

附錄四 自動化程式碼

分割程式碼:

```
createDisplayLayer -name "L_ref_plane" -number 1 -empty;
createDisplayLayer -name "L_ref_intersectline" -number 1 -empty;
string $ref_planes[];
string $ref_intersectlines[];
string $selected[]=`ls -sl`;
for ($i=1;$i<=size($selected);$i++){
    string $ref_plane[0]= `extrude -name refplane1 -ch true -rn false -po 0 -et 0 -upn 0 -d 0 1 0 -length
    10 -rotation 0 -scale 1 -dl 1 $selected[$i-1]`;
    $ref_planes[$i-1]= $ref_plane[0];
    select -r ($ref_planes[$i-1]);
    layerEditorAddObjects L_ref_plane;
    string $ref_intersectline[0]= `intersect -n proline -ch true -fs 1 -cos 0 -tol 0.01 ($ref_planes [$i-1])
    "loftedSurface1"`;
    $ref_intersectlines[$i-1]= $ref_intersectline[0];
    select -r ($ref_intersectlines [$i-1]);
    layerEditorAddObjects L_ref_intersectline;
};
```

骨架擠出程式碼:

```
createDisplayLayer -name "oa_extrude_A" -number 1 -empty;
for ($i = 1; $i < size($selected); $i++) {
    string $curveShape[0] = `listRelatives -s $selected[$i]`;
    int $degree = `getAttr ($curveShape[0] + ".degree")`;
    rebuildCurve -kcp 1 -keepRange 0 -d $degree $selected[$i];
    string $surface[0] = `extrude -ch true -rn false -po 0 -et 2 -ucp 1 -fpt 1 -upn 1 -rotation 0 -scale
    1 -rsp 1 -name ($profile[0] + "_" + $selected[$i] + "_extrude") $profile[0] $selected[$i]`;
    $surfaces[$i - 1] = $surface[0];
    select -r ($surfaces[$i - 1]);
    layerEditorAddObjects oa_extrude_A;
    if ($caps == 1) {
        int $spans = `getAttr ($surface[0] + ".spansV")`;
        select -r ($surface[0] + ".v[0]");
        catch (`planarSrf -name ($surface[0] + "_cap1") -ch 1 -d 3 -ko 0 -tol 0.00393701 -rn 0
        -po 0`);
        select -r ($surface[0] + ".v[" + $spans + "]");
        catch (`planarSrf -name ($surface[0] + "_cap2") -ch 1 -d 3 -ko 0 -tol 0.00393701 -rn 0
        -po 0`);
        parent ($surface[0] + "_cap1") ($surface[0] + "_cap2") $surface[0];
        setAttr ($surface[0] + "_cap1.inheritsTransform") 0;
        setAttr ($surface[0] + "_cap2.inheritsTransform") 0;
    }
};
```

```

    }
    currentTime -e $time;
  }
  select -r $profile[0];
  return $surfaces;

```

(修改自 open source – Oleg Alexander 的 oaExtrudeAll.mel)

凹槽製作程式碼:

```

createDisplayLayer -name "du_extrude " -number 1 -empty;
layerEditorSelectObjects oa_extrude_A;
string $oriframeA[]=`ls -s\`; select -cl ;
layerEditorSelectObjects oa_extrude_B;
string $oriframeB[]=`ls -s\`; select -cl ;
string $duframeA[]=`duplicate $oriframeA`;
select -r $duframeA; move -r 0 0.1 0;
layerEditorAddObjects du_extrude;
string $duframeB[]=`duplicate $oriframeB`;
select -r $duframeB; move -r 0 -0.1 0;
layerEditorAddObjects du_extrude;

layerEditorSelectUnused;
layerEditorDeleteLayer "";
group -n groupA1;
xform -os -piv 0 0 0;
createDisplayLayer -name "group_profile" -number 1 -empty;
layerEditorAddObjects group_profile;

```

折板編號程式碼:

```

string $selected[]=`ls -s\`;
for ($i=0; $i< size($selected); $i++) {
  rename $selected[$i] obj1;
};
string $selected[]=`ls -s\`;
for ($i=0; $i< size($selected); $i++) {
  float $XYZ[] = `objectCenter -l ($selected[$i])`;
  float $anno_x = $XYZ[0];
  float $anno_y = $XYZ[1];
  float $anno_z = $XYZ[2];
  annotate -tx $selected[$i] -p $anno_x $anno_y $anno_z ($selected[$i]);
  string $r = $i+1;
  string $annotation_name[]={"annotation"+$r};
  parent ($selected[$i]) ($annotation_name);
};

```

夾角計算程式碼:

```

string $selected[]=`ls -s\`;

```

```

for ($i=0; $i< size($selected); $i++) {
    select -r ($selected[$i] + ".vtx[0:3]");
    string $selectedCV[] = `ls -sl -fl`;

    float $pos3[] = `pointPosition -w $selectedCV[3]`;
    float $point_a_1_0 = $pos3[0];
    float $point_a_1_1 = $pos3[1];
    float $point_a_1_2 = $pos3[2];
    float $pos0[] = `pointPosition -w $selectedCV[0]`;
    float $point_a_2_0 = $pos0[0];
    float $point_a_2_1 = $pos0[1];
    float $point_a_2_2 = $pos0[2];
    float $pos2[] = `pointPosition -w $selectedCV[2]`;
    float $point_a_3_0 = $pos2[0];
    float $point_a_3_1 = $pos2[1];
    float $point_a_3_2 = $pos2[2];
    float $pos0[] = `pointPosition -w $selectedCV[0]`;
    float $point_b_1_0 = $pos0[0];
    float $point_b_1_1 = $pos0[1];
    float $point_b_1_2 = $pos0[2];
    float $pos0[] = `pointPosition -w $selectedCV[1]`;
    float $point_b_2_0 = $pos0[0];
    float $point_b_2_1 = $pos0[1];
    float $point_b_2_2 = $pos0[2];
    float $pos0[] = `pointPosition -w $selectedCV[2]`;
    float $point_b_3_0 = $pos0[0];
    float $point_b_3_1 = $pos0[1];
    float $point_b_3_2 = $pos0[2];

    vector $plane_a_line_uv1=<<$point_a_2_0-$point_a_1_0, $point_a_2_1-$point_a_1_1,
    $point_a_2_2 - $point_a_1_2>>;
    vector $plane_a_line_uv2=<<$point_a_3_0-$point_a_1_0, $point_a_3_1-$point_a_1_1,
    $point_a_3_2 - $point_a_1_2>>;
    vector $plane_a_uv = << ($plane_a_line_uv1.y) * ($plane_a_line_uv2.z) - ($plane_a_line_uv1.z)
    * ($plane_a_line_uv2.y), - (($plane_a_line_uv1.x) * ($plane_a_line_uv2.z) - ($plane_a_line_uv1.z)
    * ($plane_a_line_uv2.x)), ($plane_a_line_uv1.x) * ($plane_a_line_uv2.y) - ($plane_a_line_uv1.y)
    * ($plane_a_line_uv2.x)>>;

    vector $plane_b_line_uv1=<<$point_b_2_0-$point_b_1_0, $point_b_2_1-$point_b_1_1,
    $point_b_2_2-$point_b_1_2>>;
    vector $plane_b_line_uv2=<<$point_b_3_0-$point_b_1_0, $point_b_3_1-$point_b_1_1,
    $point_b_3_2-$point_b_1_2>>;
    vector $plane_b_uv = << ($plane_b_line_uv1.y) * ($plane_b_line_uv2.z) - ($plane_b_line_uv1.z)
    * ($plane_b_line_uv2.y), - (($plane_b_line_uv1.x) * ($plane_b_line_uv2.z) - $plane_b_line_uv1.z)
    * ($plane_b_line_uv2.x)), ($plane_b_line_uv1.x) * ($plane_b_line_uv2.y) - ($plane_b_line_uv1.y)
    * ($plane_b_line_uv2.x)>>;

    float $a_dot_b = ($plane_a_uv.x) * ($plane_b_uv.x) + ($plane_a_uv.y) * ($plane_b_uv.y) +
    ($plane_a_uv.z) * ($plane_b_uv.z);
    float $length_plan_a_uv=sqrt ((($plane_a_uv.x) * ($plane_a_uv.x)) + (($plane_a_uv.y) *

```

```

(plane_a_uv.y)) + (($plane_a_uv.z) * ($plane_a_uv.z));
float $length_plan_b_uv=sqrt ((($plane_b_uv.x) * ($plane_b_uv.x)) + (($plane_b_uv.y) *
($plane_b_uv.y)) + (($plane_b_uv.z) * ($plane_b_uv.z)));

float $angle_between_vector=acos($a_dot_b/($length_plan_a_uv*$length_plan_b_uv));
float $angle_between_plane=180-(rad_to_deg($angle_between_vector));

print ("(" + $selected[$i] + "=" + $angle_between_plane + ")");
}

```

2D 展開程式碼:

```

string $selected[]=`ls -sl`;
for ($i=0; $i< size($selected); $i++) {
    float $r = $i+1;
    string $anno[] = {"annotation"+$r};
    parent -w ($selected[$i]);
    delete $anno;

    select -r ($selected[$i] + ".vtx[0:2]");
    select -tgl plane.vtx[0:2];
    snap3PointsTo3Points(0);

    select -r ($selected[$i]);
    move -r 0 0 (10*$i) ;

    select -r ($selected[$i] + ".vtx[0:3]");
    string $selectedCV[] = `ls -sl -fl`;
    float $pos0[] = `pointPosition -w $selectedCV[0]`;
    float $anno_x = $pos0[0];
    float $anno_y = $pos0[1];
    float $anno_z = $pos0[2];
    annotate -tx $selected[$i] -p $anno_x $anno_y $anno_z ($selected[$i]);
}

string $selected[]=`ls -sl`;
for ($i=0; $i< size($selected); $i++) {
    select -r ($selected[$i] + ".vtx[0:3]");
    string $selectedCV[] = `ls -sl -fl`;

    float $pos3[] = `pointPosition -w $selectedCV[3]`;
    float $point_a_1_0 = $pos3[0];
    float $point_a_1_1 = $pos3[1];
    float $point_a_1_2 = $pos3[2];
    float $pos0[] = `pointPosition -w $selectedCV[0]`;
    float $point_a_2_0 = $pos0[0];
    float $point_a_2_1 = $pos0[1];
    float $point_a_2_2 = $pos0[2];
    float $pos2[] = `pointPosition -w $selectedCV[2]`;

```

```
float $point_a_3_0 = $pos2[0];
float $point_a_3_1 = $pos2[1];
float $point_a_3_2 = $pos2[2];

float $length_from_0=sqrt (($point_a_1_0-$point_a_2_0) * ($point_a_1_0-$point_a_2_0) +
($point_a_1_1-$point_a_2_1) * ($point_a_1_1-$point_a_2_1) + ($point_a_1_2-$point_a_2_2) *
($point_a_1_2-$point_a_2_2));
float $length_from_2 = sqrt (($point_a_1_0-$point_a_3_0) * ($point_a_1_0-$point_a_3_0) +
($point_a_1_1-$point_a_3_1) * ($point_a_1_1-$point_a_3_1) + ($point_a_1_2-$point_a_3_2) *
($point_a_1_2-$point_a_3_2));

circle -nr 0 1 0 -c $point_a_2_0 $point_a_2_1 $point_a_2_2 -r $length_from_0;
circle -nr 0 1 0 -c $point_a_3_0 $point_a_3_1 $point_a_3_2 -r $length_from_2;

select -r nurbsCircle1 ;
select -tgl nurbsCircle2 ;
CutCurve;
};
```

