

Title	計算幾何学的手法を用いた基本図形の認識
Author(s)	平識, 善弘
Citation	
Issue Date	2011-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/9617
Rights	
Description	Supervisor:浅野哲夫, 情報科学研究科, 修士

修 士 論 文

計算幾何学的手法を用いた基本図形の認識

北陸先端科学技術大学院大学
情報科学研究科情報科学専攻

平 識 善 弘

2011 年 3 月

修 士 論 文

計算幾何学的手法を用いた基本図形の認識

指導教官 浅野哲夫 教授

審査委員主査 浅野哲夫 教授
審査委員 上原隆平 准教授
審査委員 平石邦彦 教授

北陸先端科学技術大学院大学
情報科学研究科情報科学専攻

0910051 平識 善弘

提出年月: 2011 年 2 月

概要

コンピュータに正確に図形を認識させることができると、工業製品の開発において役立つことが多い。画像を使った工業機械における位置決めの問題への応用はその一例である。基本図形を含むシーンを CCD カメラで撮影した画像が入力されたとき、基本図形の中心点および形状を範囲することが本研究の主な課題である。特に一般的な方法により求められる適度な小ささの十字と円の認識に重点を置く。CCD カメラで撮影した画像には量子化誤差が必ず含まれる。よって、量子化誤差を考慮した認識も課題となる。これらの課題に対して計算幾何学的手法により、十字や円などの基本図形の位置をより高い精度で求める方法を提案する。

目次

第1章	はじめに	1
1.1	背景	1
1.2	目的	2
1.3	本論文の流れ	2
第2章	CCD センサ入力モデル	3
2.1	デジタル画像と正方格子平面の対応	3
2.2	単位正方形における重み付き図形面積の計算例	4
第3章	量子化誤差なしの白黒図形認識	6
3.1	十字の認識	7
3.1.1	一般の認識法	7
3.1.2	例外	8
3.1.3	認識できない場合	8
3.2	回転十字の認識	9
3.2.1	一般の認識法	9
3.2.2	認識できない場合	10
3.3	円の認識	11
3.3.1	一般の認識法	11
3.3.2	認識できない場合	12
3.4	画素を共有する円の認識	13
3.4.1	一般の認識法	13
3.4.2	認識できない可能性がある場合	15
第4章	量子化誤差なしの任意色図形認識	16
4.1	十字の認識	17
4.2	回転十字の認識	17
4.3	円の認識	18
4.4	量子化誤差なしの図形認識の結果	20
第5章	量子化誤差ありの図形認識	21
5.1	量子化誤差無しの方法に基づく円の中心点の区間検出	22

5.2	1画素ごとの値を用いた円の中心点の区間縮小	25
5.2.1	任意の円が任意の画素でとる値を求めるアルゴリズム	26
5.2.2	中心点を含む画素が円により完全に覆われていない場合	29
5.2.3	区切った点の座標が正しいか調べるアルゴリズム	31
5.2.4	区間を表す凸包を求めるアルゴリズム	34
5.2.5	最小包含円問題	36
5.3	量子化誤差ありの図形認識の実験	37
5.3.1	実験内容	37
5.3.2	実験結果	37
第6章	おわりに	38
6.1	まとめ	38

第1章 はじめに

1.1 背景

コンピュータに正確に図形を認識させることができると、工業製品の開発において役立つことが多い。画像を使った工業機械における位置決めの問題はその一例である。この問題は認識したい製品の一部に、あらかじめ十字や円などの図形をマークしておき、それをデジタルカメラにより撮影した画像において認識することにより、製品の位置を正確に認識するというものである。

既存の図形位置認識に関する手法としては、正規化相関法などのパターンマッチングによる認識や、エッジ検出およびサブピクセル処理（境界近傍の輝度勾配を微分することによって境界線を高い精度で求める手法）による認識などがある。それらの手法を使用すると、サブピクセル処理は0.1画素程度の精度でエッジを求めることができ、位置決めを行うことができる。とされている。

これらの方法はCCDカメラで撮影したときに起こりうる誤差を考慮した認識法である。しかし、CCDカメラや、撮影したときに起こりうる誤差について厳密なモデルを定めている訳ではない。そのため、これらの方法で正確な位置決めが行われているとはいえない。工業製品の開発現場においては、ごく小さなずれでも許されないような場面もあり、更に正確な位置決め手法の確立が求められてる。

一方、計算幾何学に基づいた方法により、入力されたラスター画像から、画素よりも細かいレベルで画像中の輪郭線の正確な形状を求める研究が行われている。例えば、Prasadら[1]は直線状の境界線および直線の交差する点について正確に検出する研究を行っている。また、画像処理への応用として、Fleischer[2]は超解像技術への応用に関する研究を行っている。

これらの研究により、CCDカメラ入力のモデルを数学的にきちんと定めておけば、量子化誤差が含まれる場合であっても、画素よりも細かいレベルで正確に輪郭線の検出が行うことができるようになっている。これらは正確な輪郭線検出の手法であるが、これらの手法を改良すれば同様の考え方で基本図形の認識についても正確に行うことができる。

1.2 目的

本論文では、画像を使った工業機械における位置決めの問題について、CCD カメラで撮影した基本図形の含まれる画像を入力とし、計算幾何学的手法により基本図形の位置をより高い精度で求める方法を提案する。ただし、この基本図形は理想的な形状であると仮定する。理想的な形状であるとは、全ての画素の輝度値が本研究で仮定する CCD センサ入力モデル（デジタル画像を正方格子平面に対応させており、ひとつの画素に対応する正方形が存在する。画素は輝度値を持つが、この値は正方形における重み付き図形面積に対応させている。詳細は本論文の第 2 章を参照）の正方格子平面上において正確な基本図形の形状を表す値をとっていることをいう。

目印として認識すべき基本図形としては、十字形と円の 2 種類を考えるが、十字形については回転も許す。これらの図形について、最初は、量子化誤差を考慮せずに図形の中心点と形状（十字なら各辺の長さ、円なら半径）を正確に求める方法論を確立する。次に円の認識については量子化誤差を考慮した場合に拡張して考える。入力画像の輝度値は整数で与えられるのが普通なので、量子化誤差は避けられない。この量子化誤差のために、輝度値だけの情報では円の中心と半径を正確に求めることは不可能である。よって、その場合には、円の中心を必ず含む円領域を出力することとする。

1.3 本論文の流れ

第 2 章では CCD センサ入力モデルについて詳細に述べる。このモデルは本論文を通して使用する。

第 3 章では量子化誤差なしの白黒図形の認識について述べる。この章では、十字形と円の 2 種類を考え、十字形については回転も許す。また、円が 2 つ存在するとき、2 つの円が 1 つ以上の画素を共有している場合についても考える。

第 4 章では量子化誤差なしの任意色図形の認識について述べる。認識すべき基本図形は第 3 章と同様である。しかし、2 つの円が存在する場合については、1 つの円の場合とほぼ同様の考え方で認識できるので省略する。また、量子化誤差なしの図形認識の結果についてもこの章でまとめる。

第 5 章では量子化誤差ありの場合について述べる。小さなスペースに配置しやすい、回転に強いなどの理由から、認識する基本図形は円のみとする。もし、量子化誤差ありの十字形の認識を行いたいならば、十字の各辺において Prasad ら [1] の直線状の境界線を正確に検出する手法を応用すれば簡単にできるだろう。この章の最後では、量子化誤差ありの図形認識を実装したプログラムを用いて、どの程度の精度が得られるか実験を行った結果を示す。

第 6 章では論文全体のまとめと、関連する解決すべき問題について述べる。

第2章 CCDセンサ入力モデル

2.1 デジタル画像と正方格子平面の対応

CCD センサ入力のモデルは，図 2.1 に示すように被写体を撮影して得られるデジタル画像を正方格子平面に対応させるものである．正方格子平面とは，単位正方形が周期的に並んだ区切りのある 2 次元平面のことをいう．

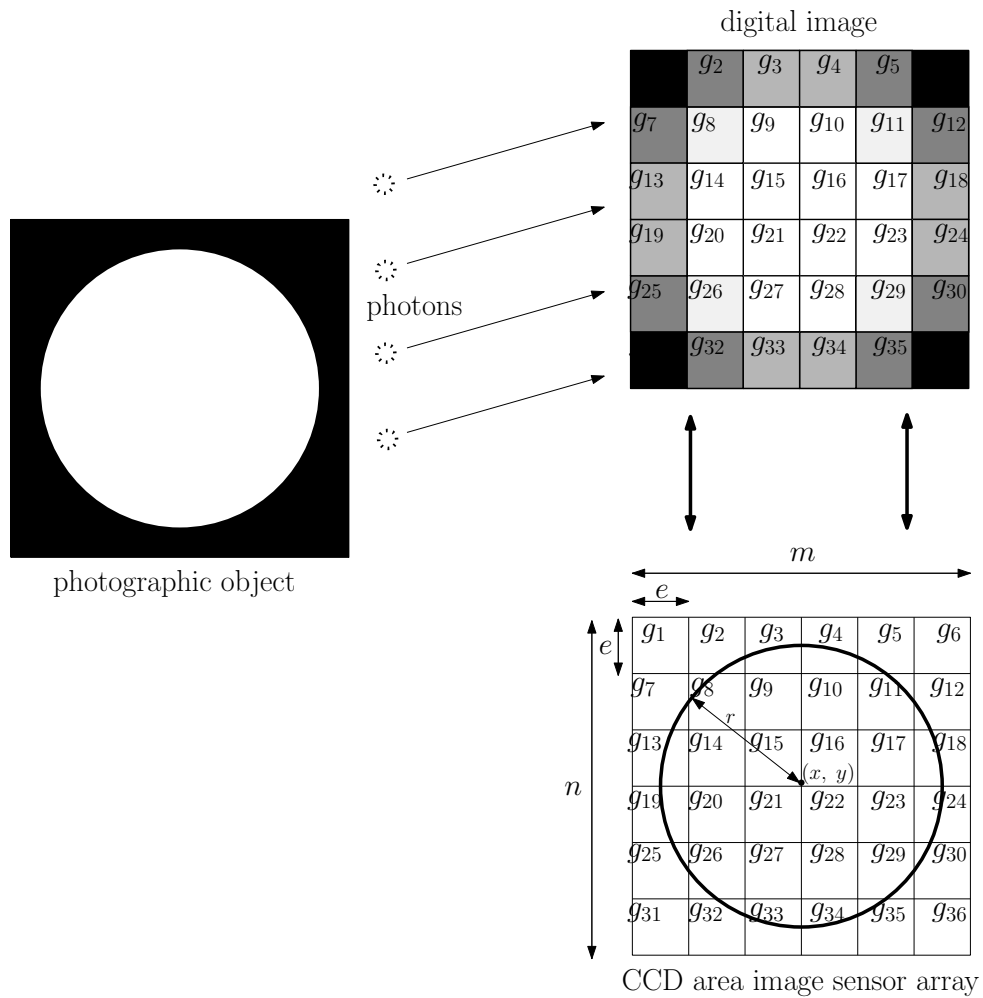


図 2.1: デジタル画像と正方格子平面の対応.

light intensity :  = 1  = 0  = 0.5

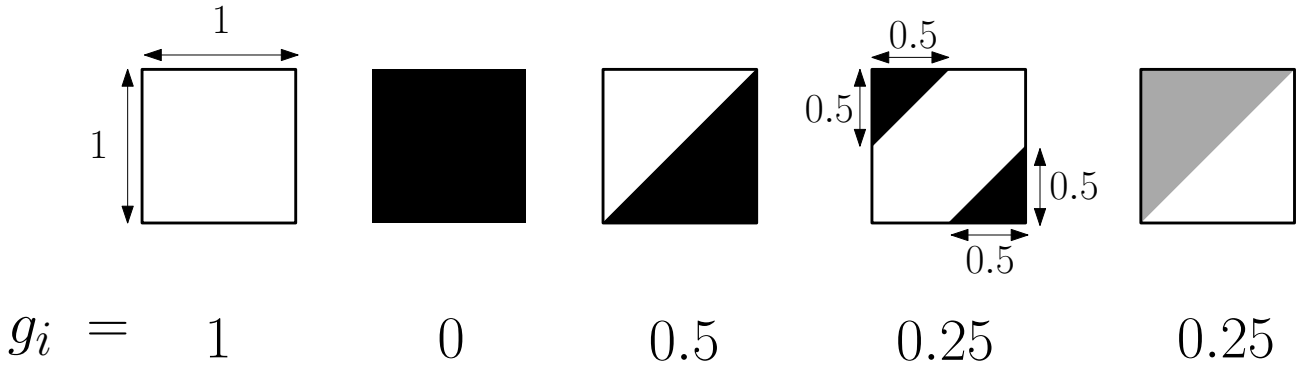


図 2.2: 1つの単位正方形における重み付き図形面積の例. 左から例1, 例2, 例3, 例4, 例5.

CCD センサはデジタルカメラなどに使用される半導体素子である。本研究では特に CCD を二次元格子状に敷き詰めたもの（CCD エリアイメージセンサ）を考える。デジタル画像の画素の輝度値は、対応する CCD に入る光子の量で決まる。このとき、撮影により得られたデジタル画像を正方格子平面に対応させる。デジタル画像の中の任意の画素に対応する正方格子平面上の単位正方形が存在する。画素は輝度値は単位正方形における重み付き図形面積に対応させている。

正方格子平面上の全ての $e \times e$ 単位正方形は、簡単の為に本論文では 1×1 とする。図形面積にかかる重みとは光強度である。光強度の最大値を 1 とする。ただし、撮影される被写体から反射する光は、光子の損失が無く、全てセンサに入るものとする。

デジタル画像の画素の列数を m 、行数を n とする。観測される輝度値は 1 次元配列 g_i (i は要素番号) で表す。輝度値 g_i は左上端から右に向かって $g_1, g_2, g_3, \dots, g_m$ と番号付けされる。右端までたどり着くと、次の行の左端から $g_{m+1}, g_{m+2}, g_{m+3}, \dots$ と番号付けを続ける。このデジタル画像の画素の配置は、ちょうど正方格子平面の単位正方形の配置と一対一対応している。

2.2 単位正方形における重み付き図形面積の計算例

図 2.2 は、1つの単位正方形における重み付き図形面積の計算の例である。被写体の撮影時に誤差無く輝度値が観測できたとすれば、そのデジタル画像の画素の輝度値 g_i に対応する正方格子平面の単位正方形における重み付き図形面積の計算結果 g_i は完全に一致する。

例 1 は単位正方形全体の光強度が 1 である。単位正方形の辺の長さは 1×1 とするため、この単位正方形に対応する画素の輝度値は 1 である。

例 2 は単位正方形全体の光強度が 0 である。この単位正方形に対応する画素の輝度値は 0 である。

例3は単位正方形がちょうど半分の領域に分けられており、一方の光強度が0、他方の光強度が1である。この単位正方形に対応する画素の輝度値は0.5である。図形と背景の境界線付近に存在する画素においては、この例のように境界線が画素中を通る場合がある。

例4は光強度が0の領域が2箇所（2箇所の面積を足すと0.25）に存在し、その他の領域は光強度が1である。この単位正方形に対応する画素の輝度値は0.25である。この例のように画素中で図形部分と背景部分が入り組む場合もある。

例5は単位正方形がちょうど半分ずつの領域に分かれており、片方の光強度が0.5、他方の光強度が0である。この単位正方形に対応する画素の輝度値は0.25である。この例のように光強度が0か1以外の場合もある。

第3章 量子化誤差なしの白黒図形認識

本章では白黒図形のみが入力として与えられると仮定し，図形の中心点を正確に求める方法について述べる．扱う図形は十字，回転十字，円，画素を共有する円であり，図形部分を白，背景を黒とする．元の図形が白と黒の部分だけで構成されていると仮定すると，CCD センサ入力モデルにおける各画素の値は，ちょうど 1×1 正方形（画素）を占める白領域の面積になることに注意する．

問題 1. 観測される輝度値 $\{g_i\}$ のみを用いて，十字と円（図形が1，背景が0）の中心点と正確な形状を求めよ．

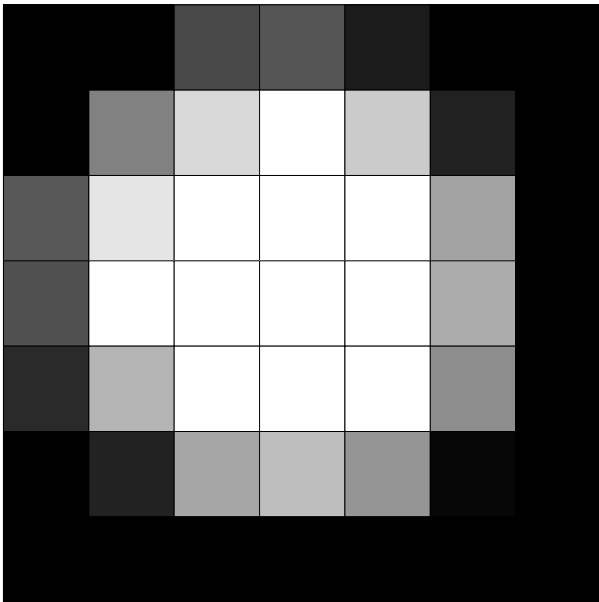


図 3.1: 入力：デジタル画像の輝度値 $\{g_i\}$ ．

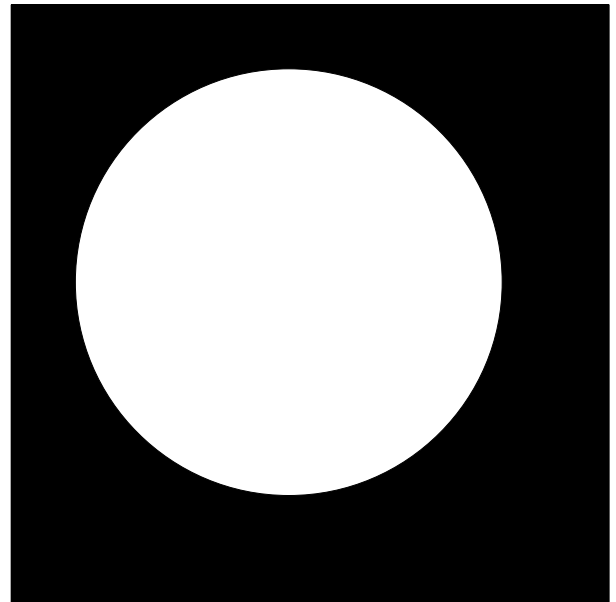


図 3.2: 出力：対象図形の正確な形状（円の場合なら中心点座標と半径）．

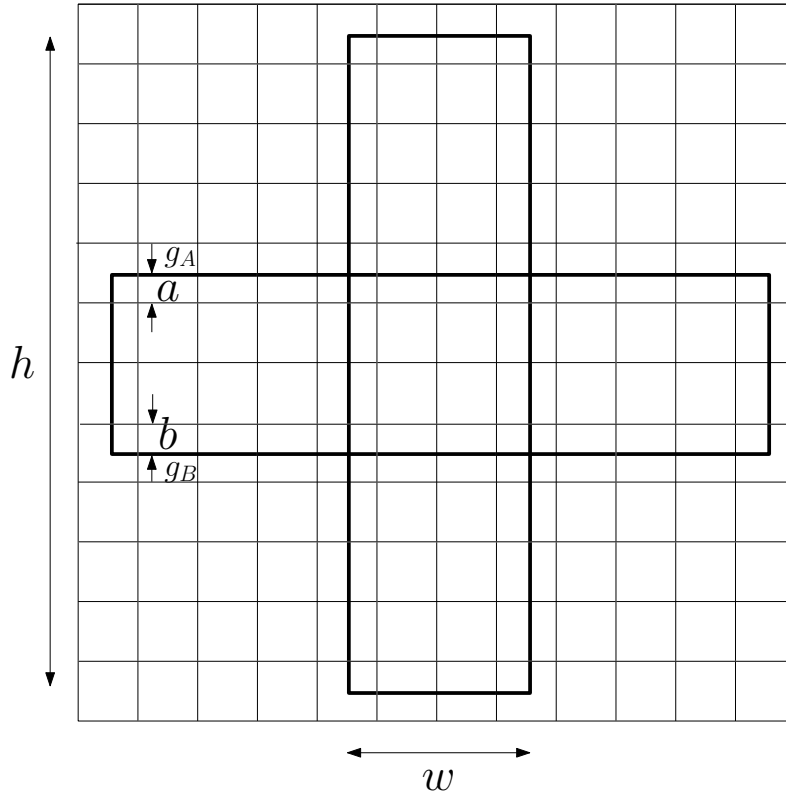


図 3.3: 画像を表す格子平面上の $w \times h$ の細長い長方形 2 個から構成される回転なしの十字。ただし、十字の横棒の上端から、上端を含む画素 g_A の底辺までの距離を a 。十字の横棒の下端から下端を含む画素 g_B の上辺までの距離を b とする。

3.1 十字の認識

3.1.1 一般の認識法

十字を、図 3.3 に示すように、 $w \times h$ の細長い長方形 2 個から構成される図形と定義したとき、十字の部分を白（値 1）、背景を黒（値 0）と仮定すると、十字を含む領域における輝度値の和は十字の面積に等しいことから、次式を得る。

$$\sum g_i = 2hw - w^2, \text{ ただし, } w \geq 1, h \geq w + 4$$

$\sum g_i$ は対象図形のみを含む最小の画素数で作られる矩形に含まれる輝度値を全て足し合わせたものであり、観測された値から計算できる。対象図形のみを含む最小の画素数で作られる矩形において、十字の横棒の上端から、上端を含む画素 g_A の底辺までの距離を a とし、十字の横棒の下端から下端を含む画素 g_B の上辺までの距離を b とすると、 a と b の値から w の値を計算することができる。また、 a と b の値は、画素 g_A 、 g_B での輝度値に等しいことがわかる。 $h \geq w + 4$ なので、 g_A 、 g_B 画素を含む列は必ず端から 2 列目に存在する。このように、 w が求まるため、上記の方程式は h だけが未知数となり、 h について代数的に解くことができる。

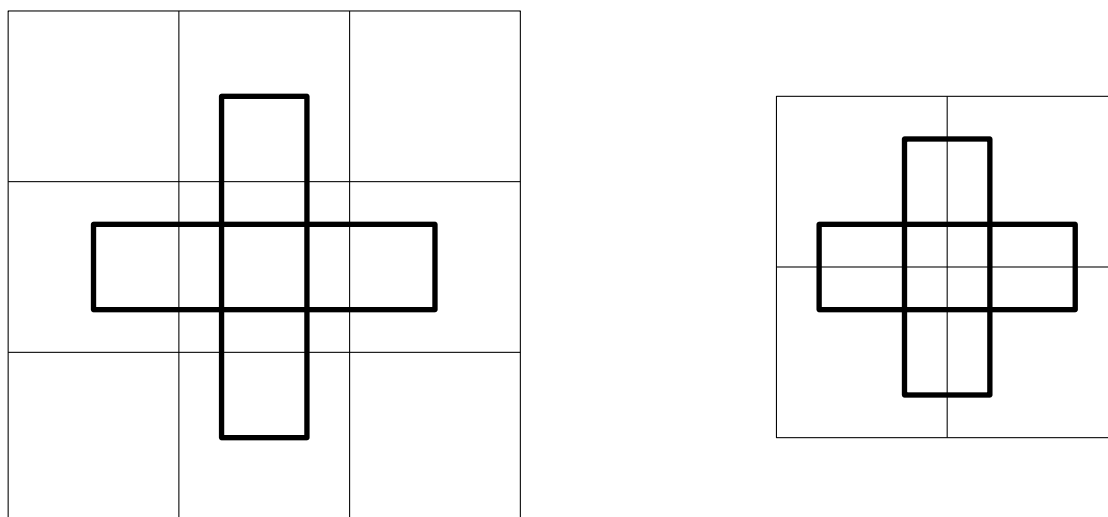


図 3.4: 十字の例外と認識できない場合

したがって，十字の中心の y 座標も求まる．同じことを十字の縦棒についても行えば，十字の中心の x 座標もわかる．

3.1.2 例外

一般の認識法には $w \geq 1$, $h \geq w + 4$ という条件が付いている．これは， $w < 1$ または $h < w + 4$ ならば，画素 g_A , g_B が存在しない可能性があるためである．ただし，別の方法ならば，もう少し小さな十字でも求めることができる．

図 3.4 の左に示しているのは，十字を構成する長方形において，その両端が他方の長方形にかぶらないように独立していて，その両端部分の面積を比較できる場合である．このような場合，両端の面積から中心点の座標が求まる．ただし，十字の w , h の正確な値は求められない．

3.1.3 認識できない場合

図 3.4 の右に示しているように，長方形の両端部分の面積が，他方の長方形の一部分の影響を受けているために比較できない場合には，十字の中心点を本質的に求めることができない．

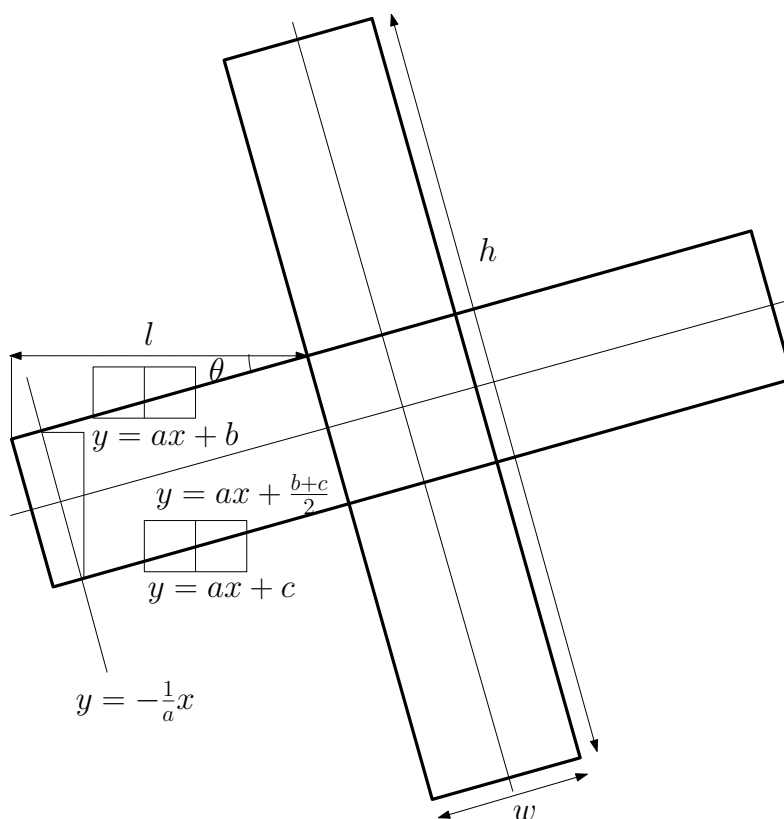


図 3.5: 画像を表す格子平面上の十字を角度 θ だけ傾けた場合. ただし, 直線 $y = ax + b$ が他方の長方形と最も左側で接触する点から直線 $y = ax + b$ の十字上の左端までの水平距離を l とする.

3.2 回転十字の認識

3.2.1 一般の認識法

回転十字は図 3.5 のように, 十字をある小さな角度 θ だけ傾けたものである. 対象図形の十字を撮影するとき, 正方格子の辺に対して水平垂直な十字が撮影できたとすれば, 通常の十字の認識法で認識可能だが, それはあまり現実的ではないだろう. また, できるだけ θ が大きくならないように注意するため, θ は 45° 以上の大きな値も取ることはないと仮定する. 全体の面積は回転なしの十字と同様に h と w の式で表されるが, $w \geq \sqrt{5}$, $h \geq \frac{8}{\cos \theta} + w$ (図 3.5 の直線 $y = ax + b$ が他方の長方形と最も左側で接触する点から直線 $y = ax + b$ の十字上の左端までの水平距離を l としたとき, 方程式: $\cos \theta = \frac{l}{\frac{1}{2}(h-w)}$, $l \geq 4$ を満たすような h) という制約が付く.

回転十字の認識には Prasad ら [1] の線形境界検出法を使用する．この方法を使うと，白と黒の領域が存在し，二つの隣り合う画素の両方を通る境界線（直線）が存在するとき，その直線を正確に認識することができる．この方法により，図中の $y = ax + b$ と $y = ax + c$ を求めたとすると，2つの直線から等距離な直線は $y = ax + \frac{b+c}{2}$ であることがわかる．この方法により十字の中心で交差する点を求めることができる．また， $y = -\frac{1}{a}x$ という直線を考えてやると， $y = ax + b$ と $y = ax + c$ のそれぞれとの交点を求めることができるので， w についても求めることができる． h の求め方は回転なしの十字と同様である．

3.2.2 認識できない場合

一般の認識法には $w \geq \sqrt{5}$, $h \geq \frac{8}{\cos \theta} + w$ という条件が付いている．図 3.6 に示すように，条件に合わない回転十字の場合，図形の正確な形状と中心点を求めることが非常に難しく，求められない場合も多いだろう．

十字を構成する長方形と背景の境界線を求めるとき，2つの隣り合う画素を用いるが， $w > \sqrt{5}$ だとすると，図 3.6 の中央の横に並ぶ2画素のように，求めたい境界線と，その反対側の境界線も画素中に含まれてしまった場合，線形境界検出法を用いることができない．また， $h > \frac{8}{\cos \theta} + w$ ならば， $l < 4$ であるため，線形境界検出法を用いるとき，十字を構成する長方形の端が含まれてしまったり，他方の長方形の一部分が含まれてしまったりする恐れがある．

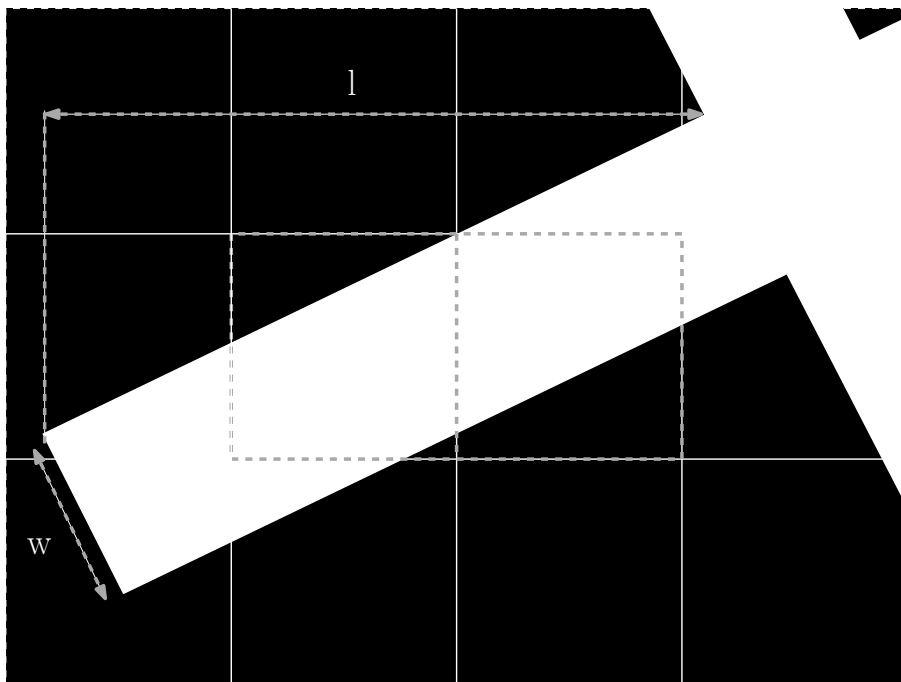


図 3.6: 回転十字の認識できない場合

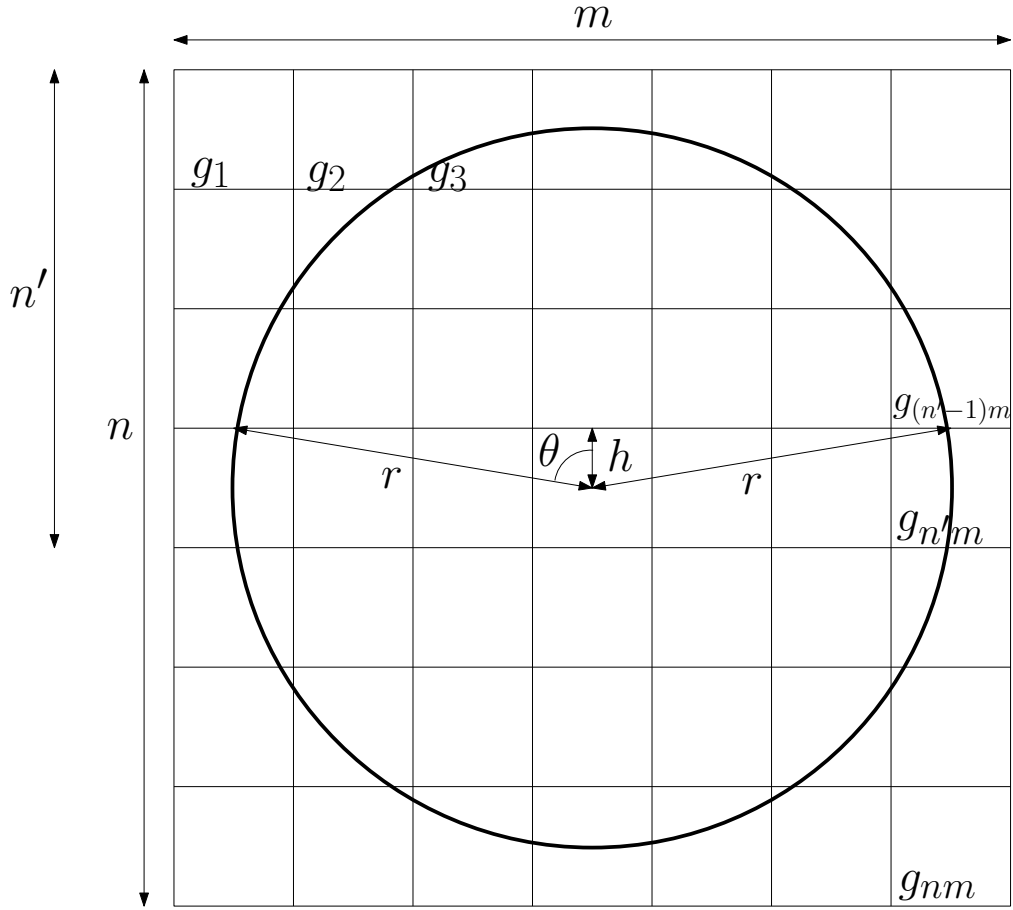


図 3.7: 画像を表す格子平面上の半径 r の円. ただし, θ は中心点を含む画素の行の上端と円の交点により定まる角度, h は円の中心から中心を含む画素の上端までの距離.

3.3 円の認識

3.3.1 一般の認識法

円を図 3.7 のように中心点を含む画素の行の上端と円の交点により定まる角度を θ とする. 半径 r とこの角度 θ によって円を定義したとき, 次式を得る. ただし, h は円の中心から, 中心を含む画素の上端までの距離である.

$$\sum_{i=1}^{nm} g_i = \pi r^2, \quad h = r \cos \theta \leq 1, \quad \sum_{i=1}^{(n'-1)m} g_i = r^2 \left(\theta - \frac{1}{2} \sin 2\theta \right),$$

$$\text{ただし, } 0 < \theta \leq \frac{\pi}{2}, \quad n \geq 2, \quad m \geq 2.$$

n, m は図形のみを含む最小の画素数で作られる矩形の長さであり, 行数と列数でもある. n', m' は中心点を含む行, 列までの行数, 列数を表している. 画像を行ごとに走査していき, 輝度値の和が最初に $\frac{1}{2}\pi r^2$ を超えた行が中心を含む行であり, 同時に n' の値もわかる. m' についても同様である.

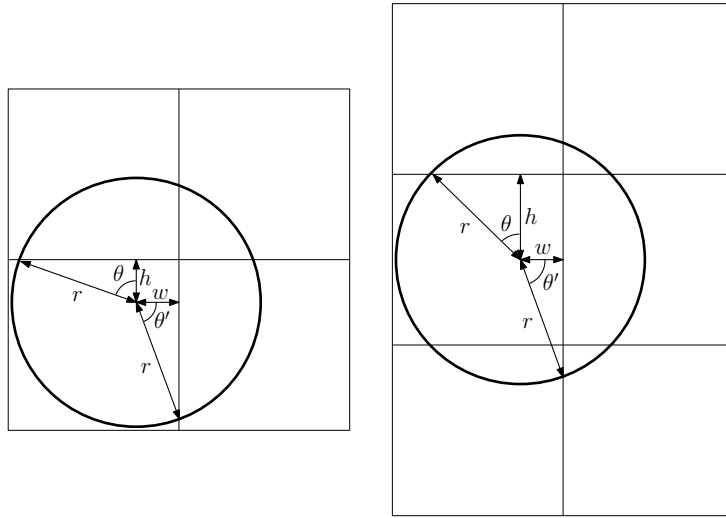


図 3.8: 認識は 2×2 画素にのる円まで可能

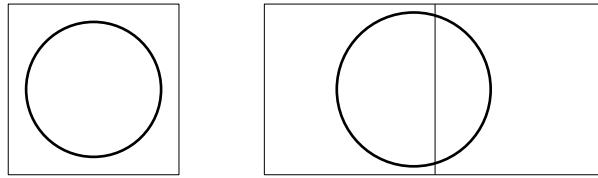


図 3.9: 円の認識できない場合

中心点の y 座標が知りたいならば, h を求めてやれば良い. 当然ながら, $\sum_{i=1}^{(n'-1)m} g_i = \frac{1}{2}\pi r^2$ ならば, $h = 0$ である. $h = r \cos \theta$ より, r, θ が求まれば, h を求めることができる. $\sum_{i=1}^{nm} g_i$ は観測される輝度値から求まるため, 半径 r の値が計算できる. 同様に $\sum_{i=1}^{(n'-1)m} g_i$ も観測される輝度値から求まり, θ だけが未知数の方程式が残る. これは代数的に解けないため, ニュートン法などの数値計算により求める. 以上で, h を計算できることがわかった.

x 座標については, 90° 回転させた同様の方法で求まるため省略する.

3.3.2 認識できない場合

一般の認識法には $n \geq 2, m \geq 2$ という条件が付いている. 図 3.8 に示すように $2 \times 2, 2 \times 3$ 画素の場合のように, 中心点を含む行 (または列) の他に行 (または列) が存在すれば求めることができる. もし, 1 行目 (または 1 列目) に中心点があるならば, 2 行目 (または 2 列目) の重み付け図形面積を用いて計算すれば良い. しかし, 図 3.9 の右に示すように 1 行 2 列のとき, x 座標は求まるが y 座標が本質的に求まらない. この行の中で y 軸方向に円が上下しても観測される値は変化しない. 図 3.9 の左に示すように 1 画素の中に入ってしまうと, この画素の中を円が動き回ったとしても観測される値が変化しないため, x 座標も y 座標も本質的に求まらない.

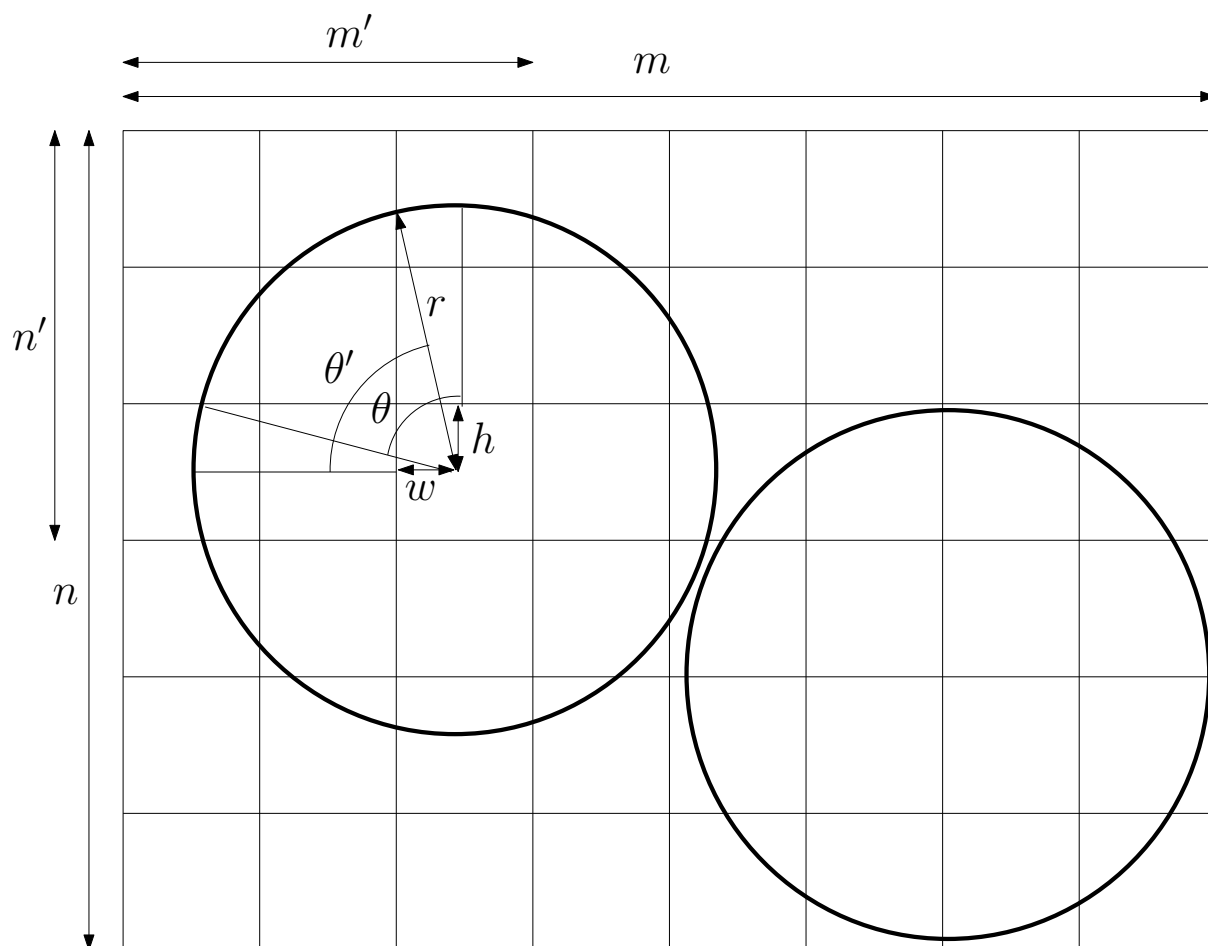


図 3.10: 画像を表す格子平面上の等しい大きさ半径 r の 2 つの円. ただし, はじめに認識する円の中心点を含む画素の行の上端と円の交点により定まる角度を θ , 中心点を含む画素の列の左端と円の交点により定まる角度を θ' , h は円の中心から中心を含む画素の上端までの距離, w は円の中心から中心を含む画素の左端までの距離.

3.4 画素を共有する円の認識

3.4.1 一般の認識法

図 3.10 のように, 大きさの等しい円が 1 つ以上の画素の一部を共有している場合のことを「画素を共有する円」という. どちらか一方の円を認識できたとすると, もう一方はひとつの円の場合と同様に求まるので, ここでは一方の円だけ認識する方法を示す.

はじめに認識する円の中心点を含む画素の行の上端と円の交点により定まる角度を θ , 中心点を含む画素の列の左端と円の交点により定まる角度を θ' とする. 半径 r とこの角度 θ, θ' によって円を定義したとき, 次式を得る. ただし, h は円の中心から中心を含む画素の上端までの距離, w は円の中心から中心を含む画素の左端までの距離である.

$$\sum_{i=1}^{n'-1} \sum_{j=1}^{m'-1} g_{(i-1)m+j} = \frac{1}{4}r^2(\pi - \sin 2\theta - \sin 2\theta' - 2\theta - 2\theta') - hw,$$

$$\sum g_i = 2\pi r^2, \quad h = r \cos \theta' \leq 1, \quad w = r \cos \theta \leq 1,$$

ただし, $r \geq \sqrt{2}$.

$r \geq \sqrt{2}$ ならば, はじめに認識する円の x 座標または y 座標を求めるとき, 少なくともどちらかの座標は他方の円の影響を受けずに, ひとつの円を認識するときと同じ方法で求めることができる. 具体的には $n > m$ のとき, h, θ' を求めることができる. $n < m$ のとき, w, θ を求めることができる. $n = m$ のときは, 1 行目と n 行目の輝度値の合計 ($\sum_{i=1}^m g_i + g_{m(n-1)+i}$) と, 1 列目と m 列目の輝度値の合計 ($\sum_{i=1}^n g_{(i-1)m+1} + g_{(i-1)m+m}$) を比べ, 他方の円の影響を必ず受けない方向を確かめることができる. はじめに認識する円の中心点からみて, 45° の位置に他方の円の中心点があるならば, どちらからでも必ず他方の円の影響を受けずに w, θ も, h, θ' も求めることができる.

上記の方法により, w, θ だけが求まったとする. 次は中心点を含む行を求めたい. 中心点が含まれる行は, $m' - 1$ 列までを使い, 画像を走査していくと, 輝度値の和が最初に $\frac{1}{4}\pi r^2 - \frac{1}{2}(w^2 \tan \theta + r^2(\frac{\pi}{4} - \theta))$ を超えた行が中心を含む行であり, 同時に n' の値もわかる.

$h = r \cos \theta'$ と置いたとき, θ' だけが未知数となる方程式が残る. この θ' をニュートン法などの数値計算で解いてやると, h, θ' についても求めることができる.

先に h, θ' が求まった場合も, 90° 回転させた同様の解き方で w, θ を求めることができる.

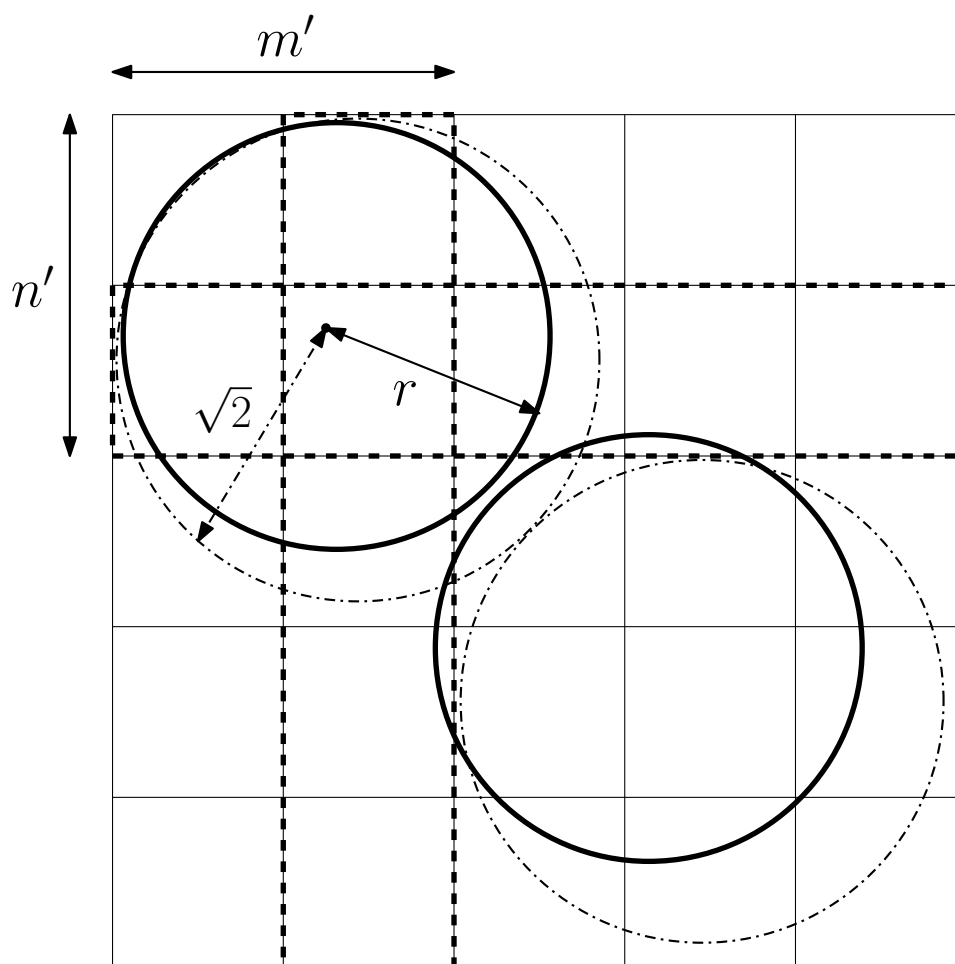


図 3.11: 画素を共有する円の認識できない可能性がある場合

3.4.2 認識できない可能性がある場合

一般の認識法には $r \geq \sqrt{2}$ という条件が付いている． $r < \sqrt{2}$ だとすると，はじめに認識する円の中心点を含む行と列の両方に，他方の円の一部が入る可能性がある．画素を共有する円は，はじめに認識する円の x または y 座標を，列（または行）ごとに足し合わせて中心点を含む列（または行）を発見して，先に求めた x か y 座標の情報を用いてもう一方を求めるが，その行と列の両方に，他方の円の一部が入ってしまう場合，先にどちらか一方を求めることができない．

図 3.11 に示すように， $r \geq \sqrt{2}$ の円ならば，そのような問題は発生しない．はじめに認識する円の中心点を含む行と列の両方に，他方の円の一部が入るならば，二つの円は重なっている．

第4章 量子化誤差なしの任意色図形認識

図形色を a 、背景色を b とし、 b の値は図形近辺の背景色を持つ画素から検出できるとする。すべての画素の輝度値から b を引くと、図形色は $(a - b)$ 、背景色は 0 になる（ただし $a > b$ ）。ここでは簡略化のために、図形色を α 、背景色を 0 とみなして考える。任意色図形の輝度値は α の重み付き面積と考えることができるため、 α を求めることができるのであれば、白黒図形の認識方法のほんの少しの改良で、任意色図形の認識が可能となる。大きな図形の場合、 α を求めることは簡単であるが、図形が小さくなったいくつかのパターンのとき、 α を求めることは困難になる。

本章では白黒図形の認識において、一般化された方法の求めることができる制約条件上の全てで、任意色図形の認識をどのように行うかを示す。

問題 2. 観測される輝度値 $\{g_i\}$ のみを用いて、十字と円（図形が α 、背景が 0）の中心点と正確な形状を求めよ。

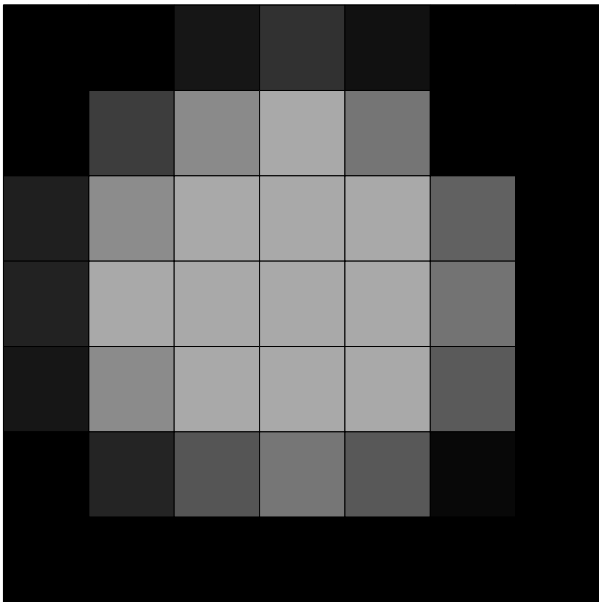


図 4.1: 入力：デジタル画像の輝度値 $\{g_i\}$.

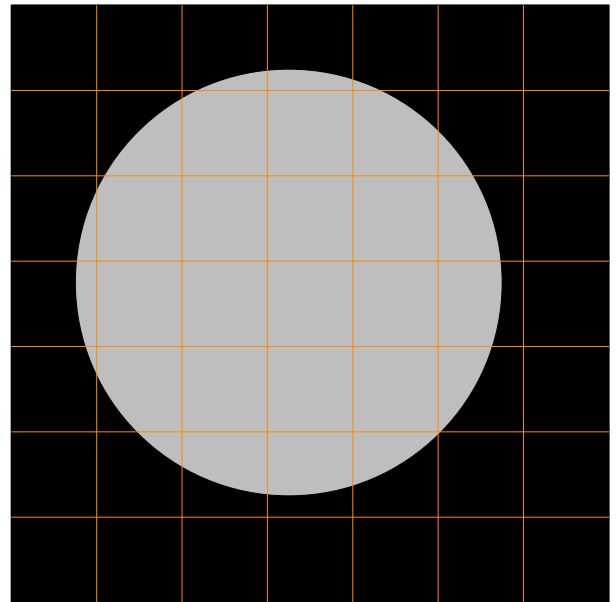


図 4.2: 出力：対象図形の正確な形状（円の場合なら中心点座標と半径）。

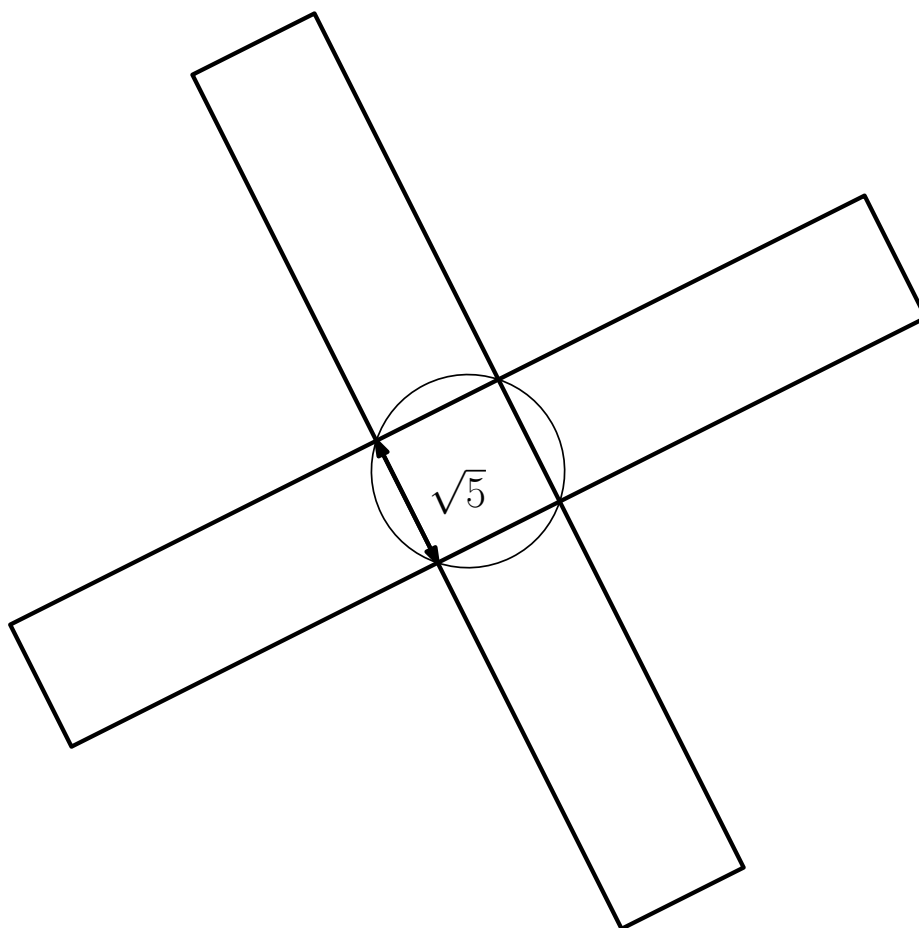


図 4.3: 中心部の幅が $\sqrt{5}$ の回転十字. 回転十字は $w \geq \sqrt{5}$ ならば, 十字がいくら回転してもちょうど α の値をもつ画素が少なくとも 1 つは存在する.

4.1 十字の認識

白黒図形のとくと同様に, 対象図形のみを含む最小の画素数で作られる矩形において, 十字の横棒の上端から, 上端を含む画素 g_A , 十字の横棒の下端から下端を含む画素 g_B と定義する. このとき, g_A, g_B の画素間に別の画素が存在していたとすると, その中間の画素は必ず α の値をもっている. α の値を持つ画素が存在しない場合, g_A, g_B の比のみが中心を求めるために必要であるため, $\alpha = 1$ と仮定しても中心点を求めることができるが, α は求めることができない.

4.2 回転十字の認識

図 4.3 に示すように, $w \geq \sqrt{5}$ ならば, 回転十字を構成する長方形 2 つが重なっている部分に一辺が $\sqrt{5}$ 以上の正方形が存在する. この正方形の内接する円が十字の中心に描ける. よって, 回転十字の θ がどんな値をとったとしても, その円の中には必ず α の値をもつ画素が存在する.

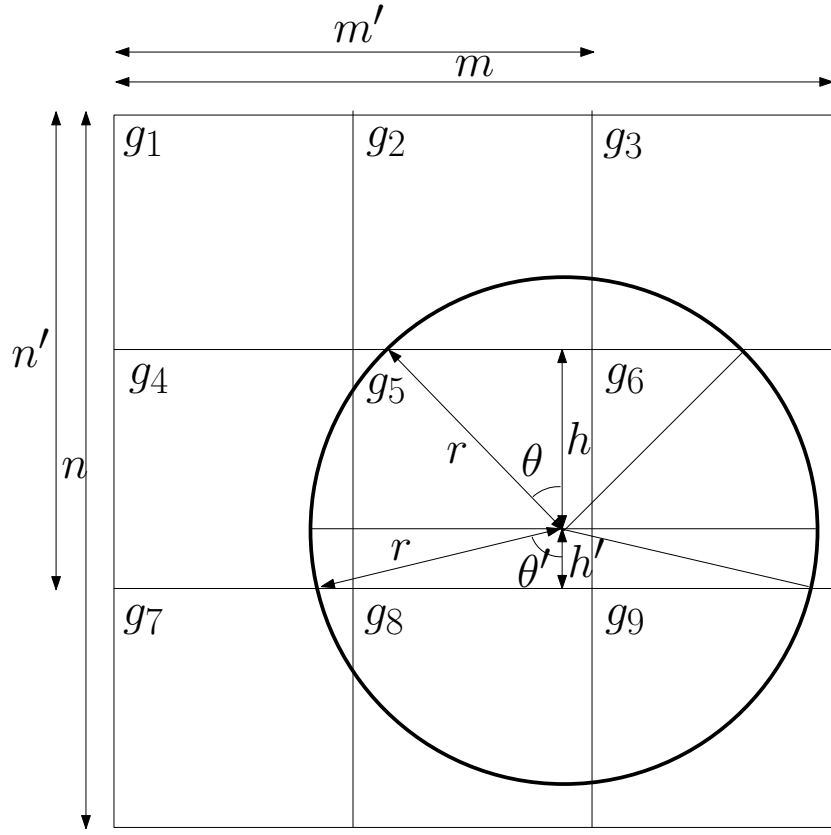


図 4.4: 画像を表す格子平面上の α の値をもつ画素が存在しない半径 r の円. ただし, θ は中心点を含む画素の行の上端と円の交点により定まる角度, h は円の中心から中心を含む画素の上端までの距離, θ' は中心点を含む画素の行の下端と円の交点により定まる角度, h' は円の中心から中心を含む画素の下端までの距離.

4.3 円の認識

$n \geq 3, m \geq 3$ の条件でも, 小さい円するとき, α の値をもつ画素が存在しないパターンがいくつか存在する. 例えば, 図 4.4 のように 8 画素にまたがるような円の場合である. このようなパターンは 12 画素以下にまたがる場合に存在し, 13 画素以上にまたがる場合には存在しない. このことを示すために次の定理を証明する.

定理 4.3.1 13 画素以上にまたがる円では, 少なくとも 1 つの画素の全体が必ず図形に覆われる.

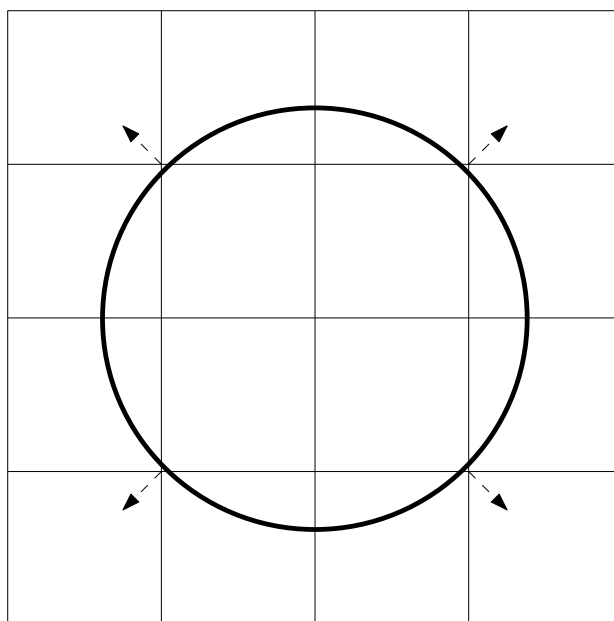


図 4.5: どの画素も図形に覆われない最も大きい円（これ以上，ほんの少しでも円を大きくすると，全体を覆われる画素が出現する）。

証明 1 正方格子平面において，13 以上の単位正方形にまたがり，かつ，どの単位正方形も完全に覆わないような真円は存在しないことを示したい．まず，中心点が含まれる単位正方形を考える．そこにある真円の半径 r を 0 から大きくしていくと考えたとき，初めて図形により全体を覆われる単位正方形は必ず中心点が含まれる単位正方形である．中心点が含まれる単位正方形が真円により全体を覆われないときの最大の半径 r を得るために，中心点を中心点が含まれる単位正方形の最も端（単位正方形の頂点のどこか）に置く．このとき，半径 r を大きくしていくと，中心点が含まれる単位正方形が初めて真円により全体を覆われる半径 r では，12 の単位正方形にまたがる円になっており，13 以上の単位正方形にまたがることは不可能である．よって，正方格子平面において，13 以上の単位正方形にまたがり，かつ，どの単位正方形も完全に覆わないような真円は存在しない．（証明終）

この証明を図に示すと，図 4.5 のようになる．このときの円の半径 r は $\sqrt{2}$ よりも，ほんの少し小さい． $r \geq \sqrt{2}$ ならば，少なくとも 1 つの画素の全体が必ず図形に覆われる．

ところで，定理 4.3.1 を使わなくても，入力画像が α の値をもつ画素を持っているパターンを，ほとんど発見することができる．

ある画素において 8 近傍のすべての輝度値が 0 よりも大きいならば，必ずその画素の全体が図形により覆われている．そのため，8 近傍の輝度値がすべて 0 よりも大きい画素は α の値をもつ．そのような画素をもつ入力画像は α の値を求めることができる．しかし， α の値の画素が存在するかわからない場合は次のように異なる方法で求めなければならない．

図 4.4 のように中心点を含む画素の行の上端と円の交点により定まる角度を θ とし、中心点を含む画素の行の下端と円の交点により定まる角度を θ' とする。半径 r とこの角度 θ によって円を定義したとき、次式を得る。ただし、 h は円の中心点から、中心を含む画素の上端までの距離であり、 h' は円の中心点から、中心を含む画素の下端までの距離であり、 $h + h' = 1$ である。

$$\frac{\sum_{i=1}^{m(n'-1)} g_i}{\frac{1}{2} \sum g_i - \sum_{i=1}^{m(n'-1)} g_i} = \frac{2\theta - \sin 2\theta}{\sin 2\theta + \pi - 2\theta}, \quad \frac{\sum_{i=mn'+1}^{nm} g_i}{\frac{1}{2} \sum g_i - \sum_{i=mn'+1}^{nm} g_i} = \frac{2\theta' - \sin 2\theta'}{\sin 2\theta' + \pi - 2\theta'},$$

$$r = \frac{1}{\cos \theta + \cos \theta'}, \quad \alpha = \frac{\sum g_i}{\pi r^2}, \quad h = r \cos \theta \leq 1, \quad h' = r \cos \theta' \leq 1.$$

θ, θ' は、それぞれニュートン法などの数値計算により解くことができる。すると、 r, α, h が求まり、中心点の y 座標もわかる。

x 座標については、 90° 回転させた同様の方法で求まるため省略する。

4.4 量子化誤差なしの図形認識の結果

以下の条件のとき、任意色図形の認識アルゴリズムについて一般化することができた。ただし、小さな十字は α が求まらないことがある。表 4.1 にその結果をまとめる。

表 4.1: 認識方法を一般化した任意色図形とその認識条件

図形名	条件
十字	$w \geq 1, h \geq w + 4$
回転十字	$w \geq \sqrt{5}, h \geq \frac{8}{\cos \theta} + w$
円	$n \geq 2, m \geq 2$
画素を共有する円	$r \geq \sqrt{2}$

第5章 量子化誤差ありの図形認識

これまでに使用していた CCD カメラ入力モデルでは、輝度値に全く誤差がないという仮定をしていたが、実際の CCD カメラからの入力には量子化による誤差がある。量子化後のデジタル値で使用されるビット数を N とし、輝度値は有効でない値が切り下げられると仮定すると、輝度値 $\{g_i\}$ の実際に取りうる $\{g'_i\}$ は以下の範囲である。(例：図 5.1)

$$\begin{aligned} g_i = 2^N - 1 \text{ ならば, } g'_i &= 1 \\ g_i \neq 2^N - 1 \text{ ならば, } \frac{1}{2^{N-1}}g_i &\leq g'_i < \frac{1}{2^{N-1}}(g_i + 1) \\ \text{ただし, } 0 &\leq g_i \leq 2^N - 1 \end{aligned}$$

本章の目的は量子化誤差の含まれる輝度値 $\{g_i\}$ のみが入力として与えられたとき、中心点座標および、図形の形状の取りうる区間について示すことである。十字の認識については、線形境界検出法の量子化誤差の区間がわかれば、十字の中心点および形状の取りうる区間も簡単にわかる。そして、線形境界検出法の量子化誤差ありの場合の区間の求め方については文献 [1] で示されているため、本章では量子化誤差ありの任意色の円の認識を行う方法のみを示す。

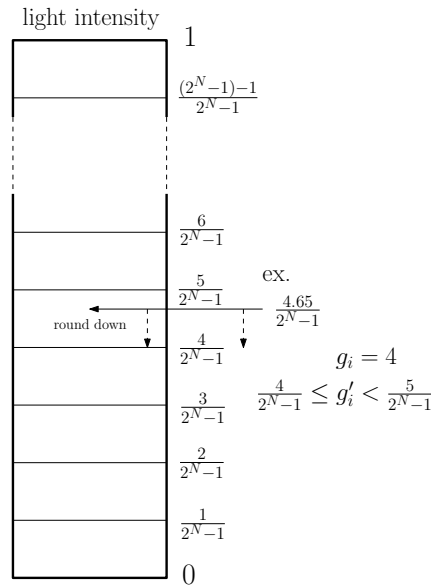


図 5.1: 例: 実際に画素へ入る光子量 (知ることができない値) が $\frac{4.65}{2^{n-1}}$ ならば、観測値は $\frac{4}{2^{n-1}}$. 切り捨てが行われていると仮定するため、実際の光子量は $\frac{4}{2^{n-1}}$ 以上、 $\frac{5}{2^{n-1}}$ 未満だとわかる。

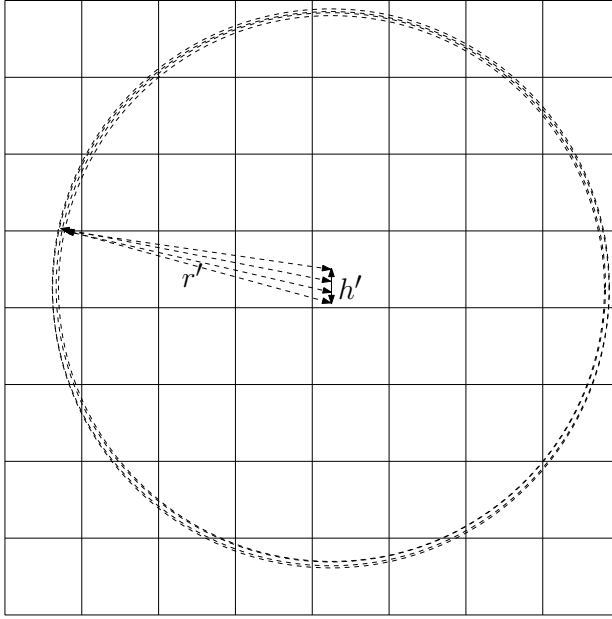


図 5.2: 量子化誤差ありの円. 範囲 h' が, ある

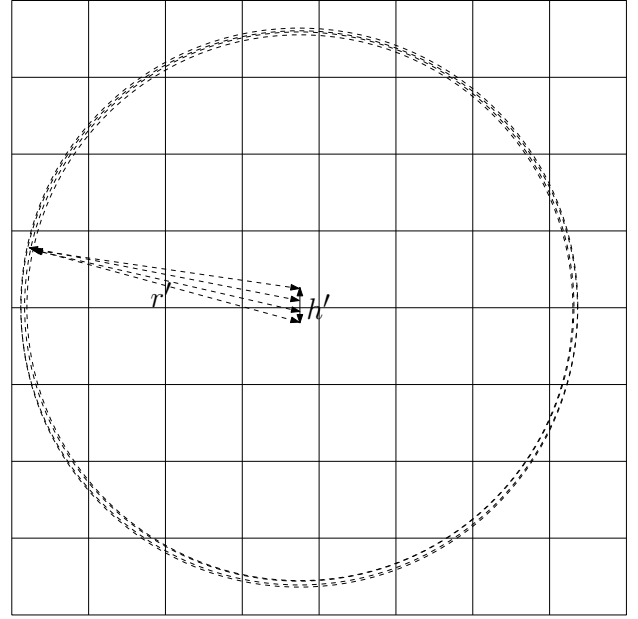


図 5.3: 量子化誤差があるとき, 範囲 h' は 2 行 1 行に収まっている.

5.1 量子化誤差無しの方法に基づく円の中心点の区間検出

前章の量子化誤差無しの方法に基づいて, 量子化誤差ありの円の中心点を取りうる区間を求める.

円の半径の区間 r' は全体の面積の最小 ($\sum \frac{1}{2^N-1} g_i$) と最大 ($\sum \frac{1}{2^N-1} (g_i + 1)$) および, 円の面積の公式 $\alpha\pi r^2$ を用いて次のように示す.

$$\sqrt{\frac{\sum \frac{1}{2^N-1} g_i}{\alpha\pi}} \leq r' < \sqrt{\frac{\sum \frac{1}{2^N-1} (g_i + 1)}{\alpha\pi}}.$$

しかし, 図 5.2 のような量子化誤差ありの円において, 中心点座標の区間を調べるとき, r' をさらに狭めることができる. y 座標の区間について示すとすると, 量子化誤差により考えられる円が最も上側に位置するパターンと, 最も下側に位置するパターンを考える.

中心点座標が中心点を含む行の上辺より下辺に近ければ, 中心点を含む行は円の半分より下側よりも上側に大きく影響するため, g_1 から中心点を含む行までの全ての画素が最大値, 以降が最小値のとき円が最も上側に位置する. また, g_1 から中心点を含む行までの全ての画素が最小値, 以降が最大値のとき円が最も下側に位置する. 中心点座標が中心点を含む行の下辺より上辺に近ければ, 同様に「 g_1 から中心点を含む行のひとつ上の行まで」で分けて求める.

中心点座標が中心点を含む行のどの付近に存在するかは、中心点を含む行より上側と下側のそれぞれで作られる扇の面積を比べればわかる。上側扇、下側扇の面積にも取りうる区間があることに注意する。上側扇、下側扇の面積の区間が重なっていなければ、 h は中心点を含む行の上辺と下辺の等距離線をまたがない範囲をとる。中心点が中心点を含む行の上辺と下辺のどちらに近いかわかる。これがわかると r' を非常に小さな区間に狭めることができる。中心点が中心点を含む行の下辺に近いとき、 r' は以下の区間である。

$$\sqrt{\frac{\sum_{i=1}^{n'm} \frac{1}{2^N - 1} g_i + \sum_{i=n'm+1}^{nm} \frac{1}{2^N - 1} (g_i + 1)}{\alpha\pi}} \leq r' < \sqrt{\frac{\sum_{i=1}^{n'm} \frac{1}{2^N - 1} (g_i + 1) + \sum_{i=n'm+1}^{nm} \frac{1}{2^N - 1} g_i}{\alpha\pi}}.$$

同様に、中心点が中心点を含む行の上辺に近いとき、 r' は以下の区間である。

$$\sqrt{\frac{\sum_{i=1}^{(n'-1)m} \frac{1}{2^N - 1} (g_i + 1) + \sum_{i=(n'-1)m+1}^{nm} \frac{1}{2^N - 1} g_i}{\alpha\pi}} \leq r' < \sqrt{\frac{\sum_{i=1}^{(n'-1)m} \frac{1}{2^N - 1} g_i + \sum_{i=(n'-1)m+1}^{nm} \frac{1}{2^N - 1} (g_i + 1)}{\alpha\pi}}.$$

実際には r' はこの区間以外も取りうるが、 h の区間を求めるとき、区間外の r' を使用しても h は h の最大値よりも大きくなり、最小値よりも小さくならない。

上側扇、下側扇の面積の区間が重なるとすると、 h は中心点を含む行の上辺と下辺の等距離線をまたぐ範囲を取り、上辺と下辺のどちらに近いかわかる。この場合は得られる h の区間は大きくなるが、上辺に近い場合、下辺に近い場合の両方を試して r' , θ を計算し、 h の最大と最小を決めなければならない。

以上の方法で求めた r' の最小値と最大値を用いて、 h の区間 h' を求めたい。 θ については h を最小にする角度 θ' (円は上寄り) を次式で数値計算により求める。ただし、この式では r' の最小値を用いる。

$$\sum_{i=1}^{(n'-1)m} \frac{1}{2^N - 1} (g_i + 1) = \alpha r'^2 (\theta' - \frac{1}{2} \sin 2\theta').$$

同様に、 h を最大にする角度 θ'' (円は下寄り) を次式で数値計算により求める。ただし、この式では r' の最大値を用いる。

$$\sum_{i=1}^{(n'-1)m} \frac{1}{2^N - 1} g_i = \alpha r'^2 (\theta'' - \frac{1}{2} \sin 2\theta'').$$

続いて、求められた r' の最小値, 最大値, θ' , θ'' から得られる h' は以下の区間である. ただし, 左辺 r' は最小値, 右辺 r' は最大値を用いる.

$$r' \cos \theta' \leq h' < r' \cos \theta''.$$

この方法を使用するとき, h' の取りうる区間が中心点を含む行の上辺または下辺をまたぐ区間をとる可能性があることにも注意しなければならない. 何行何列にまたがり得るかは量子化ビット数により変化する. 本研究では 8bit または 12bit で量子化を行うが, この検出法を実装して試したところ, 区間は 1 行 (または列) のみか, 多くとも 2 行にまたがる場合しか起こり得ないようである. つまり, h' , w' が 1 を超えるようなことはないだろう. 2 行にまたがる場合は, 少し面倒だが 1 行の場合の求め方を, またがった両方の行に対して行えば良い.

以上の方法により, 中心点の取りうる区間の検出を行うことができた. しかし, この方法で求めた区間はあまり正確ではない, 特に円が大きい場合や, 量子化ビット数が少ない場合には必要以上に大きな領域が出力される可能性がある.

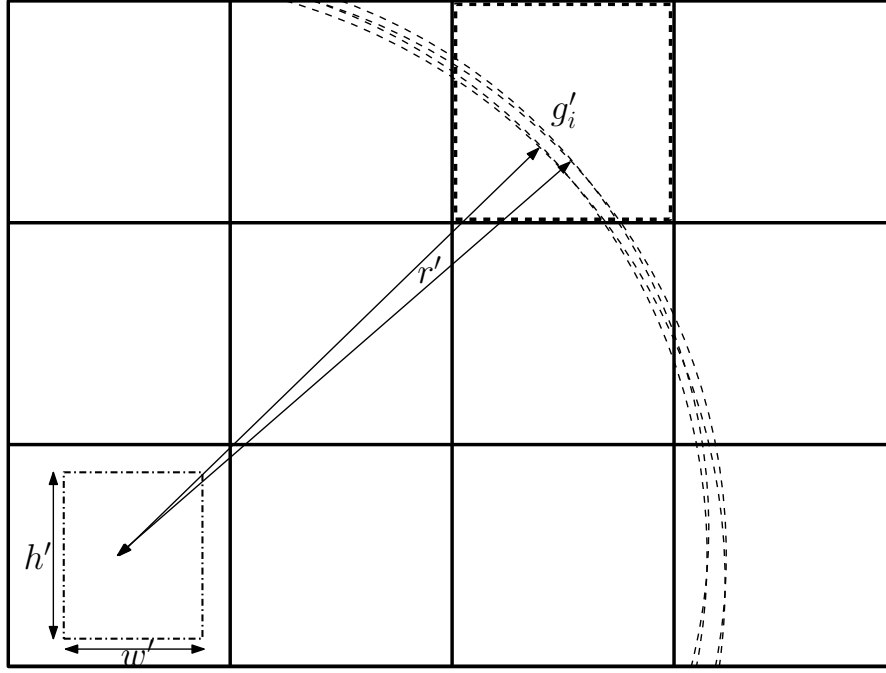


図 5.4: 中心点のある区間にある座標点と r から境界の含まれる画素中の図形領域の計算.

5.2 1画素ごとの値を用いた円の中心点の区間縮小

本節では、得られた h' , w' , r' を入力として、任意の精度で中心点の含まれる領域を縮小する．その出力結果は正確には凸包領域（凸包を構成する点 (x_1, y_1) , (x_2, y_2) , $\dots$, (x_n, y_n) ）として得られるが，出力結果の使用を簡単にするため，凸包を囲う最小包含円（座標 (x, y) と，半径 r ）についても求めることが目的である．

何故，量子化誤差無しの区間検出法が正確に区間を検出することができないかというと， x, y それぞれの座標を求めるとき，円の半分近くの領域の値を合計した値を用いているためである．そのため，さらに正確に求めたいならば，図 5.4 のように 1 画素ごとの値が，得られた区間の任意の位置で円を描いたときと同じ位置の 1 画素の値と一致する可能性があるか確かめれば良い．このために，後に示す任意の円が任意の画素でとる値を求めるアルゴリズムが必要である．

実際には得られた区間の任意の位置で計算することは難しい．そこで， h' , w' で表される中心点の取りうる矩形状の区間を任意の精度で区切り，それにより作られる座標点と半径 r を用いることにする．この方法ならば計算機を使用して計算可能である．ただし， h' , w' における最大・最小の値を用いるとは限らないため， r' は本章の初めに示した以下の区間である．

$$\sqrt{\frac{\sum \frac{1}{2^{N-1}} g_i}{\alpha \pi}} \leq r < \sqrt{\frac{\sum \frac{1}{2^{N-1}} (g_i + 1)}{\alpha \pi}}.$$

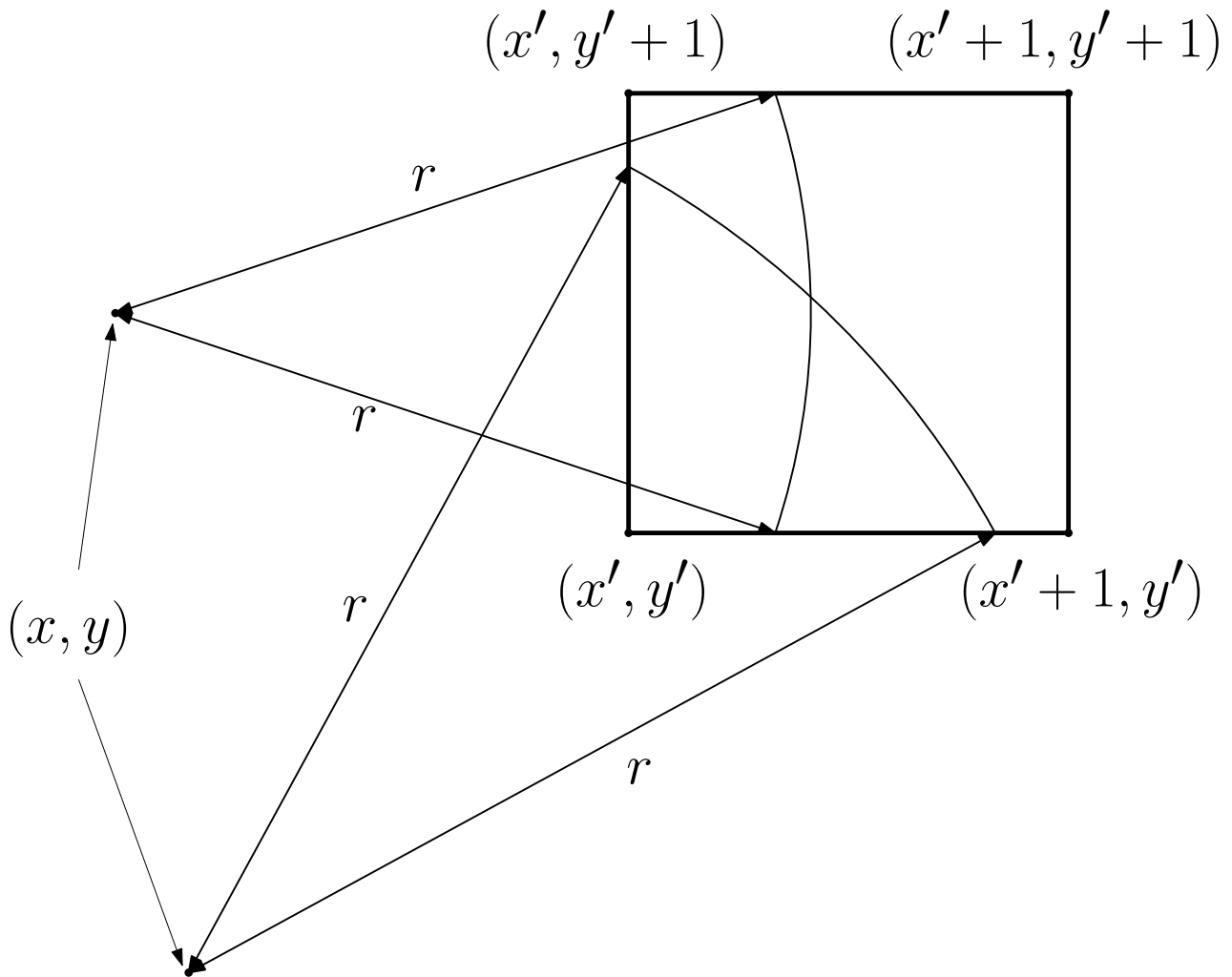


図 5.5: 点 (x, y) と画素を表す座標 (x', y') を入力したときの 2 パターン

5.2.1 任意の円が任意の画素でとる値を求めるアルゴリズム

中心点座標 (x, y) と半径 r ，目標画素の左下端点座標 (x', y') を入力としたとき，目標画素に含まれる図形の面積を計算する．画素は 1×1 の正方形としているため，目標画素の左上端点座標は $(x', y' + 1)$ ，右下端点座標は $(x' + 1, y')$ ，右上端点座標は $(x' + 1, y' + 1)$ である．円の中心点と目標画素の位置関係は斜め方向の場合と，水平垂直方向の場合の 2 つのパターンが考えられる．全方位を考えると上，下，左，右，右上，右下，左下，左上の 8 パターンがあるが，本質的には斜め方向か水平垂直方向かの位置関係がわかれば良いため，例えば図 5.5 のように右上と右のみ考え，入力を 90° ， 180° ， 270° だけ回転させて対応させれば良い．

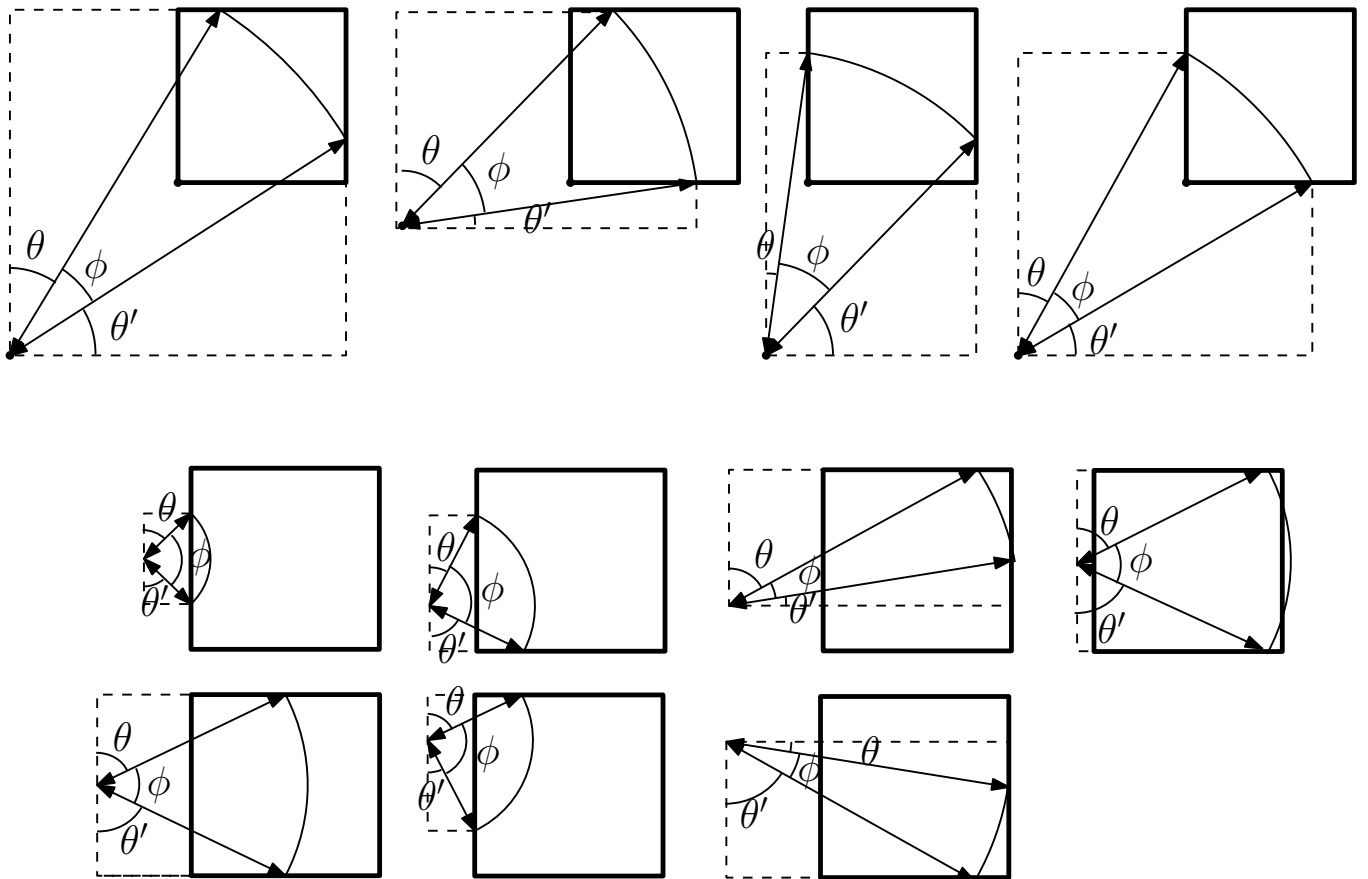


図 5.6: 点と画素の位置関係の場合分け

斜め方向と、水平垂直方向の2パターンは円周が目標画素の上辺、下辺、左辺、右辺のどこに接触しているかにより、更に斜め方向は4つ、水平垂直方向は7つに図5.6のように場合分けされる．実装においてどの辺に接しているかの判別は中心点座標と半径から、目標画素の正方形の角の4点のどれを含んでいるかで場合分けすることができる．

中心点および、目標画素と円周が接触する2点（1つだけ4点の場合もある）で作る扇形を考えたとき、扇形の中心角を ϕ とする．中心角からみて扇の左側半径を斜辺とする直角三角形を考えたとき、扇形の中心角に隣接する角度を θ ，中心角からみて扇の右側半径を斜辺とする直角三角形を考えたとき、扇形の中心角に隣接する角度を θ' とする．

斜め方向のパターンおよび、水平垂直方向であり扇形の2つの半径が上と右または右と下に接触するとき $\phi = \frac{\pi}{2} - \theta - \theta'$ である．それ以外の水平垂直方向のパターンのとき $\phi = \pi - \theta - \theta'$ である．

斜め方向のパターンの場合,
 画素の上辺に接点があるとき, $\theta = \cos^{-1}(\frac{y'+1-y}{r})$,
 画素の左辺に接点があるとき, $\theta = \sin^{-1}(\frac{x'-x}{r})$,
 画素の右辺に接点があるとき, $\theta' = \cos^{-1}(\frac{x'+1-x}{r})$,
 画素の下辺に接点があるとき, $\theta' = \sin^{-1}(\frac{y'-y}{r})$
 である.

水平垂直方向のパターンの場合,
 上側の接点が画素の左辺にあるとき, $\theta = \sin^{-1}(\frac{x'-x}{r})$,
 上側の接点が画素の上辺にあるとき, $\theta = \cos^{-1}(\frac{y'+1-y}{r})$,
 上側の接点が画素の右辺にあるとき, $\theta = \cos^{-1}(\frac{x'+1-x}{r})$,
 下側の接点が画素の左辺にあるとき, $\theta' = \sin^{-1}(\frac{x'-x}{r})$,
 下側の接点が画素の上辺にあるとき, $\theta' = \cos^{-1}(\frac{y'-y}{r})$,
 下側の接点が画素の右辺にあるとき, $\theta' = \cos^{-1}(\frac{x'+1-x}{r})$
 である.

以上をふまえると目標画素に含まれる図形部分の面積は表 5.1 のように計算できる.

表 5.1: 目標画素に含まれる図形の面積を計算する式

位置関係	目標画素に接する辺	目標画素の値を求める式
斜め	上, 右	$\frac{1}{2}(\phi - \sin \phi) + 1 - \frac{1}{2}(1 - r \sin \theta + x')(1 - r \sin \theta' + y')$
斜め	上, 下	$\frac{1}{2}(\phi - \sin \phi) + \frac{1}{2}(r \sin \theta + r \cos \theta' - 2x')$
斜め	左, 右	$\frac{1}{2}(\phi - \sin \phi) + \frac{1}{2}(r \cos \theta + r \sin \theta' - 2y')$
斜め	左, 下	$\frac{1}{2}(\phi - \sin \phi) + \frac{1}{2}(r \cos \theta - y')(r \cos' - x')$
水平垂直	左, 左	$\frac{1}{2}(\phi - \sin \phi)$
水平垂直	上, 下	$\frac{1}{2}(\phi - \sin \phi) + \frac{1}{2}(x + r \sin \theta' - x' + x + r \sin \theta - x')$
水平垂直	左, 下	$\frac{1}{2}(\phi - \sin \phi) + \frac{1}{2}(x + r \sin \theta' + x')(y' - y + r \cos \theta)$
水平垂直	上, 左	$\frac{1}{2}(\phi - \sin \phi) + \frac{1}{2}(x + r \sin \theta - x')(y' - y + r \cos \theta')$
水平垂直	上, 右	$\frac{1}{2}(\phi - \sin \phi) + 1 - \frac{1}{2}(1 - x + r \sin \theta - x')(r \cos \theta - \sin \theta')$
水平垂直	右, 下	$\frac{1}{2}(\phi - \sin \phi) + 1 - \frac{1}{2}(1 - x + r \sin \theta' - x')(r \cos \theta' - \sin \theta)$
水平垂直	上, 右, 右, 下	$[水平垂直(上, 下)] - [水平垂直(左, 左); x' は x' + 1 と置く]$

少し特殊なパターンとして, 最後の水平垂直方向(上, 右, 右, 下)の式は, 水平垂直方向(上, 下)を求める式で計算した値から, x' を $x' + 1$ と置いた水平垂直方向(左, 左)を求める式で計算した値を差し引いたものとなる. ただし, それぞれの ϕ , θ , θ' の求め方が異なることに注意すること.

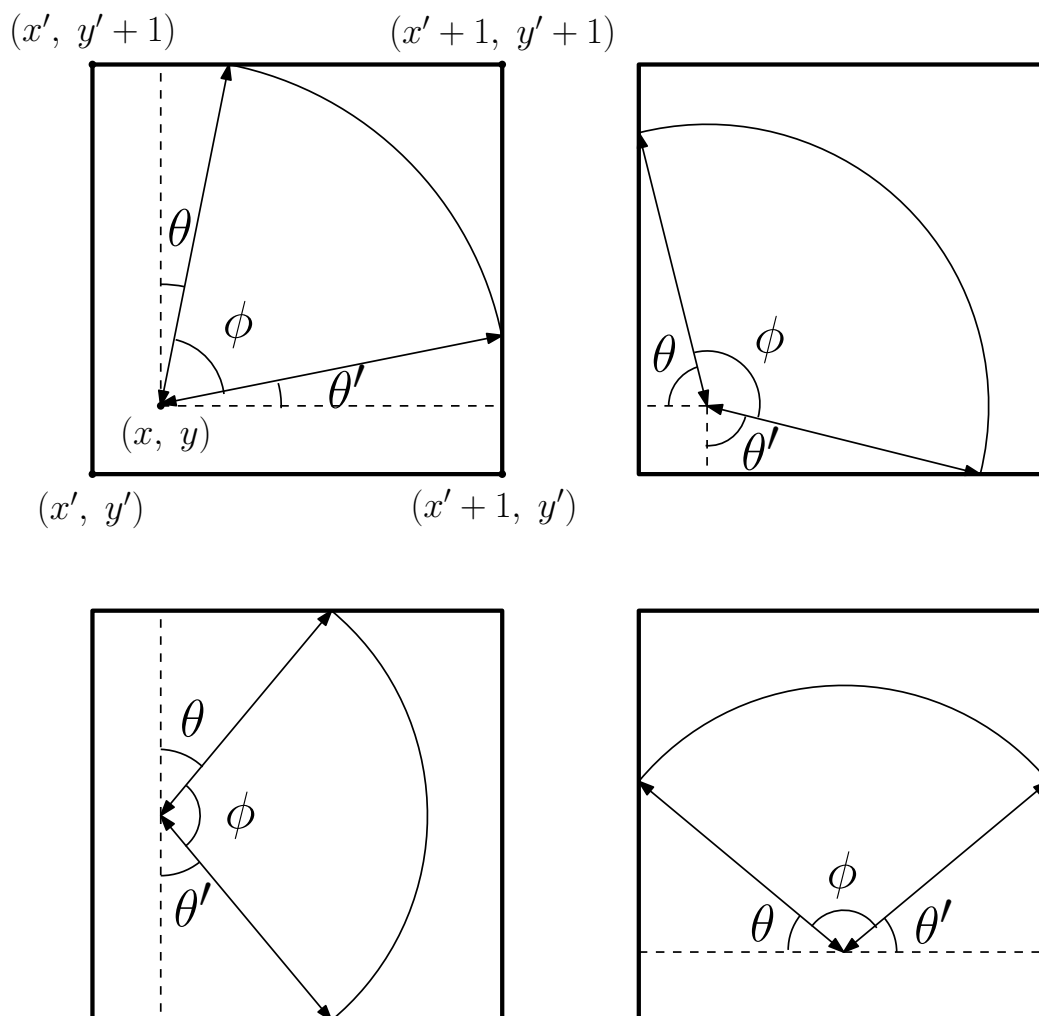


図 5.7: 中心点の含まれる画素中の図形領域の計算

5.2.2 中心点を含む画素が円により完全に覆われていない場合

前章で示した通り、12画素以下にかかる円の場合、中心点を含む画素が円により完全に覆われていないことがある。このとき、中心点を含む画素については、別の方法で画素でとる値を計算しなければならない。

対象となる画素は中心点を含む画素のみであるため、図5.7のように、中心点座標と半径から画素の4つの頂点が円により覆われるかを調べ、画素の頂点の1点が覆われない場合、2点が覆われない場合（ただし、右上端と右下端パターン、左上端と右上端パターンがある）、3点が覆われない場合を考えるだけで良い。4点が覆われないときは、1画素か2画素にまたがる円であり、円を認識することができないパターンである。図5.7は画素の左下頂点が常に覆われるパターンだが、他は回転した形であるので省略する。

角度 ϕ , θ , θ' の定義は中心点を含む画素以外の場合と同様である. ϕ , θ , θ' はそれぞれ次のように求められる.

画素の頂点のうち, 1 点が覆われない場合,

$$\theta = \cos^{-1}\left(\frac{y'+1-y}{r}\right),$$

$$\theta' = \cos^{-1}\left(\frac{x'+1-x}{r}\right),$$

$$\phi = \frac{\pi}{2} - \theta - \theta'$$

である.

画素の頂点のうち, 2 点が覆われない場合,

$$\theta = \cos^{-1}\left(\frac{x-x'}{r}\right),$$

$$\theta' = \cos^{-1}\left(\frac{x'+1-x}{r}\right),$$

$$\phi = \pi - \theta - \theta'$$

(左上端と右上端の 2 点が覆われないパターン)

または,

$$\theta = \cos^{-1}\left(\frac{y-y'}{r}\right),$$

$$\theta' = \cos^{-1}\left(\frac{y'+1-y}{r}\right),$$

$$\phi = \pi - \theta - \theta'$$

(右上端と右下端の 2 点が覆われないパターン)

である.

画素の頂点のうち, 3 点が覆われない場合,

$$\theta = \cos^{-1}\left(\frac{x-x'}{r}\right),$$

$$\theta' = \cos^{-1}\left(\frac{y-y'}{r}\right),$$

$$\phi = \frac{3\pi}{2} - \theta - \theta'$$

である.

以上をふまえると中心点の含まれる画素の図形部分の面積は表 5.2 のように計算できる.

表 5.2: 中心点の含まれる画素の図形部分の面積を計算する式

覆われない点数	画素中の図形部分の面積を求める式
1 点	$\frac{1}{2}r^2(\phi - \sin \phi) + 1 - (1 - r \sin \theta - x + x')(1 - r \sin \theta' - y + y')$
2 点	$\frac{1}{2}r^2(\phi - \sin \phi) + \frac{1}{2}(r \sin \theta + r \sin \theta')$
3 点	$\frac{1}{2}r^2(\phi - \sin \phi) + \frac{1}{2}r^2 \sin \theta \sin \theta'$

5.2.3 区切った点の座標が正しいか調べるアルゴリズム

前節で求めた目標画素に含まれる図形面積がわかれば、縮小したい区間に含まれる任意の点が円の中心点になり得るかを1画素ごとの値を用いて調べることができる。

ある点が円の中心点になり得るか調べるには、その点と半径の最大と最小（区間で与えられている）を用いてそれぞれ円を描画し、実際の入力画像の1画素ごとの輝度値の区間と重なることがあるかを調べれば良い。全ての画素において輝度値の区間と少しでも重なるならば、その点は円の中心点になり得る。逆にどこかひとつの画素において輝度値の区間と全く重ならない画素があったとすると、その座標点は円の中心点にはなり得ない。

縮小したい区間中の任意の点で円の中心点になり得るかを調べる事ができると良いが、そのような点は無限に存在するので不可能である。そのため、ある小さな値 ϵ を定め、横方向の長さが w 、縦方向の長さが h の矩形状の区間を矩形の左上角から順に ϵ ごとに縦横に区切る。このとき区切り線でできる交点を1点ずつ円の中心点になり得るか調べる事にする。交点を順に円の中心点になり得るか調べた結果の例を図5.8に示す。

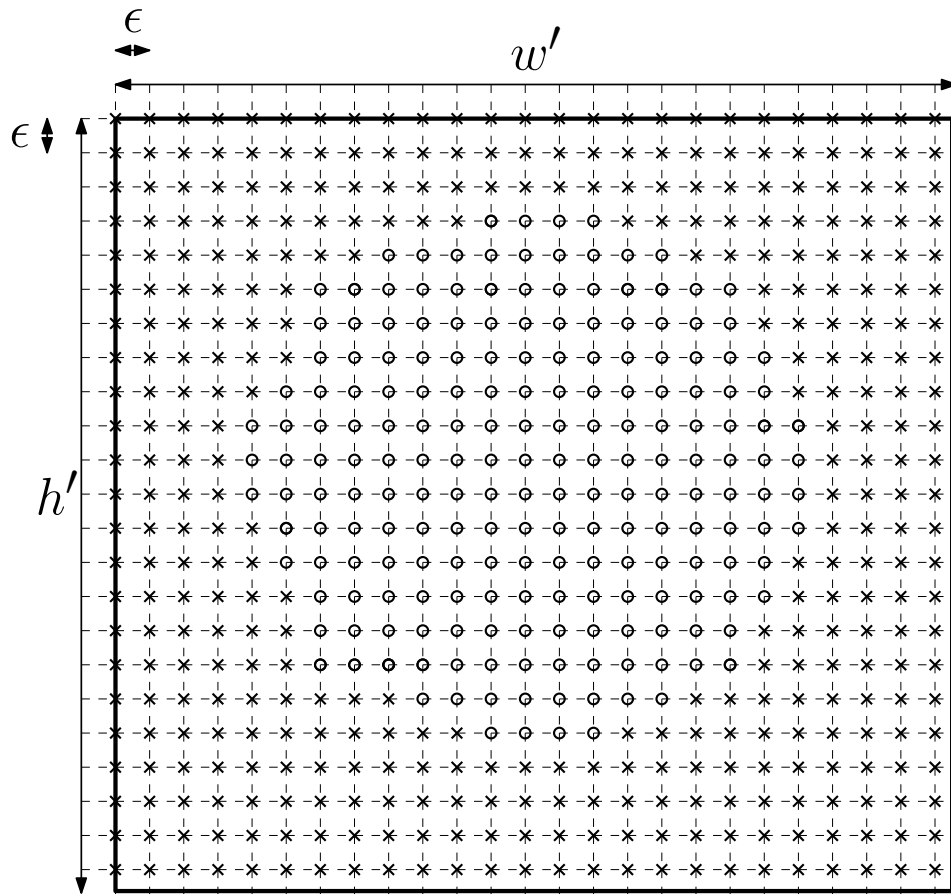


図 5.8: 区間を ϵ で区切り、点ごとに正しい（○）、正しくない（×）座標点を調べた結果。

図 5.9 に示すような最も外側の正しい座標点を用いて作る領域の中で前節のアルゴリズムを使うと，どこに中心点をおいて円を描いたとしても正しい座標点だと判断される．しかし，区間縮小により得る領域は，この領域だけでは不十分である．なぜならば，中心点になり得る点だと判断されたとしても，そのほんの少し外側は中心点になり得る点といえるかは分からない．よって，図 5.10 に示すような円の中心点になり得る点の水平垂直斜め方向に対して 1 つ外側の点を含めた領域が，少し大きい可能性があるが，正しい円の中心点になりうる領域である．

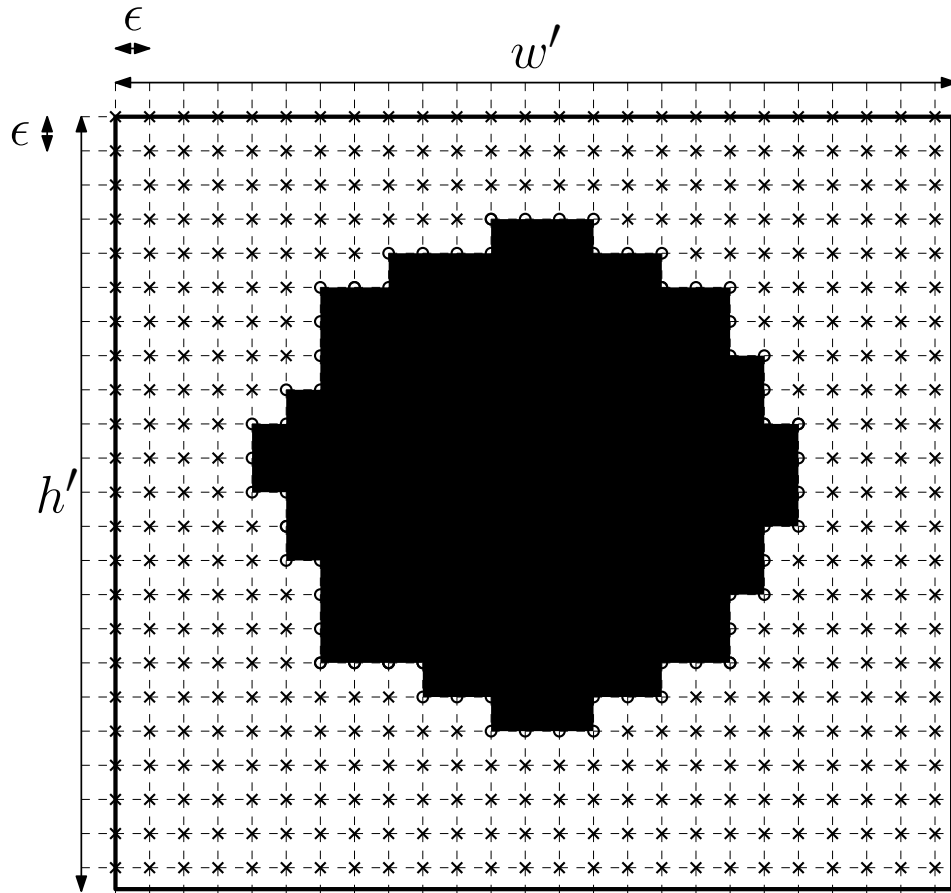


図 5.9: 正しい座標点で作る領域は必ず正しい領域となる．図中の太線は量子化誤差なしと同様の方法で求めた区間の枠．

以上までで求めた領域が正しい円の中心点になりうる領域であり，これ以外の領域は考えなくても良い．この領域を用いて，凸包（図 5.11），最小包含円（図 5.12）を求める．

ところで，1つ外側の点が w' , h' で示される矩形区間よりも外側ならば，矩形中に入るように値を丸めてやると良い．これにより， ϵ ずつ区切ったとき縦横が少し余る問題は回避される．もちろん，拡張した1つ外側の点自身は円の中心点にはなり得ないが，そのほんの少し（ ϵ よりも小さい距離）内側は円の中心点になり得る可能性がある．

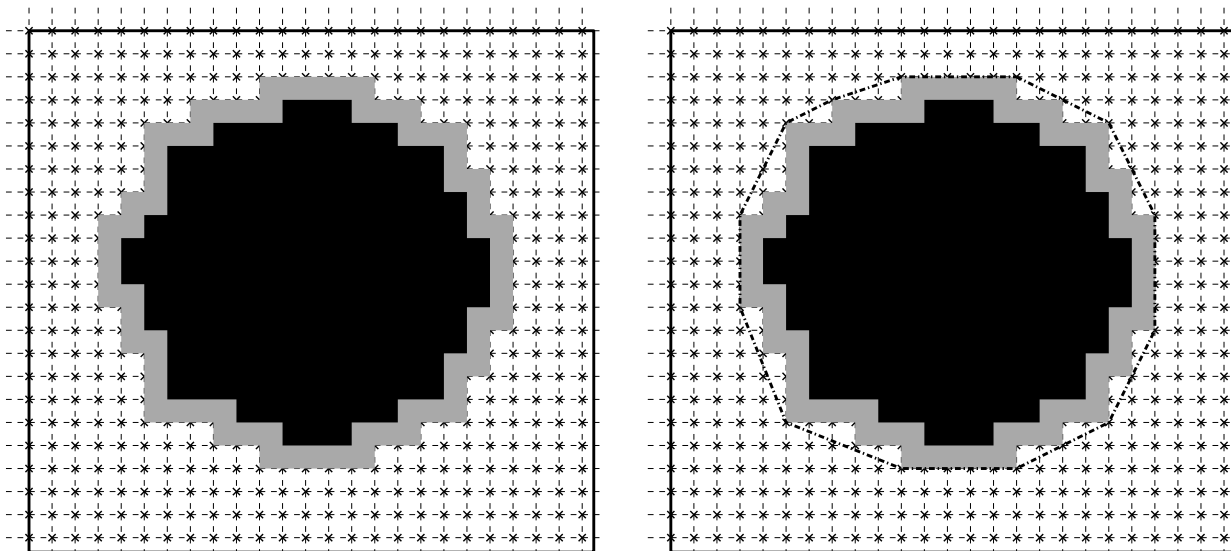


図 5.10: 正しい領域の一つ外側の点で作る領域. 図 5.11: 正しい領域が存在する可能性があるここにまだ正しい領域が存在する可能性がある．領域を囲う凸包の作成.

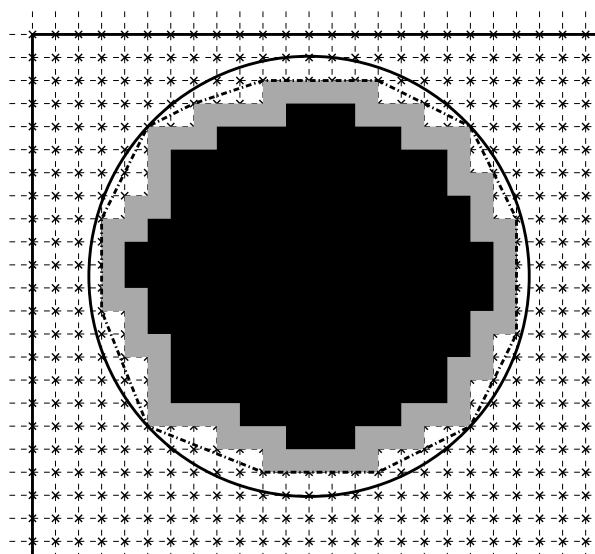


図 5.12: 最終的に求めたい円状領域.

5.2.4 区間を表す凸包を求めるアルゴリズム

これまでの方法により、量子化誤差無しの中心点検出法を応用して得られた矩形形状の区間から ϵ の精度で円の中心点になり得る区間を縮小できたが、この出力結果は全ての外側の座標点であり、応用しづらいデータである。これを扱いやすくするため、出力結果の領域を囲う最小の包含円（中心点座標と半径をもつ）を求めたい。平面上の点集合に対する最小包含円は、その点集合の凸包をつくる座標点だけを考えれば良いので、この節では凸包を求めたい。

凸包の作成は、一般的にはソートを含むため $O(n \log n)$ 時間かかるが、ここでは中心点になりうる点の最も外側の点だけを調べれば良いので、もっと高速に凸包が作成できる。凸包の作成は前節の区切ってできる交点の座標が円の中心点になり得るかのチェックと同時に行うべきである。なぜならば、 ϵ が非常に小さい値で、 w', h' の区間が大きかったとすると、交点数が数億、数兆を超えることがあるので、全ての交点を調べ、データを持つことが難しくなるためである。よって、交点のチェックと凸包作成を同時に行い、凸包を構成する点のデータのみを持つことにする。

まず、前節で見つけた中心点になりうる座標点のうち、最も外側の点のどれかを見つける。これは、点が区切った格子の交差点上に並んでいるため簡単である。どこかスタートとなる点を決めて、中心点になりうる点となりえない点の境界を上下左右に ϵ ずつ移動して（または縦方向か横方向かを決め、二分探索を用いて）発見してやれば良い。

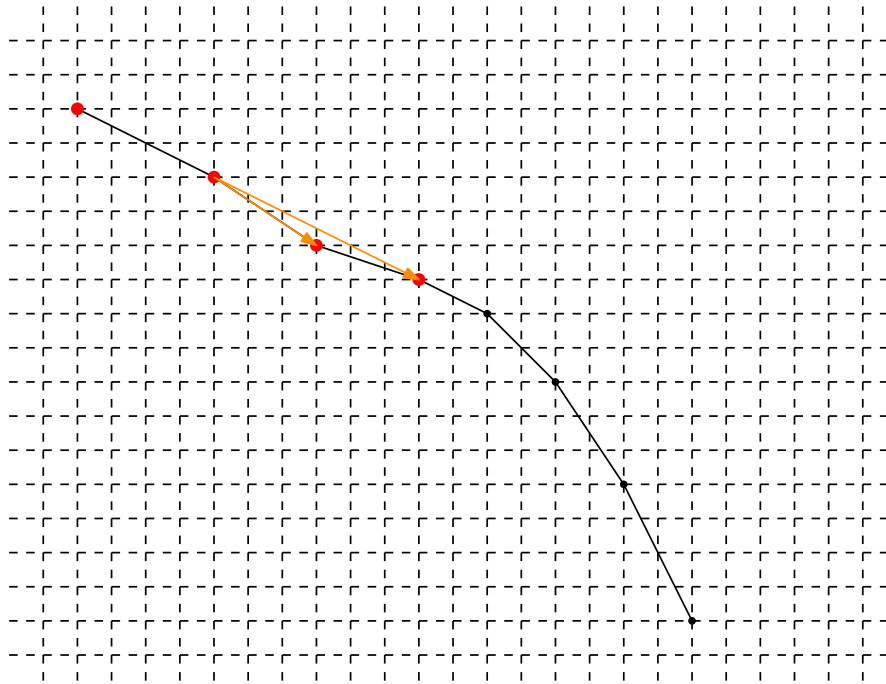


図 5.13: 2つ前の点から現在の点と1つ前のベクトルを考えて必要な点か判定する方法.

中心点になりうる座標点のうち、最も外側の点をひとつ見つけたならば、その点を始点として、領域の外側をなぞるようにして、次々に凸包の候補となる点を発見してゆく。一周して始点まで戻ってくればアルゴリズムは終了である。ただし、凸包の候補点は、明らかに候補から外れれば削除してしまえば良い。例えば、領域を時計回りになぞる場合で考える。新しい候補点を発見したとき、図 5.13 で示すように最後に発見した点から考えて、2つ前の候補点から1つ前の候補点へのベクトルが、2つ前の候補点から最後に発見した点へのベクトルからみて、右側向き、または同じ向きならば、1つ前の候補点は候補から外れる。1つ前の候補点を候補から外すならば、3つ前の候補点から2つ前の候補点へのベクトルと、3つ前の候補点から最後に発見した点へのベクトルの場合を調べて、2つ前の候補点が外れるか判断、更に4つ前、5つ前と遡り、最初に発見した点まで遡り、候補点を減らせるかを調べる。ただし、この候補点の削除の判断は、3点以上の場合しか行えない。また、最初に発見した点はこの方法では候補になりうるかは凸包作成途中では判断できないので、一周した後に判断する。

ここまでで得られたのは、図 5.11 に示したような正しい領域が存在する可能性がある領域を囲う凸包である。この凸包の最小包含円を求めて、図 5.12 に示したような目的の円状領域を得る。

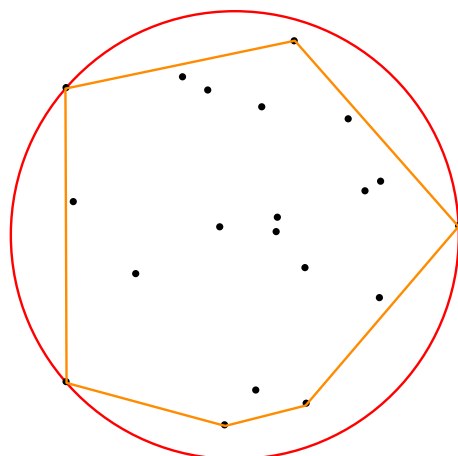


図 5.14: 凸包と最小包含円

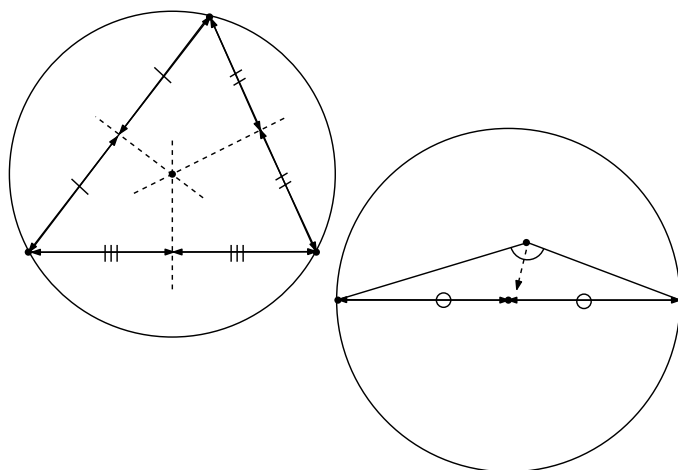


図 5.15: 三角形の最小包含円

5.2.5 最小包含円問題

図 5.14 のように，平面上の有界な閉集合 X に対して X を含む最小の円を X の最小包含円という．

最小包含円を解くための予備知識として必要となるのは，2 点および，3 点を与えられたときの最小包含円の求め方である．2 点を与えられたときの最小包含円は 2 点の midpoint が中心点であり，半径は 2 点どちらかから midpoint までの距離である．3 点を与えられた時の最小包含円は 3 点で作られる三角形が鋭角三角形か鈍角三角形かにより図 5.15 のように場合分けされる．鋭角三角形のとき，三角形の外接円が最小包含円になる．外心は三角形の 3 つの辺の midpoint で辺に直行する直線と考えたとすると，その 3 つの直線が交わる点が外心であり，外心から与えられた 3 点どれかまでの距離が半径である．鈍角三角形のとき，鈍角の対辺の midpoint が中心点であり，中心点から鈍角を作る点以外のどちらかの点までの距離が半径である．

最小包含円問題の解法は、過去に様々な手法が考え出されてきた。例えば、最小包含円は3点が定まれば一意に求まるので、 n 個の点集合の中で全ての3点の組み合わせについて、点集合全体を含んでいるか調べれば、 $O(n^3)$ 時間で求めることができる。乱択アルゴリズムを用いると、 $O(n)$ の期待時間で求めることができる（本研究ではこの手法を採用した）。また、最遠点ボロノイ図を用いると $O(n \log n)$ 時間で求めることができる。

枝狩りの方法としては与えられた点集合の凸包を作成することにより、最小包含円の候補となる点を削減できることがある。これは、最小包含円の円周上に乗る点は必ず凸包上の点となっているためである。今回の場合は凸包の作成を前節で行っているので省略する。

5.3 量子化誤差ありの図形認識の実験

5.3.1 実験内容

今回開発した量子化誤差ありの図形認識アルゴリズムを C 言語で実装し、8bit 量子化時、12bit 量子化時にどのような結果が得られるか実験を行った。

「5.2.1 任意の円が任意の画素でとる値を求めるアルゴリズム」を用いて描くことができる x, y 座標および半径 r で決まる円である、 x, y, r の値はランダムに与え、30 回試した。ただし、入力する基本図形は十分に小さいものを想定しているため、 r は 10.0 までとした。また、例外的なパターンについては手入力し、別途試している。量子化誤差無しの方法に基づく円の中心点の区間検出で得られる区間を縮小するとき、この区間をある小さな値 ϵ で区切る。 ϵ は小さくなるほど計算時間がかかるが、ある程度の小ささより小さくしたとしても、あまり結果が変わらなくなる小ささがある。そのため、 ϵ は十分に良い結果が得られ、かつ現実的な時間で実行できるように適切な値を設定した。

5.3.2 実験結果

出力結果の円の半径の上限と下限は表 5.3 のようになった。ただし、表中の値は画素の大きさ 1×1 に対する大きさである。

表 5.3: 8bit, 12bit 量子化時にランダムに理想的な円を入力した場合の結果

量子化 bit 数	出力結果の円の半径
8bit	0.0014~0.0039
12bit	0.00010~0.00027

第6章 おわりに

6.1 まとめ

本論文では、CCD センサ入力モデルを定め、量子化誤差なしの白黒図形の認識、量子化誤差なしの任意色図形の認識、量子化誤差ありの任意色図形（円のみ）の認識を行った。

今回使用した CCD センサ入力モデルは被写体を撮影して得られるデジタル画像を正方形格子平面に対応させるというものであった。既存のデジタル画像を使った位置決め手法では、このようなモデルを正確に定義するということを行わなかったため、精度の低い位置決めしかできなかった。しかし、今回の基本図形認識の手法ではきちんとモデルを定めたため、高い精度の位置決めを行うことができる結果に繋がった。

量子化誤差なしの基本図形認識の結果をまとめると、十字は $w \geq 1, h \geq w + 4$ （ただし、任意色図形ならば小さな十字は α が求まらないことがある）、回転十字は $w \geq \sqrt{5}, h \geq \frac{8}{\cos \theta} + w$ 、円は $n \geq 2, m \geq 2$ 、画素を共有する円は $r \geq \sqrt{2}$ の条件の下で、基本図形認識の一般化を行うことができた。

量子化誤差ありの円の認識アルゴリズムについては実装を行い、このプログラムを用いて中心点座標が必ず含まれる領域の大きさがどの程度になるか実験により確かめた。その結果、8 ビット量子化の場合には半径 0.0027 程度の円領域の中に中心点座標が存在し、12 ビット量子化の場合には半径 0.00018 程度の円領域の中に中心点座標が存在した。量子化誤差ありの回転十字の認識アルゴリズムは線形境界検出法 [1] を回転十字の輪郭線へ応用すれば良いだけであるので本論文では省略した。

既存の手法では、0.1 画素程度の精度でしか位置決めを行うことができなかった。つまり、1 画素を 10 個のサブピクセルに分割したとき、10 個のうち、どのあたりのサブピクセルに中心点が含まれるかわからなかった。このような既存の結果と、今回の量子化誤差ありの円の認識アルゴリズムを用いた実験で得た結果を比較するならば、出力結果の円の直径をサブピクセルの大きさとして考えれば良い。まとめると、「従来は 0.1 画素程度の精度でしか円の中心点を推定することができなかったが、計算幾何学に基づいた本手法では、量子化ビット数が 8 ビットの場合には 0.0054 画素程度、12 ビットの場合には 0.00036 画素程度の精度を達成することができた」といえる。

ただし、計算幾何学に基づいた本手法では、理想的な形状をもつ基本図形を入力としている点で従来手法とは異なることに注意すべきである。

謝辞

浅野哲夫教授，上原隆平准教授，岡本吉央特任准教授，清見礼助教をはじめ，浅野・上原研究室の皆様には公私に渡り様々な面でお世話になりました．この場を借りてお礼申し上げます．特に指導教官の浅野哲夫教授には日頃から懇切丁寧なご指導を頂きましたことを深く感謝致します．最後に，同じ時間を共にした全ての方に心からの敬意と感謝の意を表します．

参考文献

- [1] B. Prasad, A. Bishnu, and T. Asano, “Linear Boundary and Corner detection using limited number of sensor rows,” *Proc. IWCIAC 2008*, pp.250–261 (2008).
- [2] R. Fleischer, “Is the World Linear?,” *In Efficient Algorithms - Essays Dedicated to Kurt Mehlhorn on the Occasion of His 60th Birthday*, Springer LNCS 5760, pp.368–379 (2009).
- [3] “計算幾何：理論の基礎から実践まで”, 浅野哲夫 著, 共立出版 (2007).