

無人機於立體建物檢視之覆蓋路徑規劃

李莞瑜

國立政治大學 資訊科學系
台北市文山區指南路二段 64 號
E-mail : 106753009@nccu.edu.tw

李蔡彥

國立政治大學 資訊科學系
台北市文山區指南路二段 64 號
E-mail : li@nccu.edu.tw

摘要—近年來，由於無人機輕巧、機動性高且不易受地形限制之特性，開始有應用無人機於基礎設施（如電塔、老舊大樓）等檢修的案例。然而，無人機的操作大多缺乏自主運動規劃能力，進行檢修時仍是由使用者透過搖桿等介面進行人工操控，使用上不易上手。除此之外，目前市面上較無針對立體建築物巡檢為目的而設計的無人機自動化路徑規劃系統。本論文以此作為研究動機，開發無人機於立體建物檢視之覆蓋路徑規劃系統。透過使用者介面，輸入網格模型與相機參數，系統便能自動計算出無人機檢視之視點，並經由視點產生無人機巡檢路徑。該路徑混合立面與平面路徑，且能與環境不發生碰撞。最後本論文於模擬環境中使用四軸飛行器模擬飛行，證明本論文提出方法之有效性。

關鍵字：無人機、模型簡化、覆蓋路徑規劃

I. 簡介

近年來，無人機的發展已由軍事用途，延伸到廣泛民生用途，大幅降低人力與時間成本。除此之外，也開始有人利用無人機為基礎設施，如電塔、風力發電設備、橋樑、建築物等，進行檢修。臺灣建築物外牆大多以水泥為底，並貼飾瓷磚。氣溫、濕度加上自然災害如颱風、地震頻傳，磁磚剝落的潛在問題也會隨著建築物老化與日俱增。為了減少及預防外牆剝落所造成的損失與安全問題，建物定期檢測成為重要課題。現有檢測方式多依靠人工，由巡檢員藉由垂吊或搭建鷹架的方式，敲擊瓷磚表面搭配目視來判定瓷磚黏著程度。此外，亦可透過紅外線技術辨識磁磚表面溫度或統計特徵差異來進行外牆的損傷判定。然而，人工檢測工作量大、風險高且效率低。

為了提高檢測效率，近年來政府與民間機構開始使用無人機進行檢修[1]。然而，目前市面上無人機的操作大多缺乏自主運動規劃能力。若要有效完成任務，需要良好的操控技術。除此之外，目前市面上並無以立體建築物巡檢為目的而設計的無人機路徑規劃系統。

為解決以上問題，本論文提出一個無人機檢測路徑規劃系統，可以自動計算出建築物的巡檢路徑。使用者透過輸入欲檢視建築的模型檔案，系統便會對模型進行前處理，計算出一條符合使用者需求、且可避開障礙物的無人機飛行路徑。在實際飛行前，使用者也可以透過本系統預先觀看模擬的飛行路徑。

II. 相關研究

A. 模型簡化

現有模型簡化技術已有許多相關研究。目前較常見的方法有：頂點叢集法（vertex clustering）、點/邊/面刪除法（vertex/edge/face decimation）、取樣法（resampling）等[2]。若依組成網格之多邊形形狀分類，又可以分為三

角網格簡化與四邊形網格模型簡化前段敘述之方法大多針對三角形網格簡化。四邊形網格簡化如 Tarini[3]提出一種三角網格轉換為四邊形網格的方法，並提出三種局部簡化作業來達成模型簡化，包括粗糙化作業（coarsening operation）；最佳化作業（optimizing operation）；清除作業（cleaning operation）。

B. 二維覆蓋路徑規劃

覆蓋路徑規劃(Coverage Path Planning)任務為在避開環境中障礙物的同時，經過使用者選定感興趣區域中所有點的路徑。目前這項技術廣泛應用於掃地機器人、割草機、自動收割機...等。由於無人機技術的興起，也有許多研究將無人機應用於覆蓋路徑規劃。如 Torres[4]提出一個應用於三維地形重建的覆蓋路徑演算法。該演算法預設起飛點與降落點為同一點，並提出四種覆蓋掃描的方式。在計算出欲掃描區域的最佳掃描方向（optimal line sweep direction）後，由起飛點開始以四種掃描方法求出距離最短的作為最後的路徑。

C. 三維覆蓋路徑規劃

無人機三維覆蓋路徑規劃需考慮環境中的障礙物與欲掃描物體之結構。現有的無人機三維覆蓋路徑規劃多用於立體結構的建模與檢視。如 Jung[5]提出一個覆蓋路徑演算法 ML-CPP。該算法先將現有的模型轉換成體素圖（volumetric map）。對此體素圖進行分層，並由每個體素的中心點計算出視點，透過濾波器對重疊的視點進行採樣後，由旅行推銷員（traveling salesman problem, TSP）問題解法連結每層的視點形成路徑，接著組合每層路徑得出最後的結果。

III. 系統架構與設計

本論文之系統架構與流程如圖 1 所示。第一部分為模型前處理（model preprocessing），降低模型複雜度，使模型較為易為後續路徑規劃所用。第二部分為路徑規劃（path planning），由簡化的模型、相機參數與無人機起飛點，產生檢視建築物的視點（viewpoints）。將視點分段之後，系統計算覆蓋路徑，再由此覆蓋路徑產生無人機飛行之軌跡。第三部分為路徑模擬或執行，由第二部分產生的無人機路徑檔案，在模擬器中進行飛行模擬或送到實機執行，藉此確認該路徑的正確性。

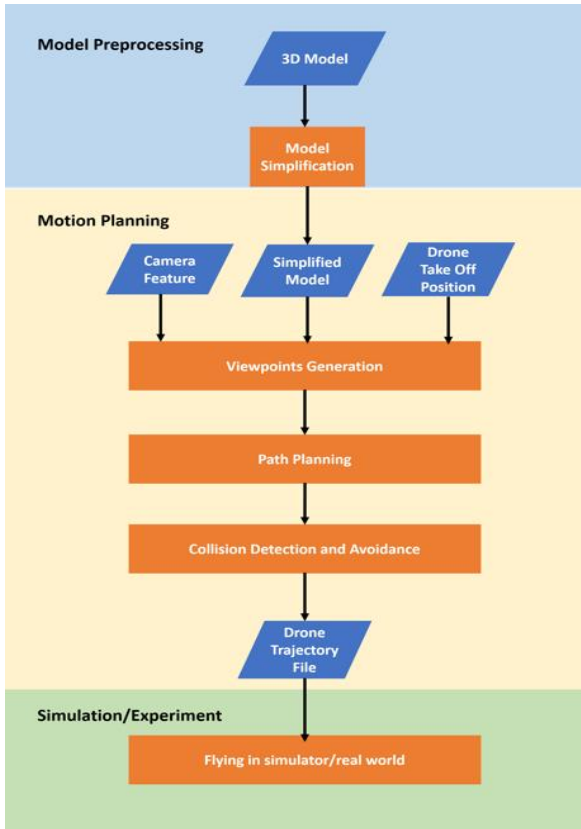


圖 1 系統架構與流程

A. 模型前處理

1) 合併共面

合併共面演算法如 Algorithm1 所示，目的在於透過合併相似共面，減少模型面數。

Algorithm 1 Merge Coplanar Faces

Input: original mesh M_b
Output: mesh after merging coplanar faces M_{co}

```

1: for face  $f$  in  $M_b$  do
2:   if  $f$  has coplanar face then
3:     for coplanar face  $f_{co}$  of  $f$  do
4:       find shared edges and vertices between  $f$  and  $f_{co}$ 
5:       delete shared edges and vertices between  $f$  and  $f_{co}$ 
6:        $f_{new}$  = combine vertices and edges of  $f$  and  $f_{co}$ 
7:     end for
8:     update  $f_{new}$ 's coplanar
9:     if  $f_{new}$  has coplanar face then
10:      add  $f_{new}$  into  $M_b$ 
11:     else if  $f_{new}$  has hole inside polygon then
12:      decompose  $f_{new}$  and add into  $M_{co}$ 
13:     else
14:      add  $f_{new}$  into  $M_{co}$ 
15:     end if
16:   else
17:     add  $f$  into  $M_{co}$ 
18:   end if
19: end for
20: return  $M_{co}$ 

```

檢查原始模型 M_b 中的每個面 f ，若 f 的相鄰面中存在 f 之共面 f_{co} 則將 f 與 f_{co} 進行合併。共面定義為與 f 平面方程式相同的面，或與 f 之間的夾角極小，將此面視為 f 的共面 f_{co} ， f_{co} 可能存在多個。若 f 不存在共面，則將 f 加進新模型 M_{co} 中。

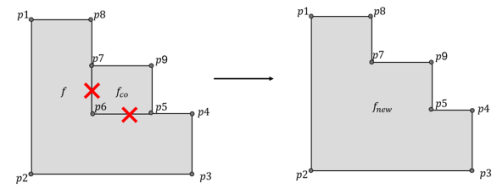


圖 2 合併共面

f 與共面 f_{co} 之合併方法示意圖如圖 2。 f 與 f_{co} 之間存在在共享邊 p_5p_6 與 p_6p_7 。刪除共享邊，並合併剩餘的邊及點，得到合併後的新多邊形 f_{new} 。合併完成後，將 f 與 f_{co} 從模型中刪除。更新 f_{new} 在模型中與其他面的相鄰關係，若 f_{new} 已無其他共面的相鄰面，則將 f_{new} 加進新模型 M_{co} 中，反之則繼續與共面的相鄰面進行合併，直到無共面之相鄰面。在合併共面的過程中，若合併後的面 f_{new} 為有洞的多邊形，如窗戶等，由於在模型檔案中無法表示多邊形內的洞，故將 f_{new} 進行分解後加進新模型 M_{co} 中。

2) 簡化細節

Algorithm 2 Simplify Detail of Mesh

Input:
 M_{co} : Mesh after merge coplanar faces;
 d_s : Safe distance between drone and building;
 l_d : Drone size;
 $hfov/vfov$: horizontal/vertical field of view
Output: Mesh after simplify detail M_s

```

width of camera view  $fpw = 2 * d_s * \tan(hfov/2)$ 
height of camera view  $fpl = 2 * d_s * \tan(vfov/2)$ 
camera footprint  $fp = fpw * fpl$ 
for each face  $f$  in  $M_{co}$  do
  if  $Area(f) < fp$  or  $f$  has edge  $< (d_s + l_d)$  then
    find neighbor face  $f_n$  which  $Area(f_n) < fp$  or  $f_n$  has edge  $< (d_s + l_d)$ 
    if  $f_n \neq None$  then
      find shared edges between  $f$  and  $f_n$ 
      delete shared edges between  $f$  and  $f_n$ 
      merge  $f$  and  $f_n$ 
    else
      find  $f_n$  from  $f$ 's neighbor which has max area
      merge  $f$  and  $f_n$ 
    end if
  end if
end for
return  $M_s$ 

```

簡化模型細節演算法如 Algorithm2 所示，其目的在於排除模型中面積過小的細節，如窗框等。若面積符合但無人機無法進入或可以進入但無法保持安全距離的空間，如開口過小的凹洞等，其相關面也會在這階段被合併。

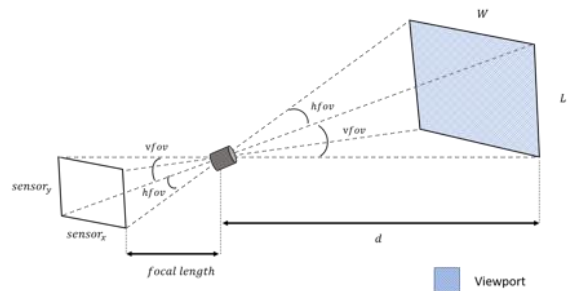


圖 3 視野範圍

基於針孔成像原理，如圖 3 所示，可計算出相機畫面對應到真實世界的視野範圍 (viewport)。以預設之安全距離 d_s 帶入圖中的 d ，計算出以安全距離 d_s 進行拍攝得到的視野範圍 a_{safe} 。

面積過小的面定義為多邊形面積小於 a_{safe} 的面。若模型中有多邊形 f ， f 無共平面的相鄰面，且 f 的相鄰面中，非共面的數量大於等於3面，則判斷 f 可能位於模型中凹陷或凸起位置。若 f 內有小於 $(d_s + l_{drone})$ 的邊，則判斷 f 所形成的空間內，無法讓無人機保持安全距離拍攝，須被簡化。由以上條件，檢查由前一節合併共面後的模型 M_{co} ，若 M_{co} 中有面 f 符合上述條件，則判定 f 需要被簡化。將 f 與凹洞中的相鄰面合併達到簡化效果。

B. 產生視點

視點 (viewpoint) 由無人機組態與相機旋轉角度組成。無人機組態包含位置座標 (x, y, z) 、旋轉角度 ϕ (roll)、 θ (pitch)、 ψ (yaw) 組成。在本論文中由於僅使用平視與俯視來檢視建物表面，故相機角度僅使用 θ_{camera} (pitch) 角。

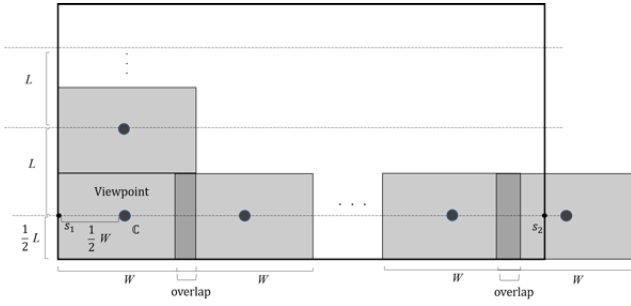


圖 4 多邊形求視點

將簡化後的網格模型 M_s 中的面分為立面與平面。為使無人機檢視時保持水平掃描，則會透過視野範圍之長 L 、寬 W ，由掃描線法 (sweep line) 將多邊形分割並求出視點。分割方法如圖 4 所示，由多邊形底邊開始計算，由距底邊 $\frac{1}{2}L$ 處新增第一條掃描線，之後往上平移，每隔 L 新增一條掃描線。再由每條掃描線與多邊形之切點連線，如圖 4 中的 $\overline{s_1s_2}$ ，從 $\frac{1}{2}W$ 處開始，以固定的覆蓋率 o (overlap)，每隔 $(1-o)*W$ 新增一點，視點位於該點 d 公尺處。若 $\overline{s_1s_2}$ 無法整除 $(1-o)*W$ 時，則以無條件進位的方式新增一視點。

為使無人機在檢視的過程中都可以朝向要檢視的表面，在檢視立面時， ψ (yaw) 為相機拍攝方向相對於+y軸之旋轉角度。拍攝時相機維持固定不動，故旋轉角度皆保持 0° ，若檢視面為平面，則將相機 θ_{camera} (pitch) 角旋轉 90° 朝向下方。

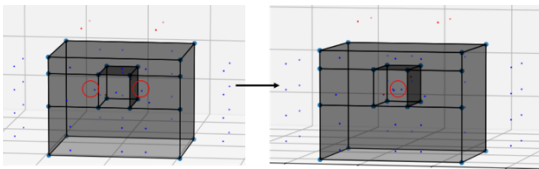


圖 5 更新發生碰撞之視點位置

當以原檢視距離 d 所產生的視點會與模型產生碰撞 (如視點在模型內部等) 時，則系統會將原本由此產生碰撞的視點 vp 所拍到的區域，根據安全檢視距離 d_s ，以前述方法重新計算該區域之視點，如圖 5 所示。

C. 路徑規劃

Algorithm 3 Inspection Path Planning

Input:

- 1: vp : viewpoints of mesh M_s
- 2: M_s : mesh model

Output: ζ : inspection path

- 3: Grouping viewpoints into nodes
- 4: Generate undirected weighted graph G
- 5: $order$ = Solve TSP problem from G
- 6: ζ = Generate inspection path from $order$
- 7: Collision detection and avoidance between ζ and M_s
- 8: return ζ

本論文所提出之路徑規劃演算法如 Algorithm 3 所示。在計算路徑前，先將計算出的視點進行分段。若為立面，將拍攝同一面且高度差小於 $\frac{1}{2}$ 視野範圍長 L 的連續視點視為同一段；若為平面，則以 Torres 於[4]中所提出的最佳掃描方向 (Optimal line sweep direction) 方法，所得的邊作為底邊，將連線與底邊平行的視點們視為同一段。

我們將此路徑規劃問題視為一 TSP 問題。由節點生成無向加權圖 G 。 $G = (N, E)$ ， N 為節點集合， E 為邊集合。假設節點 n_a 與 n_b 為 N 內任意兩節點， n_a 由視點 $\{vp_1, vp_2, \dots, vp_m\}$ 組成， n_b 由視點 $\{vp_1, vp_2, \dots, vp_n\}$ 組成。兩節點之間的連通關係定義如下：若節點 n_a 與節點 n_b 拍攝面相同或相鄰時，則定義為相連， n_a 與 n_b 之間由邊 $e(n_a, n_b)$ 相連。

我們希望針對檢視路徑長度與無人機移動所消耗的能量進行最佳化。由於節點由多個視點組成，會先計算兩節點內，視點與視點之間移動所作的功，並取最小值作為兩節點之間的成本。任意兩相連節點 n_a 與 n_b 之間的邊 $e(n_a, n_b)$ 成本計算如下：

$$\begin{aligned}
 & n_a, n_b \in N \\
 & \forall vp_i \in n_a, i \in 0 \text{ or } \|n_a\| - 1 \\
 & \forall vp_j \in n_b, j \in 0 \text{ or } \|n_b\| - 1 \\
 & W(vp_i, vp_j) = w_{rise} F_{rise} d_{rise} \text{ or } w_{desc} F_{desc} d_{desc} \\
 & \quad + w_{trans} F_{trans} d_{trans} + w_{rotate} \tau \psi \\
 & cost(n_a, n_b) = \min(W(vp_i, vp_j))
 \end{aligned}$$

vp_i 、 vp_j 為節點 n_a 、 n_b 內的視點，可以為第一點或最後一點。 w_{rise} 、 w_{desc} 、 w_{trans} 、 w_{rotate} 分別為上升、下降、水平移動與旋轉的權重。 d_{rise} 、 d_{desc} 、 d_{trans} 分別為上升、下降與水平距離。 F_{rise} 、 F_{desc} 、 F_{trans} 分別為上升、下降與水平移動時無人機所受的力。 τ 為無人機左轉或右轉時產生的力矩， ψ 為無人機左轉或右轉產生的角位移。由於無人機左轉與右轉時，僅受yaw角影響，故僅計算yaw角所作的功。

將無向加權圖 G 使用 Helsgaun 提出的 LKH[6]演算法解決此 TSP 問題，得到每個節點的拜訪順序後，依序連接節點形成路徑。為維持節點內視點的連續性，在連接節點時，分別由兩節點之第一點與最後一點的組合，取距離最短的兩個視點連接。將得到的路徑與模型進行碰撞檢查與排除，得到最終檢視路徑。

IV. 實驗結果與分析

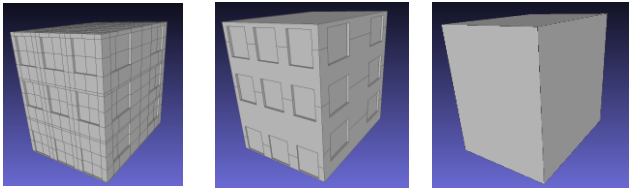


圖 6 透天厝 左：原始模型 中：合併共面後 右：簡化後

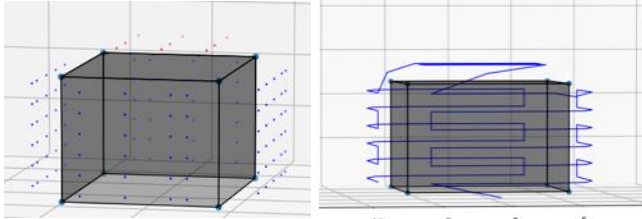


圖 7 透天厝 左：檢視視點 右：檢視路徑

透天厝模型簡化結果如圖 6 所示。原始模型簡化前總面數為 856 面，總頂點數為 877 點。經簡化後總面數為 5 面，總頂點數為 8 點。由簡化後模型所生成之視點由圖 7 左所示，視點皆可保持原先設定之檢視距離，由視點所產生之檢視路徑由圖 7 右所示，該路徑除檢視到所有模型表面外，亦保持穩定的上升或下降，並無大幅度垂直升降，在平面與立面之過渡時亦與模型保持安全距離。

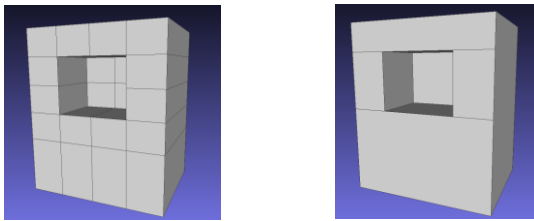


圖 8 凹洞建築 左：原始模型 右：簡化後

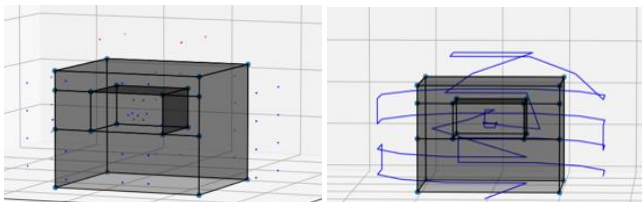


圖 9 凹洞建築 左：檢視視點 右：檢視路徑

凹洞模型簡化結果如圖 8 所示。原始模型簡化前，總面數為 62 面，總頂點數為 68 點。合併共面後，總面數為 13 面，總頂點數為 20 點。因合併共面後之模型表面皆符合所設條件，故不需再簡化細節。由簡化後模型所生成之視點如圖 9 左所示，視點皆可保持原先設定之檢視距離，亦考量凹洞內的面。該模型之檢視路徑如圖 9 右所示，該路徑除檢視到包含凹洞內部之所有模型表面外，亦保持穩定的上升或下降，並無大幅度垂直升降，在平面與立面之過渡時亦與模型保持安全距離。

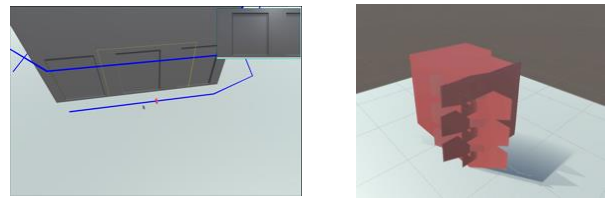


圖 10 透天厝 左：模擬畫面 右：視野範圍軌跡

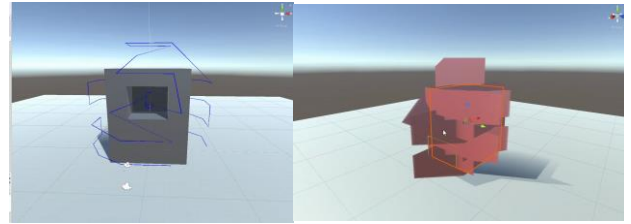


圖 11 凹洞建築 左：模擬畫面 右：視野範圍軌跡

本論文在 Unity3D 模擬與驗證路徑之正確性。由圖 10 與圖 11 所示。圖 10 左與圖 11 左中，藍色線段為由前述演算法所得之檢視路徑，黃色框線為無人機相機之視野範圍。執行時，以子母畫面顯示，右上方之畫面為無人機檢視畫面。除此之外，我們以視野範圍軌跡來驗證路徑是否可以覆蓋到模型的所有表面，如圖 10 右與圖 11 右所示，該圖亦驗證本論文所提出之方法可以讓無人機檢視到模型所有表面，包含凹洞內之表面。

V. 結論與未來研究

本論文完成一建築物檢測路徑規劃系統。此系統提供使用者圖形化介面，使用者輸入欲檢測模型、相機參數等資訊，整合模型前處理、視點產生與路徑規劃等功能進行路徑計算，並讓使用者預先檢視後續檢測之模擬路徑。根據實驗結果，可證實我們提出的方法可檢視到排除無法進入之空間（如開口過小的凹洞）以外的所有表面，且該路徑不會與建築物產生碰撞。未來希望加入多機規劃以提升檢視效率。

致謝

本研究經人工智慧普適研究中心 (PAIR Lab) 授權科技部計畫「前瞻無人機智慧系統：核心技術研發與情境應用展示 (3/4)」執行，於計畫編號 MOST109-2634-F-004-001 補助下完成，特此致謝。

參考文獻

- [1] 張竣堯, "運用無人飛行載具搭載紅外線進行建築外牆磁磚檢測," *建築研究簡訊*, vol. 99, 2018. 中華民國內政部建築研究所.
- [2] P. Cignoni, C. Montani, and R. Scopigno, "A comparison of mesh simplification algorithms," *Computers & Graphics*, vol. 22, no. 1, pp. 37-54, 1998.
- [3] M. Tarini, N. Pietroni, P. Cignoni, D. Panozzo, and E. Puppo, "Practical quad mesh simplification," in *Computer Graphics Forum*, 2010, vol. 29, no. 2: Wiley Online Library, pp. 407-418.
- [4] M. Torres, D. A. Pelta, J. L. Verdegay, and J. C. Torres, "Coverage path planning with unmanned aerial vehicles for 3D terrain reconstruction," *Expert Systems with Applications*, vol. 55, pp. 441-451, 2016.
- [5] S. Jung, S. Song, P. Youn, and H. Myung, "Multi-Layer Coverage Path Planner for Autonomous Structural Inspection of High-Rise Structures," presented at the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018.
- [6] K. Helsgaun, "An effective implementation of the Lin-Kernighan traveling salesman heuristic," *European Journal of Operational Research*, vol. 126, no. 1, pp. 106-130, 2000.

