

# 大量の楕円形の葉を対象とした 動的プロジェクションマッピングの自動生成

末吉知樹<sup>1)</sup> (学生会員)

森本有紀<sup>2)</sup> (正会員)

1) 九州大学芸術工学府

2) 九州大学芸術工学府

## Automatic Generation of Dynamic Projection Mapping for a Large Number of Elliptic Leaves

Tomoki Sueyoshi<sup>1)</sup>(Student Member)

Yuki Morimoto<sup>2)</sup>(Member)

1) Graduate School of Design, Kyushu University

2) Faculty of Design, Kyushu University

morimoto@design.kyushu-u.ac.jp

### アブストラクト

大量の楕円形の葉への動的なプロジェクションマッピングを自動生成する手法を提案する。近年、プロジェクションマッピングの投影対象は、建物から魚など様々であるが、大量の葉で構成される植物のように、多数の動的で可変な対象への投影は非常に困難である。本手法では、多数の葉を投影領域として選出・追跡し、自動で投影領域の位置合わせを行う。また、ユーザー入力を用いることで、多数の葉へのエフェクトアニメーションの向きや順番を一括で適用するシステムを提案する。

キーワード：動的プロジェクションマッピング、画像処理、ビジュアルエフェクト

### Abstract

We propose a method to generate dynamic projection mapping for a large number of elliptical leaves by automatically tracking the projection areas and the image registrations of projection areas. Recently, projection mapping targets have included a wide variety of objects, ranging from buildings to fish. However, it is difficult to project images onto a large number of dynamic and deformable targets such as plants composed of many leaves. Our system automatically specifies and tracks dynamic and deformable plants as target projection areas for registration to them. We also propose a system user interface for designing the direction and order of effect animations onto many leaves at once using user input.

**Keywords:** Dynamic projection mapping, image processing, visual effects

## 1. はじめに

Projections in the Forest [1]はプロジェクションマッピング (PM) で森の植物や生き物を投影対象とした幻想的な映像作品であり、生物発光をモチーフとしたエフェクトアニメーション (エフェクト) を対象の形状に合わせて投影している。この作品では、屋外の森で映像と投影対象の位置合わせを手作業で行う必要があり、膨大な労力と時間がかかる。また本研究では、大量の葉で構成されるような植物を投影対象として扱うが、投影対象数の増加に伴って、位置合わせに要する労力と時間も増加する。このように PM の対象は近年多様化しているが、植物の葉は困難な対象物の一つである。そこで本システムでは、屋外に大量

にあり動的である葉、を投影対象とした動的プロジェクションマッピング (DPM) の生成を支援するシステムを提案する。

本研究では、風や人が触るなどの外的要因による動きを考慮し、多数の葉をそれぞれ追跡することで、DPM を実現する。DPM を行うための追跡手法は複数提案されている[2, 3]が、事前の入力 (投影対象の 3D モデルや素材の物理特性など) なしで動的で可変な対象へ追跡を行うことは、非常に困難である。事前入力無しで未知の形状に DPM を行う事例もある[4]が、高価な機材が複数必要である。よって本システムでは、より安価な機材で DPM をできるだけ自動的に生成する。また、事前に投影対象のデータをシステムに入力することなく、指定した葉に対して自動的に投影領域の位置合わせと追跡を行う。一方、

これまでの葉を対象とした DPM の提案手法[5,6]では、10 枚程度の対象までしか追跡できなかつたため、本手法では追跡を高速化することで、投影する葉の枚数を増やす。大量の対象の位置を高速に求めるため、投影対象を楕円形の葉に限定し、リアルタイムで対象を近似した楕円を最小二乗法で求めることで約 50 枚の葉を追跡する。また、対象同士における接触・重なりの可能性が高い位置を判定し、可能性が低い位置から、初期形状を維持する位置を求めることで、対象同士の接触・遮蔽にも頑健な追跡手法を提案する。

さらに、ユーザーが枝情報を描画することで、多数の葉へ投影するエフェクトの動きや向きを直感的かつ一括で指定できる UI を提案する。投影映像については、生物発光のように植物自身が発光しているような表現を目指し、投影対象の形状と入力した枝情報から自動生成するエフェクトを 2 種類作成する。

## 2. 先行事例

### 2.1 ダイナミックプロジェクションマッピング

DPM を行うための追跡手法として、非剛体への DPM における自動的な位置合わせの研究は様々な手法が提案されている。Narita らの研究[2]では、紙や布の形状変化も考慮した投影を実現しているが、追跡には赤外線マーカーを必要とする。また、Bermano らの研究[3]では、人の顔に対して変形を考慮した位置合わせを高速に行っているが、投影対象である顔の三次元モデルを事前に入力する必要がある。このように、投影対象の 3D モデルや素材の物理特性など事前の入力を用いた既存手法が動的で可変な対象の追跡手法としては多い。本研究ではマーカーレスかつ事前入力無しで、投影対象である葉を追跡し、位置合わせを行う。

一方で、MIDAS[4]では、事前準備なしで、動的で可変な対象への DPM を実現している。3 台の赤外線カメラを用いて、対象の法線情報を取得することで、手や粘土、液体などの様々な物体の形状変化を追跡し、マテリアルやテクスチャを変更する表現を行うことができる。しかし機材が非常に高価であるという制約がある。本手法では、将来的な社会実装を期待し、比較的に安価で一般的な機材で DPM を行う。

### 2.3 植物を用いたインタラクション作品

植物を用いたメディアアートの事例を複数紹介する。Interactive Plant Growing[7]は、鑑賞者が植物に触れることで、仮想空間の植物が育っていくインスタレーションである。また、Botanicus Interacticus[8]は、植物に触れた位置等を推測できるセンシング技術によって、ディスプレイにエフェクトを表示するインスタレーションである。本研究では、対象の形状情報等から生成したエフェクトを現実の植物に投影し、植物自体の見え方を変えることで、より対話的なコンテンツの作成を目指す。

一方で、The Royal Botanic Garden[9]は木に投影を行った作品である。対象の木を 3D スキャンし、映像の投影範囲などを事前にシミュレートすることで PM を制作しており、木の表面

に様々な模様を投影している。本研究では、植物の葉それぞれを追跡し、各対象の形状に合わせて生成した映像の提示を行う。

## 3. 本手法

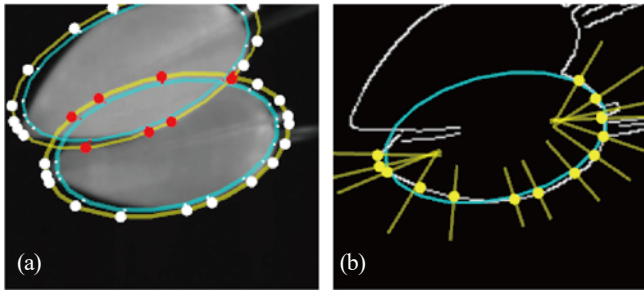
本研究では、3.1節で述べるシステム構成によって、位置合わせと、投影領域の選出と追跡、エフェクトの生成を自動で行い、多数の葉への DPM を実現する。投影面をカメラで取得し、投影面に適応させた映像をプロジェクターで投影するシステム (ProCam) を用いる。ProCam の位置合わせでは、事前に投影のキャリブレーション情報を取得しておくことで、投影時にかかる労力を軽減する手法[10]を採用する。本システムの手順として、最初に、初期画面( $t=0$ )において映像を投影したい葉の領域をユーザーがインターフェース上で選択する。その後( $t>0$ )、追跡中は毎フレーム、一つ前のフレームの各輪郭の頂点を追跡する。つづいて、輪郭形状とユーザー入力を用いて生成したエフェクトを追跡された投影領域にテクスチャマッピングする。最後に生成された結果をプロジェクターで投影する。

### 3.1 システム構成

本システムは、ProCam に加え、赤外線照明を用いる[10] (図 1)。ProCam は、双方の光軸のズレを最小限に抑えるために可能な限り近くに配置し、葉の全面が見える位置に設置する。また、PM の鑑賞に適した暗い環境で投影を行うことが多い。さらに、街灯の白熱電球の指向性が強い光や、太陽光の強さの変化は、投影領域の選出や追跡を困難にする。そのため、本システム以外の照明がほとんどない環境を想定し、夕方から夜間などの時間、街灯の少ない場所を選定した上で赤外線カメラ撮影のために赤外線照明を設置する。また、赤外線カメラによる撮影で葉と背景が同様にならないよう、対象物体である植物の背景には、赤外線を反射する物体が無い環境で投影する。対象に投影したプロジェクターの映像の光の波長は約 360 nm ~ 830 nm であり、おおそ赤外線カメラ (780nm-) の波長外なので、投影された映像の本手法の画像処理への影響を無視することができる。



図 1. ProCam (上), システム設置図 (下).



分割と削除の結果, 残った各輪郭の頂点を毎フレーム追跡す

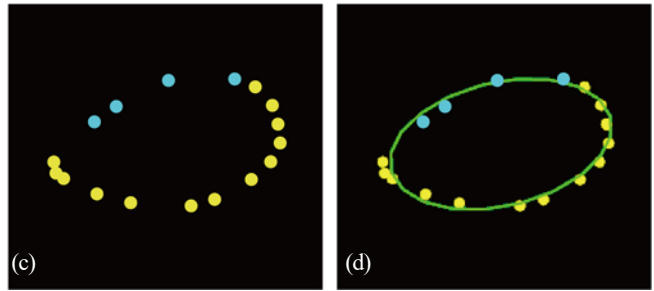


図 2. 楕円近似を用いた追跡手法の概要.

### 3.2 投影領域の選出

最初に得たシーンの赤外線画像に対し, 画像処理による自動の投影領域抽出後, そのうち投影対象としない領域の削除や, 2枚の葉が接触し一つの領域となっている部分の分割などの修正を, マウスを用いたユーザー入力によって柔軟に行う.

初期フレーム( $t=0$ )において, 以降追跡する投影領域を以下のように決定する.

- A. 赤外線カメラで取得した初期フレーム画像 $I^0$ に対し, ガウシアンフィルタを適用し, 平滑化画像を取得する.
- B. 平滑化画像から canny 法によってエッジ画像 $E^0$ を取得する.
- C. 平滑化画像に閾値処理を適用し二値化画像を取得する.
- D. 二値化画像からエッジ画像 $E^0$ を減算した画像に対し, モルフォロジー処理のオープニングを適用し, 白領域の輪郭を抽出する.
- E. 抽出された輪郭に対し, 凹部分の輪郭の頂点を用いて, 輪郭の分割と削除を行う.
- F. E で求めた輪郭に対し, マウス入力によって分割と削除を行う.

Dの結果では, 葉同士の隣接や重なりによって, 複数の対象が一つの領域として繋がっている輪郭や, 枝等の投影対象ではない物体との隣接によって, 投影対象では無い領域を含む輪郭が抽出されることがある. そのため, E では, 輪郭における内角が $\pi$ 以上の頂点を凹部分の頂点として求め, 輪郭の分割と削除を行う. まず, D で抽出された輪郭に対し, 制約付きドロネー分割を行う. 次に, 分割された各三角形において, 凹部分の頂点を2つ以上持つとき, その凹部分の頂点を結ぶ線分によって輪郭を分割する.

Fにおける, マウス入力を用いた分割では, マウスドラッグの押し放しで輪郭上の2つの頂点を選択し, 2点を結ぶ線分によって輪郭を分ける. 削除は, 輪郭を遮るようにマウスドラッグで線を描画し, 入力線と各輪郭が2回以上交差する時に実行する.

また, Fの選出結果, 追跡には適さないような小さい輪郭も選出される. そのような輪郭を削除するため, 輪郭面積が閾値以下の時, 追跡対象から外す. 閾値は, UI上のスライドバーを移動させることによって, 変更することができる.

る輪郭の頂点 $p_i^0 \in \mathbb{R}^2, i=1,2,\dots,n, n$ : 頂点数, とする. つづいて, 選出された各領域の輪郭頂点の点列を楕円 $O^0$ で近似する. また, 近似楕円 $O^0$ の長軸と短軸を定数倍 (本研究では 1.1 倍) に拡大した, 大きい楕円 $Q^0$ を求める. さらに, それぞれの楕円 $O^0, Q^0$ において, 楕円中心から $p_i^0$ 方向に伸ばした半直線との交点を $o_i^0, q_i^0 \in \mathbb{R}^2$ とする. 本手法の追跡では, 毎フレーム, 前フレームの楕円 $O^{t-1}$ 上の点 $o_i^{t-1}$ を用いながら, 楕円 $O^t$ を追跡する.

### 3.3 投影領域の追跡

追跡する各対象の領域はその頂点を楕円で近似し, その周囲のエッジ上のみ次のフレームの対象の頂点があると想定することで, 計算速度および追跡の精度を上げる. 前フレームの近似楕円から現フレームの頂点 $p_i^t$ を求め, 点列を用いて映像をテクスチャマッピングする. このとき, 他の対象が重なると, 他の対象の頂点を現フレームの頂点 $p_i^t$ として求める可能性があり, 追跡の精度が下がる. そのため, 他の対象との重なった頂点の位置を修正した上で楕円近似する. この処理は毎フレーム行う. 以下はその詳細である.

楕円近似は, 点集合に対して二乗誤差が最小になる楕円を求める[11]. これまでの追跡手法[5, 6]ではオプティカルフローやテンプレートマッチングなど画像ベースの方法でフレーム間の類似する箇所を求め追跡することから, 計算時間を要する. 本手法では, エッジ画像から抽出したベクタ情報ベースで一つ一つの対象を楕円近似することで高速な追跡を目指す.

毎フレーム, 追跡に用いるためのエッジ画像 $E^t$ をまず求める. cannyエッジ画像を初期フレームと同様に求めた後 (3.2節AB), 追跡のノイズとなる輪郭内部のエッジを削除するため, 3.2節ACと同様の方法で求めた二値化画像をcannyエッジ画像から減算し, エッジ画像 $E^t$ とする.

エッジ画像 $E^t$ 上での追跡処理を, 下記のA~Eに示す. Aでは投影領域同士の重なりによる他の対象の頂点を修正するため, 各頂点における他の投影領域と重なりを楕円の衝突判定によって求める. B, Cでは追跡結果 $p_i^t$ を求める. Aで衝突しない点の追跡結果を近傍のエッジ画像 $E^t$ 上で求め, 衝突した点の追跡結果は初期フレームの位置関係から相対的に求める. Dではこれらの点に対して楕円近似を行い, 現フレームの楕円 $O^t$ を求める. 具体的には以下の流れで追跡する (図2).

- A. 大きい楕円 $Q^{t-1}$ 同士の衝突判定を行い, 楕円上の点 $q_i^{t-1}$

の中で他の投影領域と衝突した点を求める (図2a) .

- B. 点 $o_i^{t-1}$ で、 $O^{t-1}$ における法線とエッジ画像 $E^t$ のエッジの交点を計算し、 $p_i^t$ とする (図2b) .
- C. 他の投影領域と衝突した頂点の追跡結果 $p''_i^t$ を計算する (図2c) .
- D. B, Cで求めた頂点 $p_i^t$ ,  $p''_i^t$ を追跡結果 $p_i^t$ とし、Kalmanフィルタを適用したものを出力結果とする.
- E. 頂点 $p_i^t$ を楕円 $O^t$ で近似する (図2d) .

**A) 投影領域同士の重なり** 複数の投影対象が干渉する場合、 $Q^{t-1}$ 同士の衝突判定を行うことで、投影対象同士で重なる可能性が高い箇所を疑似的に求める。まず、 $Q^{t-1}$ 同士の衝突判定を行う (図1a) .  $Q^{t-1}$ 同士が衝突しているとき、頂点 $q_i^{t-1}$ ごとに内外判定を行い、他方の楕円の内側にあるとき衝突した点 (図2a赤点) とする。

**B) 近似楕円の法線とエッジの交点の計算** 衝突しない箇所について、現フレームにおける対応箇所をエッジ上で求める。まず、 $o_i^{t-1}$ の位置で $O^{t-1}$ における法線 (図2b黄線) を計算する。次に、エッジ画像 $E^t$ から輪郭抽出によってエッジベクタを抽出する。このとき、枝と葉が接する箇所では、エッジベクタを抽出できない可能性がある。そのため、抽出したエッジベクタの連結情報を用いて、隣接点で囲まれる内角が $0 \sim 2\pi/3$ ,  $4\pi/3 \sim 2\pi$ の点を特徴点として計算し、特徴点同士で結んだ線をエッジベクタとして追加する。このエッジベクタと法線の交点 $p_i^t$  (図2b黄点) を求める。一つの法線に対し交点が複数ある時は、 $p_i^{t-1}$ とのユークリッド距離が最も短い交点を $p_i^t$ とする。また、交点が見つからない時は、前フレームの追跡結果 $p_i^{t-1}$ を $p_i^t$ とする。

**C) 衝突した点の追跡** Bで求めた対応点 $p_i^t$  (図2c黄点) を制御点としたリジッド変形[12]を初期形状に行うことで、衝突した点の追跡結果 (図2c青点) を計算し、点 $p''_i^t$ とする。

**D) Kalmanフィルタの適用** B, Cで求めた頂点 $p_i^t$ ,  $p''_i^t$ を追跡結果 $p_i^t$ には動きとしてブレがあるため、Kalmanフィルタを適用し、投影エフェクトの描画時にこの予測値を用いることで、追跡によって生じるブレを減らし、バードウェアによって生じる遅延を軽減する。

**E) 近似楕円の追跡** まず、B, Cで求めた点 $p_i^t$ ,  $p''_i^t$ を用いて、最小二乗法による楕円近似を行う (図2d) . 求めた楕円 (図2d緑線) を近似楕円の追跡結果 $O^t$ とする。また、初期フレームと同様、大きい楕円 $Q^t$ 、各楕円中心から $p_i^t$ 方向に伸ばした半直線との交点を $o_i^t$ ,  $q_i^t$ を計算する。

### 3.4 ユーザー入力を適用したエフェクト生成

本研究では、生物発光のように、投影対象自身が発光しているような表現を目指し、葉自身から得られる輪郭形状、植物や自然のモチーフを用いてエフェクトを自動生成する。

また、葉が繋がっている枝に沿ったエフェクトを生成するため、枝に対する葉の位置・順番が必要である。画像処理による枝情報の抽出を行えないため、UI上でユーザーが葉と枝の接す

る位置を頂点 $r_j \in \mathbb{R}^2, j=1,2,\dots,m, m$ : 頂点数、としてマウスクリックで順番に指定 (図3赤点) することによって、枝にあたる線を描画し (図3白線) , 枝と葉の位置関係を取得する。本システムでは、この情報を用いた2つのエフェクトを提案する。

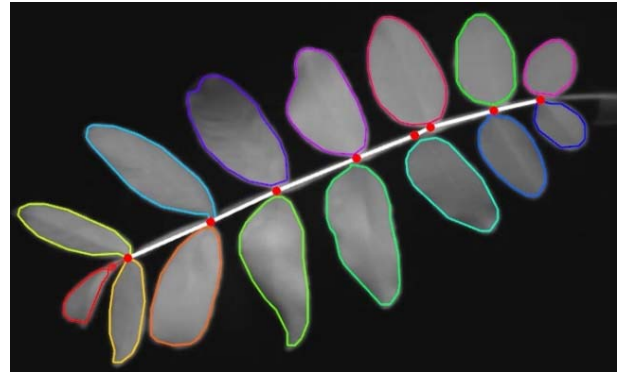


図 3. マウスクリック指定 (赤点) による線の描画 (白線) .

**1) 葉脈をモチーフとしたエフェクトにおける始点指定** 葉脈をモチーフとしたエフェクトを生成する。画像情報から葉脈を抽出できないため、Runionsらの手法[13]を用いて葉脈を生成し、葉脈から計算した速度場上で粒子を動かす。葉脈の始点の指定に、枝状にマウスクリックで順番に指定した頂点 $r_j$ の連結情報を用いる。輪郭の各頂点 $p_i^0$ と頂点 $r_j$ の距離 $d$ を計算し、距離 $d$ が最小の時の頂点 $p_i^0$ を始点 $p_{origin}^0$ として採用し、葉脈を生成する (図4) .



図 4. 枝入力で始点を指定することによって生成した葉脈。

**2) ホタルをモチーフとしたエフェクトにおける光る順番の指定** ホタルの発光リズムに見られる1/fゆらぎを用いて投影領域全体が点滅するエフェクトを作成する。また、光が枝に沿って伝搬していくようなエフェクトを作成するため、ユーザーが描画した線を用いる。まず、葉脈をモチーフとしたエフェクトと同様に、輪郭の各頂点 $p_i^t$ と指定した頂点 $r_j$ の距離 $d$ を計算し、距離 $d$ が最小の時の頂点 $r_l$  ( $l$ : 距離 $d$ が最小の時の頂点番号) を求める。描画した線の始点 $r_0$ から求めた点 $r_l$ までの長さを用いて、エフェクトの光の大きさを変えることで、投影領域が枝に沿って順番に光るようなエフェクトを生成する (図5) .

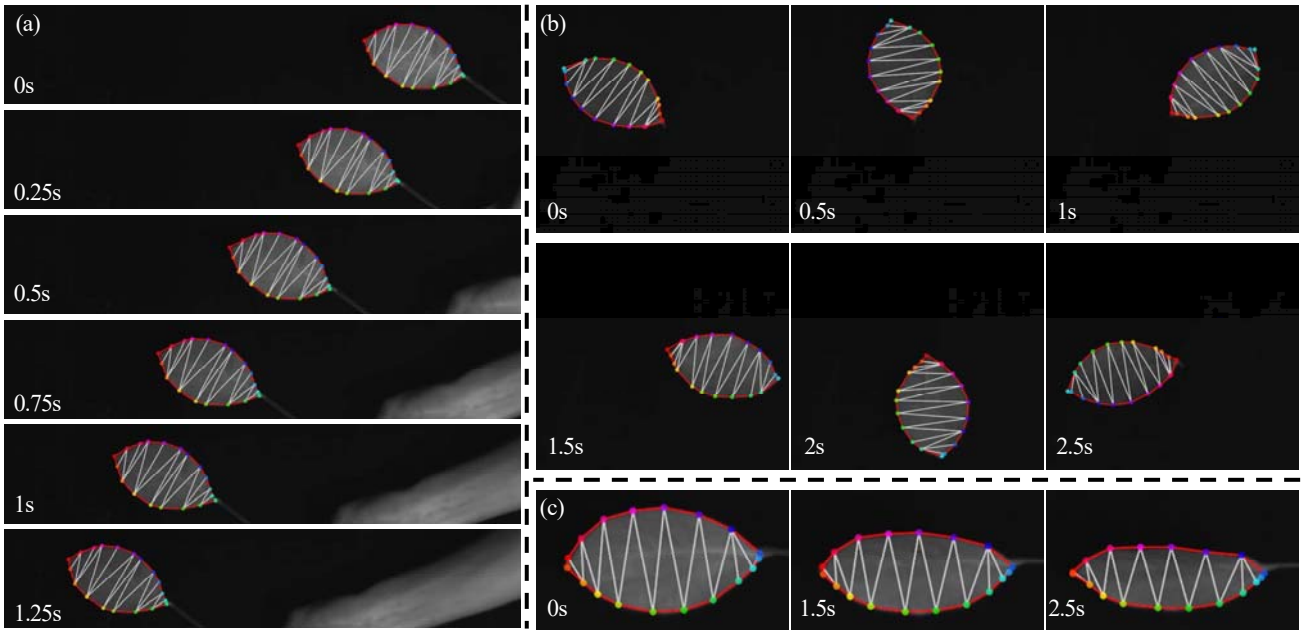


図6. 動く楕円形の葉に対する追跡結果 (a)平行移動させた葉, (b)カメラ方向の軸を中心に約360度回転させた葉, (c)葉の長軸方向を軸に回転させた葉.

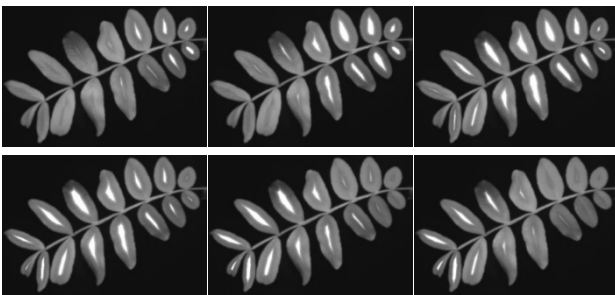


図5. 枝入力を適用したホタルモチーフのエフェクト.

#### 4. 結果考察

本研究によって作成した追跡結果を図6, 投影結果を図7, 9, 10に示す. 本手法で示す結果の作成には, Windows PC(Intel Core i9-9900(3.1GHz) PC)を使用した. カメラは解像度2048×1088pxでフレームレート169fpsのSTC-MBCM200U3V-NIRを, プロジェクターは解像度 1280×720pxのTaxan KG-PL081W, 赤外線照明はS8100-30-B/C-IRを用いた. また, 実験は, 照明の無い屋内や夜間の屋外など, 赤外線照明以外の光源の影響が少ない環境で行った. これによって意図しない光の影響を避けることができた.

まず, 投影領域の選出では, UI上のマウスドラッグで投影領域の削除と分割を行うシステムによって, 多数の葉が含まれるシーンにおいて柔軟に領域を選出することができた. また, 5分程で, 多数の葉から投影領域を選出することができた. 一方で, 削除の際, 入力線の長さに関わらず2回交差した輪郭が削除されるため, ユーザーが気づかずに極端に短い距離の交差を発生させてしまうことで意図しない削除が起きることがある. この誤動作は, 5回に1回程度の割合で起こるが, 投影対象領域周辺のみで削除を丁寧に行うことで回避できる. また, 入力線の長さ

に関わらず削除を行うことで, 小さい領域も一括で削除できるという利点もある.

つづいて, 追跡では, 本追跡手法と既存手法を用いて, 様々な形状をもつ対象や動きに対して実験を行い比較した. 本追跡手法と既存手法の比較を表1に示す.

表1. 葉や花を対象とした4つの追跡手法の比較

項目/手法	手法[10]	手法[5]	手法[6]	本手法	
形状	楕円形	○	○	○	○
	ハート形	—	○	○	—
	カエデ	—	○	○	—
	イチョウ	—	○	○	△
	花	—	○	○	—
動き	平行移動	○	○	○	○
	z軸回転	○	○	○	○
	x軸回転	○	○	○	○
	局所変形	△	○	○	—
速度	追跡速度	4.97 ms	6.22 ms	4.57 ms	2.02 ms

投影対象の形状においては, 楕円近似を用いていることから, 楕円の葉のほか, イチョウ等の楕円に近い形状は追跡できたが, 楕円形とは大きく異なる形状であるハート形の葉を追跡することはできなかった. 一方, 投影対象の動きに対しては, 平行移動, z軸回転, x軸回転する楕円形の葉を追跡することができた(図6). 一方で, 楕円形の葉でも, 図8左のように, 画面上で楕円と判定できない程歪んだものは追跡できない. また, 楕円形でも, 動くことで長軸と短軸が容易に入れ替わるような, 正円に近い形状の葉に対しては, 楕円の長軸と短軸を同定することができず, 追跡できない. 本手法でどの程度の形状の違いまで追跡対象として許容できるか, その条件を特定することは今後の課題である.

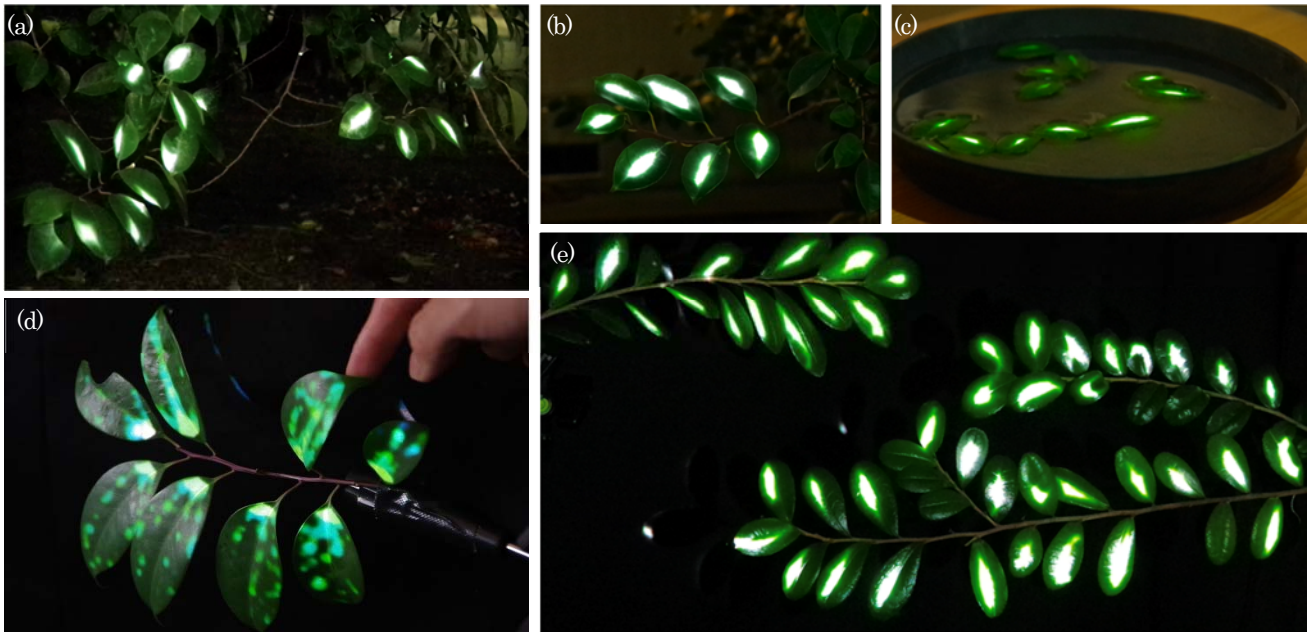


図7. 本手法の投影結果.

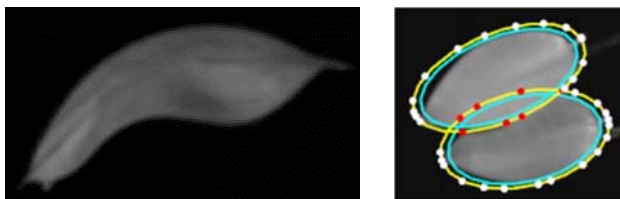


図8. 追跡が困難な程歪んだ楕円形の葉 (左), 楕円による衝突箇所の判定 (右).

速度においては, 1枚の葉に対して, 1フレームあたり2.02 ms (494 fps) で追跡することができ, 既存手法[6]の4.57 ms (219 fps) (表1) と比較すると倍以上高速である. また既存手法[6]では, 60 fps以上で9枚の葉までしか追跡できなかったが, 本手法では, カメラの範囲内で取得できる90枚の葉まで, 1フレーム15.7ms (63fps) で追跡できた. 一方で, ハードウェアのシステム遅延によって, 対象と映像の位置のズレが生じてしまう. 今後より高速化するには, 映像の転送で遅延が少ないプロジェクターの使用など, 高速な機材の利用が求められる. また, 葉と枝の連結情報などを用いるなど, マクロな視点での予測を取り入れることで不自然な動き予測を排除し, 遅延を解消できる可能性がある.

また, 投影領域同士の重なりを考慮し, 楕円によって衝突した箇所を計算した結果, 図8右赤点のように, 正しく判定することができた. そのため, 複数の葉が重なった時も5割程度の頂点であれば重なっていても安定して追跡することができた. 一方で, 追跡対象でない葉や物体と重なる時の衝突を判定していないため, 追跡対象以外の物体と重なった箇所を正しく追跡できない.

エフェクトにおいては, マウスドラッグ等のユーザー入力を適用し, 入力に応じたアニメーションを生成することができた (図9, 図10). また, 屋外の葉 (図7ab) や大量の葉 (図7e),

水上に浮かぶ葉(図7c)に対して投影することができた. さらに, 対象を人が触れて動かしても追跡することができた (図7d).

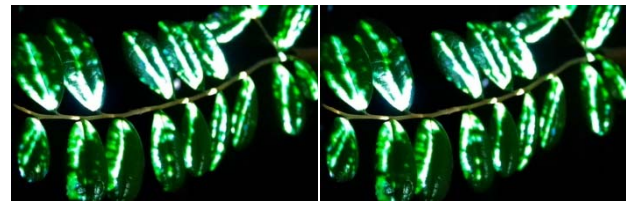


図9. 葉脈モチーフのエフェクトの投影結果.

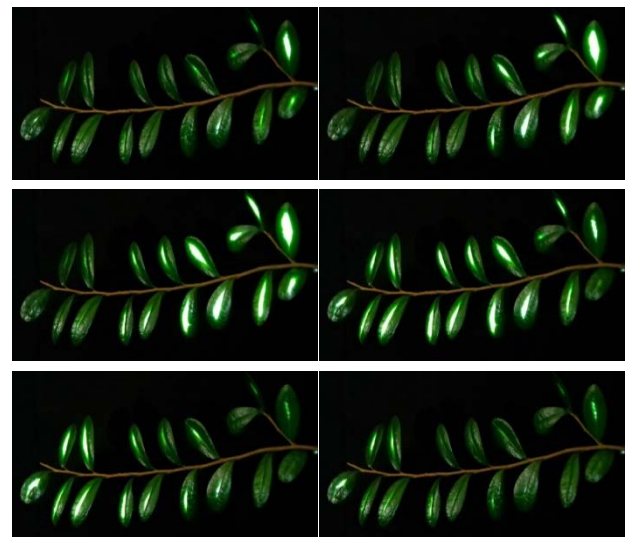


図10. ホタルモチーフのエフェクトの投影結果.

## 5. 結論

本研究では, 大量の楕円形の葉へのDPMを支援するシステムを提案した. 投影領域の選出においては, 多数の葉を簡単に

選択するため、画像処理と簡易なユーザー入力を組み合わせた半自動の選出システムを提案した。追跡においては、投影領域を楕円近似することで高速に追跡できる手法を提案した。さらにエフェクトは、ユーザーが枝情報を入力することで、枝に沿ったアニメーションを作成することができた。

今後の課題として、様々な形状の植物を大量に追跡することがあげられる。また、人とのインタラクションを取り入れたイルミネーションなど、エンターテインメント分野での応用が期待できる。

## 参考文献

- [1] Friedrich van Schoor, Tarek Mawad. 2014. Projections in the Forest
- [2] Gaku Narita, Yoshihiro Watanabe, and Masatoshi Ishikawa. 2015. Dynamic projection mapping onto a deformable object with occlusion based on high-speed tracking of dot marker array. In Proceedings of the 21st ACM Symposium on Virtual Reality Software and Technology (VRST '15), Stephen N. Spencer (Ed.). ACM, 149-152. DOI: <https://doi.org/10.1145/2821592.2821618>.
- [3] Amit H. Bermano, Markus Billeter, Daisuke Iwai, and Anselm Grundhöfer. 2017. Makeup Lamps: Live Augmentation of Human Faces via Projection, 2017. Eurographics. DOI: <https://doi.org/10.1111/cgf.13128>.
- [4] Leo Miyashita, Yoshihiro Watanabe, and Masatoshi Ishikawa. 2018. MIDAS projection: markerless and modelless dynamic projection mapping for material representation. ACM Trans. Graph. 37, 6, Article 196, 12 pages. DOI: <https://doi.org/10.1145/3272127.3275045>.
- [5] Tomoki Sueyoshi and Yuki Morimoto. 2019. Tangible Projection Mapping onto Deformable Moving Thin Plants via Markerless Tracking. In The Adjunct Publication of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19). Association for Computing Machinery, New York, NY, USA, 105–106. DOI: <https://doi.org/10.1145/3332167.3356887>.
- [6] Tomoki Sueyoshi, and Yuki Morimoto. 2021. Interactive Dynamic Projection Mapping onto Thin Plants with Bioluminescent Effect Animations. NICOGRAPH International 2021. DOI: <https://doi.org/10.1109/NICOINT52941.2021.00008>.
- [7] Christa Sommerer, and Laurent Mignonneau. 1993. Interactive Plant Growing. SIGGRAPH Machine Culture.
- [8] Ivan Poupyrev, Philipp Schoessler, Jonas Loh, and Munehiko Sato. 2012. Botanicus Interacticus: interactive plants technology. In ACM SIGGRAPH 2012 Emerging Technologies (SIGGRAPH '12). Association for Computing Machinery, New York, NY, USA, Article 4, 1. DOI: <https://doi.org/10.1145/2343456.2343460>.
- [9] Technical Direction Company, 2016, Sydney Royal Botanic Gardens Vivid Sydney 2016 - 'Synthesis' by Ample Projects
- [10] Tomoki Sueyoshi and Yuki Morimoto. 2018. Automatic

generation of interactive projection mapping for leaves. In SIGGRAPH Asia 2018 Posters (SA '18). Association for Computing Machinery, New York, NY, USA, Article 2, 1–2. DOI:

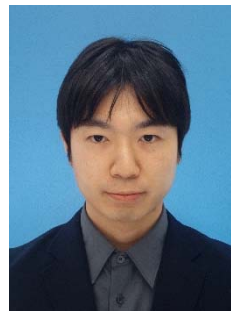
<https://doi.org/10.1145/3283289.3283310>.

[11] Andrew Fitzgibbon and Robert Fisher. 1970. A Buyer's Guide to Conic Fitting. In Proceedings of the 6th British conference on Machine vision (Vol. 2) (BMVC '95). BMVA Press, GBR, 513–522.

[12] Scott Schaefer, Travis McPhail, and Joe Warren. 2006. Image deformation using moving least squares. In ACM SIGGRAPH 2006 Papers (SIGGRAPH '06). Association for Computing Machinery, New York, NY, USA, 533–540. DOI: <https://doi.org/10.1145/1179352.1141920>.

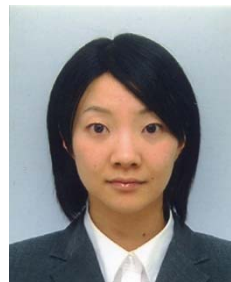
[13] Adam Runions, Martin Fuhrer, Brendan Lane, Pavol Federl, Anne-Gaëlle Rolland-Lagan, and Przemyslaw Prusinkiewicz. 2005. Modeling and visualization of leaf venation patterns. In ACM SIGGRAPH 2005 Papers, Markus Gross (Ed.). ACM, 702-711. DOI: <https://doi.org/10.1145/1073204.1073251>.

## 末吉 知樹



2018年 九州大学芸術工学部画像設計学科卒業。2020年 九州大学芸術工学府芸術工学専攻修士課程修了。同年より九州大学芸術工学府芸術工学専攻博士課程在学中。

## 森本 有紀



2008年九州大学芸術工学府修了, 同年東京大学にて日本学術振興会特別研究員, 2009年より独立行政法人理化学研究所研究員, 2012年より東京電機大学未来科学科講師, 芝浦工業大学助教などを経て, 2016年より九州大学助教。コンピュータグラフィクスに関する研究に従事。情報処理学会, ACM SIGGRAPH 各会員。博士(芸術工学)。