Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Optimization for 3D Model-based Multi-Camera Deployment ⋆

Xuebo Zhang [*,**] **Jose Luis Alarcon-Herrera** [*]
Xiang Chen(✉) [*]

[*] *Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON, N9B 3P4 Canada (e-mail: {alarconj, xchen}@uwindsor.ca).*
[**] *Institute of Robotics and Automatic Information System(IRAIS) and Tianjin Key Laboratory of Intelligent Robotics (tjKLIR), Nankai University, Tianjin 300071, China. (e-mail: zhangxuebo@nankai.edu.cn)*

Abstract: Based on convex optimization techniques, we propose a new multi-camera deployment method for optimal visual coverage of a three-dimensional (3D) object surface. Different from existing methods, the optimal placement of a single camera is formulated as two convex optimization problems, given a set of covered triangle faces. Moreover, this idea is incorporated into a recursive framework to expand the covered area for each camera, wherein initially covered triangle faces are elegantly chosen using an importance criterion for the first recursion. By placing cameras one by one using the same method, the object surface is gradually covered by iteratively removing the covered partition of the previously deployed camera. Due to the usage of convex optimization, each camera is guaranteed to be placed at an optimal pose for a group of triangle faces other than a single one. This merit, together with the importance criterion-based selection of initially covered triangle faces, reduces the number of required cameras while satisfying various constraints including the resolution, field of view, focus and occlusion. Simulation results on two real 3D computer-aided design (CAD) models are presented to verify the effectiveness of the proposed approach.

Keywords: Multi-camera deployment, convex optimization

1. INTRODUCTION

As a typical kind of non-contact sensor, visual cameras provide rich information for interested objects or scenes with a relatively low cost, thus they are employed in various applications, such as visual surveillance, vision-based control, and so on. Since a single camera has a limited sensing range and ability, multi-camera networks are highly demanded in many applications such as precise inspection of industrial products. In this case, one fundamental problem is the optimal multi-camera deployment while satisfying various task requirements.

For multi-camera deployment, a classical survey can be found in Tarabanis et al. (1995) for earlier research. After that, many recent results have also been reported in the literature, see Sheng et al. (2003); Chen et al. (2004); Alarcon-Herrera et al. (2013). Though some feasible or near-optimal approaches are provided, the research on this topic is far from mature.

One key factor is the appropriate choice of the coverage model, which takes into account various coverage criterions and task requirements such as dimension of the inspected objects (2D or 3D), camera field of view (FOV), focus, resolution, occlusion, and so on. Interested readers can refer to the recent survey in Mavrinac et al. (2013) to know more about the state of the art on mainstream coverage models. By using discretization of the inspecting space, the optimal deployment problem is successfully reduced to a "set cover problem"in Erdem et al. (2006). Yet, the scene considered in this work is restricted to the planar case. Jiang et al. (2010) consider the FOV constraint and possible occlusion in their weighted coverage model. Nevertheless, the constraints for resolution and focus are not included. A. Mavrinac proposes a new coverage model to take into account almost all realistic constraints in Mavrinac (2012). This model is validated and successfully applied in many scenarios, such as deployment of range cameras in Alarcon-Herrera et al. (2013) and real-time view selection for large-scale surveillance systems in Mavrinac et al. (2014).

In addition to the coverage model, optimization strategies and their related solution spaces for searching, are also important for optimal multi-camera deployment. Angella et al. (2007) propose an optimal camera placement approach which achieves excellent performance by considering many realistic coverage constraints. However, they use

⋆ This work is supported in part by Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant, in part by National Natural Science Foundation of China under Grant 61203333, in part by Specialized Research Fund for the Doctoral Program of Higher Education of China under Grant 20120031120040, in part by Tianjin Natural Science Foundation under Grant 13JCQNJC03200.

discrete position sampling to generate a potential list of candidate camera viewpoints as the solution space, and then the optimal solution is found within this discrete solution space. Sheng et al. (2003, 2005) propose to find a solution by using generate-and-test approach, wherein the search space is constituted by discrete points generated according to visual constraints. Chen et al. (2004) use genetic algorithm (GA) to find optimal poses for multiple cameras. The work in Morsly et al. (2012) explore the utilization of particle swarm algorithm (PSO) to search the optimal solution in a discrete 3D space. From above analysis, it is shown that most existing methods are optimal only in a discrete space, and most optimization methods are meta-heuristic which have no theoretic guarantee on optimality of the solution. Therefore, optimization techniques in a continuous solution space still need to be further studied.

Considering a known 3D object model, we make an attempt to refine existing coverage criterions in this paper, and we further find that convex optimization techniques can be successfully employed to obtain the optimal solution for orientation and position of each camera in the continuous solution space. The general idea is to place cameras one by one such that the covered area for every newly deployed camera is increased as large as possible, given the practical visual coverage constraints specified by the user. Specifically, on the basis of the previously validated coverage model in Mavrinac (2012), the coverage criterions for the view angle and resolution are further combined together to form a new more realistic resolution criterion, which is of explicit physical meaning and more friendly to end users. By using the polygonal mesh to express the 3D object surface as triangle faces, it is shown that optimal deployment of a single camera can be formulated as two convex optimization problems, given a set of triangle faces that can be covered. Hence, by using a recursive approach, the pose of this camera is updated using convex optimization and then the number of covered triangle faces is increased for each recursion. The recursive process stops when the increasing number of the covered triangle faces is less than a threshold for another recursion. Note that it is significant to properly choose the starting triangle faces to be covered for the first recursion, and thus we propose a coverage importance criterion to elegantly choose the initially covered triangles as highly important ones along the object boundary. By placing new cameras one by one using the same method, the whole object surface is gradually covered by iteratively removing the covered partition of the previously deployed camera. The task of camera deployment continues until the increasing of the covered area is too small to justify the cost of adding a new camera. Two simulation examples on automotive body parts are provided to demonstrate the effectiveness of the proposed approach.

2. PROBLEM STATEMENT

The problem we consider is the multi-camera deployment for visual coverage of a 3D object with a known CAD model, which is usually encountered in industrial inspection tasks. Hence, inputs of this problem include a known CAD model and a calibrated camera model with known internal parameters(note that the proposed approach can be extended to cope with a list of different cameras easily, and we omit this part in this paper due to the limited space). The outputs include the number of required cameras and poses (position and orientation) for these cameras.

2.1 Object model and camera model

The object 3D CAD model is expressed using a polygon mesh, which is a collection of vertices, edges and triangle faces to depict the shape of the object surface. There exist many kinds of representations for polygon meshes, and we choose the render dynamic mesh, which can be easily utilized to dynamically manipulate the polygon mesh such as traversing the CAD topology in linear time, boundary finding and subdivision, see Tobler (2006).

A pin-hole camera model is utilized, where $f \in \mathbb{R}$ is the focal length, $s_u,\ s_v \in \mathbb{R}$ denote the horizontal and vertical pixel dimensions, and $\boldsymbol{o} = (u_0,\ v_0)$ represents the principle point (as called "image center") in pixels. In addition, the image resolution is $w \times h$, with $w \in \mathbb{Z}^+$ and $h \in \mathbb{Z}^+$ being the width and height of the image, respectively. Other lens parameters include the effective aperture diameter $A \in \mathbb{R}$ and the subject distance $z_S \in \mathbb{R}$. Please refer to Mavrinac (2012) for detailed meaning of these parameters.

2.2 Notation for the camera pose representation

To facilitate the setup of cameras, the output camera poses are usually expressed in the world coordinate frame. However, visual criterions are usually evaluated in the camera frame. Therefore, we briefly introduce the coordinate relationships in this subsection.

Let $\mathcal{F}_w$ and $\mathcal{F}_c$ denote the world coordinate system and the camera coordinate system, respectively. The vertices of the 3D object are expressed using $\mathcal{P}_i$ $(i = 1,\ 2,\ \cdots, n)$ with $n \in \mathbb{R}$ being the total number of those vertices of the triangles. The world coordinate of $\mathcal{P}_i$ is $\boldsymbol{P}_{wi} = [x_{wi}\ \ y_{wi}\ \ z_{wi}]^{\mathrm{T}}$ expressed in $\mathcal{F}_w$, with its camera coordinate being $\boldsymbol{P}_{ci} = [x_{ci}\ \ y_{ci}\ \ z_{ci}]^{\mathrm{T}}$ expressed in $\mathcal{F}_c$, then we have the following relationship

$$\boldsymbol{P}_{wi} = \boldsymbol{R}\boldsymbol{P}_{ci} + \boldsymbol{T} \tag{1}$$

where the pose of the camera is expressed by a rotation matrix $\boldsymbol{R} \in SO(3)$ and a translation vector $\boldsymbol{T} \in \mathbb{R}^3$ with respect to the frame $\mathcal{F}_w$. It follows from (1) that

$$\boldsymbol{P}_{ci} = \boldsymbol{R}_c\boldsymbol{P}_{wi} + \boldsymbol{T}_c \tag{2}$$

with $\boldsymbol{R}_c \in SO(3)$ and $\boldsymbol{T}_c = [t_{cx}\ \ t_{cy}\ \ t_{cz}] \in \mathbb{R}^3$ being the pose expressed in $\mathcal{F}_c$ as

$$\boldsymbol{R}_c = \boldsymbol{R}^{\mathrm{T}},\ \boldsymbol{T}_c = -\boldsymbol{R}^{\mathrm{T}}\boldsymbol{T} \tag{3}$$

Hence, the pose $(\boldsymbol{R},\ \boldsymbol{T})$ and $(\boldsymbol{R}_c,\ \boldsymbol{T}_c)$ can be easily converted to each other using (3).

3. VISUAL COVERAGE CRITERIONS

On the basis of the previously developed and validated coverage model in Mavrinac (2012), we combine the criterions for the resolution and view angle to give a new realistic resolution criterion in this section. Other criterions for FOV, focus and occlusion are the same as those in Mavrinac (2012), which are briefly introduced for self-completeness of this paper.

3.1 A new resolution criterion

Existing methods usually use two separate criterions to evaluate the resolution, namely the distance z_{ci} of the measured point along the z axis of $\mathcal{F}_c$, and the view angle of the camera with respect to the measured triangle face. However, to evaluate the realistic resolution, these two criterions are actually strongly coupled with each other. As shown in Figure 1, for two line segments L_1 and L_2 with the same length, though the distances of points on L_1 are smaller, the resolution of L_1 is less than that of L_2 since the view angle for the segment L_2 is better. Hence, it is difficult to use separate criterions to accurately evaluate the realistic resolution of an object.

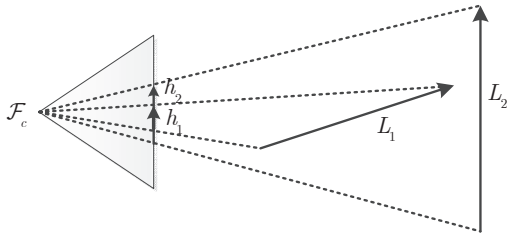


Fig. 1. resolution with different distances and view angles

Inspired by this fact, instead of using two separate criterions, we combine the distance criterion and the view angle criterion together to form a new resolution criterion:

$$r(\boldsymbol{P}_{ci}) = \frac{f\cos\theta_i}{z_{ci}\max(s_u, s_v)} \tag{4}$$

where $r(\boldsymbol{P}_{ci}) \in \mathbb{R}$ denotes the resolution of point $\boldsymbol{P}_{ci}$, $\theta_i \in [0,\ \pi)$ is the angle between the camera axis and the normal of the measured triangle. z_{ci} is the camera coordinate of $\boldsymbol{P}_{ci}$ along the z axis of $\mathcal{F}_c$. Note that this resolution criterion is a good representative to reflect the worst direction of resolution, and the proof is omit in this paper due to the space limit.

With this new criterion, we can specify the task constraints. For general applications, a task parameter r_a (with unit being pixel/mm), which denotes an acceptable value of the resolution, is given such that the inspected point should satisfy

$$r(\boldsymbol{P}_{ci}) \geq r_a. \tag{5}$$

3.2 Camera field of view

Generally, a monocular camera has a limited camera field of view(FOV), which is expressed using four limit angles $\alpha_l,\ \alpha_r,\ \alpha_t,\ \alpha_b$ along the left, right, top and bottom direction. These angles are computed by using the calibrated camera parameters $\boldsymbol{o},\ f,\ s_u,\ s_v$ as described in Mavrinac (2012). It is assumed that the type of cameras has been well selected and these parameters are constant.

In terms of the camera coordinate $\boldsymbol{P}_{ci} = [x_{ci}\ \ y_{ci}\ \ z_{ci}]^{\mathrm{T}}$, a covered vertex should fall in the camera FOV, which is analytically expressed as

$$-\tan\alpha_l \leq \frac{x_{ci}}{z_{ci}} \leq \tan\alpha_r, \tag{6}$$

$$-\tan\alpha_t \leq \frac{y_{ci}}{z_{ci}} \leq \tan\alpha_b. \tag{7}$$

3.3 Focus

To inspect the object clearly, the camera should be placed such that the inspected object lies in a range around its focus plane. Hence, a constraint should be enforced as

$$z_n(c_a,\ z_S) \leq z_{ci} \leq z_f(c_a,\ z_S) \tag{8}$$

where $z_n,\ z_f \in \mathbb{R}$ denotes two distance specifying the focus range with $z_n < z_f$, and they can be previously computed using the maximal allowable blur circle c_a and the subject distance z_S in Mavrinac (2012).

3.4 Occlusion handling

The static occlusion is also considered. To judge if a vertex is occluded, we only need to judge if the line segment connecting this vertex and the optical center is intersected with other triangles. Other than directly considering the occlusion when placing the camera, we usually use other criterions to conduct the camera deployment, and then the occlusion situation is checked posteriorly to further update the coverage results.

With these aforementioned coverage requirements for resolution (5), FOV (6), focus (8) and occlusion, we provide some definitions to facilitate the subsequent analysis.

Definition 1: Covered vertex. A vertex is said to be "covered" if all the coverage criterions are satisfied including resolution (5), FOV (6), focus (8) and occlusion.

Definition 2: Covered triangle face. A triangle face is said to be "covered" if its three vertices are covered vertices.

4. CONVEX OPTIMIZATION-BASED POSE DETERMINATION

Given a list of triangle faces, the objective of this section is to determine the optimal pose of a single camera. It is shown that this pose determination problem is formulated as two successive convex optimization problems, which are solved to obtain the orientation and position, respectively.

4.1 Rotation determination algorithm

For a number of k_f known triangle faces with different unit normal vectors $\boldsymbol{n}_i = [n_{xi}\ \ n_{yi}\ \ n_{zi}]^{\mathrm{T}}$ $(i = 1,\ 2,\ \cdots,\ k_f)$, the goal is to find the axis of the minimum cone that contains all these normals, then the camera orientation is determined to be along this axis. In this way, we can not only find the "average" normal lying in the center of these k_f normals, but also minimize the angle between this average and the worst one.

Let $\boldsymbol{d} = [d_x\ \ d_y\ \ d_z]^{\mathrm{T}}$ denote the direction of the cone axis, then the orientation determination problem is transformed into a convex optimization problem as follows:

$$\begin{aligned}
&\text{Find} \quad d_x,\ \ d_y,\ \ d_z,\ \ S && (9)\\
&\text{subject to} \\
&\quad S \leq 1 && (10)\\
&\quad \epsilon||\boldsymbol{d}||_2 \leq S && (11)\\
&\quad \text{for} \quad i = 1,\ 2,\ \cdots,\ k_f \\
&\qquad n_{xi}d_x + n_{yi}d_y + n_{zi}d_z \geq S && (12)
\end{aligned}$$

where $S \in \mathbb{R}$ denotes a lower bound for inner product of $\boldsymbol{n}_i$ and $\boldsymbol{d}$, with its value being greater than 1 to constrain the

amplitude of $\boldsymbol{d}$. Note that $\epsilon \in (0, 1]$ is given as the cosine of the angle between an edge and the axis of a second-order cone containing all the normals, where $\epsilon > 0$ is set as a necessary considtion for visibility. By testing the feasibility of the convex optimization problem for a given ϵ , we use the bisection method to gradually find the maximum value of ϵ which preserves feasibility. Hence, the axis direction $\boldsymbol{d}$ of the minimum cone is obtained, and the camera rotation matrix $\boldsymbol{R}_c$ can be computed so that its orientation is $-\boldsymbol{d}$.

4.2 Translation determination algorithm

Once the camera orientation is obtained, the angle θ_i between the normal of the i^{th} triangle face and the camera axis can be determined as follows:

$$\theta_i = \text{acos}(-\boldsymbol{d} \cdot \boldsymbol{n}_i)/||\boldsymbol{d}||_2. \tag{13}$$

To facilitate the following analysis, we introduce an auxiliary vector $\boldsymbol{P}_{mi} = [x_{mi} \;\; y_{mi} \;\; z_{mi}]^{\text{T}}$ as

$$\boldsymbol{P}_{mi} = \boldsymbol{R}_c \boldsymbol{P}_{wi}. \tag{14}$$

Since $\boldsymbol{R}_c$ has been obtained by using the rotation determination algorithm, the vector $\boldsymbol{P}_{mi}$ can be directly computed from the known 3D coordinates of the i^{th} point P_{wi} in frame $\mathcal{F}_w$. By further substituting (14) into (3), we obtain

$$x_{ci} = x_{mi} + t_{cx}, \tag{15}$$
$$y_{ci} = y_{mi} + t_{cy}, \tag{16}$$
$$z_{ci} = z_{mi} + t_{cz} \tag{17}$$

wherein x_{mi}, y_{mi}, z_{mi} are all known from (14). Hence, the visual coverage constraints can be expressed by using x_{mi}, y_{mi}, z_{mi} and the translation $\boldsymbol{T}_c = [t_{cx} \; t_{cy} \; t_{cz}]$, by substituting (15)−(17) into the (5), (6) and (8).

With the objective of minimizing the worst cases for resolution, focus and FOV, the determination of the translation vector $\boldsymbol{T}_c$ then becomes a constrained convex optimization problem as follows:

Minimize $w_{xy}\left(\sigma_x^2 + \sigma_y^2\right) + w_{foc}\sigma_{zf}^2 + w_z\sigma_{zfov}^2 + w_R\sigma_r$

subject to

for $i = 1, \; 2, \; \cdots, \; 3k_f$

$$-\sigma_x \le x_{mi} + t_{cx} \le \sigma_x \tag{18}$$
$$-\sigma_y \le y_{mi} + t_{cy} \le \sigma_y \tag{19}$$
$$-x_{mi} - t_{cx} - \tan\alpha_l(z_{mi} + t_{cz}) \le 0 \tag{20}$$
$$x_{mi} + t_{cx} - \tan\alpha_r(z_{mi} + t_{cz}) \le 0 \tag{21}$$
$$-y_{mi} - t_{yx} - \tan\alpha_t(z_{mi} + t_{cz}) \le 0 \tag{22}$$
$$y_{mi} + t_{yx} - \tan\alpha_b(z_{mi} + t_{cz}) \le 0 \tag{23}$$
$$z_n \le z_{mi} + t_{cz} \le z_f \tag{24}$$
$$-\sigma_{zf} \le z_{mi} + t_{cz} - \frac{z_n + z_f}{2} \le \sigma_{zf} \tag{25}$$
$$z_f - z_{mi} - t_{cz} \le \sigma_{zfov} \tag{26}$$
$$z_{mi} + t_{cz} - \frac{f\cos(\theta_i)}{r_a \min(s_u, s_v)} \le 0 \tag{27}$$
$$z_{mi} + t_{cz} - \frac{f\sigma_r\cos(\theta_i)}{\min(s_u, s_v)} \le 0 \tag{28}$$

where w_{xy}, w_{foc}, w_z, $w_R \in \mathbb{R}$ are positive weights, and σ_x, σ_y, $\sigma_{zfov} \in \mathbb{R}$ denote the worst-case performance index relating to the FOV constraint, while $\sigma_{zf} \in \mathbb{R}$ and $\sigma_r \in \mathbb{R}$ represent those relating to the focus and resolution constraints. This convex optimization problem is essentially a linear constrained quadratic program, which can be easily solved to obtain the optimal translation $\boldsymbol{T}_c$.

Algorithm 1 Single Camera Deployment Algorithm

Input: O_{bj}, U_c, α_l, α_r, α_t, α_b, f, s_u, s_v, z_n, z_f, r_a, N_{bd}, N_Δ

Output: $\boldsymbol{R}_c$, $\boldsymbol{T}_c$, C

Initialization. $C = C_{tmp} = \{\}$, $B_d = \{\}$, $\boldsymbol{R}_c = \boldsymbol{R}_{tmp} = \boldsymbol{0}$, $\boldsymbol{T}_c = \boldsymbol{T}_{tmp} = \boldsymbol{0}$. Subscript *_tmp* is for temporal use.
Find the boundary B_d for U_c.
set $flag = 0$
for E_i in B_d **do**
  find the related face f_0 for E_i, set $C = \{f_0\}$.
  Using C as the faces to be covered, run the **convex optimization-based pose determination algorithm** to compute a pose $\boldsymbol{R}_{tmp}$ and $\boldsymbol{T}_{tmp}$.
  Using $\boldsymbol{R}_{tmp}$ and $\boldsymbol{T}_{tmp}$, find covered triangles as C_{tmp} by **FOV, focus, occlusion and resolution checking.**
  if $C \in C_{tmp}$ and $len(C_{tmp}) \ge N_{bd}$ **then**
    $\boldsymbol{R}_c \leftarrow \boldsymbol{R}_{tmp}$, $\boldsymbol{T}_c \leftarrow \boldsymbol{T}_{tmp}$, $C \leftarrow C_{tmp}$, **break**
  else if $C \in C_{tmp}$ and $len(C_{tmp}) > flag$ **then**
    $\boldsymbol{R}_c \leftarrow \boldsymbol{R}_{tmp}$, $\boldsymbol{T}_c \leftarrow \boldsymbol{T}_{tmp}$, $C \leftarrow C_{tmp}$, $flag = len(C_{tmp})$
  end if
end for
while 1 **do**
  Using C as the faces to be covered, run the **convex optimization-based pose determination algorithm** to compute a pose $\boldsymbol{R}_{tmp}$ and $\boldsymbol{T}_{tmp}$.
  Using $\boldsymbol{R}_{tmp}$ and $\boldsymbol{T}_{tmp}$, find covered triangles as C_{tmp} by **FOV, focus, occlusion and resolution checking.**
  $J_1 \leftarrow C \in C_{tmp}$
  $J_2 \leftarrow (len(C_{tmp}) - len(C) > N_\Delta)$
  if J_1 and J_2 **then**
    $C \leftarrow C_{tmp}$, $\boldsymbol{R}_c \leftarrow \boldsymbol{R}_{tmp}$, $\boldsymbol{T}_c \leftarrow \boldsymbol{T}_{tmp}$
  else if J_1 **then**
    $C \leftarrow C_{tmp}$, $\boldsymbol{R}_c \leftarrow \boldsymbol{R}_{tmp}$, $\boldsymbol{T}_c \leftarrow \boldsymbol{T}_{tmp}$
    break
  else
    break
  end if
end while
Return C, $\boldsymbol{R}_c$ and $\boldsymbol{T}_c$

5. CAMERA DEPLOYMENT

5.1 Single Camera Deployment

Let O_{bj} denote the 3D object model, with U_c being the uncovered part. A threshold N_{bd} is to evaluate the importance of a edge in the boundary of uncovered area, which denotes the number of covered triangles for using the corresponding selected edge to place the camera. If the resulting camera pose can cover a number of triangles larger than N_{bd}, then we think this edge is important enough to be used to provide the initial covered faces for the first recursion. The threshold N_Δ is set for the newly added number of "covered triangles", if this number is less than N_Δ, the recursion will be stopped.

Using the single camera deployment method as described in Algorithm 1, we can compute the pose of a newly deployed camera $\boldsymbol{R}_c$, $\boldsymbol{T}_c$, and also the corresponding list of "covered triangles" C at this pose. Note that the FOV,

Algorithm 2 Multi-Camera Deployment Algorithm

Input: O_{bj}, α_l, α_r, α_t, α_b, f, s_u, s_v, z_n, z_f and r_a, N_{uc}, N_{sc}

Output: n_c, $\boldsymbol{T}_c(i)$, $\boldsymbol{R}_c(i)$ with $i = 1,\ 2,\ \cdots,\ n_c$.

1: Initialization. $U_c = O_{bj}$, $n_d = 0$.
2: **while** $len(U_c) \geq N_{uc}$ **do**
3: $n_d \leftarrow n_d + 1$
4: run **Algorithm 1** to compute the pose $\boldsymbol{R}_c(i)$ and $\boldsymbol{T}_c(i)$ and the covered triangles C within U_c at this viewpoint.
5: **if** $len(C) \leq N_{sc}$ **then**
6: **break**
7: **end if**
8: update the uncovered model U_c by removing the covered triangles C.
9: **end while**
10: $n_c \leftarrow n_d$
11: Return n_c, $\boldsymbol{R}_c(i)$ and $\boldsymbol{T}_c(i)$, and the task is completed.

focus, occlusion and resolution checking is to judge if (5), (6) and (8) are satisfied without occlusion.

5.2 Multi-Camera Deployment

On the basis of the single camera deployment algorithm, we propose a divide-and-conquer based multi-camera deployment approach, which is described in Algorithm 2. N_{uc} and N_{sc} are two thresholds to terminate the algorithm. N_{uc} denotes the allowable number of uncovered triangles for the 3D object, and N_{sc} represents the number of triangles that justifies the deployment of an additional camera. Using Algorithm 2, we will obtain the number of deployed cameras n_c, the position $\boldsymbol{T}_c(i)$ and orientation $\boldsymbol{R}_c(i)$ of the cameras, with $i = 1,\ 2,\ \cdots,\ n_c$. Once $\boldsymbol{T}_c(i)$ and $\boldsymbol{R}_c(i)$ of the cameras are computed, we utilize the transformation in (3) to compute the corresponding position $\boldsymbol{T}(i)$ and orientation $\boldsymbol{R}(i)$ of these cameras in the world coordinate system for easy setup in practice.

6. SIMULATION RESULTS

To verify the performance of the proposed multi-camera deployment approach, we conduct two simulations using the 3D models of the "door" and the "hood" of an automotive body. The camera image size is 1280×1024, with the principle point being $\boldsymbol{o} = (640,\ 512)$. The focal length is $f = 8$ mm and the pixel dimension is $s_u = s_v = 0.0053$ mm/pixel, while the aperture and the subject distance are $A = 2$ mm and $z_S = 1168.4$ mm, respectively. The acceptable resolution is set as $r_a = 0.34$ pixel/mm, and the acceptable blur is set as $c_a = 2.5$ pixel. Using the aforementioned camera and task parameters, other parameters, such as α_l, α_r, α_t, α_b, z_n, z_f, can be easily computed. In addition, the thresholds are set as $N_{bd} = 50$, $N_\Delta = 5$, $N_{sc} = 15$, $N_{uc} = 20$.

The results are shown in Figure 2 and Figure 3, respectively, from which we can qualitatively see that the cameras are deployed in a good manner with a limited number of cameras. In these two figures, the red triangles represent the covered triangles, with stronger brightness representing larger resolution. To quantitatively analyze the performance, we demonstrate the decrease of the number of uncovered triangle faces with respect to the number

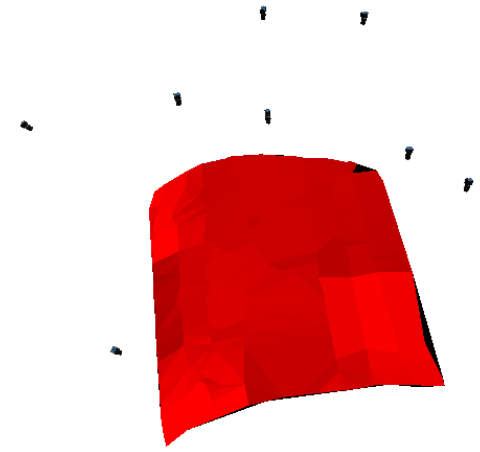


Fig. 2. Multi-camera deployment for the door model

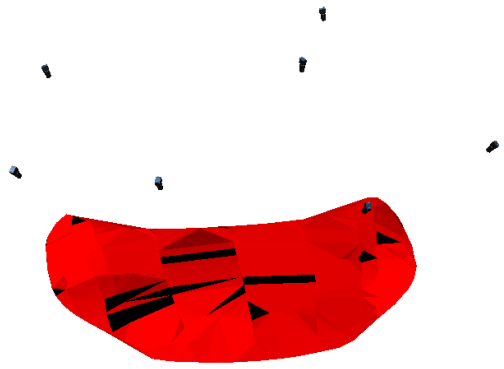
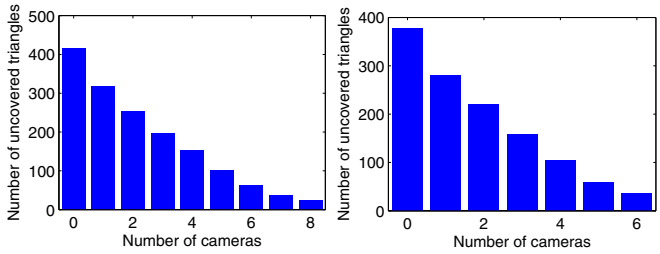


Fig. 3. Multi-camera deployment for the hood model



(a) the "door" model (b) the "hood" model

Fig. 4. Uncovered number of triangles w.r.t. number of deployed cameras

of used cameras in Figure 4. For the door model, 8 cameras are utilized to cover 393 of 417 triangles (covered ratio 94.24%), while 6 camera are deployed to cover 341 of 378 triangles (covered ratio 90.21%) of the hood model.

To reveal the detailed process of the camera placement, we show the covered area for successively placed cameras for the door model in Figure 5, from which it is seen that cameras are placed to cover the triangles from the boundary to central part of the object surfaces, in an order that is generally consistent with the "importance", namely the newly covered triangles for each camera. For inspection tasks, one important criterion is the resolution, therefore, we present the resolution distribution using (4) to compute the resolution along the worst direction as histograms in Figure 6. It is clear that the resolution for most triangles are more than 0.8 pixel/mm, which is much better than the required resolution 0.34 pixel/mm.

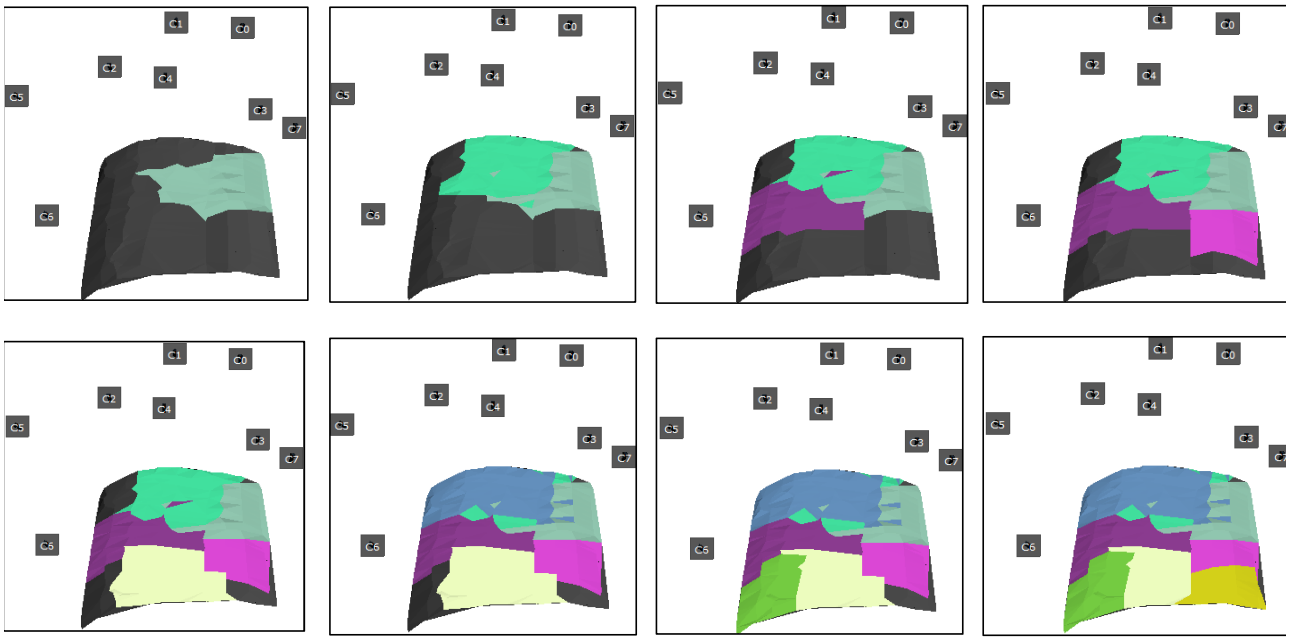
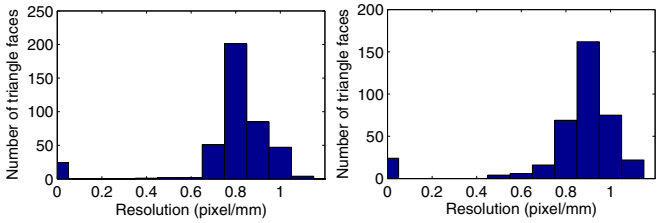


Fig. 5. The placement sequence and the corresponding covered area for each camera



(a) the "door" model (b) the "hood" model

Fig. 6. Resolution distribution

7. CONCLUSION

A 3D model-based optimal multi-camera deployment approach is presented in this paper. A new coverage criterion for resolution is proposed by combining the distance and the view angle criterions to be more realistic. Multiple cameras are placed one by one, with each single camera placement problem being solved by a recursive convex optimization approach. By carefully selecting initial triangles for the first recursion, multiple cameras are placed in a reasonable importance-based order. Simulation results on two automotive body parts are presented to demonstrate the effectiveness of the proposed approach.

REFERENCES

A. Mavrinac. Modeling and Optimizing the Coverage of Multi-Camera Systems. Ph.D. Thesis, University of Windsor, 2012.

A. Mavrinac, X. Chen. Modeling coverage in camera networks: a survey. *International Journal of Computer Vision*, 101(1):205–226, 2013.

A. Mavrinac, X. Chen, and Y. Tan. Coverage quality and smoothness criteria for real-time view selection in a multi-camera network. *ACM Trans. Sensor Networks*, 10(2): 33, 2014.

J. L. Alarcon-Herrera, A. Mavrinac, and X. Chen. Sensor planning for range cameras via a coverage strength model. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2011, pp. 838-843.

W. Sheng, N. Xi, M. Song and Y. Chen. CAD-Guided sensor planning for dimensional inspection in automotive manufacturing. *IEEE/ASME Transactions on Mechatronics*, 8(3):372–380, 2003.

W. Sheng, N. Xi, M. Song and Y. Chen. Robot path planning for dimensional measurement in automotive manufacturing. *ASME Journal of manufacturing science and engineering*, 127(2):420–428, 2005.

S. Y. Chen, Y. F. Li. Automatic sensor placement for model-based robot vision. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics,* 34(1):393–408, 2004.

K. A. Tarabanis and R. Tsai. A survey of sensor planning in computer vision. *IEEE Trans. Robot. Autom.*, 11(1): 86–104, 1995.

Y. Morsly, N. Aouf, M. S. Djouadi, and M. Richardson. Particle swarm optimization inspired probability algorithm for optimal camera network placement. *IEEE Sensors Journal*, 12(5):1402–1412, 2012.

U. Erdem and S. Sclaroff Automated camera layout to satisfy taskspecific and floor plan-specific coverage requirements. *J. Comput. Vis. Image Understanding*, 103:156–169, 2006.

F. Angella, L. Reithler, F. Gallesio. Optimal deployment of cameras for video surveillance systems. *In Proceedings of IEEE conference on advanced video and signal based surveillance*, 2007:388–392.

Y. Jiang, J. Yang, W. Chen, W. Wang. A coverage enhancement method of directional sensor network based on genetic algorithm for occlusion-free surveillance. *In Proceedings of international conference on computational aspects of social networks*, 2010:311–314.

R. F. Tobler, S. Maierhofer. A mesh data structure for rendering and subdivision. *Proc. of International Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision*, Plzen, Czech, 2006:157–162.