

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2011-81663

(P2011-81663A)

(43) 公開日 平成23年4月21日(2011.4.21)

(51) Int.Cl.

G 06 T 11/20 (2006.01)

F 1

G 06 T 11/20

1 O O

テーマコード(参考)

5 B 08 0

審査請求 未請求 請求項の数 9 O L (全 15 頁)

(21) 出願番号

特願2009-234394 (P2009-234394)

(22) 出願日

平成21年10月8日 (2009.10.8)

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(74) 代理人 100090273

弁理士 國分 孝悦

(72) 発明者

赤井 宏行

東京都大田区下丸子3丁目30番2号 キ
ヤノン株式会社内

F ターム(参考) 5B080 AA03 DA06 EA04 FA09 FA15

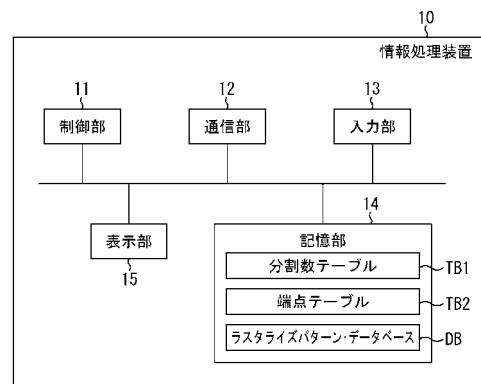
(54) 【発明の名称】 画像描画装置、画像描画方法、及びコンピュータプログラム

(57) 【要約】

【課題】真円のエッジの部分を複数の直線に分割して真円を描画するに際し、当該直線の端点の情報を高速に生成する。

【解決手段】円图形が幾つの直線に分割されるかを、当該円图形の半径 r と対応付けて分割数テーブル TB 1 に登録しておく。また、分割された直線の端点の X 座標算出用データ D_x と Y 座標算出用データ D_y とを、直線分割数 N と対応付けて端点テーブル TB 2 に予め登録しておく。したがって、描画图形が円图形の場合には、外形(曲線)をベジエ曲線で近似する必要がなく、三角関数の演算により、分割された直線の各分割点(端点)を直接に且つ高速に求めることができる。また、分割された直線の端点の座標の計算処理で負荷の掛かる計算処理の効率化を図ることができる。

【選択図】図 1



【特許請求の範囲】**【請求項 1】**

真円のベクトルデータの一部または全部を、複数の直線に分割し、分割した複数の直線に基づいて、当該真円の一部または全部をラスタライズする画像描画装置であって、

前記真円の分割数を、真円の大きさに係る情報に対して分割数に係る情報が予め登録された第1の記憶領域を参照して導出する第1の導出手段と、

前記真円のベクトルデータの一部または全部を、前記第1の導出手段により導出された分割数で分割することにより得られる複数の直線の端点の位置を、分割数に係る情報に対して端点の位置に係る情報が予め登録された第2の記憶領域を参照して導出する第2の導出手段と、

前記第2の導出手段により導出された位置に端点を有する複数の直線に基づいて、前記真円の一部または全部をラスタライズする描画手段と、を有することを特徴とする画像描画装置。

【請求項 2】

前記第2の記憶領域には、予め定められているM個の分割数に係る情報に対して端点の位置に係る情報が登録されており、

前記第2の導出手段は、前記第1の導出手段により導出された分割数であるNがMよりも小さい場合には、M/Nの間隔で、前記第2の記憶領域に登録されている端点の位置に係る情報を読み出すことを特徴とする請求項1に記載の画像描画装置。

【請求項 3】

前記第2の記憶領域には、分割数に応じた端点の位置であって、真円の一部分に対応する端点の位置に係る情報が登録されており、

前記第2の導出手段は、前記第2の記憶領域に登録されていない真円の他の部分に対応する端点の位置に係る情報を、前記第2の記憶領域に登録されている情報を用いて導出することを特徴とする請求項1又は2に記載の画像描画装置。

【請求項 4】

前記第1の導出手段は、前記真円の大きさが、予め定められた大きさよりも小さい場合に、当該真円の分割数を、前記第1の記憶領域を参照して導出することを特徴とする請求項1~3の何れか1項に記載の画像描画装置。

【請求項 5】

前記描画手段は、前記第2の導出手段により導出された位置に端点を有する直線の描画内容を表すラスタライズパターンを、直線の両端点の相対的な位置関係に係る情報に対して当該直線のラスタライズパターンに係る情報が予め登録された第3の記憶領域を参照して導出することを特徴とする請求項1~4の何れか1項に記載の画像描画装置。

【請求項 6】

前記第3の記憶領域には、直線の両端点の相対的な位置関係に応じた当該直線のラスタライズパターンであって、真円の一部分に対応する当該直線のラスタライズパターンに係る情報が登録されており、

前記描画手段は、前記第3の記憶領域に登録されていない真円の他の部分に対応する直線のラスタライズパターンに係る情報を、前記第3の記憶領域に登録されている情報を用いて導出することを特徴とする請求項5に記載の画像描画装置。

【請求項 7】

前記第3の導出手段は、前記真円の大きさが、予め定められた大きさよりも小さい場合に、前記第2の導出手段により導出された位置に端点を有する直線の描画内容を表すラスタライズパターンを、前記第3の記憶領域を参照して導出することを特徴とする請求項5又は6に記載の画像描画装置。

【請求項 8】

真円のベクトルデータの一部または全部を、複数の直線に分割し、分割した複数の直線に基づいて、当該真円の一部または全部をラスタライズする画像描画方法であって、

前記真円の分割数を、真円の大きさに係る情報に対して分割数に係る情報が予め登録さ

れた第1の記憶領域を参照して導出する第1の導出工程と、

前記真円のベクトルデータの一部または全部を、前記第1の導出工程により導出された分割数で分割することにより得られる複数の直線の端点の位置を、分割数に係る情報に対して端点の位置に係る情報が予め登録された第2の記憶領域を参照して導出する第2の導出工程と、

前記第2の導出工程により導出された位置に端点を有する複数の直線に基づいて、前記真円の一部または全部をラスタライズする描画工程と、を有することを特徴とする画像描画方法。

【請求項9】

請求項1～7の何れか1項に記載の画像描画装置の各手段としてコンピュータを機能させることを特徴とするコンピュータプログラム。 10

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、画像描画装置、画像描画方法、及びコンピュータプログラムに関し、特に、真円の全部または一部を描画するために用いて好適なものである。

【背景技術】

【0002】

ベクトル形式で記述されている図形データ等を表示装置に表示するためには、与えられたベクトルデータを、論理データから計算される「表示装置における座標」に変換し、描画するレンダリング処理が必要となる。従来から、ベクトルデータ（各図形データ）の処理に関する効率化の提案がなされている。特許文献1では、円を含む曲線の分割間隔を当該曲線の領域により変更することで滑らかに曲線を描画する手法が提示されている。また、特許文献2では、線分の描画について、線分自身の線幅、傾き、形状等の描画属性によって最適な処理を選択し、効率化を図る手法が提示されている。 20

【0003】

また、真円（以下、円图形と称する）の描画に着目した場合、従来は、ベクトルデータで与えられる円图形を表示装置に描画するために、まず円图形をベジエ曲線等の曲線ベクトルで近似する。そのため、円图形の半径を用いて曲線ベクトルでの近似数（分割数）を計算する。次に、特許文献3で示されているように、計算した分割数（分割点）に応じて、各曲線を描画するための制御点を求めて、円图形を曲線ベクトルデータの組合せに分割する。次に、特許文献4で示されているように、各曲線の曲線ベクトルデータを複数個の直線ベクトルデータに分割する。 30

【先行技術文献】

【特許文献】

【0004】

【特許文献1】特開平8-297747号公報

【特許文献2】特開平10-11592号公報

【特許文献3】特開平7-168945号公報

【特許文献4】特開平11-58871号公報

【発明の概要】

【発明が解決しようとする課題】

【0005】

しかしながら、前述した従来の手法では、描画する円图形のサイズに依らず一律に前述の曲線分割処理と、直線分割処理とが行われる。そのため、例えば、描画対象の円图形のサイズが小さい場合には、分割後の曲線が直線分割処理により1本の直線に分割（近似）され、ベジエ曲線の制御点や導出処理が冗長になるという課題があった。 40

また、複数の円图形の描画が指定された場合には、処理の繰り返しが増えるため、多大な負荷となる場合があるといった課題があった。

また、円图形から分割された直線の座標は、ラスター毎に計算されていた。そのため、

10

20

30

40

50

ラインジョイン処理・ラインキャップ処理において補間図形として円が指定された場合等、多数の円图形の描画が指定された場合には、多大な負荷が掛かるという課題があった。

本発明は、このような問題点に鑑みてなされたものであり、真円のエッジの部分を複数の直線に分割して真円を描画するに際し、当該直線の端点の情報を高速に生成することを目的とする。

【課題を解決するための手段】

【0006】

本発明の画像描画装置は、真円のベクトルデータの一部または全部を、複数の直線に分割し、分割した複数の直線に基づいて、当該真円の一部または全部をラスタライズする画像描画装置であって、前記真円の分割数を、真円の大きさに係る情報に対して分割数に係る情報が予め登録された第1の記憶領域を参照して導出する第1の導出手段と、前記真円のベクトルデータの一部または全部を、前記第1の導出手段により導出された分割数で分割することにより得られる複数の直線の端点の位置を、分割数に係る情報に対して端点の位置に係る情報が予め登録された第2の記憶領域を参照して導出する第2の導出手段と、前記第2の導出手段により導出された位置に端点を有する複数の直線に基づいて、前記真円の一部または全部をラスタライズする描画手段と、を有することを特徴とする。

10

【発明の効果】

【0007】

本発明によれば、真円の分割数に係る情報と、直線の端点の位置に係る情報とを予め登録しておくようにしたので、真円のエッジの部分を複数の直線に分割して真円を描画するに際し、当該直線の端点の情報を高速に生成することができる。

20

【図面の簡単な説明】

【0008】

【図1】情報処理装置の構成を示す図である。

【図2】分割数テーブルを示す図である。

【図3】円图形を分割した直線の端点と、端点テーブルを示す図である。

【図4】ラスタライズデータを得る手法を説明する図である。

【図5】ラスタライズデータを生成する際の処理を説明するフローチャートである。

【図6】図5のフローチャートの詳細を説明するフローチャートである。

30

【図7】図6のステップS17に続くフローチャートである。

【図8】図6のステップS12に続くフローチャートである。

【図9】図8に続くフローチャートである。

【図10】ラインジョイン処理及びラインキャップ処理を説明する図である。

【図11】隙間がある折れ線部分と、隙間を補間する円图形の端点を示す図である。

【図12】直線分割処理の一例を説明するフローチャートである。

【発明を実施するための形態】

【0009】

以下に、図面を参照しながら、本発明の各実施形態について説明する。

(第1の実施形態)

まず、本発明の第1の実施形態について説明する。

40

図1は、情報処理装置10の構成の一例を示す図である。情報処理装置10は、制御部11と、通信部12と、入力部13と、記憶部14と、表示部15とを有している。

制御部11は、例えばCPU等を用いて構成され、情報処理装置10を統括制御する。通信部12は、例えば通信インターフェース等を用いて構成され、外部装置との通信を行う。入力部13は、例えばキーボード及びマウス等を用いて構成され、ユーザによる情報処理装置10に対する入力操作を受け付ける。記憶部14は、例えばROM、RAM、及びHDD等を用いて構成され、情報処理装置10におけるデータ及びコンピュータプログラム等を記憶する。本実施形態では、記憶部14には、後述する分割数テーブルTB1、端点テーブルTB2、及びラスタライズパターン・データベースDBが予め設定(記憶)されている。この他、最大分割半径Rd、最大分割数M、及び最大ラスタライズ半径Rr

50

も記憶部 14 に予め設定（記憶）されている。ただし、分割数テーブル TB1、端点テーブル TB2、ラスタライズパターン・データベース DB、最大分割半径 Rd、最大分割数 M、及び最大ラスタライズ半径 Rr の少なくとも何れか 1 つは、情報処理装置 10 と通信可能な外部装置に記憶されていてもよい。また、これらのパラメータの値は、実施される形態や環境、システムの処理能力に依存する。

【0010】

以下、これらのパラメータについて説明を行う。

最大分割半径 Rd、最大分割数 M (M は 2 以上の整数) は、それぞれ、後述する分割数テーブル TB1、端点テーブル TB2 が許容する最大円半径、分割数の最大値である。描画する円図形（真円）の半径 r が最大円半径 Rd を超える場合、当該円図形については、分割数テーブル TB1、端点テーブル TB2、ラスタライズパターン・データベース DB を用いたレンダリング処理は不可と判断される。この場合、当該円図形については、ベジエ曲線への分割を行う従来の処理が選択される。10

【0011】

図 2 は、分割数テーブル TB1 の一例を示す図である。

分割数テーブル TB1 は、与えられた円図形の半径 r から、円の直線分割数 N (N は 2 以上の整数) を得るためのテーブルである。図 2 に示すように、分割数テーブル TB1 は、直線分割数 N と、当該直線分割数 N を持つ円の最大半径 Rmax と、当該円の最小半径 Rmin とを構成要素とする。与えられた円図形の半径 r を最小半径 Rmin から最大半径 Rmax までの間に含む直線分割数 N が分割数テーブル TB1 に含まれる。ここで、分割数テーブル TB1 の各直線分割数 N は、後述する端点テーブル TB2 の効率化のために、最大分割数 M の公約数となるようにすることができる。20

【0012】

図 3 は、円図形を複数の直線に分割する場合の当該直線の端点の一例（図 3 (a)）と、端点テーブル TB2 の一例（図 3 (b)）とを示す図である。

端点テーブル TB2 は、円図形を複数の直線に分割する際に、各直線（線分）の端点の座標を予め行いテーブル化したものである。各直線の端点の座標は、三角関数等の計算を行うことにより求めることができる。通常、N 個に直線分割された線分の端点 EPn(xn, yn) (n = 0, 1, ..., N-1) は、図 3 (a) に示すように、円の中心点の座標 (xo, yo)、円の半径 r、及び基準点からの角度 n [rad] により三角関数を用いて求められる。基準点からの角度 n とは、中心点と基準点 EP0 とを結ぶ直線と、中心点と端点 EPn とを結ぶ直線とのなす角度である。基準点からの角度 n は、円の分割数が定まっているれば基準点 EP0 を 0 (ゼロ) とする、N 個に直線分割された線分の端点を特定する端点番号 n を用いて求めることができ、その値を用いた三角関数の計算も予めテーブル化することが可能である。30

【0013】

本実施形態では図 3 (b) に示すように、基準点 EP0 からの端点番号 n (分割数) をインデックスとして、当該端点番号 n の端点の X 座標算出用データ、Y 座標算出用データをそれぞれ、D_x、D_y としてテーブル化したものが端点テーブル TB2 である。

ここで、端点テーブル TB2 を、円図形の半径 r、直線分割数 N 毎に複数保持する必要はなく、最大分割数 M で作成された端点テーブル TB2 を、異なる直線分割数 N の端点の X、Y 座標を求める際にも利用でき、单一のテーブルのみで対応できる。その方法として、まず、所望する円図形の直線分割数 N と最大分割数 M とから、端点テーブル TB2 の参照間隔を M / N とする。そして、端点番号 n の端点の座標を求める際には、n × M / N をインデックスとして端点テーブル TB2 からデータを取得するようにする（図 3 (b) の左端を参照）。この方法により、单一の端点テーブル TB2 のみで、直線分割数 N に依存せずに、N 個に直線分割された線分の端点の座標を得ることができる。したがって、複数のテーブルに冗長な情報を格納するといったことが無く、効率良くデータを保持することが可能である。直線分割で対応可能な分割数（円半径）は、端点テーブル TB2 の分割数に依存するため、端点テーブル TB2 の分割数（最大分割数 M）は出来得る限り大きな値4050

であることが望ましい。このように本実施形態では、端点テーブルTB2には、M個のインデックスがある。

【0014】

直線分割された線分の端点EPnのX座標xn及びY座標ynと、基準点からの角度nは、それぞれ(1)式～(3)式で表され、端点テーブルTB2から得られる端点EPnのX座標xn及びY座標ynは、それぞれ(4)式、(5)式で表される。

$$X_n = x_0 + r \times \sin(n) \quad \dots \quad (1)$$

$$Y_n = y_0 - r \times \cos(n) \quad \dots \quad (2)$$

$$n = n \times 2 / N \quad \dots \quad (3)$$

$$X_n = x_0 + r \times TB2[n \times M / N, 1] \quad \dots \quad (4)$$

$$Y_n = y_0 - r \times TB2[n \times M / N, 2] \quad \dots \quad (5)$$

ここで、TB2[n × M / N, 1]は、端点テーブルTB2のインデックスn × M / NのD_xの値を示し、TB2[n × M / N, 2]は、端点テーブルTB2のインデックスn × M / NのD_yを示す。

【0015】

最大ラスタライズ半径Rrは、分割後の各線分をラスタライズする際に、後述するラスタライズパターン・データベースDBを用いた手法が許容する円图形の最大半径である。

ラスタライズパターン・データベースDBは、分割された線分の描画内容を表すラスタライズデータをデータベース化したものである。

図4は、分割後の線分からラスタライズパターン・データベースDBを用いてラスタライズデータを得る手法の一例を説明する図である。

図4に示すように、ラスタライズパターン・データベースDBは、開始位置CP及び終了位置NP(両端点)の相対的な位置関係の一例として、両端点の座標を整数化したものの差分(dx, dy)と該差分の最大値MAX(dx, dy)をインデックスとする。また、ラスタライズパターン・データベースDBは、分割された線分のラスタライズパターン・データとして、例えば、直前のラスターに対するX方向の変化分及びY方向の変化分(|_dx, |_dy)を格納する。

【0016】

図4に示す例では、データ81に示すように、両端点の座標を整数化した際の差分の最大値MAX(dx, dy)をインデックス1として、ラスタライズパターン・データを大きさ毎に分類・格納したデータ82へのエントリを格納する。データ82は、各線分の大きさ毎に両端点の差分[dx, dy]をインデックス2として、各ラスタライズパターン・データ83へのエントリを有する。各ラスタライズパターン・データ83は、線分の開始点を基準とした番号をインデックス3とし、直前のラスターの座標に対するX方向の変化分とY方向の変化分をそれぞれ_|_dx、_|_dyとして格納している。

ここで、円图形から分割される線分の長さ、傾きは限られている。したがって、ラスタライズパターン・データ83として、線分の全てのラスタライズパターンをデータベース化する必要はない。このように、必要なラスタライズパターン・データのみを保持することで、格納するデータ量を削減し、処理の効率化を図ることができる。

【0017】

端点テーブルTB2、及びラスタライズパターン・データベースDBの少なくとも何れか一方には、円周全体分のデータ(真円の全体に対応するデータ)を格納する必要はない。円の対称性を利用し、真円の一部分に対応するデータ(例えば1象限分(1/4円周分)のデータ)を格納する。そして、異なる象限内の端点(真円の他の部分に対応する端点)の計算や、線分(真円の他の部分に対応する線分)のラスタライズ時には、符号、XY値を入れ替える処理を行うことで、格納するデータ量を削減し、処理の効率化を図ることができる。

【0018】

図5は、円图形のベクトルデータからラスタライズデータを生成する際の情報処理装置10の処理の一例を説明するフローチャートである。

10

20

30

40

50

円図形のベクトルデータの描画が指定されると、ステップ S 1において、制御部 1 1は、描画対象の円図形の半径 r が最大分割半径 R_d 以下であるか否かを判定する。この判定の結果、指定された円図形の半径 r が最大分割半径 R_d 以下である場合には、ステップ S 2に進み、分割数テーブル $T_B 1$ と端点テーブル $T_B 2$ を用いた後述する円図形の直線分割処理が実行される。

一方、指定された円図形の半径 r が最大分割半径 R_d 以下でない場合には、ステップ S 3に進み、後述する曲線分割処理を経て直線分割を行う従来手法に基づく直線分割処理が実行される。

【0019】

ステップ S 2 の処理の後のステップ S 4 では、制御部 1 1は、円図形の半径 r が最大ラスタライズ半径 R_r 以下であるか否かを判定する。この判定の結果、円図形の半径 r が最大ラスタライズ半径 R_r 以下である場合には、ステップ S 5に進み、後述するラスタライズデータパターン・データベース D_B を用いた分割線分のラスタライズ処理が実行される。10

ステップ S 4 の判定の結果、円図形の半径 r が最大ラスタライズ半径 R_r 以下でない場合、又はステップ S 3 の処理が実行された場合には、ステップ S 6 に進む。そして、各線分について、ラスター単位で表示装置における座標の算出を行う従来手法に基づくラスタライズ処理が実行される。

【0020】

次に、図 6～図 9 のフローチャートを参照しながら、図 5 のフローチャートの詳細を説明する。20

まず、円図形のベクトルデータの描画が指示されると、図 6 のステップ S 1 1 において制御部 1 1は、描画対象の円図形の半径 r と、最大分割半径 R_d とを取得する。

次に、ステップ S 1 2 において、制御部 1 1は、描画対象の円図形の半径 r が最大分割半径 R_d 以下であるか否かを判定する。この判定の結果、描画対象の円図形の半径 r が最大分割半径 R_d 以下である場合には、分割数テーブル $T_B 1$ 及び端点テーブル $T_B 2$ を用いた端点の座標の算出が可能と判断してステップ S 1 3 に進む。一方、描画対象の円図形の半径 r が最大分割半径 R_d 以下でない場合には、分割数テーブル $T_B 1$ 及び端点テーブル $T_B 2$ を用いた端点の座標の算出は不可と判断し、図 8 のステップ S 2 5 に進む。後述するように、ステップ S 2 5 以降では、円図形を複数の曲線ベクトルに分割した後に直線分割する従来手法に基づく直線分割処理が行われる。30

【0021】

ステップ S 1 3 に進むと、制御部 1 1は、描画対象の円図形の半径 r を用いて分割数テーブル $T_B 1$ から、描画対象の円図形の直線分割数 N を直接取得すると共に、最大分割数 M を取得する。

このように本実施形態では、例えば、ステップ S 1 3 の処理を実行することにより第 1 の導出手段の一例が実現され、分割数テーブル $T_B 1$ を用いることにより第 1 の記憶領域の一例が実現される。

次に、ステップ S 1 4 において、制御部 1 1は、最大分割数 M で作成された单一の端点テーブル $T_B 2$ から所望の直線分割数 N に対応したデータの参照を可能とするための参照間隔 M / N を計算する。40

次に、ステップ S 1 5 において、制御部 1 1は、端点番号 n ($n = 0, 1, \dots, N-1$) と参照間隔 M / N とから、端点テーブル $T_B 2$ の参照インデックスを $n \times M / N$ として所望の端点の座標算出用データ D_x, D_y を取得する。

【0022】

次に、ステップ S 1 6 において、制御部 1 1は、ステップ S 1 5 で取得した座標算出用データ D_x, D_y を用いて、端点 EP_n の座標 (x_n, y_n) を、前述した(4)式、(5)式により計算する。このとき、端点テーブル $T_B 2$ の保持するデータが第一象限分であり、所望するデータが第二象限の端点であった場合、制御部 1 1は、端点の Y 座標算出用データ D_y の符号を反転させる。このように、制御部 1 1は、異なる象限の端点テーブル T_B 50

2の座標算出用データを所望の端点の座標算出用データに適応させる手段を有する。次に、ステップS17において、制御部11は、現在の端点番号nに1を加算したものが、描画対象の円图形の直線分割数N以上であるか否かを判定する。この判定の結果、現在の端点番号nに1を加算したものが、描画対象の円图形の直線分割数N以上でない場合には、ステップS15に戻り、現在の端点番号nに1を加算したもの現在の端点番号nとし、当該端点番号nについて処理を実行する。このようにして、N個の端点の座標が計算されると、図7のステップS18に進む。

以上のように本実施形態では、例えば、ステップS14～S17の処理を実行することにより第2の導出手段の一例が実現され、端点テーブルTB2を用いることにより第2の記憶領域の一例が実現される。

10

【0023】

図7のステップS18に進むと、制御部11は、最大ラスタライズ半径Rrを取得する。

次に、ステップS19において、制御部11は、描画対象の円图形の半径rが最大ラスタライズ半径Rr以下であるか否かを判定する。この判定の結果、描画対象の円图形の半径rが最大ラスタライズ半径Rr以下である場合には、ラスタライズパターン・データベースDBを用いたラスタライズ処理が可能と判断してステップS20へ進む。一方、描画対象の円图形の半径rが最大ラスタライズ半径Rr以下でない場合には、図9のステップS31に進む。ステップS31では、後述するように、各線分について、表示装置における座標の算出をラスター単位で行う従来手法に基づくラスタライズ処理が実行される。

20

【0024】

ステップS20に進むと、制御部11は、複数に分割された線分の両端点の座標を整数化し、その差分dn=(dx, dy)を計算する(図4を参照)。

次に、ステップS21において、制御部11は、ステップS20で計算した差分dnの最大値MAX(dx, dy)を計算する。

次に、ステップS22において、制御部11は、差分dnと、当該差分dnの最大値MAX(dx, dy)とをインデックスとしてラスタライズパターン・データベースDBから、所望の線分のラスタライズパターン・データを一意に取得する。

次に、ステップS23において、制御部11は、線分の開始端点CPに、ステップS22で取得したラスタライズパターン・データを順次加算することで線分の表示装置における座標を高速に求める。

30

【0025】

次に、ステップS24において、制御部11は、現在の端点番号nに1を加算したものが、描画対象の円图形の直線分割数N以上であるか否かを判定する。この判定の結果、現在の端点番号nに1を加算したものが、描画対象の円图形の直線分割数N以上でない場合には、ステップS15に戻り、現在の端点番号nに1を加算したもの現在の端点番号nとし、当該端点番号nについて処理を実行する。このようにして、N個の線分のラスタライズデータが得られ(円图形のラスタライズが行われ)、描画対象の円图形の直線分割数N以上になると、本フローチャートによる処理を終了する。

以上のように本実施形態では、例えば、ステップS20～24の処理を実行することにより第3の導出手段の一例が実現され、ラスタライズパターン・データベースDBを用いることにより第3の記憶領域の一例が実現される。

40

【0026】

前述したように、図6のステップS12において、描画対象の円图形の半径rが最大分割半径Rd以下でないと判定されると、分割数テーブルTB1及び端点テーブルTB2を用いた端点の座標の算出は不可と判断し、図8のステップS25に進む。

ステップS25に進むと、制御部11は、円图形のベジエ曲線(以下の説明では必要に応じて分割曲線Cnと称する)への分割数Ncを1象限1曲線として計算する。

次に、ステップS26において、制御部11は、分割曲線Cn(n=1, 2, ..., Nc)の開始点、終了点を含む制御点を計算する。制御点の計算は、特許文献3に記載され

50

ているように公知の技術であるので、ここでは、詳細な説明を省略する。

次に、ステップ S 2 7において、制御部 1 1は、直線近似方法を用いて、ステップ S 2 6で計算された制御点から分割曲線 C n の直線分割数 N n を計算する。直線近似方法は、特許文献 4 に記載されているように公知の技術であるので、ここでは、詳細な説明を省略する。

【0027】

次に、ステップ S 2 8において、制御部 1 1は、分割曲線 C n の i 番目の分割線分 S i ($i = 1, 2, \dots, N_n$) の両端点の座標を、前述した(1)式～(3)式を用いて計算する。

次に、ステップ S 2 9において、制御部 1 1は、現在の i に 1 を加算したものが、分割曲線 C n の直線分割数 N n 以上であるか否かを判定する。この判定の結果、現在の i に 1 を加算したものが、分割曲線 C n の直線分割数 N n 以上でない場合には、ステップ S 2 8 に戻り、現在の i に 1 を加算したものを現在の i とし、当該 i 番目の分割線分 S i の両端点の座標を計算する。10

一方、現在の i に 1 を加算したものが、分割曲線 C n の直線分割数 N n 以上となり、分割曲線 C n の各分割線分 S i の両端点の座標が計算されると、ステップ S 3 0 に進む。ステップ S 3 0 に進むと、制御部 1 1は、現在の端点番号 n に 1 を加算したものが、分割曲線 C n の分割数 N c 以上であるか否かを判定する。この判定の結果、現在の端点番号 n に 1 を加算したものが、分割曲線 C n の分割数 N c 以上でない場合には、ステップ S 2 6 に戻り、現在の端点番号 n に 1 を加算したものを現在の端点番号 n とし、当該端点番号 n について処理を実行する。このようにして、ステップ S 2 6 ~ S 2 9 の処理を N c 個の分割曲線毎に繰り返し、円图形の直線分割が行われると、図 9 のステップ S 3 1 に進む。図 8 のフローチャートが実行された場合、円图形の半径 r が大きく、直線分割後の各線分のラスタライズパターンが多数あると判断される。そこで、本実施形態では、図 8 のフローチャートが実行された場合には、図 9 のフローチャートを実行し、各線分について、ラスター単位で表示装置における座標の算出を行う従来手法に基づくラスタライズ処理が行われる。20

【0028】

図 9 のステップ S 3 1 に進むと、制御部 1 1は、線分 S n ($n = 0, 1, \dots, N-1$) の処理ラスター位置 L n として開始点 C P n をセットする。30

次に、ステップ S 3 2 において、制御部 1 1は、線分 S n の処理ラスター位置 L n における表示装置上の座標を計算する。その後、ステップ S 3 3 において、制御部 1 1は、現在の処理ラスター位置 L n に 1 を加算し、現在の処理ラスター位置を次の位置(ラスター)へ移動させる。

次に、ステップ S 3 4 において、制御部 1 1は、現在の処理ラスター位置 L n が、線分 S n の終了点 N P n を超えたか否かを判定する。この判定の結果、現在の処理ラスター位置 L n が、線分 S n の終了点 N P n を超えていない場合には、ステップ S 3 2 に戻り、現在の処理ラスター位置 L n について処理を行う。そして、現在の処理ラスター位置 L n が、線分 S n の終了点 N P n を超え、線分 S n の終了点 N P n の座標が計算されると(線分 S n のラスタライズ処理が行われると)、ステップ S 3 5 に進む。40

ステップ S 3 5 に進むと、制御部 1 1は、現在の端点番号 n に 1 を加算したものが、描画対象の円图形の直線分割数 N 以上であるか否かを判定する。この判定の結果、現在の端点番号 n に 1 を加算したものが、描画対象の円图形の直線分割数 N 以上でない場合には、ステップ S 1 5 に戻り、現在の端点番号 n に 1 を加算したものを現在の端点番号 n とし、当該端点番号 n について処理を実行する。このようにして、ステップ S 3 1 ~ S 3 4 の線分のラスタライズ処理が N 個の線分毎に繰り返し行われ、円图形のラスタライズ処理が行われると、本フローチャートによる処理を終了する。

【0029】

以上のように本実施形態では、円图形が幾つの直線に分割されるかを、当該円图形の半径 r と対応付けて分割数テーブル T B 1 に登録しておく。また、分割された直線の端点の50

X 座標算出用データ D_x と Y 座標算出用データ D_y とを、直線分割数 N と対応付けて端点テーブル T_B_2 に予め登録しておく。したがって、描画図形が円図形の場合には、外形（曲線）をベジエ曲線で近似する必要がなく、分割された直線の各分割点（端点）を直接に且つ高速に求めることができる。また、分割された直線の端点の座標の計算処理で負荷の掛かる計算処理の効率化を図ることができる。よって、真円のエッジの部分を複数の直線に分割して真円を描画するに際し、当該直線の端点の情報を高速に生成することができる。

また、本実施形態では、線分をラスタライズしたデータをラスタライズパターンとして、線分の両端点の座標の差分と対応付けてラスタライズパターン・データベース D_B に予め登録しておく。したがって、線分とスター（スキャンライン）との交点の座標をその都度（スキャンライン毎に）計算することなく線分を描画することが可能になる。10

【0030】

尚、本実施形態で説明した図 5 のステップ S_3（図 8）、ステップ S_6（図 9）は、それぞれ、従来の手法で実現できる直線分割処理、ラスタライズ処理であれば、前述した方法に限定されるものではない。

また、本実施形態では、円図形が幾つの直線に分割されるかを、当該円図形の半径 r と対応付けて分割数テーブル T_B_1 に登録した場合を例に挙げて説明した。しかしながら、円図形の半径 r の代わりに、円図形の直径や円周等、円図形（真円）の大きさの情報を用いるようにしてもよい。

また、本実施形態では、円図形の半径 r と、最大分割半径 R_d 及び最大ラスタライズ半径 R_r との比較結果に応じて処理を切り替えるようにした。しかしながら、最大分割半径 R_d 及び最大ラスタライズ半径 R_r の少なくとも何れか一方を設定せずに、図 6 のステップ S_13、S_20 以降の処理を無条件で実行するようにしてもよい。20

また、本実施形態では、情報処理装置 10 が画像描画装置の一例である場合について説明したが、画像描画装置は、印刷装置等であってもよい。

【0031】

（第 2 の実施形態）

次に、本発明の第 2 の実施形態について説明する。前述した第 1 の実施形態では、一般的な円図形（の全部）を描画する場合を例に挙げて説明した。これに対し、本実施形態では、ラインジョイン処理、ラインキャップ処理において、補間図形として円が指定された場合の描画について説明する。この場合には、円図形の必要な一部だけを選択し描画する必要がある。このように本実施形態と前述した第 1 の実施形態とは、ラスタライズ処理の対象となる図形が異なることによる処理（曲線の図形を複数の直線に分割し、分割した直線の端点の座標を導出する処理）が主として異なる。したがって、本実施形態の説明において、前述した第 1 の実施形態と同一の部分については、図 1～図 9 に付した符号と同一の符号を付す等して詳細な説明を省略する。30

【0032】

図 10 は、ラインジョイン処理及びラインキャップ処理を説明する図である。

図 10 (a) に示すように、線幅のある線の描画を行う場合、折れ線部分に隙間 5_1 が発生する。この隙間 5_1 を直線、折れ線、円等の所定の形状で、図 10 (b) に示すように補間する処理がラインジョイン処理である。一方、図 10 (a) に示すように、線幅のある線の描画を行う場合に線端 5_2 を直線、円等の所定の形状で図 10 (b) に示すように補間する処理がラインキャップ処理である。40

【0033】

図 11 は、隙間が生じている折れ線部分と、ラインジョイン処理により当該隙間を補間する円図形の端点の一例を示す図である。

図 11 (a) に示すように、線幅 w を持つ折れ線を、ラインジョイン処理において、円で補間して描画する場合、折れ線の隙間を破線 141 で示す形状に円図形で補間する必要がある。以下の説明では、必要に応じてこの円図形を補間円図形と称する。

図 12 は、直線分割処理の一例を説明するフローチャートであり、図 6 の代わりに実行

されるものである。本実施形態においても、第1の実施形態で説明したように、予めパラメータが設定される。

まず、図12のステップS41において、制御部11は、補間円图形の半径rと、最大分割数Rdとを取得する。このとき、ラインジョイン処理、ラインキャップ処理における補間円图形の半径rは、描画線分(折れ線)の線幅wの2分の1となるので、制御部11は、まず線幅wを取得し、取得した線幅wを1/2倍したものを補間円图形の半径rとして求める。

【0034】

次に、ステップS42において、制御部11は、描画対象の円图形の半径rが最大分割半径Rd以下であるか否かを判定する。この判定の結果、描画対象の円图形の半径rが最大分割半径Rd以下でない場合には、前述した図8のステップS25に進む。これにより、図11(a)の破線141に示す補間円图形が図8に示した処理に従って直線分割される。

一方、描画対象の円图形の半径rが最大分割半径Rd以下である場合には、ステップS43に進む。ステップS43に進むと、制御部11は、描画対象の円图形の半径rを用いて分割数テーブルTB1から、描画対象の円图形(1周分の円)の直線分割数Nを直接取得すると共に、最大分割数Mを取得する。

このように本実施形態では、例えば、ステップS43の処理を実行することにより第1の導出手段の一例が実現され、分割数テーブルTB1を用いることにより第1の記憶領域の一例が実現される。

次に、ステップS44において、制御部11は、最大分割数Mで作成された单一の端点テーブルTB2から所望の直線分割数Nに対応したデータ参照を可能とするための参照間隔M/Nを計算する。

【0035】

図11に示す例では、図11(a)における折れ線の隙間を円图形で補間するため、図11(b)に示すように、当該隙間を円图形で補完した場合に、折れ線の角点SP、FPの間に挟まれる端点EPns～EPneまでを求める必要がある。

そこで、ステップS45において、制御部11は、必要な端点の開始端点番号nsと終了端点番号neとを計算する。その方法として、まず、制御部11は、基準点EP0から各角点SP、FPまでの角度s、e[rad]を求める。ここで、基準点EP0から角点SPまでの角度sは、中心点と基準点EP0とを結ぶ直線と、中心線と角点SPとを結ぶ直線とのなす角度である。また、基準点EP0から角点FPまでの角度eは、中心点と基準点EP0とを結ぶ直線と、中心線と角点FPとを結ぶ直線とのなす角度である。次に、制御部11は、端点の開始端点番号nsを $(n - 1) \times 2 / M < s \leq n \times 2 / M$ を満たすnとして計算する。同様に制御部11は、端点の終了端点番号neを $n \times 2 / M < e < (n + 1) \times 2 / M$ を満たすnとして計算する。これにより、円图形の描画に必要な端点番号が求まる。端点の開始端点番号nsが求まると、ステップS46において、制御部11は、端点座標計算の開始点をn=nsとしてセットする。

【0036】

次に、ステップS47において、制御部11は、端点番号n(n=ns, ns+1, ..., ne)と参照間隔M/Nとから、端点テーブルTB2の参照インデックスを $n \times M / N$ として所望の端点の座標算出用データD_x、D_yを取得する。

次に、ステップS48において、制御部11は、ステップS47で取得した座標算出用データD_x、D_yを用いて、端点EPnの座標(xn, yn)を、前述した(4)式、(5)式により計算する。本実施形態でも、制御部11は、異なる象限の端点テーブルTB2の座標算出用データを所望の端点の座標算出用データに適応させる手段を有する。

【0037】

次に、ステップS49において、制御部11は、現在の端点番号nに1を加算したものが、端点の終了端点番号ne以上であるか否かを判定する。この判定の結果、現在の端点番号nに1を加算したものが、端点の終了端点番号ne以上でない場合には、ステップS47

10

20

30

40

50

に戻り、現在の端点番号 n に 1 を加算したものを現在の端点番号 n とし、当該端点番号 n について処理を実行する。このようにして、($ne - ns$) 個の端点の座標が計算されると、図 7 のステップ S 1 8 に進む。そして、補間図形である弧SP-FPの各分割線分（線分SP-EP ns 、EP ne -FPを含む）を第 1 の実施形態で説明したのと同様にしてラスタライズ処理を行うことで必要な部分だけを描画する。

以上のように本実施形態では、例えば、ステップ S 4 4 ~ S 4 9 の処理を実行することにより第 2 の導出手段の一例が実現され、端点テーブル TB 2 を用いることにより第 2 の記憶領域の一例が実現される。

【 0 0 3 8 】

以上のように本実施形態では、ラインジョイン処理・ラインキャップ処理を行うに際し、線の線幅の $1 / 2$ 倍を半径 r とする円図形として、分割数テーブル TB 1 から直線分割数 N を取得する。そして、補間する領域（破線 1 4 1 ）に対応する端点の開始端点番号 ns と終了端点番号 ne を求め、開始端点番号 ns から終了端点番号 ne までの各端点の座標を、端点テーブル TB 2 から取得した端点の座標算出用データ D_x 、 D_y に基づいて計算する。したがって、通常の円図形だけでなく、円図形を用いてラインジョイン処理・ラインキャップ処理を行う場合にも、前述した第 1 の実施形態で説明した効果が得られる。

【 0 0 3 9 】

尚、前述した実施形態は、何れも本発明を実施するにあたっての具体化の例を示したものに過ぎず、これらによって本発明の技術的範囲が限定的に解釈されなければならないものである。すなわち、本発明はその技術思想、又はその主要な特徴から逸脱することなく、様々な形で実施することができる。

【 0 0 4 0 】

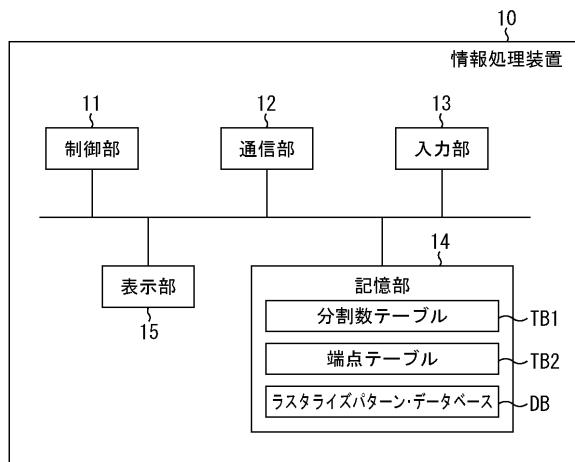
（ その他の実施例 ）

本発明は、以下の処理を実行することによっても実現される。即ち、上述した実施形態の機能を実現するソフトウェア（プログラム）を、ネットワーク又は各種記憶媒体を介してシステム或いは装置に供給し、そのシステム或いは装置のコンピュータ（または C P U や M P U 等）がプログラムを読み出して実行する処理である。

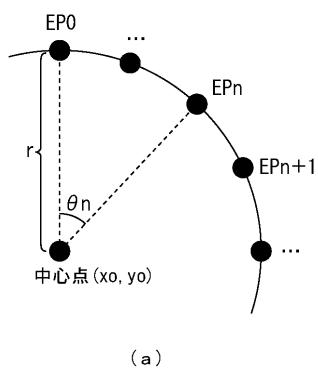
10

20

【図1】



【図3】



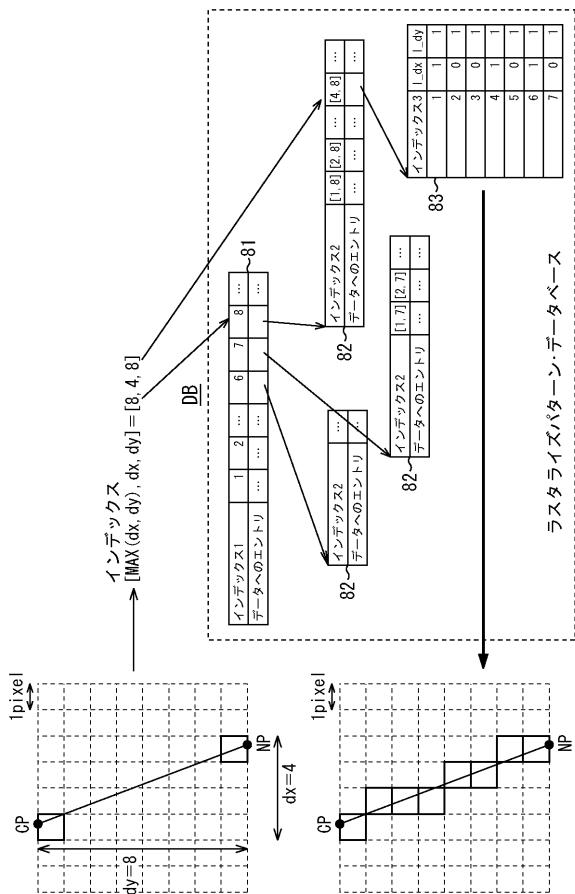
【図2】

最小半径Rmin	最大半径Rmax	直線分割数N
1	7	4
8	14	8
15	28	16
...

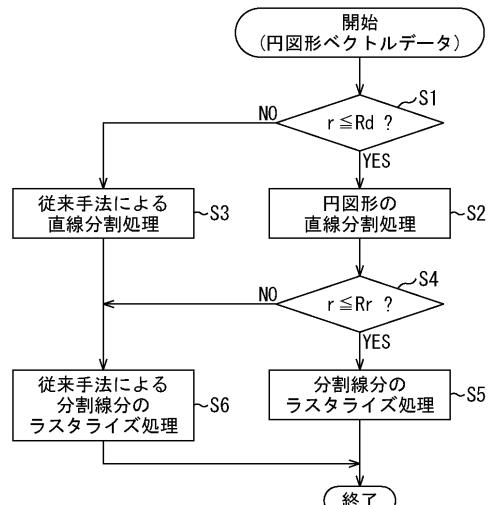
インデックス	D_x	D_y
0	$\sin 0$	$\cos 0$
1	$\sin(2\pi/M)$	$\cos(2\pi/M)$
...
n	$\sin(n*2\pi/M)$	$\cos(n*2\pi/M)$
...
$n*M/N$	$\sin(n*M/N*2\pi/M)$	$\cos(n*M/N*2\pi/M)$
...
...
...

(b)

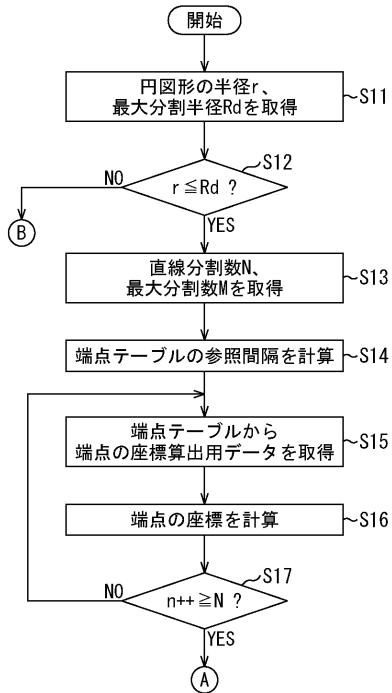
【図4】



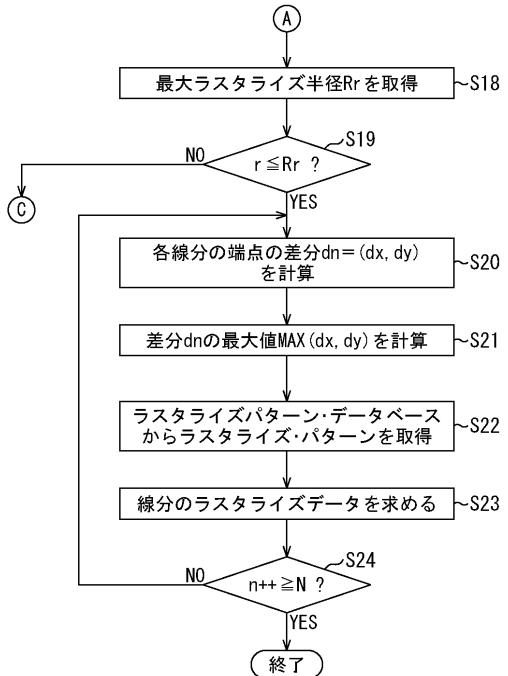
【図5】



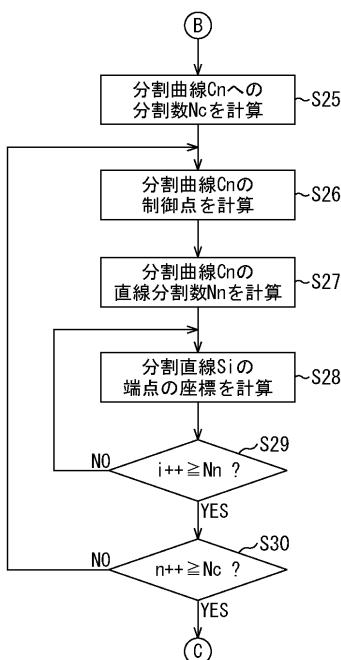
【図6】



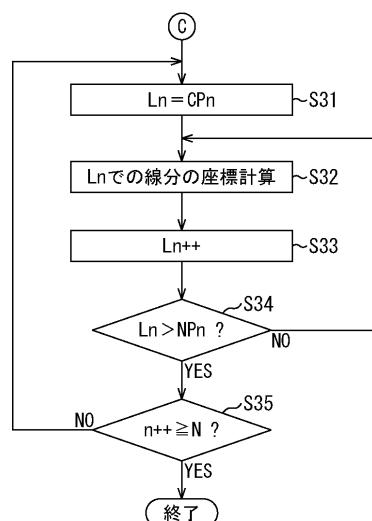
【図7】



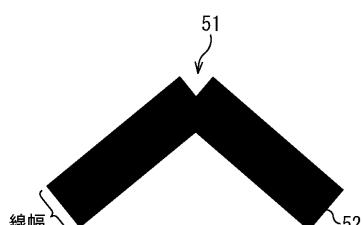
【図8】



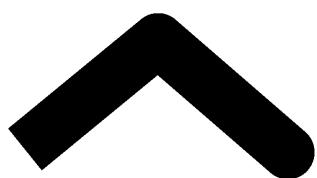
【図9】



【図 1 0】

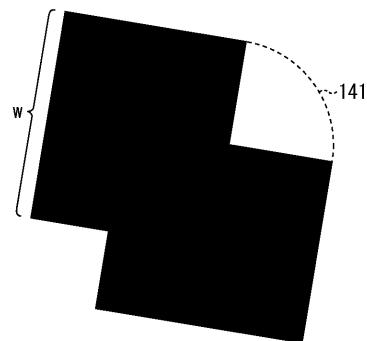


(a)

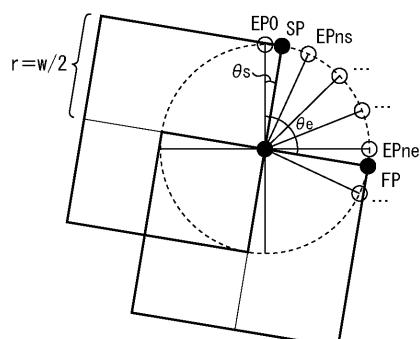


(b)

【図 1 1】



(a)



(b)

【図 1 2】

