

A Cable-driven Parallel Mechanism for Capturing Object Appearance from Multiple Viewpoints

J.-D. Deschênes, P. Lambert, S. Perreault, N. Martel-Brisson, N. Zoso, A. Zaccarin, P. Hébert
Computer Vision and Systems Laboratory, Laval University

{desche07, hebert}@gel.ulaval.ca

S. Bouchard, C. M. Gosselin

Robotics Laboratory at the Department of Mechanical Engineering, Laval University

samuel.bouchard.1@ulaval.ca, gosselin@gmc.ulaval.ca

Abstract

This paper presents the full proof of concept of a system for capturing the light field of an object. It is based on a single high resolution camera that is moved all around the object on a cable-driven end-effector. The main advantages of this system are its scalability and low interference with scene lighting. The camera is accurately positioned along hemispheric trajectories by observing target features. From the set of gathered images, the visual hull is extracted and can be used as an approximate geometry for mapping a surface light field. The paper describes the acquisition system as well as the modeling process. The ability of the system to produce models is validated with four different objects whose sizes range from 20 cm to 3 m.

1. Introduction

Appearance modeling for a large variety of object types is currently a challenge that motivates intense research. One obvious application is the realistic real-time visualization of objects under free viewpoint selection. For this purpose, the appearance is captured, modeled and rendered while taking into account light rays arising from all of the viewpoints. One well-known approach consists in capturing a light field of the object [12]. The light field can be further compressed by capturing an approximate geometry of the object before encoding the set of rays that emerge from the object's surface [6, 16]. In order to produce such a model, a large quantity of images must be captured from all of the viewpoints surrounding an object.

Capturing and processing all of these images essentially requires the automation of the whole acquisition and modeling process. Several systems have thus been implemented in the last 15 years, including [3, 9]. For moving or deforming objects, a large set of cameras is typically used [9].

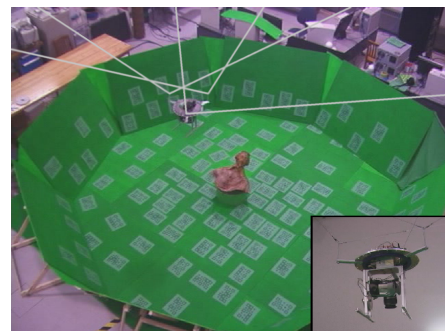


Figure 1. The cable-driven parallel mechanism in action. The end-effector holds a SLR camera that is used to acquire images of the object from multiple viewpoints. The end-effector is suspended by six cables passing through pulleys attached to the walls of the room. The object is surrounded by a green enclosure to allow easy extraction of the object silhouette as well as precise positioning.

However, when the object is static, it is advantageous to use a robotic arm supporting a camera in the end-effector. This makes it possible to capture a dense set of views for modeling. For more flexibility, it is also possible to complement with a turntable as long as the light sources remain rigid with the object. All of these systems differ in their intrinsic precisions and flexibility to capture objects of various sizes.

In this paper, we propose an acquisition and modeling system based on a cable-driven parallel mechanism for positioning and orienting a camera. One implementation of the acquisition system is illustrated in Fig. 1 where one can see the camera module that moves around the object. There are significant advantages for developing such a system. Indeed, cable-driven robots can reach a large workspace such as a quasi-hemisphere around the object. More importantly,

they are highly scalable, thus enabling them to capture objects of various sizes ranging from a few centimeters to several meters. Moreover, the cables do not interfere significantly with the environment lighting.

Cable-driven cameras are commercially available but are not well suited for object modeling. Skycam [2] and Cablecam [1] are two examples mainly dedicated to television events. Major limitations of such devices are their workspace which is mostly constrained in a plane and the lack of accurate positioning.

In the next sections, a full proof of concept is described and demonstrated from acquisition to rendering for objects whose sizes range from 20 cm to 3 m. The robot including the end-effector apparatus and its kinematic characteristics are described in section 2. Although the robot can be controlled and positioned accurately for planning trajectories around the object, more accurate positioning is obtained using photogrammetry. This is presented in section 3. In section 4, a passive appearance modeling approach is described. Section 5 presents an evaluation of the system.

2. Description of the Cable-Driven Robot

In order to obtain the different viewpoints necessary for building a 3D model, the camera is typically moved along a hemispherical surface centered on the object, as shown in Fig. 2. Spherical coordinates are used to define the position of the mechanism in that workspace. Ideally, the camera should also be oriented such that its optical axis intersects with the center of the object for every photograph. Thus, the mechanical device used to move the camera should be able to reach such a workspace. The radius of the hemisphere depends on the size of the digitized object and on the settings of the camera. It is worth noting that the six degrees of freedom describing the workspace are coupled in such a way that if two of them are specified (e.g. ϕ and θ), the other four can be deduced. This greatly simplifies the analysis and design of the mechanism.

A six-cable suspended robot [5] is adapted for this task. The geometry of the mechanism is defined as the position of the attachment points of the 6 cables. The cables are attached to the camera platform (the end-effector) and pass through pulleys attached to the walls of the room. These points on the wall are referred to as departure points. The challenge in the design is that this workspace involves relatively large orientation changes ($\pm 90^\circ$ tilt) as well as large translations.

2.1. Kinematic Optimization

Physically, the goal of the optimization process is to obtain a mechanism with a static equilibrium workspace as close as possible to the desired workspace, without the

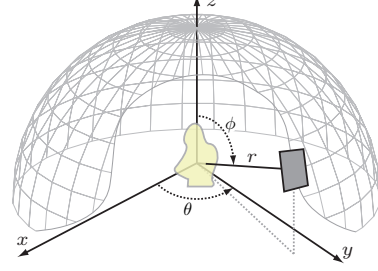


Figure 2. Workspace of the mechanism around the object. The front portion has been removed for clarity.

cables interfering with each other or with the end-effector. In a static equilibrium workspace, all cables are in tension. In the proposed system, the camera will be moved slowly from one pose to the next, thus the static equilibrium workspace defined in [14] is considered. The departure points are constrained to lie on the walls of the room. To avoid interference between the cables and the camera or the electronic devices on the end-effector, the attachment points are constrained to lie on a circle with a variable radius r between 0 and 25 cm. Their positions are optimized under these constraints following a procedure similar to the one described in [4]. For a given robot configuration, the objective function to be minimized is

$$\Phi = \sum_{\phi=0}^{\pi/2} \sum_{\theta=0}^{2\pi} -s(\phi, \theta) + c(\phi, \theta) \quad (1)$$

which is the sum of two penalty terms over the workspace. In this equation, $s(\phi, \theta) = 1$ if the mechanism can be in static equilibrium at the pose defined by coordinates ϕ and θ . If not, $s(\phi, \theta) = 0$. Similarly, $c(\phi, \theta) = 1$ if a cable crossing occurs between two steps of the procedure. Otherwise, $c(\phi, \theta) = 0$.

In order to verify that the robot is in static equilibrium, the following equations are defined,

$$[\mathbf{w}_1 \quad \mathbf{w}_2 \quad \cdots \quad \mathbf{w}_6] \mathbf{t} = -\mathbf{r}, \quad (2)$$

$$\mathbf{W}\mathbf{t} = -\mathbf{r} \quad (3)$$

where \mathbf{t} is a vector containing all of the cable tensions and \mathbf{r} is the wrench applied by gravity to the mechanism. Vector \mathbf{w}_i is the unit wrench vector of cable i which can be written as

$$\mathbf{w}_i = \begin{bmatrix} \mathcal{O} & \mathbf{u}_i \\ \mathcal{Q} & \mathcal{E} \mathbf{b}_i \times \mathbf{u}_i \end{bmatrix} \quad (4)$$

where \mathcal{O} and \mathcal{E} refer to the fixed reference frame $\mathcal{R}_{\mathcal{O}}$ and the frame $\mathcal{R}_{\mathcal{E}}$ attached to the end-effector, respectively.

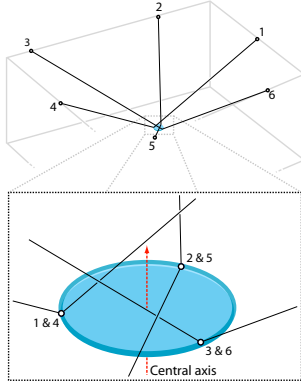


Figure 3. Illustration of the attachment points of the system. Top: Departure points ${}^0\mathbf{a}_i$. Bottom: End-effector points ${}^E\mathbf{b}_i$.

Moreover, ${}^E\mathbf{b}_i$ is the position vector of the attachment point of cable i on the end-effector expressed in \mathcal{R}_E , ${}^O\mathbf{Q}_E$ is the rotation matrix from \mathcal{R}_O to \mathcal{R}_E , and \mathbf{u}_i is a unit vector along cable i originating from the end-effector. It is expressed by

$$\mathbf{u}_i = \frac{\mathbf{a}_i - \mathbf{b}_i}{\|\mathbf{a}_i - \mathbf{b}_i\|}, \quad (5)$$

with \mathbf{b}_i equal to $\mathbf{b}_i = \mathbf{c}_E + {}^O\mathbf{Q}_E {}^E\mathbf{b}_i$. In this equation, \mathbf{a}_i is the position vector of the attachment point of cable i on the wall expressed in \mathcal{R}_O , and \mathbf{c}_E is the position vector of the origin of \mathcal{R}_E expressed in \mathcal{R}_O . The vectors with no prefix are expressed in \mathcal{R}_O . The wrench exerted by gravity on the end-effector \mathbf{r} can be balanced by the mechanism if the tension vector \mathbf{t} that satisfies (3) also satisfies the condition: $\mathbf{t} \succeq 0$. In this case, it means that all of the cables must be in tension to balance the wrench and $s(\phi, \theta) = 1$.

With regards to the objective function, the best architecture that was found encompasses 3 different attachment points on the end-effector and could reach the complete translational workspace with a maximum end-effector tilt angle of 30° , which is far from the ideal 90° . This architecture is illustrated in Fig. 3. To avoid the orientation limitation, an actuated device could be embarked on the platform to orient the camera. This would have the drawback of adding weight and complexity at the end-effector. For the sake of simplicity, minor modifications were made to the concept without adding a complete level of actuation on the end-effector. These modifications are detailed in section 2.3.

The robot is controlled using joint coordinates. For a desired Cartesian pose ${}^O\mathbf{T}_E$, the cable lengths are calculated and then actuated without real-time feedback on the actual pose of the end-effector. The cable lengths are cal-

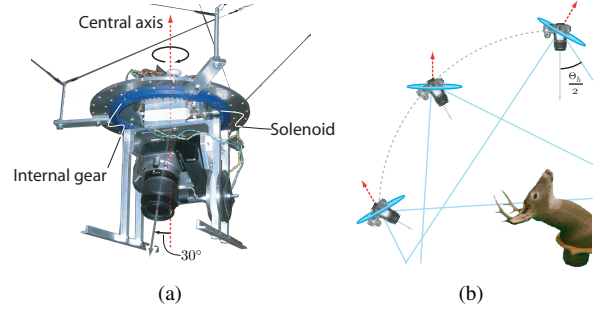


Figure 4. a) End-effector apparatus. b) Polar trajectory showing camera offset angle. The rays indicate the larger field of view of the camera. The arrow represents the optical axis of the camera.

culated using the geometric parameters of the robot ξ_k , a set of variables that comprises all of the attachment points. The attachment points on the end-effector ${}^E\mathbf{b}_i$ are measured directly. The departure points on the wall, \mathbf{a}_i , can be calibrated using a photogrammetric approach or directly measured with a theodolite. Although both approaches were tested, this will not be discussed any further in this paper. Errors on the geometric model can result in errors in the pose of the camera. However, since it is an important issue for appearance modeling, the precise pose of the camera will be obtained by photogrammetry. For this reason, inaccuracy in the robot pose does not affect the modeling of the object.

2.2. Trajectory definition

The system scans over a hemisphere. For a given azimuth angle, the mechanism goes step by step from $\phi = 0$ to $\phi = 90^\circ$. At each step, the mechanism stops smoothly, stabilizes and a picture is taken. Stability measures for such trajectories are presented in section 5.1. At the end of a polar trajectory, the azimuth angle is incremented and the mechanism is moved back up. This continues until the complete hemisphere shown in Fig. 2 has been covered. A part of the trajectory and a more detailed view of the device can be seen in the video extension 1 on the website of the authors ¹. Since the desired orientation is not achievable, in the next section we show how the tilt angle ϕ_t of the end-effector can be made to differ from the polar angle ϕ .

2.3. End-effector apparatus

The mechanism can tilt the end-effector to a maximum angle of 30° , which is insufficient for the application. Instead of having the optical axis of the camera coincide with the central axis of the end-effector, the axis is offset. The tilt

¹<http://vision.gel.ulaval.ca/~vision3d>

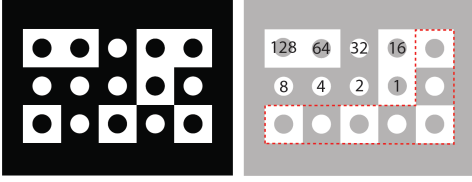


Figure 5. Matrix code representing the value 46. The view at the right shows the pattern used to determine orientation (dashed L) and the weight of each bit. Dark and light color circles indicate a value of 0 and 1 respectively.

angle ϕ_t of the end-effector is chosen to be different from the polar angle ϕ such that:

$$\phi_t = f\phi + \delta. \quad (6)$$

The trajectory can then be adapted so that the tilt of the end-effector is -30° when ϕ is 0° , and 30° when ϕ corresponds to 90° . It varies linearly between both extremes so variables $f = \frac{2}{3}$ and $\delta = -\frac{\pi}{6}$ in eq. 6. This is illustrated in Fig. 4 b).

To ensure that the camera takes portrait pictures for all azimuth angles, its rotation angle needs to be adapted around the central axis of the end-effector. To achieve this, the implemented device shown in Fig. 4 a) exploits gravity to passively orient the camera around that axis. To do so, the camera is installed on a flat bearing whose rotation axis coincides with the central axis of the end-effector. The part of the end-effector that is attached to the cables does not rotate and is equipped with a large internal gear. A solenoid is installed on the rotating part. While unactuated, it blocks the gear and hinders rotation of the camera. When it is actuated, the camera can rotate freely around the end-effector's central axis. Since the center of mass of the rotating part is not on the rotation axis, the camera will rotate under the action of gravity when the solenoid is activated and the end-effector is tilted. This orientation change is undergone each time that ϕ reaches a maximum. A demonstration of the orienting device is shown in the video extension 1.

The camera mounted on the end-effector is a 6.3 megapixel Canon Rebel 300D with an extended battery pack to provide an autonomy of approximately one thousand pictures. The camera and solenoid are remotely controlled by a base station via a RS-232 wireless link. The data stream is decoded by a μ controller which activates the camera or the solenoid depending on the requested command. The camera was adapted so that it can be activated by the μ controller. The total weight of the end-effector including the camera, lens, batteries, electronics, and end-effector components is 5.75kg.

3. Data Processing and Object Modeling

At the end of the acquisition, images are transferred to a computer and the offline modeling process begins.

3.1. Camera Pose Computation

The camera's pose is obtained by observation of automatically recognizable patterns scattered around the object. A model of the 3-D coordinates of these targets is computed from the set of acquired images along the trajectory. Given the model of target coordinates, one can compute the image \mathbf{x} of a known 3-D point \mathbf{X} with the classical projection model [8]:

$$\underbrace{\begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} sf & 0 & u_0 & 0 \\ 0 & f & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\mathbf{K}} \underbrace{\begin{bmatrix} \mathbf{Q} & \mathbf{c} \\ \mathbf{0} & 1 \end{bmatrix}}_{\mathbf{T}} \underbrace{\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}}_{\mathbf{X}} \quad (7)$$

This projection model is augmented with a nonlinear lens distortion model expressing corrected image coordinates \mathbf{a}_c in terms of their distorted counterparts \mathbf{a}_d :

$$\mathbf{a}_c = \mathbf{a}_d + \mathcal{F}_D(\mathbf{a}_d, \delta). \quad (8)$$

It is shown in [8] how to invert the distortion model and obtain \mathbf{a}_d from \mathbf{a}_c .

For positioning, a target pattern that is easy to detect and identify has been devised in the form of matrix codes. It is shown in Fig. 5. Its circular features' image coordinates are accurately measured using [13].

Matrix code targets are placed around the object of interest such that at least five are visible from each point of view. The cable robot then displaces the camera around this static scene, taking a high number of pictures. The problem is then to estimate each camera pose as well as the 3-D target coordinates. This is done using iterated bundle adjustment (see [7] for instance). The intrinsic parameters of the camera can be calibrated beforehand. Although not essential for visualisation purpose, the scale factor of the scene can also be obtained from the target pattern design.

The 3-D model of the positioning targets is computed from a subset of the images which includes typically 100 to 150 images. Once a model of the 3-D coordinates of the target features has been built, any camera pose from the whole set of images can be quickly computed by using the *direct linear transform* [7]. This well-known method computes an estimate of the 3×4 projection matrix \mathbf{M} . From this, the pose \mathbf{T} is extracted by premultiplying by the inverse of known intrinsics matrix \mathbf{K} . Note that this estimate does not respect the constraints associated with a rotation matrix. It is instead used as initialization for an iterative algorithm minimizing the reprojection error.

3.2. Image segmentation

This system requires two steps of extraction from images. First, target silhouettes are extracted for pose computation and then, the object silhouette is extracted for modeling purposes. The extraction of a silhouette S requires a low level segmentation approach. Considering the observation of an RGB vector at a given pixel, x' , the binary classification process consists in choosing the label $Y \in \{0 = \text{background}, 1 = \text{foreground}\}$ maximizing $P(Y|x')$. In this case, color keying is prioritized over background subtraction techniques since camera positioning can hardly be reproduced.

Due to lighting conditions, cast shadows and paint color sprayed non uniformly, the acquisition enclosure shows many shades of green. To model this complex samples distribution in the RGB space, a non parametric approach, such as the k -nearest neighbors algorithm, is well suited. This approach requires a training set where samples of both categories are correctly labeled. The background training set is built by taking pictures of the empty enclosure from different point of views before beginning the acquisition. Since the object's color properties are *a priori* unknown, its training set was generated following a uniform distribution within the RGB cube. To keep the ratio of background/foreground pixels observed in the images, the number of foreground samples was set to an approximate value of 20% of the number of background samples.

The nearest neighbor approach subsumes that the label associated with an unknown sample (x') is very likely to be the same label as the closest training sample (x) in the RGB space, $P(Y|x') \approx P(Y|x)$. To reduce the training sample influence over the decision boundaries in the RGB space, that hypothesis leads to a k -nearest neighbors formulation where, for a given number of neighbors k ,

$$P(Y = 1|x') = \frac{1}{k} \sum_{x \in N_{x'}} Y(x). \quad (9)$$

In the equation, $N_{x'}$ is the neighborhood of x' . If $P(Y = 1|x') > 0.5$, the pixel is labeled as foreground. The choice of k was set empirically to give the best overall results over a small number of pictures. Fig. 6 b) shows an extracted object's silhouette.

Based on the same approach, matrix code targets are extracted for camera positioning. In this case, the two training sets are respectively given by RGB samples from the enclosure and by RGB samples from the targets themselves. Extracted targets are shown in Fig. 6 c). It is worth noting that small artifacts that appear outside the object's silhouette will be automatically eliminated at the visual hull reconstruction stage.

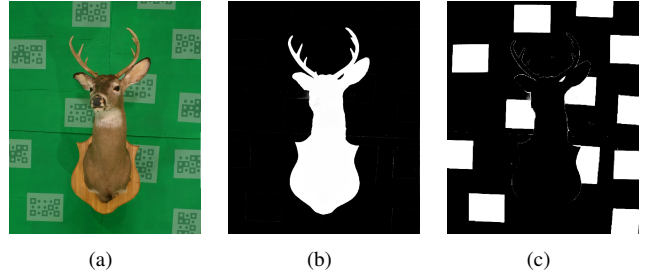


Figure 6. a) Captured image, b) Extracted object's silhouette, c) Extracted calibration targets, in all cases without post-processing.

4. Passive Appearance Modeling

The present system relies on a hybrid approach between geometry-based and image-based modeling: it uses an approximate geometry of the object and compensates for its imprecision with view interpolation. View interpolation is further exploited to enable the object to be modeled without assuming its reflectance properties. The approximate geometry used here is based on the visual hull [11]. It will be used to parameterize a *surface light field* [16] that implements the required view interpolation. These stages are described next.

Let S_i be the i^{th} image silhouette and M_i the corresponding camera projection matrix. The generalized cone C_i generated by the silhouette S_i is defined as the set of points that verifies $C_i = \{\mathbf{X} \in \mathbb{R}^3 | M_i \mathbf{X} \in S_i\}$. The visual hull \mathcal{V} can therefore be computed by the following cone intersections :

$$\mathcal{V} = \bigcap_{i=1, \dots, n} C_i = \{\mathbf{X} \in \mathbb{R}^3 | M_i \mathbf{X} \in S_i, \forall i\}. \quad (10)$$

A partial result of this operation on two images is illustrated in Fig. 7. Cone intersections are computed using implicit functions since the intersection $\phi_3(\mathbf{X})$ of two sets expressed by two implicit functions $\phi_1(\mathbf{X})$ and $\phi_2(\mathbf{X})$ is $\phi_3(\mathbf{X}) = \max(\phi_1(\mathbf{X}), \phi_2(\mathbf{X}))$, which is easily computed.

Every generalized cone C_i is thus represented by an implicit function $\phi_i(\mathbf{X})$ where $\mathbf{X} \in \Gamma$ and $\Gamma \subset \mathbb{R}^3$ is a region of space known to contain the object. For computational purposes, Γ is discretized on a regular three-dimensional grid of points \mathbf{X}_k .

An efficient way to compute $\phi_i(\mathbf{X}_k)$ is to set the value at a gridpoint \mathbf{X}_k by projecting \mathbf{X}_k in an image, transformed into a 2D signed distance field, $s_i(\mathbf{x})$, of the boundary of the silhouette S_i , thus $\phi_i(\mathbf{X}_k) = s_i(M_i \mathbf{X}_k)$. This is similar to [10] and illustrated in Fig. 7. The implicit functions $\phi_i(\mathbf{X})$ are therefore built from the backprojection of 2D

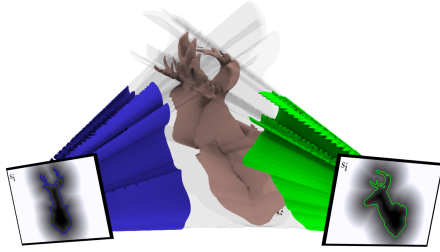


Figure 7. The visual hull is obtained by intersecting all generalized cones corresponding to the object’s silhouette in each image. Two generalized cones are shown with their corresponding distance fields. Silhouette contours are marked in blue and green.

signed distance fields. An approximation of the intersection of every generalized cone C_i is simply $\max_i(\phi_i(\mathbf{X}))$ and thus the implicit function representing the visual hull \mathcal{V} is $\phi(\mathbf{X}_k) = \max_i(\phi_i(\mathbf{X}_k))$. A mesh K_0 of the boundary of the visual hull is needed for the *surface light field* and it can be generated by finding the zero level set of $\phi(\mathbf{X}_k)$ using marching cubes.

A *surface light field* parameterized on the visual hull allows rendering from any desired view of the object. Simplifying, surface light fields are an extension of view-dependent texture mapping. The surface used in our case is the mesh K_0 of the visual hull. Therefore, the *surface light field* is a function $L : K_0 \times \mathcal{S}^2 \rightarrow RGB$ where \mathcal{S}^2 is the sphere of unit vectors in \mathbb{R}^3 and the radiance is described by points in \mathbb{R}^3 corresponding to *RGB* triples. The Light Field Mapping software [6] was used to store and render the *surface light fields* of our models. The quality of the rendered light field can be improved by refining the visual hull using multi-view techniques [15] to reduce aliasing artefacts at rendering.

5. Experimental validation

This section validates the operation of the cable-driven mechanism. To begin with, stability and repeatability experiments are shown. Then, models that were built from four different objects are presented and their quality is discussed.

5.1. End-effector Stability

Since both the 3D modeling and the camera calibration processes use the same dataset, both the object and the positioning targets must be on focus in images. This requires a large depth of field obtained by using a small lens aperture combined with a long exposure. This means that the robot, at each point of the trajectory, must be stable to avoid

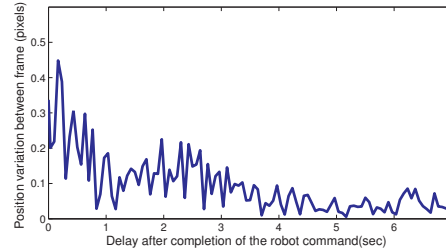


Figure 8. Target position variation (in pixels) after completion of the robot command.

blurred images.

An experiment has been conducted to assess the stability of the robot. The SLR camera was replaced by a Sony 3 CCD progressive scan video camera on the end-effector. On the ground, a 15 cm diameter white sphere has been placed over a contrasting background. The experiment consisted in filming the sphere at a distance of 1.5 m while doing several stop and go trajectories with the robot. The perceived movement of the robot has been obtained by computing the displacement of the sphere target between each frame of the video sequence. Fig. 8 shows the decay of the movement with time after command termination. A precise positioning algorithm has been used for the sphere detection and estimation [13]. This result is typical along the trajectory and suggests that after a 7 second delay, the robot is stable enough to capture sharp images of the scene. Typically, between each camera pose, the displacement requires 5 seconds and then the system pauses for 7 seconds for stabilization before capturing an image. Exposure times up to 2 seconds are applied. Thus, approximately 14 sec. is required for each image.

5.2. Repeatability

The end-effector gear is blocked for the repeatability test. The robot stabilizes at four different poses 25 times. The four poses are at the corner of a 3m square. Images are captured and the poses are precisely determined using the photogrammetric method. Mean positions and orientations are calculated for each pose. The distance and orientation to the means are summarized in Fig. 9. From this figure, one can see that the repeatability is below the 6cm in position and below 7° in orientation. This is sufficient for the application, especially considering that the poses are precisely measured using the camera.

5.3. Experimental setup

The robot is installed in a room of approximately 10 m wide, 12 m long and 6 m high. The object is surrounded by an enclosure made of eight rigid panels painted in green

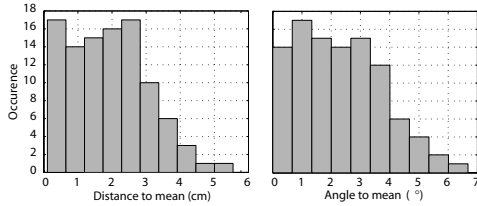


Figure 9. Repeatability: Variation in position (left) and orientation (right).

Table 1. Object dimensions and acquisition parameters.

Name	Object size (m)	# of photos	Radii
Snowmobile	2.8 x 1.0 x 1.3	600	2.7m
Deer head	0.6 x 0.8 x 1.0	105	1.5m
Bust	0.7 x 0.6 x 0.5	212	1.5m
Small statue	0.1 x 0.1 x 0.2	242	0.8m

as shown in Fig. 1. The overall enclosure radius is approximately 3 m. The matrix code positioning targets are printed in dark green on letter format light green paperboard and they are distributed randomly over the entire enclosure. An ink jet printer is used to avoid the specularly of laser printing that could cause a problem in the color keying. Within the enclosure, each object is fixed on a pedestal that facilitates view collection for larger ϕ angles.

5.4. Objects modeling results

Four different objects acquired with the proposed system are presented. They were selected to show the variety of objects that can be captured and especially the scalability of the system. Table 1 summarizes the experimental conditions of each acquisition. The number of images acquired for each object varies from 105 to 600 and the radius of the trajectories varies from 0.8 m to 2.7 m. Fig. 10 shows the result of the modeling process. For each object, the first image represents the mesh computed as a geometry approximation and the two others are rendered views of the final 3D model. To build the visual hull of each object, the bounding box was divided into 150^3 voxels. From this volumetric representation, a mesh was extracted. Every step of the modeling process are shown in the video extension 2.

The first object presented is a snowmobile. With its length of nearly 3 m, this is the largest object that can be acquired with the system. A 10 mm focal length lens providing a field of view of ($\Theta_v = 73.6^\circ$, $\Theta_h = 96.6^\circ$) was used to allow the entire snowmobile to be seen as well as a sufficient number of targets. This object is particularly interesting with respect to its size as well as its shininess.

The deer head and the bust of the Cardinal of Richelieu are two medium sized objects. As shown in Fig. 10 d), the wood of the deer exhibits a small structure which demonstrates the accuracy of positioning. Another interesting characteristic of this model is the captured reflectance of the fur. The bust of the Cardinal of Richelieu is also a challenging object because it exhibits complex reflectance properties. One can note the displacement of the specularities on the forehead of the bust in Fig. 10 h) and i). For these smaller objects, an 18 mm focal lens was used. This leads to a field of view of ($\Theta_v = 46.8^\circ$, $\Theta_h = 67.07^\circ$).

The fourth object is a dark small statue of less than 20 cm. For such a small object, the density of targets was increased around the object to allow better camera positioning. The resulting model can be seen in Fig. 10 k) and l).

One problem that arose in the experiments is the color bleeding effect of the background on the object. This caused a segmentation artifact that could not be corrected automatically on the snowmobile model. Actually, this can be observed at the tip of the left ski of the snowmobile. Color bleeding also affects the appearance of the models. For this purpose, image pixels that are in the same region of the color space as the green enclosure have been desaturated.

6. Conclusion

We have presented the full proof of concept of a cable robot system for automatically acquiring hundreds of calibrated images around objects of various sizes on a hemispherical trajectory. These calibrated images allow a surface light field to be built on the visual hull of the object and the resulting model is used for free viewpoint rendering. The extraction of silhouettes of the object can be further improved to avoid holes or missing parts on the visual hull. Moreover, the overall quality can be increased by using two cameras with lenses of different focal length. One camera can be pointed at the targets so as to position the end-effector while the other can be used to collect better pictures of the object. It is finally worth noting that this type of system is relatively cheap and could be scaled up for modeling objects of more than 10 m. For large systems, the controller will be adapted to take into consideration the elasticity of the cables.

Acknowledgements

The authors express their gratitude to the NSERC/CRSNG and NATEQ/FQRNT for supporting this research and to Annette Schwerdtfeger for proof reading.

References

- [1] Cablecam. <http://www.cablecam.com/>.

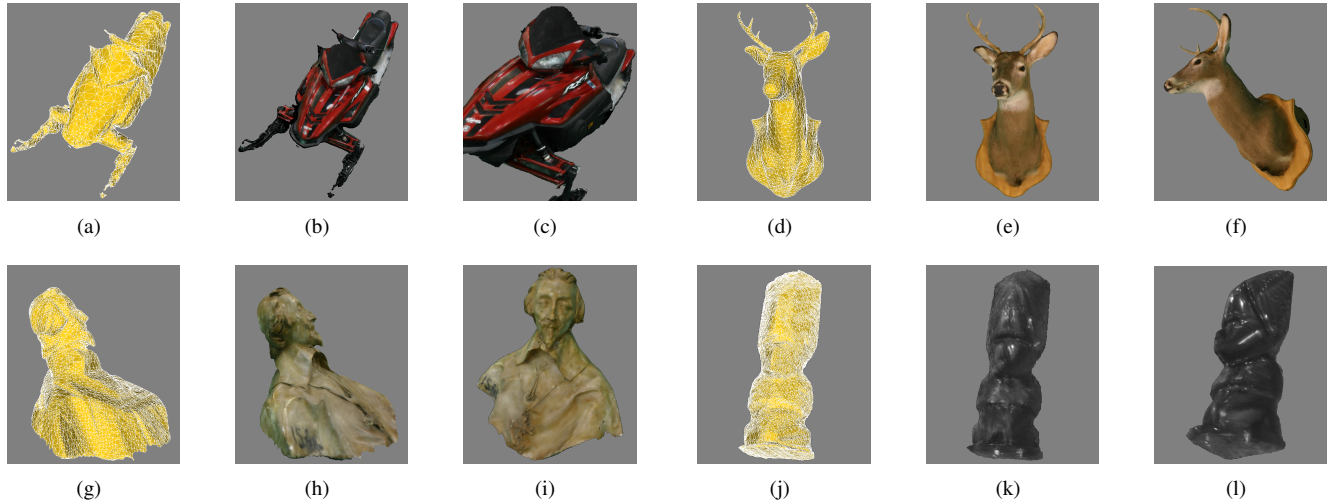


Figure 10. Object modeling result for the acquisition of a snowmobile, a deer head, a bust and a small statue. For each object, the mesh and two rendered views from the final 3D model are shown.

- [2] Skycam. <http://www.skycam.tv/>.
- [3] The Stanford Spherical Gantry. <http://graphics.stanford.edu/projects/gantry/>.
- [4] M. Arsenault and R. Boudreau. Synthesis of Planar Parallel Mechanisms While Considering Workspace, Dexterity, Stiffness and Singularity Avoidance. *Journal of Mechanical Design*, 128:69–78, Jan. 2006.
- [5] S. Boucard and C. M. Gosselin. Kinematic Sensitivity of a Very Large Cable-driven Parallel Mechanism. *Proc. of the ASME International Design Engineering Technical Conferences, Mechanics and Robotics Conference*, Sept. 2006.
- [6] W. Chen, J. Bouguet, M. Chu, and R. Grzeszczuk. Light Field Mapping: Efficient Representation and Hardware Rendering of Surface Light Fields. *ACM Transactions on Graphics*, 21(3):447–456, 2002.
- [7] R. Hartley and A. Zisserman. *Multiple View Geometry*. Cambridge University Press, 2000.
- [8] J. Heikkilä. Geometric Camera Calibration Using Circular Control Points. *IEEE Transactions on PAMI*, 22(10):1066–1077, Oct. 2000.
- [9] T. Kanade, P. Rander, and P. J. Narayanan. Virtualized Reality: Constructing Virtual Worlds from Real Scenes. *IEEE MultiMedia*, 4(1):34–47, – 1997.
- [10] H. Kuck, W. Heidrich, and C. Vogelgsang. Shape from Contours and Multiple Stereo—A Hierarchical, Mesh-Based Approach. In *First Canadian Conference on Computer and Robot Vision*, pages 76–83. IEEE, 17–19 may 2004.
- [11] A. Laurentini. The Visual Hull Concept for Silhouette-Based Image Understanding. *IEEE Transactions on PAMI*, 16(2):150–162, 1994.
- [12] M. Levoy and P. Hanrahan. Light Field Rendering. *Computer Graphics*, 30(Annual Conference Series):31–42, 1996.
- [13] J. Ouellet and P. Hébert. A Simple Operator for Very Precise Estimation of Ellipses. In *Fourth Canadian Conference on Computer and Robot Vision*, Montreal, Quebec, CANADA, 28 - 30 may 2007. IEEE.
- [14] A. T. Riechel and I. Ebert-Uphoff. Force-Feasible Workspace Analysis for Underconstrained, Point-Mass Cable Robots. *Proc. of the 2004 IEEE International Conference on Robotics and Automation*, 5:4956–4962, Apr. 2004.
- [15] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. *CVPR 2006*, pages 519–528, June 2006.
- [16] N. Wood Daniel, I. Azuma Daniel, et al. Surface Light Fields for 3D Photography. *Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH'00*, July 2000.