 15: *A flower with a long, thin stem (left) is scanned and reconstructed (right) with our system. Elongated objects such as this are a worst case for a turntable-based system with equally-spaced views.*

Cultural heritage. Figure 16 shows a reconstructed model of a reproduction cuneiform tablet, which along with fresco fragments forms another important category of objects in archaeological digitization. The inscriptions on the tablet are clearly captured by our high-fidelity structured-light scanner, and with the scalability of our system we believe it would be easy to digitize these artifacts en masse with little human interaction, thus setting archaeologists and conservators free from tedious tasks.

Figure 16: *An $8 \times 6 \times 3$ cm reproduction cuneiform tablet (left) is scanned and reconstructed (right) with our system.*

8 Conclusion and Discussion

We propose a scalable prototype that automates the 3D acquisition of multiple objects with novel view and path planning algorithms. Our system significantly reduces the per-object human interaction time associated with 3D acquisition, which should lead to the broader use of 3D scanning in a variety of fields.

Scalability. Our system is designed as a prototype for large-scale 3D acquisition, and we have shown a set of evaluations on how our system scales up. Currently the computation time of our simulated annealing based view plan algorithm for each object ranges from several seconds to several minutes, single threaded on a CPU, depending on the density of candidate views and surface samples. We believe that the design of independently computing the optimal set of views for each object makes it easier for the view plan to scale up, since the optimization for multiple objects can be computed in parallel.

In addition, we provide a surface-sample clustering strategy to further reduce the view plan time consumption. Given a set of candidate views and a surface sample set, we compute a feature vector for each surface sample based on its response to all the candidate views. A bottom-up clustering is then performed to group "similar" samples, and the number of clusters are controlled with a distance threshold. Preliminary results show that the objective function is still well-preserved when we halve the number of surface samples, as shown in Figure 17.

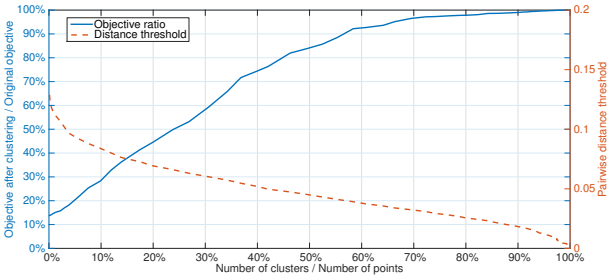


Figure 17: *Clustering to* $50-60\%$ *of the original points does a good job of preserving the objective function (blue curve). The red curve shows how the number of clusters is related to the distance threshold chosen for clustering. The graph is generated starting from a sampling rate comparable to the resolution of the coarse model provided by scene exploration.*

Generalization. Currently our system focuses on acquiring a surface geometry model. It would be interesting to generalize the view planning to support appearance acquisition as well. This would involve augmenting our current view quality function with a new term representing the expected response of a point sample to controlled illumination, which would evaluate whether a given view is also good for photometric capture.

Registration and object flipping. Registration is always a required part of the post process in a standard acquisition pipeline. Scanning multiple objects at a time provides more global information for registering scans for the same scene, compared to scanning with a single object system. However, our current prototype requires flipping the objects to scan their under-sides, which in fact creates a new scene. There is no easy way of aligning the front side to the back side globally. The strategy we adopt now is to perform global alignment within the front side scene and the back side scene to obtain models integrated for both sides, and then to segment out each object to perform the back-to-front alignment independently. One possible future direction is to explore global back-to-front registration algorithms that automatically account for the user interaction of flipping each fragment.

A different approach to solving this problem is to entirely avoid the flipping interaction. Our system can be augmented by replacing the platform with transparent material so that a second scan head can be employed to scan objects from below. We have run experiments to test the feasibility of this technique, as shown in Figure 18.

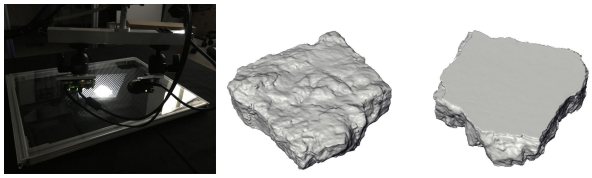


Figure 18: *We scan the front side of an object using our system as usual but the back side through a sheet of transparent thin acrylic (left) and obtained the final models nicely reconstructed for both the front (right) and back (middle) side, which are then merged together.*

Limitations. The quality of our initial scan alignments is limited by the precision of the scan-head's motor control. While the existing initial estimates of alignment are usually sufficient for automatic registration using ICP, inaccurate initial poses complicate both automatic segmentation and registration of flat (and otherwise underconstrained) objects such as fresco fragments. Adding encoders to the motors to precisely read off their positions would lead to greater robustness in post-processing.

Our system is also limited in the motion ability of the scan head, since we only have three automatic degrees of freedom in our positioning system. For objects with significant self-occlusion, this could require a number of manual adjustment on the platform height, and/or re-positioning the objects in order to acquire the scene.

By using two scissor-jack lifting platforms, we have demonstrated the possibility of introducing an additional degree of freedom (vertical translation) with the system still calibrated, and we believe it would be simple to motorize the axes of the lifting platform. Because the view planning algorithm supports arbitrary scan-head motion, a more complex gantry design can use the same planner to scan a more diverse group of objects at once.

The limitation can also be ameliorated by guiding the user how objects should be moved for optimal scanning. A global computation based on our (currently per-object) view quality metric and a search over potential ways to re-position the objects could be used.

Acknowledgements

We would like to thank all the people who have provided helpful suggestions, encouragement, and feedback for this project, particularly Tim Weyrich, Camillo J. Taylor, James Bruce, David Radcliff, Joanna Smith, and the members of the Princeton Graphics Group. We also thank all the reviewers for their constructive comments. This work is supported by NSF grants CCF-1027962, IIS-1012147, and IIS-1421435.

References

BERKITEN, S., FAN, X., AND RUSINKIEWICZ, S. 2014. Merge2-3D: Combining multiple normal maps with 3D surfaces. *Proc. Int. Conf. 3D Vision (3DV)* (Dec.), 440–447.

BERNARDINI, F., AND RUSHMEIER, H. 2002. The 3D model acquisition pipeline. *Computer Graphics Forum 21*, 2, 149–172.

BERNARDINI, F., RUSHMEIER, H., MARTIN, I. M., MITTLEMAN, J., AND TAUBIN, G. 2002. Building a digital model of Michelangelo's Florentine Pietà. *IEEE Computer Graphics and Applications 22*, 59–67.

BOWYER, K. W., AND DYER, C. R. 1990. Aspect graphs: An introduction and survey of recent results. In *Proc. SPIE: Close-Range Photogrammetry Meets Machine Vision*, vol. 1395, 200 – 208.

BROWN, B. J., TOLER-FRANKLIN, C., NEHAB, D., BURNS, M., DOBKIN, D., VLACHOPOULOS, A., DOUMAS, C., RUSINKIEWICZ, S., AND WEYRICH, T. 2008. A system for high-volume acquisition and matching of fresco fragments: Reassembling Theran wall paintings. *ACM Trans. Graphics (Proc. SIGGRAPH) 27*, 3.

CHEN, S., LI, Y., AND KWOK, N. M. 2011. Active vision in robotic systems: A survey of recent developments. *Int. J. Robotics Research 30*, 11, 1343–1377.

CHENG, P., KELLER, J. F., AND KUMAR, V. 2008. Time-optimal UAV trajectory planning for 3D urban structure coverage. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2750–2757.

CHRISTOFIDES, N. 1976. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical Report 388, Graduate School of Industrial Administration, Carnegie Mellon University.

CURLESS, B., AND LEVOY, M. 1996. A volumetric method for building complex models from range images. In *Proc. ACM SIGGRAPH*, 303–312.

ENGLOT, B., AND HOVER, F. 2010. Inspection planning for sensor coverage of 3D marine structures. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 4412–4417.

FEIGE, U. 1998. A threshold of Ln N for approximating set cover. *J. ACM 45*, 4, 634–652.

GONZALEZ-BARBOSA, J.-J., GARCÍA-RAMÍREZ, T., SALAS, J., HURTADO-RAMOS, J.-B., AND RICO-JIMÉNEZ, J.-D.-J. 2009. Optimal camera placement for total coverage. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 3672–3676.

GUROBI OPTIMIZATION, I., 2015. Gurobi optimizer reference manual. http://www.gurobi.com.

KARP, R. M. 1972. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, Eds. Plenum, 85–103.

KAZHDAN, M., AND HOPPE, H. 2013. Screened Poisson surface reconstruction. *ACM Trans. Graph. 32*, 3, 29:1–29:13.

KIRKPATRICK, S., GELATT, C. D., AND VECCHI, M. P. 1983. Optimization by simulated annealing. *Science 220*, 671–680.

KRIEGEL, S., BRUCKER, M., MARTON, Z. C., BODENMLLER, T., AND SUPPA, M. 2013. Combining object modeling and recognition for active scene exploration. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2384–2391.

LAURENTINI, A. 1994. The visual hull concept for silhouette-based image understanding. *IEEE Trans. PAMI 16*, 2, 150–162.

LAWLER, E. L., LENSTRA, J. K., KAN, A. R., AND SHMOYS, D. B. 1985. *The traveling salesman problem: a guided tour of combinatorial optimization*, vol. 3. Wiley.

LEVOY, M., PULLI, K., CURLESS, B., RUSINKIEWICZ, S., KOLLER, D., PEREIRA, L., GINZTON, M., ANDERSON, S., DAVIS, J., GINSBERG, J., SHADE, J., AND FULK, D. 2000. The Digital Michelangelo Project: 3D scanning of large statues. In *Proc. ACM SIGGRAPH*, 131–144.

OLSON, E. 2011. AprilTag: A robust and flexible visual fiducial system. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 3400–3407.

RUSINKIEWICZ, S., AND LEVOY, M. 2001. Efficient variants of the ICP algorithm. In *Proc. 3D Digital Imaging and Modeling (3DIM)*, 145–152.

RUSINKIEWICZ, S., HALL-HOLT, O., AND LEVOY, M. 2002. Real-time 3D model acquisition. *ACM Trans. Graph. 21*, 3, 438–446.

SALVI, J., PAGÈS, J., AND BATLLE, J. 2004. Pattern codification strategies in structured light systems. *Pattern Recognition 37*, 827–849.

SCOTT, W. R., YZ, W. R. S., ROTH, G., AND RIVEST, J.-F. 2001. View planning as a set covering problem. Tech. Rep. 44892, NRC Canada.

SCOTT, W. R., ROTH, G., AND RIVEST, J.-F. 2003. View planning for automated three-dimensional object reconstruction and inspection. *ACM Computing Surveys 35*, 1, 64–96.

TARBOX, G. H., AND GOTTSCHLICH, S. N. 1995. Planning for complete sensor coverage in inspection. *Computer Vision and Image Understanding 61*, 1, 84–111.

TAYLOR, C. 2012. Implementing high resolution structured light by exploiting projector blur. In *Proc. IEEE Workshop on Applications of Computer Vision (WACV)*, 9–16.

URRUTIA, J. 2000. Art gallery and illumination problems. In *Handbook of Computational Geometry*, J. Sack and J. Urrutia, Eds. Elsevier.

VÁZQUEZ, P.-P., FEIXAS, M., SBERT, M., AND HEIDRICH, W. 2001. Viewpoint selection using viewpoint entropy. In *Proc. Vision Modeling and Visualization (VMV)*, 273–280.

WANG, P., KRISHNAMURTI, R., AND GUPTA, K. 2007. View planning problem with combined view and traveling cost. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 711–716.

WEISE, T., WISMER, T., LEIBE, B., , AND GOOL, L. V. 2009. In-hand scanning with online loop closure. In *Proc. 3D Digital Imaging and Modeling (3DIM)*.

WU, S., SUN, W., LONG, P., HUANG, H., COHEN-OR, D., GONG, M., DEUSSEN, O., AND CHEN, B. 2014. Quality-driven Poisson-guided autoscanning. *ACM Trans. Graph. 33*, 6, 203:1–203:12.

XU, K., HUANG, H., SHI, Y., LI, H., LONG, P., CAICHEN, J., SUN, W., AND CHEN, B. 2015. Autoscanning for coupled scene reconstruction and proactive object analysis. *ACM Trans. Graph. 34*, 6, 177:1–177:14.

YAN, F., SHARF, A., LIN, W., HUANG, H., AND CHEN, B. 2014. Proactive 3D scanning of inaccessible parts. *ACM Trans. Graph. 33*, 4, 157:1–157:8.

ZHAO, J., YOSHIDA, R., CHING SAMSON CHEUNG, S., AND HAWS, D. 2013. Approximate techniques in solving optimal camera placement problems. *Int. J. Distributed Sensor Networks*, Article ID 241913.