

# 建築画像の消失点検出手法の開発と それに基づく3次元建築モデルの再構成手法

## DEVELOPMENT OF A METHOD FOR DETECTING VANISHING POINTS OF AN ARCHITECTURAL IMAGE AND RECONSTRUCTING A 3D ARCHITECTURAL MODEL

山中 俊介\*, 加藤 直樹\*\*, 藤澤 克樹\*\*\*

*Syunsuke YAMANAKA, Naoki KATOH and Katsuki FUJISAWA*

We present a method for detecting vanishing points of an architectural image, which consists of mainly parallel and orthogonal lines, and reconstructing a 3D architectural model. For this, we implement an algorithm for line detection from an architectural image, based on the Hough Transform employing the plane sweep technique and test its efficiency and ability of the line detection from digital images. We then apply it to architectural images in order to see the practical usefulness of the proposed method.

**Keywords :** *Computer Vision, Reconstruction, Vanishing Point, 3D model, Camera Calibration, Binary Edge Image, Line Detection, Hough Transform, Plane Sweep*  
コンピュータ・ビジョン, 再構成, 消失点, 3次元モデル, カメラ・キャリブレーション,  
2値化エッジ画像, 直線検出, ハフ変換, 平面走査法

### 1 序

#### 1.1 研究の背景

コンピュータ・グラフィックス(以下CG)によって建築物を仮想空間上で3次元的に扱う要求の増加とあいまって、建築分野において、CGを用いた建築表現の必要性が増してきている[1]。を仮想空間上で3次元的に扱うことができるということが大きい。

例えば都市景観のシミュレーション等において、既存の建築物群の3次元モデルを作ってそれらを忠実に表現することは、都市のスケール感を把握したり、都市の中の場所を特定するためには非常に有効である[2]。しかし実際のところ、3次元モデリングは一般に大変な時間と労力を伴う作業であるので、それらを緻密にモデリングすることはあまり現実的ではない。したがって特に単純な建築物などに対しては、美しさ・リアルさを損なうことなく、より速くより簡単に3次元モデルを作ることが望まれる。

そこで近年、2次元画像である写真から直接3次元モデルを作り出すという技術が注目されている[3][4]。この技術は再構成(reconstruction)とよばれ、コンピュータ・ビジョン<sup>1</sup>(以下CV)の分野

における最も重要なテーマのひとつである。

#### 1.2 既往の研究

3次元モデルの再構成において、対象となるオブジェクトが建築物のように線分や面の平行性や垂直性の関係を豊富に含んでいる場合、3次元から2次元への写像によって生じる見かけ上のオブジェクトの形の変化から、面の傾きや奥行きなどの3次元情報を得るという手法が有効であり、実際にこれに即した方法で建築画像を扱ったいくつかの研究がなされており[3][4]、3次元再構成のためのソフトウェアも開発されている<sup>2</sup>。

このとき問題となるのは、2次元画像から対象オブジェクトを構成する線分や面などの成分をいかにして抽出するかということである。しかしながら[3][4]等による従来の研究や市販のソフトウェアでは、2次元画像における建築物の構成成分、すなわち点・エッジ・面などを画像から認識し、選択しているのはユーザーであり、コンピュータはその選択誤差を最小にするために補助的に用いられているに過ぎない。これはCVとはいうものの、「人間の目」に頼る部分が大きく、コンピュータが画像を認識しているとはとても言い難い。

しかし建築画像に限らず、コンピュータによって画像の中の対象

\* 戸田建設 工修

\*\* 京都大学大学院工学研究科建築学専攻 教授・工博

\*\*\* 京都大学大学院工学研究科建築学専攻 助手・博士(理)

Toda Corporation, M. Eng.

Prof., Dept. of Architecture and Architectural Systems, Kyoto Univ.,  
Dr. Eng.

Instructor, Dept. of Architecture and Architectural Systems, Kyoto Univ.,  
Dr. Sc.

を抽出するという事は困難な問題である。その理由としては、画像に含まれるノイズやオクルージョン<sup>3</sup>によって、抽出されるべき情報の精度が大きく左右されてしまうということが挙げられる。その中で Hough 変換とよばれる図形の検出アルゴリズムは、それらの障害に対しても安定した処理を行なうことができるということで知られており、特に直線検出用 Hough 変換については多数の研究が行なわれている [5]。しかし実際には従来の研究で考案されたアルゴリズムでは効率面で問題があるということが指摘されている [6][7][8]。

### 1.3 研究の目的

1.2 節で述べた既往の研究の問題点を考慮して、本研究では 2 次元建築画像における対象建築物の構成成分の検出に適した手法を実装する。その際に、一般に広く用いられている直線検出手法との比較実験を行ない、その有効性を検証する。そして最終的に、提案手法によって得られた情報から、より精度が高く、より美しい建築物の 3 次元モデルを再構成することを目的とする。

### 1.4 研究の特徴

本研究の特徴としては、以下の点が挙げられる。

- デジタル画像からの直線検出のために、平面走査法を用いた Hough 変換 [6][7] を実装・適用する。
- 建築画像の消失点検出にも、本研究で実装した Hough 変換を利用する。
- 上記の手法の実装において、建築物を対象とした画像の特徴を利用して、3 次元建築モデルの再構成手法を開発する。

また、本論文で扱う建築画像について次の条件を与える。

- (1) 対象建築物は、水平面に対して 1 面が平行な直方体成分から成り立つ。すなわち建築物を構成する線分や面が、平行性や直交性の関係を豊富に含む。
- (2) 地平線がほぼ水平方向である。すなわち写真撮影時に、カメラの 1 辺を地平線と平行にする。
- (3) 消失点を 3 つ含む (3 点透視)。すなわち写真撮影時に、カメラの光軸を水平線に対して傾ける。

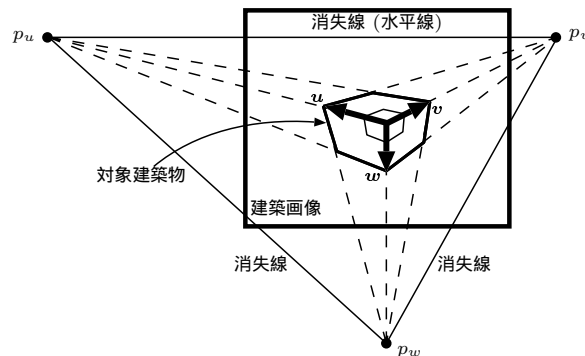


図 1: 建築画像と消失点

このとき建築画像は図 1 のように、水平線上の 2 つの消失点  $p_u, p_v$  と、垂直線上の 1 つの消失点  $p_w$  が存在し、画像平面上でそれらに収束する線分に対応する対象建築物の構成成分の方向ベクトル  $u, v, w$  は互いに直交する。

### 1.5 3 次元建築モデル再構成のプロセス

本論文では、2 次元建築画像を解析することによって、透視図として歪んだ立面のテクスチャを復元し、それらを適切な位置関係に配置して 3 次元建築モデルを再構成する (図 2)。2 次元建築画像の解析においては、原画像をエッジ検出して 2 値化した 2 値化エッジ画像を扱い<sup>4</sup>、直線成分および消失点を検出し、それらの情報をもとにカメラ・キャリブレーション (5.1 節参照) を行ない、立面の復元に用いる。

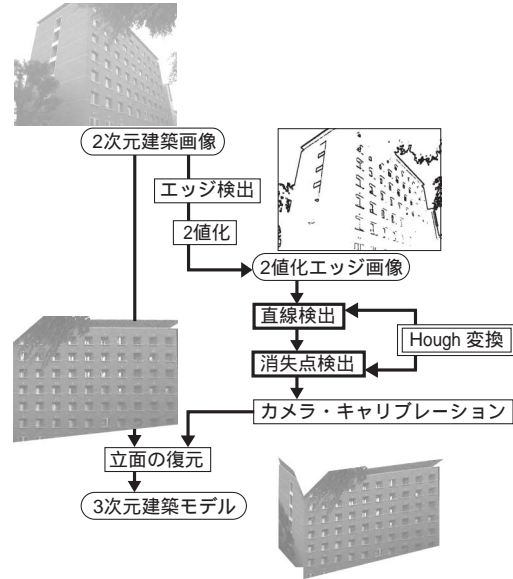


図 2: 3 次元建築モデル再構成のプロセス

## 2 建築画像と消失点

ピンホール・カメラ・モデル<sup>5</sup>による 3 次元シーンの 2 次元画像平面への投影は透視変換<sup>6</sup>で表される (図 3)。したがって、カメラ座標の点  $P$  が画像平面上の点  $p$  に対応するとき、 $P$  を  $p$  の同次座標として、画像平面を 2 次元射影空間<sup>6</sup>とみなすことができる。2 次元画像から 3 次元空間のオブジェクトを再構成するためには、それらの射影幾何学的関係に基づいて、画像平面からもとのオブジェクトの情報を得ることが必要である。そこで、画像平面上の消失点の性質を利用する。

1 つの消失点に収束する画像平面上の線分は、3 次元空間の平行な直線に対応している。さらに、2 次元射影空間としての画像平面上の消失点の同次座標が得られると、その消失点に収束する線分に対応する 3 次元空間の直線の方向ベクトルが分かる [9]。

消失点を得るためには、3 次元空間の平行線に対応する線分が画像平面上に存在している必要があるが、建築画像について考えた場合、建築物の外観には平行な線分や平面が豊富に含まれているので、それを利用して消失点を検出できると考えられる。さらに本論文では、画像の対象建築物が主として直方体成分から成り立つという条件を与えているため、画像平面から 3 つの消失点を得られたとき、それらに収束する線分に対応する 3 次元空間の直線の組は互いに直交して

いと仮定することができる。すなわち図 3 において、カメラのレンズ中心を原点  $O$  としたカメラ座標の  $XY$  平面に平行な画像平面上の 3 つの消失点  $p_u, p_v, p_w$  から同次座標ベクトル  $u, v, w$  が得られ、それらは対象建築物の互いに直交する 3 辺の方向ベクトルに対応するというのである。したがって、次の関係式を得る。

$$(u, v) = 0, \quad (v, w) = 0, \quad (w, u) = 0 \quad (1)$$

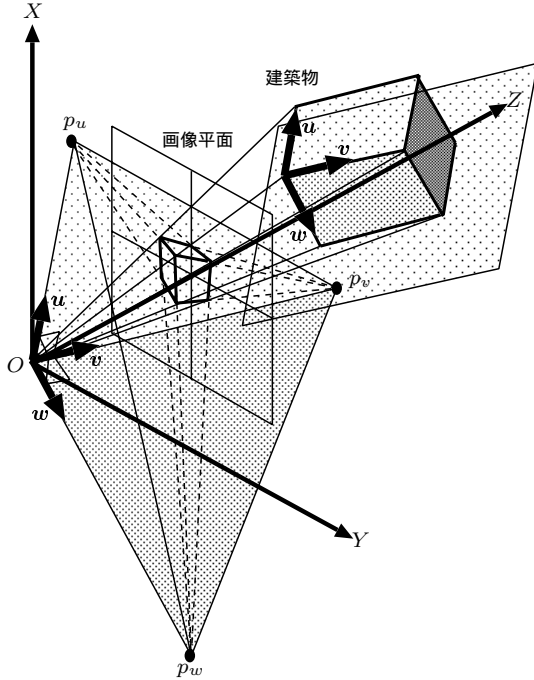


図 3: 直交性の 3 次元的解釈

ここで、 $(u, v)$  はベクトル  $u, v$  の内積を表す。

したがって、消失点  $p_u, p_v, p_w$  が高精度で推定できると、対象建築物の構成成分の方向ベクトルが正しく得られ、その 3 次元モデルが再構成できる。

### 3 建築画像からの直線検出

#### 3.1 Hough 変換

建築画像から消失点を検出するためには、まずその建築物を構成する直線成分を検出しなければならない。そこで直線検出用の Hough 変換を用いる。Hough 変換の概要を以下に述べる。

##### 3.1.1 Hough 変換の原理

画像平面上的点  $p = (x_p, y_p)$  を通る任意の直線の傾きを  $a$ 、 $y$  切片を  $b$  とすると次の関係が成り立つ。

$$y_p = ax_p + b \iff b = -x_p a + y_p \quad (2)$$

Hough 変換ではこれを、 $x-y$  画像平面上の点  $p$  から  $a-b$  パラメータ平面上の、傾き  $-x_p$ 、 $b$  切片  $y_p$  の直線  $L_p$  への変換とみなす。このとき  $L_p$  上の点  $(a, b)$  の集合は、点  $p$  を通る直線群のパラメータに対応しており、この  $L_p$  を、点  $p$  の特徴直線とよぶ。したがって、点

$p$  と点  $q$  を通る直線のパラメータは、それぞれの特徴直線  $L_p$  と  $L_q$  の交点の座標に対応することになる (図 4)。この性質を利用すると、画像平面において直線成分を求める問題は、パラメータ平面において特徴直線が多数交差する点を求める問題に変換することができる。

一般に特徴直線が多数交差する点を求める代表的な手法として、パラメータ平面を格子状に離散化し、特徴直線が通過する格子を投票箱として投票を行ない、所与の値 (しきい値) を超える票を得た格子に対応するパラメータを検出するという投票法がある。Hough 変換の安定性は、このような「多数決原理」に基づいている [5]。すなわち、画像平面上で多くの点から成り立つ直線ほど検出されやすいということである。この場合、格子のサイズを小さくすると直線検出の能力は向上するが、計算時間は増加する。このため、適切な離散化が必要である。

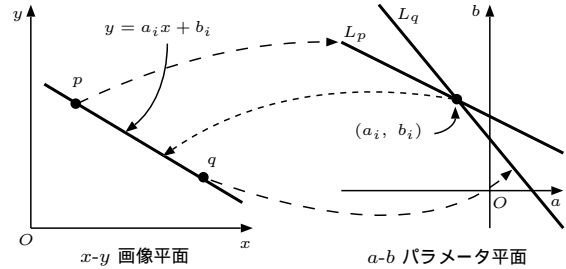


図 4:  $a-b$  パラメータ平面を用いた Hough 変換

##### 3.1.2 2 値化エッジ画像に対する Hough 変換のモデル

2 値化エッジ画像のピクセル中心を整数格子点とした座標において、塗りつぶされたピクセルに対応する格子点をエッジ点とよぶことにする。このとき直線  $L$  のデジタル成分 ( $D(L)$  と記す) とは、その直線からの水平距離と垂直距離の小さい方の値が 0.5 以下であるようなエッジ点の集合である [6]。これを逆に考えると、あるエッジ点をデジタル成分とする直線は、そのエッジ点からの水平距離あるいは垂直距離が 0.5 の範囲内にある直線であるといえる。本論文ではこのことを考慮して、次のように問題定義をする。

2 値化エッジ画像のエッジ点の集合  $G$  と、あるしきい値  $t$  が与えられたとき、直線  $L$  のデジタル成分  $D(L)$  と  $G$  の共通集合の要素数が  $t$  以上であるような  $L$  をすべて求める

直線  $y = ax + b$  に関していえば、傾き  $a$  について次の関係が成り立つ。

$$|a| \leq 1 \iff (\text{水平距離}) \leq (\text{垂直距離})$$

$$|a| > 1 \iff (\text{水平距離}) > (\text{垂直距離})$$

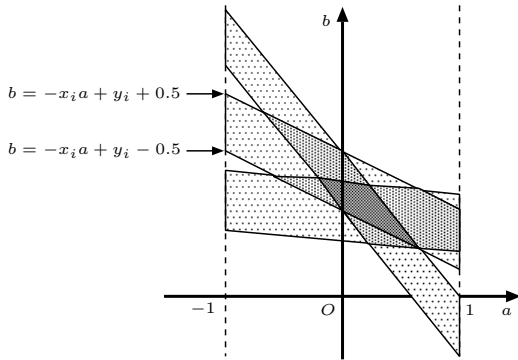
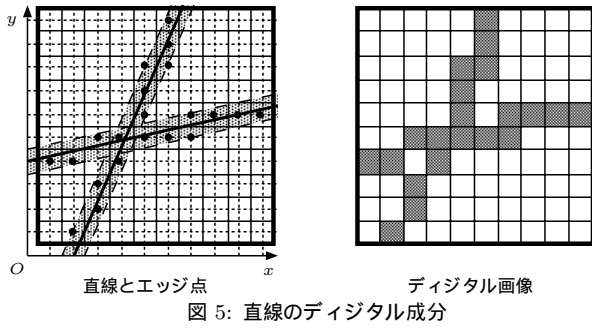
したがって  $|a| \leq 1$  とすると、エッジ点  $p_i = (x_i, y_i)$  が直線  $y = ax + b$  のデジタル成分に含まれるとき、次の式が成り立つ。

$$-0.5 \leq y_i - ax_i - b \leq 0.5$$

$$\iff -x_i a + y_i - 0.5 \leq b \leq -x_i a + y_i + 0.5 \quad (3)$$

式 (3) は、 $a-b$  パラメータ平面において、傾きが  $-x_i$  の 2 本の平行線で囲まれた、 $b$  軸方向の幅が 1 の帯領域に対応している。この帯

領域は先述の特徴直線を拡張したものであり、直線検出の問題は多数の帯領域が重なっている部分を求める問題に帰着される。 $|a| > 1$  の場合は  $x$  と  $y$  を入れ替えることによって同様に処理できる。



### 3.2 投票法を用いた Hough 変換の問題点

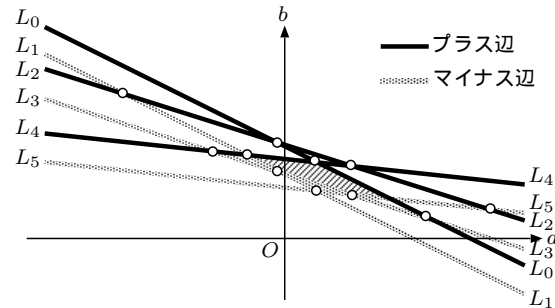
投票法によると、格子状に離散化したパラメータ平面に対して、帯領域に含まれる格子を投票箱として投票していくのだが、画像サイズを  $N \times N$  とすると、画像に表現する直線の傾きは  $O(N^2)$  通り、 $y$  切片は  $O(N^4)$  通り存在することが知られており、それら全てを検出できるようにするためにはパラメータ平面は  $O(N^6)$  個の格子に離散化する必要がある [7] (この方法を 3.4 節では、 $a$ - $b$  Vote.1 と記す)。これは、画像に含まれるエッジ点の数に関わらず、画像のサイズ  $N$  が大きくなるほど計算量が  $O(N^6)$  のオーダーで増えるということを示している。

しかし直線検出を行なう画像は、特に建築画像についていえば画像サイズに対してエッジ点の数が少ない「疎」な画像であるので、このような離散化は非効率であるといえる。したがって通常用いられている投票法では、厳密な離散化を行わずに  $a$ - $b$  平面を適当なサイズの幅で格子状に区切って  $O(N^6)$  よりも少ない投票箱を用意するが、そうすると検出できないパラメータが存在するという問題が生じる [8]。(この方法を 3.4 節では、 $a$ - $b$  Vote.2 と記す。) また直線検出は、パラメータ平面において帯領域が局所的に最も重なっている領域に対応する極大直線成分のみを検出すれば十分であるが [6]、投票法の場合は投票後の各格子の得票数が極大であるかどうかの判定が困難であり、しきい値のみで検出を判定すると極大成分以外のパラメータまで検出されてしまうおそれがある [7][8]。

### 3.3 平面走査法を用いた Hough 変換

上述の投票法による直線検出アルゴリズムの欠点を克服するために、Asano, Katoh and Tokuyama[6] は平面走査法を用いたアルゴリズムを開発した。この手法の計算量は画像サイズに依存せず、エッジ点の数に比例する計算量で済むという特徴を有する。しかし、このアルゴリズムの実装や計算機実験による有効性の検証は現在のところ報告されていない。本論文では、建築画像の 2 値化エッジ画像においては比較的エッジ点が少ないこと、このアルゴリズムの直線検出能力が理論的に保証されていることから、今回初めて実装を行なった。以下ではこの平面走査法を用いた直線検出アルゴリズムの概略を紹介する (詳細は [6] 参照)。

式 (3) より、エッジ点  $p_i = (x_i, y_i)$  に対応する帯領域はパラメータ平面において平行な 2 直線  $b = -x_i a + y_i + 0.5$  および  $b = -x_i a + y_i - 0.5$  によって囲まれており、前者をプラス辺、後者をマイナス辺とよぶことにする。そしてこれらの直線群は、パラメータ平面を大小様々な凸多角形領域に分割する (一部の領域は無限遠に延びる領域となる)。この凸多角形領域をセルとよぶことにする。ここで極大直線成分とは、帯領域が局所的に最も重なっている領域に対応する直線成分であるので、あるセルに対応する直線成分が極大であるためには、そのセルの上部境界がすべてプラス辺であり、下部境界がすべてマイナス辺でなければならない。



帯領域を構成する直線群は、 $a = -\infty$  においては  $b$  軸の正方向から傾きの昇順に並んでおり、 $a$  が増加するにつれて各交点で他の直線と順番に入れ替わり、 $a = \infty$  においてはすべての順番が逆転して (当然平行な直線群の順番は不変)、 $b$  軸の正方向から傾きの降順に直線群が並ぶことになる (図 7)。したがって  $a$  軸の負方向から直線群の交点を順次走査して、そのつど直線の上下関係を調べていけば、極大成分に対応するセルを見つけることができる。そのセルにおける帯領域の重複度は、 $b$  軸の正方向からそのセルに到達するまでに通過するプラス辺の数をカウントすれば得られ、これがあるしきい値以上であれば、そのセルに対応する直線パラメータを出力する。本研究における実装では、セルの頂点座標の重心をそのセルを代表する直線パラメータとして出力するものとした。この手法を用いることによって、パラメータ平面を厳密に離散化しなくともあらゆる直線パラメータが表現でき、また極大成分のみを検出できる。

### 3.4 実験 — 直線検出手法の比較

#### 3.4.1 極大成分の検出能力

図 8 の 2 値化エッジ画像 (サイズ:  $100 \times 100$ , エッジ点の数: 840)

の直線検出に対して、本研究で実装した平面走査法を用いた Hough 変換 (Plane Sweep) と、従来の投票法を用いた Hough 変換のうち、[7] によるパラメータ平面の厳密な離散化を行なう手法 (*a-b Vote.1*) と、厳密でない離散化を行なう手法 (*a-b Vote.2*) を適用した結果を示す。パラメータ検出のためのしきい値は 75 とする。以下の数値実験は DEC Alpha (CPU21164: 600 MHz, メモリ: 1 Gbyte) で行なった。

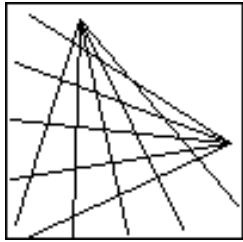


図 8: 入力画像

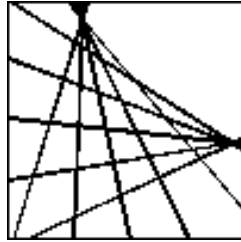


図 9: Plane Sweep

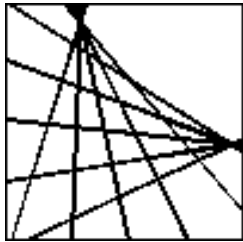


図 10: *a-b Vote.1*

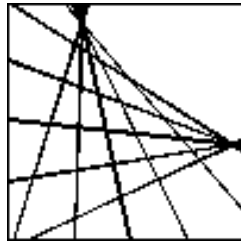


図 11: *a-b Vote.2*

表 1: 直線検出手法の比較 1

	time(sec.)	para	line	$a_{size}$	$b_{size}$
Plane Sweep	26.6	210	10	-	-
<i>a-b Vote.1</i>	3727.5	11908	10	-	-
<i>a-b Vote.2</i>	29.7	33	10	1/100	1/5

表 1 の para は検出された直線パラメータの数であり、line はその直線パラメータをもとにデジタル成分を求めて、実際に画像に出力したときに「人間の目」で確認できた直線の本数である。また  $a_{size}$ ,  $b_{size}$  は、*a-b Vote.2* においてパラメータ平面を離散化するときの各パラメータ軸の分割サイズである (Plane Sweep や *a-b Vote.1* ではそのようなパラメータは不要である)。

図 9, 10 より、Plane Sweep と *a-b Vote.1* は精度にそれほど差はないように見えるが、表 1 より、Plane Sweep の方がはるかに計算時間が短い。また、両者は同じ本数の直線 (line) を出力しているにも関わらず、Plane Sweep の方が検出パラメータ数 (para) が少なく抑えられているので、効率面で *a-b Vote.1* よりも優れているといえる。これは Plane Sweep がしきい値以上の票を得た極大成分の重心のみを検出しているのに対し、*a-b Vote.1* は得票数がしきい値以上であれば極大成分以外のパラメータでも検出してしまうからである。したがって、図 10 には表れていないが、*a-b Vote.1* は多くの無駄な直線パラメータを検出していることになる。

一方 *a-b Vote.2* は、実行時間が Plane Sweep とほぼ同じになるように、パラメータ平面の  $a$  軸を 1/100 間隔に、 $b$  軸を 1/5 間隔に離散化した投票法であるが、図 11 より、一見うまく直線検出できて

いるように見える。しかも表 1 より、出力された直線の本数のわりに検出パラメータ数も少ない。しかしこれは検出パラメータ数が少なく抑えられているというよりもむしろ、この手法では検出可能なパラメータが限られているためと考えられる。またパラメータ平面の離散化が厳密でないため、検出パラメータの精度は Plane Sweep や *a-b Vote.1* に比べて信頼性が低い。つまり、パラメータ平面を厳密に離散化しない限り、どんなに細かい格子状にしても、格子の境界線上の点に対応する直線をパラメータ表現できないのである。次の実験ではこの問題について検証する。

### 3.4.2 厳密に離散化しない投票法の問題点

図 12 の 2 値化エッジ画像 (サイズ: 100×100, エッジ点の数: 769) の直線検出に対して、先述の Plane Sweep および *a-b Vote.1*, *a-b Vote.2* を適用した結果を示す。パラメータ検出のしきい値は 100 とする。

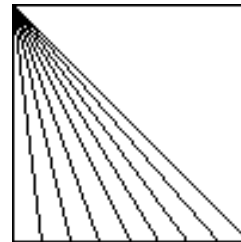


図 12: 入力画像

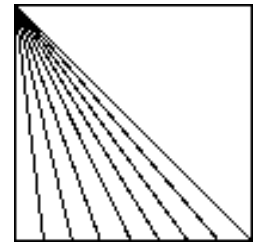


図 13: Plane Sweep

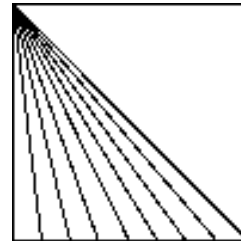


図 14: *a-b Vote.1*

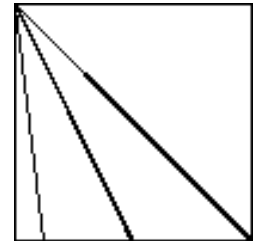


図 15: *a-b Vote.2*

表 2: 直線検出手法の比較 2

	time(sec.)	para	line	$a_{size}$	$b_{size}$
Plane Sweep	13.7	17	8	-	-
<i>a-b Vote.1</i>	1135.6	58	8	-	-
<i>a-b Vote.2</i>	11.6	19	3	1/50	1/10
	1143.3	426	8	1/2500	1/20

図 13, 14 より、Plane Sweep と *a-b Vote.1* はともに 8 本すべての直線を出力しているが、Plane Sweep の実行時間とほぼ同じになるように離散化を行なった *a-b Vote.2* は 3 本の直線しか検出できていない。離散化をより細かくして  $a$  軸を 1/2500 間隔に、 $b$  軸を 1/20 間隔にしたときに、はじめて 8 本の直線が出力できたが、実行時間が *a-b Vote.1* とほぼ同じであるにも関わらず、検出されたパラメータ数はそれよりも多くなっており、結局厳密に離散化した *a-b Vote.1* の方が効率が良いといえる。以上の実験結果より、従来の投票法を用いた手法よりも Plane Sweep の方が優れているといえる。

### 3.5 建築画像への適用

本研究で建築画像の直線検出を行なう目的は、対象建築物の構成成分の消失点を検出することであるので、検出された直線群の中から

各消失点で交わる直線束を選ばなければならない。従来の研究ではこの作業をユーザーが行っていたが、本論文では建築画像の特徴に着目し、計算機によってこれを行なう手法を提案する。

本論文で扱う典型的な建築画像に含まれる直線成分は、対象建築物の構成成分に対応するものが支配的であると考えられるので、その2値化エッジ画像から「多数決原理」に基づく Hough 変換によって適当なしきい値で直線成分を検出すれば、それらの成分が残るはずである。さらに検出する直線の傾きを限定することによって、そのうちの垂直成分と水平成分とをほぼ独立した形で抽出することができる。

以下に、建築画像から各消失点に収束する直線束を抽出した例を示す。画像から検出する直線  $y = ax + b$  が図 18 の座標系にしたがうとき、傾きを  $|a| \leq 1$  に限定すると対象建築物の垂直成分が主に検出され、それらは垂直方向の1つの消失点に収束する(図 19)。同様に傾きを  $a \leq -1$  あるいは  $a \geq 1$  に限定すると対象建築物の水平成分が主に検出され、それぞれ水平線上の2つの消失点に収束する(図 20, 21)。図 20 には明らかに例外的な直線が見られるが、これは原画像に写っている街灯に対応する直線成分である。



図 16: 原画像 (480×640)

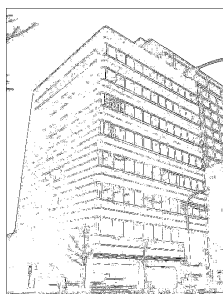


図 17: 2 値化エッジ画像

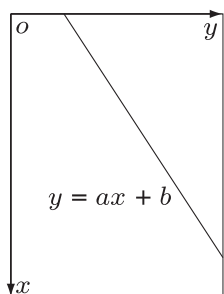


図 18: 画像座標系

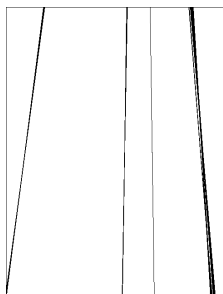


図 19: 垂直成分 ( $|a| \leq 1$ )

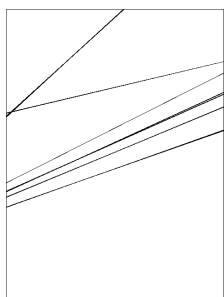


図 20: 水平成分 ( $a \leq -1$ )

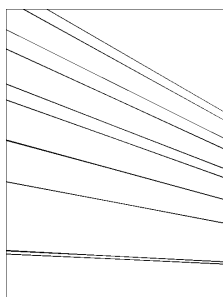


図 21: 水平成分 ( $a \geq 1$ )

画像にはノイズだけでなく、対象とする建築物以外のものに対応する成分も含まれるため、図 20 のように例外的な直線まで検出されるおそれがあるが、それらは直線検出のしきい値を適当に設定したり、検出する直線の傾きをさらに限定したりすることによって、取り除くことができる。また、次節でも述べるが、検出結果に例外的な直線が多少含まれていたとしても、後の消失点検出にはほとんど支障がない。これは先に述べたように、建築物を対象とした画像においては、その建築物の構成成分に対応する直線成分が、その他の直線成分よりも支配的、すなわち大部分を占めているからである。

### 3.6 直線検出のしきい値

平面走査法では、極大成分のセルにおける帯領域の重複度が与えられたしきい値以上であれば、そのセルに対応する直線パラメータを出力する。パラメータ平面の帯領域は画像平面のエッジ点に対応しているため、あるセルにおける帯領域の重複度とはすなわち、そのセルに対応する直線の成分となるエッジ点の数である。したがって、しきい値は画像のサイズとエッジ点の分布を考慮して決定しなければならないが、現段階ではその値の適切な設定方法は確立されておらず、非常に困難な問題とされている。本論文においても、予備実験を通して経験的にしきい値を決定しており、その点でまだ「人間の目」に頼る部分があるといえる。

## 4 建築画像からの消失点検出

### 4.1 Hough 変換による消失点検出

Hough 変換は、画像平面上において1直線上に並ぶ点集合が、パラメータ平面上において1点で交わる直線集合に対応するという幾何学的双対変換<sup>7</sup>の性質を利用したものである。これを逆に考えると、画像平面上において1つの消失点に収束する直線集合のパラメータは、パラメータ平面上において画像平面の消失点に対応する直線上にあることが分かる。

3.5 節より、Hough 変換のしきい値を適当に設定することによって、建築画像における支配的な3つの直線束、すなわち画像平面上で3つの消失点に収束する直線束がそれぞれ独立に検出できることが示された。したがって、そのパラメータ平面に対してさらに直線検出を行なうことによって、画像平面上の消失点が得られる(図 22)。そこで再び本研究で実装した Hough 変換を適用する。  
先にも述べたように Hough 変換は、情報の誤りや欠落に対して安定した処理を行なうことができるので、消失点検出に利用する直線束の中に例外的な直線が多少含まれていたとしても、検出される消失点の座標はほとんど影響を受けない。消失点検出に Hough 変換を適用する理由はこの点にある。

ここで消失点検出に用いる直線パラメータは、その数が多すぎると消失点検出の計算時間が長くなり、逆に少なすぎると、その中に例外的な直線が含まれている場合に、誤差の影響が大きくなるおそれがある。したがって直線検出の際のしきい値は、これらのことを考慮して決定されなければならない。

### 4.2 実験 — 実写画像への適用

3.5 節の図 16 および次の図 23 の実写画像の消失点検出に対して、本論文の提案手法を適用した結果 (PS) と、「人間の目」でエッジを



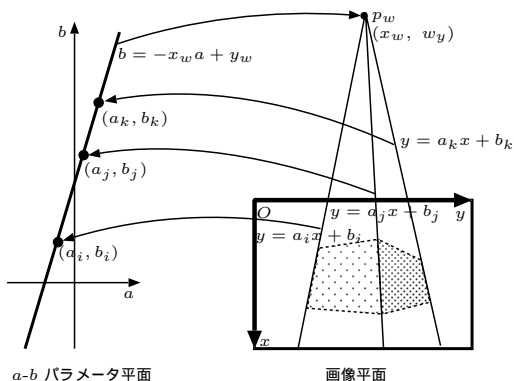


図 22: 消失点とパラメータ平面

認識して計算した結果 (HV) を表 3 に示す。なお、画像座標系は図 18 に従うものとし、表 3 において、 $p_w$  は垂直成分が収束する消失点、 $p_u$ 、 $p_v$  は水平成分が収束する消失点とする。



図 23: 原画像 (480×640)

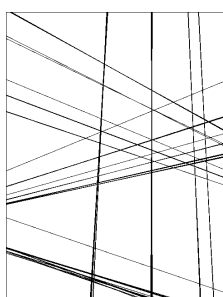


図 24: 直線検出結果

表 3: 消失点検出結果の比較

		$p_w$	$p_u$	$p_v$
図 16	PS	-1521.4, 286.0	561.2, -353.8	590.1, 1102.6
	HV	-1510.4, 288.9	558.5, -353.1	588.5, 1102.9
図 23	PS	-1737.9, 330.0	510.0, -392.1	582.5, 1064.1
	HV	-1761.5, 322.3	510.5, -393.2	579.5, 1048.5

PS: Plane Sweep による消失点 HV: 「人間の目」による消失点

画像のサイズから考えると、PS と HV の消失点の値の差はほとんど無視できる。したがって本論文の提案手法は、消失点検出に関して「人間の目」とほぼ同程度の検出能力があるといえる。

#### 4.3 既存の消失点検出手法との比較

従来の消失点検出手法は、画像から選んだエッジ点列に対して最小 2 乗法などによって線分を当てはめ、それらの線分を延長した直線の交点を計算するというものである [3][10]。しかし画像に含まれるノイズのために、それらの直線は必ずしも 1 点で交わるとは限らない。そこで Liebowitz ら [3] は、エッジ点に当てはめた線分の端点から、ある 1 点を通る直線群までの距離の 2 乗和が最小になるように最大尤度推定を行ない、その点を消失点として求めている。また金谷 [11] は、これを射影幾何学的に考えて、エッジ点に当てはめた線分の同次座標ベクトルとの内積の 2 乗和が最小になるような点を消失点として求め、さらにそれらの線分にその消失点を通る直線を当てはめている。

先にも触れたが、これらの手法が消失点の計算に用いる線分を「人間の目」で選び、あらかじめ指定しているのに対して、本論文の手法は、「建築物を簡略化・単純化していけば、ほとんどが直方体成分の集まりになる」という特徴を利用して、画像から建築物の構成成分を自動的に認識して消失点を計算しているため、CV という言葉そのものの意味に近いと思われる。しかしあらゆる建築物に対する融通性という点では、建築画像の認識にはまだ「人間の目」、すなわち人の介入が不可欠であるだろう。

### 5 3次元建築モデルの再構成

#### 5.1 カメラ・キャリブレーション

再構成とは、カメラによる 3 次元シーンから 2 次元画像への投影の逆問題を解くことであるので、その変換過程を正確に記述することが必要となる。そのために、カメラの内部パラメータからなるカメラ行列を決定することをカメラ・キャリブレーションという [12]。本論文ではカメラのレンズひずみを無視するため、カメラの内部パラメータは視心座標  $(u_0, v_0)$  と焦点距離  $f$  の 3 つとなる。本論文の手法によって、建築画像から直交する 3 方向の消失点を検出できるので、これを利用してそれらの 3 つの内部パラメータを求める。

#### 5.2 立面の復元

カメラ行列から、画像平面上の任意の点の 2 次元射影平面上における同次座標が求められる。そこで Liebowitz らの論文 [3] および Faugeras の著書 [13] に基づいた立面復元のアルゴリズムを用いて、画像平面を射影変換し、対象建築物の立面を復元する。なお、2 つの建築物が隣接する画像においてはそれらの立面の境界線の切り出しは人手でおこなった。

#### 5.3 実験 — 立面の復元

例 1: 3.5 節の図 16 の実写画像に対して、本論文の手法によって得られた消失点 (表 3) を用いてカメラ・キャリブレーションを行ない、立面の復元を行なった (図 25, 26)。両立面ともほぼ歪みのない長方形になっていることが分かる。また、(縦) : (横) の比は北側 1 : 0.75, 西側 1 : 0.68 であり、実測値 (北側 1 : 0.71, 西側 1 : 0.62) にほぼ等しいことが確認できた。



図 25: 北側立面復元図



図 26: 西側立面復元図

例 2: 図 27 のように対象外の建築物や道路・車などの障害物を多く含んでいる場合は、対象建築物の構成成分のみを検出することは困難であるため、図の黒枠のようにクリッピングを行ない、その 2 値化エッジ画像 (図 28) を用いて消失点検出を行なった。その結果復元された 2 つの立面の縦横比も、やはり実測値とほぼ等しくなった。



図 27: 原画像 (640×480)

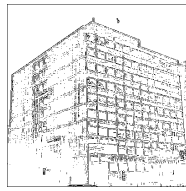


図 28: 2 値化エッジ画像

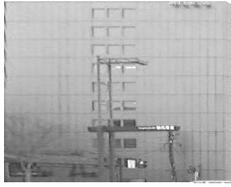


図 29: 東側立面復元図



図 30: 北側立面復元図

例 3: 図 31 の建築物はベランダと壁面との奥行き差があるため、下から見上げた画像から立面を復元すると不自然さが残ってしまうが、柱間隔や階高のプロポーションは正確に表現されているので、建築物の雰囲気は損なっていないと思われる。



図 31: 原画像 (480×640)



図 32: 西側立面復元図



図 33: 南側立面復元図

5.4 3次元モデルの再構成 VRML などによって仮想空間内で3次元モデルを作る場合には、絶対的な方向性やスケールはあまり重要ではない。そのため、建築画像から復元された立面のうちのひとつを参照平面として選び、それに対する他の立面の相対的な法線方向とスケールを求めて3次元モデルの再構成を行なった。以下に本研究で得られた3次元建築モデルをVRML<sup>8</sup>によって表した例を示す。



北西方向



南西方向



南東方向

図 34: 例 1

図 34 は図 16 および図 23 を含む複数枚の建築画像から建築物のすべての立面を復元した3次元モデルである。高さの異なる部分がある立面が存在するが、画像平面上ではそれらの部分を構成する直線成分の消失点は等しいので、それらは同時に復元できる。画像に写らない部分の立面は直方体のボリュームとして補っている。本論文では写真の撮影条件を厳密に設定しなかったため、立面の陰影に違いが見られる<sup>9</sup>。

図 35 は図 29,30 を用いて復元した3次元モデルである。2つの立面しか復元されていないが、建築物のスケール感や雰囲気は十分にうかがえる。

図 36 は図 32,33 を用いて復元した3次元モデルであるが、立面のデザインに共通性が見られるような場合は、このようにひとつの立面を別の立面に配置することによって補うことができる。

## 6 結語

本研究の成果を以下にまとめる。

- 2値化エッジ画像からの直線検出のために、平面走査法を用いた Hough 変換を実装・適用した。
- 直線検出において、本研究で実装した手法が従来の手法と比べて、精度および計算時間の面で優れていることを示した。
- 建築画像の特徴を利用して、その消失点検出において、本論文で提案した手法が有効であることを示した。
- 本論文の提案手法によって得られた消失点を用いて、十分に精度が高く美しい3次元建築モデルが得られた。

今後の研究課題としては、まず平面走査法の導入によって直線検出用 Hough 変換の計算時間は飛躍的に改善されたが、それでもまだ実写画像への適用となると多くの計算時間を必要とするので、アルゴ



リズムやデータ構造を一層工夫することによって実用レベルに近づけなければならないことが挙げられる。

また、本論文では直線検出のしきい値を経験的に決定しているため、検出される直線パラメータの数は恣意的である。先述のように、検出される直線パラメータの数は後の消失点検出に大きく影響してくるので、しきい値についての明確な基準を設けることが重要であり、これも今後の研究課題である。

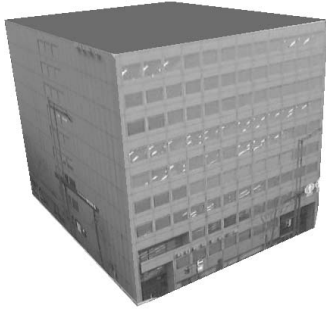
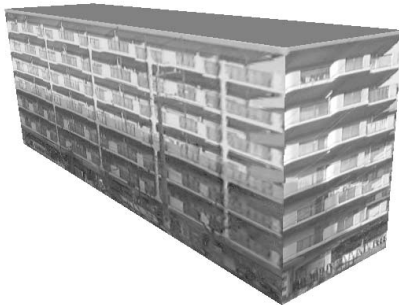


図 35: 例 2



南西方向



南東方向

図 36: 例 3

最後に、3次元モデルの再構成の段階でひとつの建築画像から復元された建築物の2つの立面は、一般に相対的なスケールが異なっており、一方の立面を参照平面として他方の立面のスケールを決定しなければならない。これは原画像における立面の境界線と視心座標との距離によって決まると推定されるが、詳細については今後の研究課題としたい。もしそうであれば、原画像から建築物の構成成分だけでなく、隣り合う立面の境界線を検出する必要があるが、これを計算機によって自動化することは、現段階では非常に困難であるだろう。

#### 注釈

- 注 1. CG が 3 次元シーンから 2 次元画像への写像を行なうのに対し、CV は 2 次元画像から 3 次元シーンへの写像を行なうものとされる [5]。
- 注 2. アボソフトウエア社の Photo3D などが代表的な例である。  
<http://www.photo3d.com/> 参照。
- 注 3. 観測の視点位置によって、ある視点から見えるものが他のものに隠されて見えないこと。
- 注 4. 本研究では予備実験を通して検証した結果、原画像の画像処理用ソフトとして、Pineapple USA 社の Image Magick を採用した。
- 注 5. ピンホール (レンズ中心) を通る光線のみで結像を考えるカメラ・モデルである。モデルパラメータとしてはピンホールと画像平面間の距離を表す焦点距離  $f$  を用いる。3 次元空間と 2 次元画像平面の幾何学的対応関係を記述する最も単純なモデルである。
- 注 6. 3 次元空間の原点  $O$  を視点 (レンズ中心) とし、焦点距離を  $f$  とする透視変換では、 $O$ - $XYZ$  座標系 (カメラ座標系) における 3 次元空間の点  $P = (X, Y, Z)$  は、 $P$  と原点  $O$  とを結ぶ直線と平面  $Z = f$  との交点  $\tilde{p}$  に投影される。点  $\tilde{p}$  の座標を  $(x, y, f)$  とすれば、次の幾何学的関係が得られる。  
$$x = f \frac{X}{Z}, \quad y = f \frac{Y}{Z}$$
このとき、平面  $Z = f$  を 2 次元射影空間とよび、カメラの投影における画像平面に対応する。そして  $\tilde{p}$  および  $P$  を、画像平面上の点  $p = (x, y)$  の同次座標とよぶ。同次座標はその要素に任意の 0 でない実数をかけたものと同じ点を表すという性質をもつ。
- 注 7. ここでいう双対変換  $\tau$  とは、点  $p : (x, y)$  を直線  $\tau(p) : b = -xa + y$  に、直線  $l : y = ax + b$  を点  $\tau(l) : (a, b)$  に写像する変換である。
- 注 8. Virtual Reality Modeling Language, 3D のオブジェクト群を記述し、それらを組み立てることによって仮想シーンを構築するための言語である。
- 注 9. この問題を解消するレンダリング手法についての研究は Debevec ら [4] によって行なわれている。

#### 参考文献

- [1] 渡辺一正 他: “1999 年グラフィックスユーザーのコンピュータ利用環境調査”, 日経 CG, Vol.156, pp.58-89, Sep. 1999.
- [2] 北原英雄, 佐藤宏介, 千原國宏: “景観シミュレーション CG のための建物壁面データの合成”, 日本建築学会計画系論文集, No.515, pp.305-311, Jan. 1999.
- [3] David Liebowitz, Antonio Criminisi and Andrew Zisserman: “Creating Architectural Models from Images”, EUROGRAPHICS '99, Vol.18, No.3, 1999.
- [4] Paul E. Debevec, Camillo J. Taylor, Jitendra Malik: “Modeling and Rendering Architecture from Photographs: A hybrid geometry- and image-based approach”, SIGGRAPH '96, 1996.
- [5] 松山隆司, 和田俊和: “Hough 変換: 投票と多数決原理に基づく幾何学的対象の検出と識別”, コンピュータビジョン 技術評論と将来展望, pp.149-165, 松山隆司 久野義徳 井宮淳 編, 新技術コミュニケーションズ, 1998.
- [6] Tetsuo Asano, Naoki Katoh, Takeshi Tokuyama: “A Unified Scheme for Detecting Fundamental Curves in Binary Edge Images”, Proc. of the 2nd Annual European Symposium on Algorithms, pp. 215-226, Springer-Verlag, 1994 (to appear in Computational Geometry: Theory and Applications).
- [7] 浅野哲夫, 河村泰之: “デジタル直線検出問題の計算量に関するアルゴリズム論的考察”, 電子情報通信学会論文誌, Vol. J83-D-1, No.1 pp.80-89, Jan. 2000.
- [8] Tetsuo Asano, Naoki Katoh: “Variants for the Hough transform for line detection”, Computational Geometry: Theory and Applications, Vol.6, 231-252, 1996.
- [9] 金谷健一: “画像理解 — 3 次元認識の数理 —”, 森北出版, 1990.
- [10] 丸山 保, 金谷健一: “カメラ焦点距離の最適キャリブレーションシステム”, 情報処理学会研究報告, 93-CV-82, 1993.
- [11] 金谷健一: “コンピュータビジョンのためのくりこみ法”, 情報処理学会研究報告, 93-CV-84, 1993.
- [12] 出口光一郎: “カメラキャリブレーション手法の最近の動向”, 情報処理学会研究報告, 93-CV-82, 1993.
- [13] Oliver Faugeras: “Three-Dimensional Computer Vision — A Geometric Viewpoint —”, The MIT Press, 1993.