

# 植物画像の三次元復元による芽かき位置に関する研究

石丸 大稀 （指導教員 西村 俊二）

平成 31 年 1 月 25 日

## A study on sprout position by three dimensional reconstruction of plant image

HIROKI ISHIMARU (ACADEMIC ADVISOR SYUNJI NISHIMURA)

**概要：**菊を栽培する際、枝の分かれ目から脇芽と呼ばれる必要のない芽が生えてくる。この芽を取り除く「芽かき」は栄養の分散や枝の混み合いを防ぐための重要な作業の 1 つである一方、手間のかかる作業であり農業従事者には負担となっている。本研究では菊の芽かきを自動化するために、茎と枝分かれ部分の検出及び三次元認識を行った。三次元で認識するために画像を用いて対象までの距離を測定することができる Depth カメラを使用した。入力映像から色による菊の検出を行った後ヒストグラムから茎の位置を概算し、得られた値を元により詳細に茎の探索を行う。2 次元画像から検出した茎を深度データから三次元復元を行うことで認識を行う。復元された茎の 3 次元モデルを複数枚の画像から形状を復元する技術である SfM(Structure from Motion)を用いて復元されたモデルと比較し、考察を行った。

キーワード: 芽かき, Depth カメラ, 三次元復元

## 1. 緒言

日本における農家戸数は昭和 25 年をピークに減少を続けている。近い将来、農業を支えてきた高齢農業者の多くが引退することが見込まれており、農業労働力は脆弱化の進行が懸念されている。一方で、現場では依然として人手に頼る作業や熟練者でなければできない作業が多く、効率かつ安定的な農業経営が相当部分を担う強靱な農業構造の確立を目指す必要がある。このためにロボット技術や情報通信技術 (ICT) を活用した「スマート農業」が注目されており、農作業の省力化・軽労化だけでなく新規就農者の確保や栽培技術力の継承が期待されている。

本研究では菊の栽培に注目した。菊に限らず、植物の中には葉の付け根部分から小さな芽が生えるものがあり、脇芽と呼ばれている。大菊（一輪菊）では脇芽を摘み、一枝に対して一輪だけ残す。脇芽を摘み取る作業は芽かきと呼ばれ、栄養の分散を防ぐほか、枝の混み合いを防ぐ効果がある。このため菊栽培において非常に重要な作業となる。現在は人の手作業によって行われており、手間となっていることから芽かきの自動化を検討した。

芽かきを自動化するためには、脇芽を認識し、その部分の三次元座標を取得する必要がある。一般的に、植物には個体差が存在し、茎の角度や枝の位置などが一意に決まらない。また、工業製品と異なり、より繊細であるからその取扱いには注意を払わねばならない。そういった条件の下での高精度な三次元復元が必要である。

植物の三次元復元に関する研究として、植物を複数方向から撮影した画像から植物の三次元「枝構造」を正確に再現する手法が提案されている [1]。この手法では

深層学習を用い、葉などに隠された枝の存在確率を推定することで三次元復元を行う。しかしこの手法はあくまで推定であり、また複数枚の画像を必要とするため機構が複雑になってしまう。そのためよりシンプルで高精度な三次元復元手法が必要とされている。

本研究の目的は菊における茎と枝分かれ部分の三次元座標取得である。脇芽は枝分かれの部分にできることが多く、この部分と茎を検出することで脇芽のしやすい位置を三次元的に特定することができる。この情報を基にして他のセンサを利用するなどすれば、より高精度に脇芽を検出することができる。

## 2. Depth カメラ

### 2.1 Depth カメラの仕様

茎と枝分かれ部の三次元座標取得に際し、画像と深度データを容易に取得できる Depth カメラを使用した。

使用するのは Intel 社が提供する RealSense D435 である (図 1)。主な仕様を表 1 に示す。

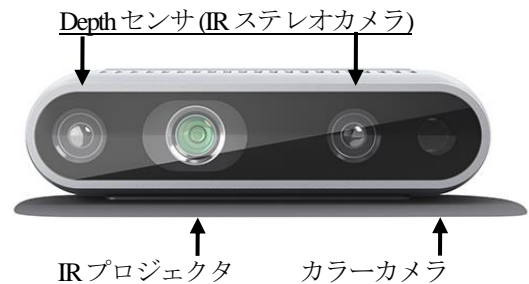


図 1 RealSense D435 (Intel 社ホームページより)

表1 RealSense D435 の主な仕様 (Depth カメラ)

画素数(pixels)	1280 x 720
有効距離(m)	0.2 ~ 10
有効画角(deg)	H:91.2, V:65.5

## 2.2 Depth カメラの技術概要

本研究で使用する RealSense D435 は深度を計算するためにステレオビジョンを使用している。ステレオビジョンは左右の IR カメラ (IR ステレオカメラ) と IR プロジェクタの構成で実装されており、左右の IR カメラから入力された画像に対し、画像上の点を相関させることによって画像内の各ピクセルの深度値を計算している。IR プロジェクタは目に見えない IR の静的パターンを投影することでステレオビジョンにおける対応点探索を補助し、テクスチャの乏しいシーンで深度取得の精度を改善している。図 2 は IR カメラが撮影した画像を拡大したものである。白い点に見えるものが投影された静的パターンである。



図 2 IR カメラ画像



図 3 カラー画像

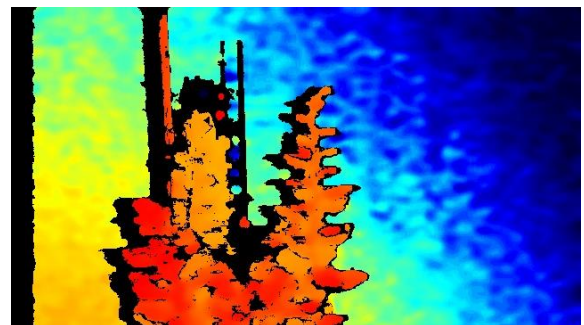


図 4 深度画像

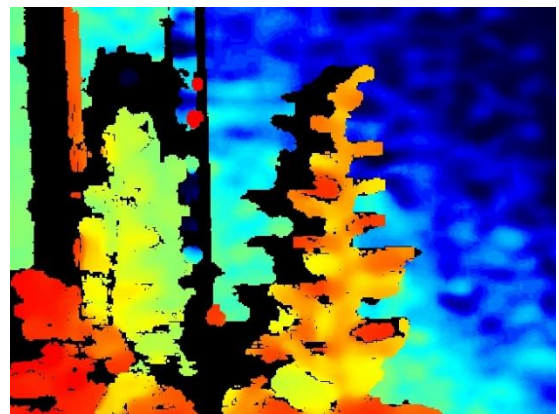


図 5 深度画像 (一致)

## 2.3 画像

Depth カメラから取得できるデータは、カラー画像と各ピクセルの深度値である。図 3、図 4 にはそれぞれカラー画像と深度値に色を付け視覚化したものを示す。尚、カメラに近い物体が赤、遠いものが青で表現されている。

図 3 と図 4 から分かるように、カラーカメラと IR ステレオカメラは画角が異なり、本研究の様にカラー画像のピクセルに対するデータ値を扱おうとする場合、適切な深度値を得ることができない。そこで画角の狭いカラーカメラに IR ステレオカメラの表示領域を一致させる。一致させた結果を図 5 に示す。

## 3 検出方法

### 3.1 三次元復元の流れ

図 6 に三次元復元のアルゴリズムを示す。まず、カラー画像と深度データを紐付けるために位置合わせを行う。次にカラー画像から菊を検出し、背景と菊とに 2 値化する。その後 2 値画像から茎の位置を概算し、画像下部からウィンドウを上方へスライドさせる方式で探索を行う。探索と同時に枝の分かれ目部分の検出も行い、それぞれ画像上の座標を記録しておく。記録された画像上の座標と深度データから 3 次元空間に展開し、三次元復元を行う。

また、図7は撮影環境である。三脚にカメラを固定し撮影を行った。

の背景が白い壁となるような位置で撮影を行った。

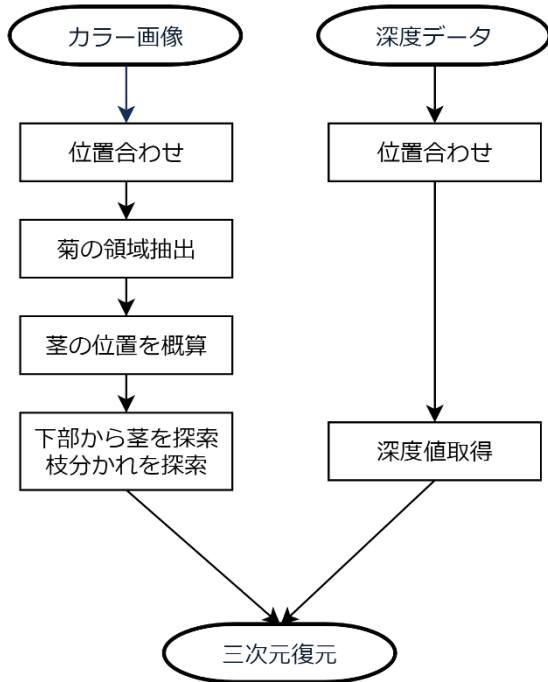


図6 三次元復元の流れ



(a) カラー画像 (b) マスク画像

図8 分類器への入力画像

実行後、色相(Hue)・彩度(Saturation)・明度(Value)の解析結果をそれぞれ図9、図10、図11に示す。また解析結果を元にカラー画像を2値化した一例を図12に示す。今回の環境下では入力画像中央が緑色に変色が見られたが、一方で図12では問題なく分類できていることがわかる。



図7 撮影環境

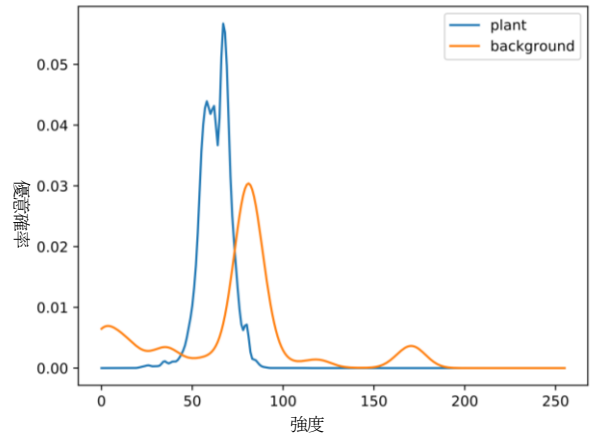


図9 色相(Hue)

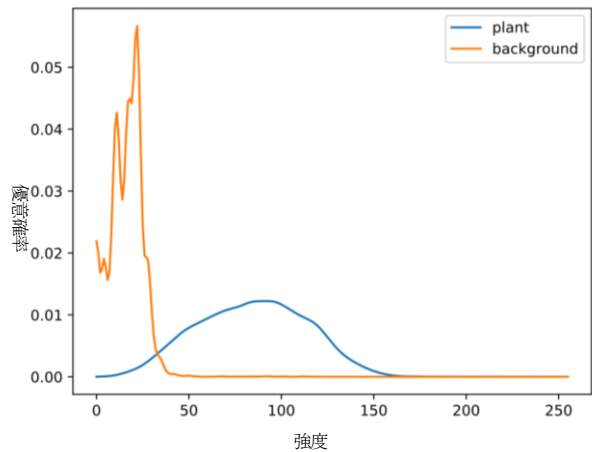


図10 彩度(Saturation)

### 3.2 分類(2値化)

Depth カメラから入力されたカラー画像を菊領域と背景領域に分類する。分類には PlantCV の単純ベイズ分類器を用い、色情報から分類を行った。PlantCVは OpenCV や Numpy, Matplotlib といったオープンソースソフトウェアによって構築された植物向けの画像処理パッケージである [2].

Depth カメラから取得した 4 枚の静止画を菊領域と背景領域に 2 値化し、元のカラー画像とセットで分類機への入力とした。図8では入力した画像の一部を示している。尚、今回は屋内・蛍光灯下での撮影であり、また菊

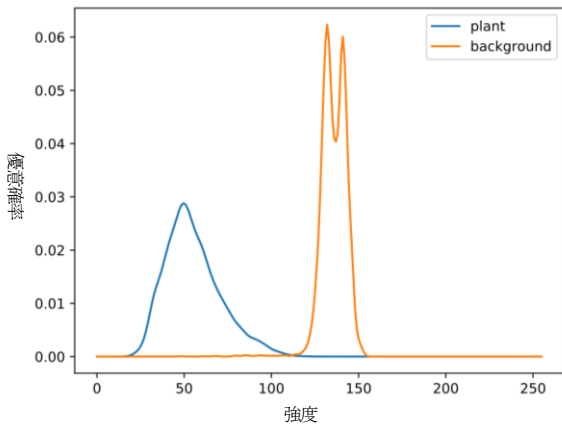
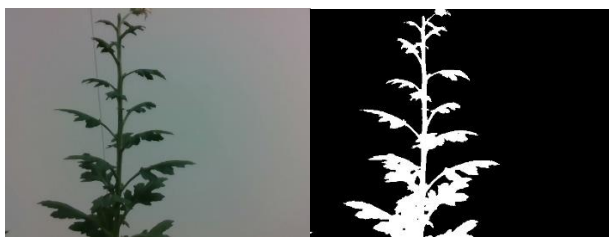


図 11 明度(Value)



図 14 ヒストグラム

また、この手法では図 15 の様に Depth カメラに対して菊(茎)が斜めに入力された場合はヒストグラムが図 16 となり、茎の横方向の位置を推定することが難しくなる。



(a) 入力画像 (b) 出力画像

図 12 分類結果

### 3.3 ヒストグラム

3.1 によって 2 値化された画像において縦方向の画素値の度数分布を求める。茎が垂直に近い状態で自立していると仮定すると、画像縦方向の画素値の度数分布を求めることで茎の横方向の位置を推定することができる、入力画像 (図 13) のヒストグラムを図 14 に示す。



図 15 入力画像



図 13 入力画像

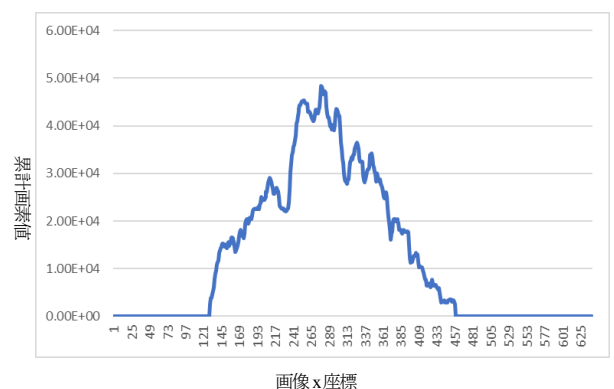


図 16 ヒストグラム

### 3.4 探索

#### 3.4.1 茎判定アルゴリズム

3.3 で概算した画像横方向の茎の位置を起点に茎の探索を行う。画像下部からウィンドウを上方向にスライドさせながら茎を辿る。図 17 では 2 値画像上にウィンド



ウを緑枠で表し、ウィンドウ内の菊領域を青で示している。

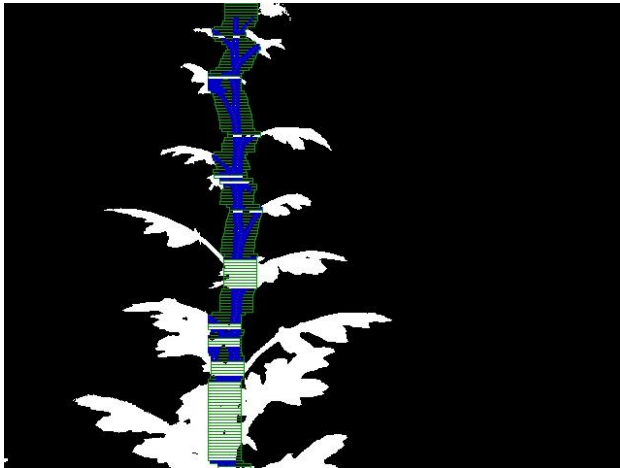


図 17 実行結果

一段上のウィンドウの横座標を決定する流れについて述べる。

ウィンドウ内の殆どが菊領域であるとき (例えば茎と Depth カメラの間に葉が存在している場合), ウィンドウ内で茎を判別することができない. この場合は横座標の更新を行わず, さらに一段上のウィンドウへ推移する. 図 17 ではウィンドウ内に菊領域があるにもかかわらず青でなく白で描画されている部分があるがこの部分が茎を判別できないと判定されたウィンドウである.

また, ウィンドウ内の 2 値画像データに対してグルーピングを行う. 背景領域と菊領域に 2 値化されたデータ内で菊領域をグループごとにラベリングする, 例えば図 18 の様に, ウィンドウ内に茎と枝が存在する場合, 2 値データは図 19 となるが区別できず扱いづらい. 図 20 の様に変換し, 菊領域を区別することで後の茎と枝の判定時に利用する. 尚以下の図は簡略化している.



図 18 ウィンドウ内の簡略図

1	1	1	1	0	0	0	0	0	1	1	1	1	0
0	1	1	1	1	0	0	0	0	1	1	1	1	0
0	0	1	1	1	1	0	0	0	1	1	1	1	0

図 19 ウィンドウ内の 2 値データ

1	1	1	1	0	0	0	0	0	2	2	2	2	0
0	1	1	1	1	0	0	0	0	2	2	2	2	0
0	0	1	1	1	1	0	0	0	2	2	2	2	0

図 20 ラベリング結果

次のウィンドウ座標に関して, ウィンドウ内の菊領域 (グループごと) に重心を求め, グループが 1 つならその座標, 複数ならその平均値の座標に更新する. この際, 各グループの面積も計算し, 大きなものは除外する. また, 3.3 でも述べたように, このアルゴリズムでは垂直に近い状態で自立していると仮定している. 垂れ下がった枝葉による誤判定を防ぐために, 概算された茎の位置 (ウィンドウのスタート位置) から移動できる上限を設けている.

### 3.4.2 枝判定アルゴリズム

下方から探索を進めると, 重心の数 (菊領域のグループ数) が 1 から 2 に増える箇所がある. この部分を枝の分かれ目と判定する. 図 21 では枝の分かれ目付近を簡略化し示している. なお緑枠でウィンドウ, 赤点で重心を表している.

さらにこの 2 点中で枝と茎を判定する. 枝の分かれ目と判定された部分の重心 2 点 (図 22 中の A と B) の横方向座標と分かれ目以下に存在する数個の横方向重心座標の絶対差を累計し, 値が小さい方を茎と判定する. このとき差を計算する対象となる重心は, 以下のすべての条件が成り立つものとし, A, B 直下のウィンドウから下方向へ探索する. 成立しないウィンドウがあれば以降の探索は行わない. ①重心が計算されている (菊領域が大きすぎない) ②重心の数が 1 つである 図 22 では重心 A との絶対差を緑線, 重心 B との絶対差を青線で示している. 5 本分の絶対差を合わせたとき重心 A より重心 B の値が小さくなるため茎は B 点であると判定する.

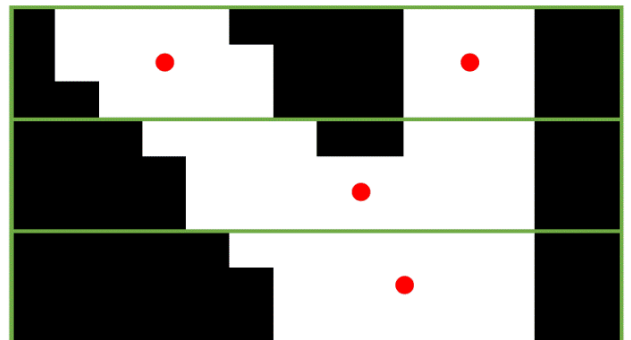


図 21 枝分かれ付近の簡略図

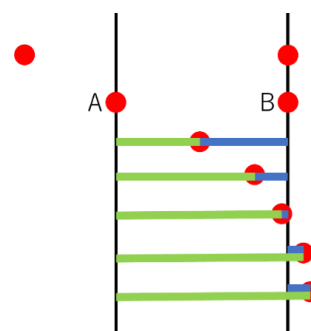


図 22 枝判定手法

図 23 では実際に入力された画像上に検出された枝の分かれ目を赤丸で重ね、一部切り出した上で示している。検出に成功した点では茎と枝の判定もできている。一方で茎を辿りそこから分かれている枝を検出しているため葉などの茎から離れた要素を考慮していない。そのため茎に葉が重なる部分から伸びる枝の分かれ目は検出できていない。

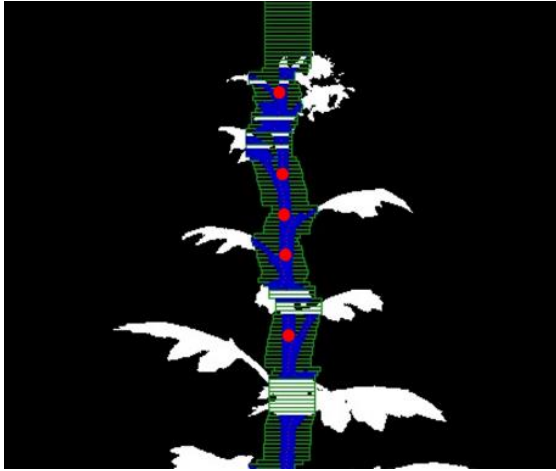


図 23 出力結果

### 3.5 空間への展開

画像上の座標と、対応する深度値、カメラの内部パラメータを使用することで三次元の点に写像する。

実際に取得したパラメータを表 2、値を計算する式を (3.5) に示す。WEB 上に公開されているプログラムを一部編集し実装した [3]。なお式中の  $c_n$  は表 2 の  $coeffs[n]$  に相当し、変換前の画像上の  $x$  座標,  $y$  座標, 深度値をそれぞれ  $X, Y, Z$  で表している。求める座標は

$$\left. \begin{aligned} x &= \frac{X - ppx}{fx} \\ y &= \frac{Y - ppy}{fy} \\ r2 &= x^2 + y^2 \\ f &= 1 + c_0 \cdot r2 + c_1 \cdot r2^2 + c_4 \cdot r2^3 \\ ux &= x \cdot f + 2 \cdot c_2 \cdot x \cdot y + c_3 \cdot (r2 + 2 \cdot x^2) \\ uy &= y \cdot f + 2 \cdot c_3 \cdot x \cdot y + c_2 \cdot (r2 + 2 \cdot y^2) \end{aligned} \right\} (3.5)$$

として

$$(Z \cdot ux, Z \cdot uy, Z)$$

である。

また、この式により三次元座標に変換される例を表 3 に示す。すべての重心に対してこの変換を行い、空間へ展開する。図 24 には得られた 3D モデルを示している。小さな点が検出されたすべての重心で、大きな点はその重心の中で枝の分かれ目がかつ茎であると判定された点

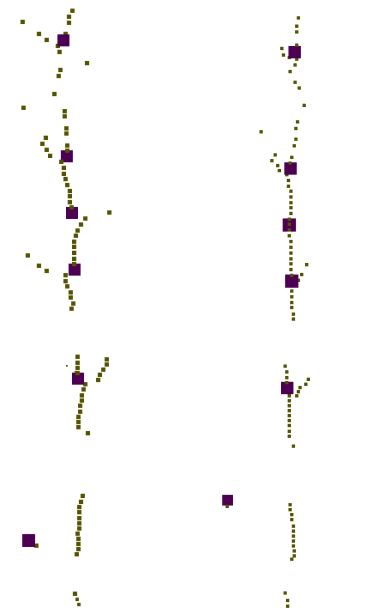
である。画像では 6 つの枝の分かれ目が検出されていることがわかるが一番下に位置するものは誤って検出されたものである。葉が茎に重なりウィンドウ内に微細な領域が現れ、それを茎と判定してしまったことが原因と考えられる。

表 2 内部パラメータ

width (幅)	640
height (高さ)	360
ppx (画像上の主点 x 座標)	318.817
ppy (画像上の主点 y 座標)	178.692
fx (焦点距離)	320.892
fy (焦点距離)	320.892
model (歪みモデル)	Brown Conrady
coeffs (係数)	[0,0,0,0]

表 3 変換例

	変換前	変換後
x	318	-0.008278
Y	114	-0.655202
Z	3.25	3.25



(a) 正面 (b) 右側面

図 24 3D モデル

### 3.6 誤検出

本研究で用いた Depth カメラを使用する手法では誤検出も多々見られた。それぞれの誤検出についてその原因と改善策について考察した。

はじめに、図 25, 26 の様に茎が葉によって隠されている場合と茎が見えていてもその背景に同系色の葉や枝が存在する場合、その部分の茎の位置が取得できていない。手法では入力された画像内を更に分割するウィンド

ウを用いてウィンドウごとに処理を行った。そのためウィンドウ内部に特徴がないと検出することができない。菊領域と背景領域を色情報により分割しているため、色が被る部分の特徴を失ってしまった。人間は普通茎が隠れていてもその付近の情報から推測することができる。そのため検出された他の茎の情報を利用して茎の位置を推定することができる。また、同様に上記の条件下では枝の分かれ目も検出できなかった。茎を辿りながら分かれ目を検出する手法であるため茎が検出できていなければ分かれ目も検出できない。これを解決するためにはDepthカメラの移動を制御し、各フレームから検出された枝の分かれ目を保存することが有効である。Depthカメラの自己位置と空間に展開された検出点を紐づけて処理することができれば更に検出数が増えるであろう。



(a) カラー画像 (b) 抽出結果  
図 25 茎の前に葉がある場合



(a) カラー画像 (b) 抽出結果  
図 26 茎の後に葉がある場合

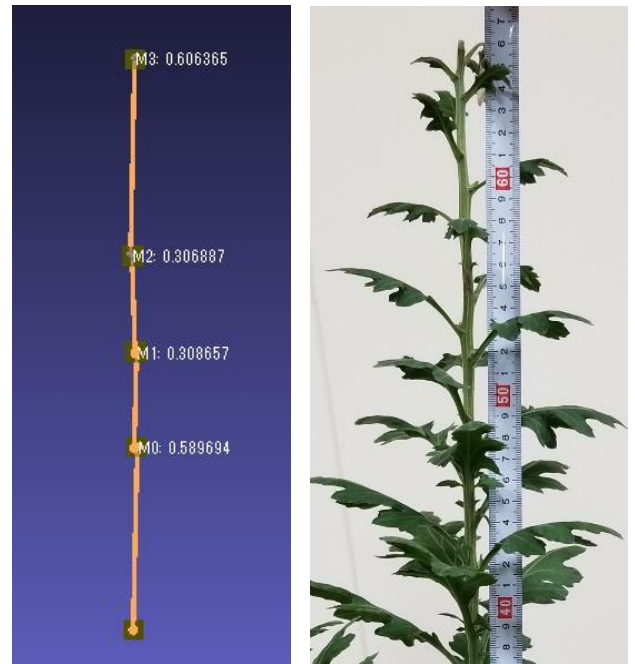
## 4. 実験

### 4.1 長さの比較

まず、図 27(a)に示すように得られた 3D モデル (点群) から枝の分かれ目間の距離を測定する。そのうちの 1 区間を基準としたときの他区間の比率を求める。次に実際の菊でも同様に距離を測定する。図 27(b)のように茎に沿わせてメジャーを設置し正面から撮影を行った後、画像から測定を行った。同様に 3D モデルと同じ 1 区間を基準として他区間の比率を求めた。この手法により得られたデータを表 4 に示す。

表 4 より、一番下の重心から一番上の重心までの長さは、3D モデルで 1.811603、画像で 100 となる。この数値を基準に他区間の比率を求めた。表 5 に結果を示す。また求められた比率のうち、3D モデルと画像の各区間の誤差と測定されたデータを用い比率から長さへ変換した。得られた結果を表 6 に示す。平均して、およそ 1.25mm の誤差があった。緒言で述べたように、後の脇芽検出を

補助するデータとして 3D モデルを使用する場合、1.25mm の誤差が与える影響は小さいものであると考える。



(a) 3Dモデルから (b) 画像から  
図 27 測定方法

表 4 測定データ

	3D モデル	画像(mm)
区間 1	0.589694	31
区間 2	0.308657	18
区間 3	0.306887	16
区間 4	0.606365	35

表 5 比率

	3D モデル	画像	誤差
区間 1	0.3255	0.31	0.0155
区間 2	0.1704	0.18	0.0096
区間 3	0.1694	0.16	0.0094
区間 4	0.3347	0.35	0.0153

表 6 誤差

区間 1	1.55 mm
区間 2	0.96 mm
区間 3	0.94 mm
区間 4	1.53 mm
平均	1.25 mm

### 4.2 形状比較

検出された枝の分かれ目の 3D モデルと複数枚の画像から生成された 3D モデルを比較した。後者の手法は、異なる角度から撮影された複数枚の画像を用いオブジェクトを三次元復元するもので、SfM (Structure from Motion)

という技術である。この比較ではオープンソースの SfM プログラムである Regard3D [4]を使用して三次元復元を行った。スマートフォンで撮影した 25 枚の画像を最高設定で実行したところ 1 枚に平均で 36838 個の特徴点が検出され、三角測量方式で 73507 個からなる点群が得られた。この点群と重心の 3D モデルを比較した。図 28 に得られた点群と枝の分かれ目 (図中の赤丸) の 3D モデルを重ねて 2 つの角度から示している。また、この 2 つの点群を重ねる際に互いの位置、全体のスケール、角度を調整する必要があるがこの作業は手動で行った。2 モデル間の誤差は重ねる際の調整に大きく依存するため数量的な評価は行っていない。

この比較では検出された枝の分かれ目の位置に大きな誤りは見られなかった。

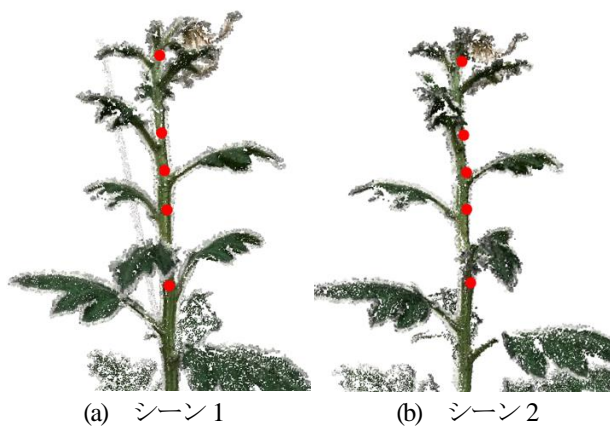


図 28 3D モデルでの比較

この SfM の手法では 25 枚の処理に 7 分以上かかっている。正確性には優れるが複数枚の画像を必要とする点や実行時間から大量の菊を処理する必要がある現場では実用性にかける。一方で Depth カメラを使用した枝の分かれ目の検出法ではおよそ 1 秒で処理が完了するため、リアルタイム性に優れている。

## 5. 結言

本研究では菊の芽かきを自動化するために茎と枝の分かれ目の認識及び三次元復元を目的とした。Depth カメラから入力されるカラー画像から検出を行い、検出された点に対応する深度値を取得した後、得られたデータを空間に展開し、実際に菊と比較を行った。

三次元に復元された点間の長さの誤差はおよそ 1.25mm であった。また、SfM を用いて復元された菊と比較を行い、形状の類似を確認した。

SfM は菊を高精度に三次元復元することができ、得られるモデルの信頼度も高いが、その分実行に時間を要す。複数枚の画像からマッチング処理を行っているのが当然であるが、本研究の目的においては必要ない処理も含まれている。提案した手法はおよそ 1 秒で 1 フレームの処理を完了でき、多くの菊が栽培される農業現場での芽か

きを自動化するための一次処理として効果があると考えられる。

## 謝辞

本研究に際して、様々なご指導を頂きました西村俊仁准講師に深謝いたします。また、この研究の機会をくださった情報工学科の先生方、そして多くの知識やご指摘を下さいました同研究室の先輩・同期の皆様にも厚く御礼申し上げます。

## 参考文献

- [1] T. Isokane, F. Okura, A. Ibe, Y. Matsushita, Y. Yagi, “Probabilistic Plant Modeling via Multi-View Image-to-Image Translation,” 2018.
- [2] D. D. P. S. Center, “PlantCV,” 2014. [オンライン]. Available: <https://plantcv.danforthcenter.org/>.
- [3] V. PTERNEAS, “Intel RealSense D415/435: Coordinate Mapping in C# | LightBuzz,” LIGHTBUZZ INC, 10 10 2018. [オンライン]. Available: <https://lightbuzz.com/intel-realsense-coordinate-mapping/>.
- [4] R. Hiestand, “Home,” 2015. [オンライン]. Available: <http://www.regard3d.org/>.