

設計過程における グラフィックインターフェイス

福井幸男

工業技術院 製品科学研究所

設計の上流過程における計算機支援技術のあり方を考察するために、ケーススタディとしてリレー光学系の設計においてアイデア段階での計算機利用を試みた。即ち、要求仕様を与えて設計案を提示する部分、提示例に従い数値計算によってその正当性を確認する部分、確認した結果をグラフィック出力する部分に分けてプログラム化し、分散処理により効率的な処理を行った。その結果要求仕様を与えて、計算機から提示される設計案をグラフィック出力することにより、容易に設計者が内容を理解検討し、また新しいアイデアを生む源となりうることが判った。開発設計構想時のような創造的な場では、計算機と設計者がごく自然に対話できることが必要であり、グラフィックインターフェイスはその必要不可欠な要素であることが実感できた。

Graphic interface in design process

Yukio FUKUI

Industrial Products Research Institute
(1-1-4 Higashi, Yatabe, Tsukuba, Ibaraki 305)

We have tried computer aided design in primitive process of optical relay system for the feasibility study of introduction of computer in early stage of design process. Our program is a distributed system consisting of three independent routines which run simultaneously and cooperating to one another. The first routine proposes basic design solutions for the given requirements while the second checks numerically if the solutions are valid or not, and the last is graphic interface which visualize the solutions three dimensionally from any desired viewing angle or position. It was clarified that the visualization of the solutions made it easy for the user to understand them and even stimulated user's imagination for improvement or new ideas. The graphic interface between user and the computer is essential especially for dealing with vague ideas in early stage of design process.

1. はじめに

新装置開発に於て計算機を利用した作業は、形状設計段階における各種解析、製作図面作成、製作段階における工程設計、数値制御加工等があり、更に品質管理工程でも技術データ、事務データの処理に用いられている。今後は設計者のアイデアを引き出す設計の上流工程に計算機が利用されてゆくことが予想される。この上流過程ではユーザインターフェイスが特に重要な意味を持つはずである。なぜならば、この過程においては設計者は自己の持つポテンシャルを最大限に活性化させ知的活動に専念しなければならず、それ故に計算機との対話上の些事に気をそがれてはならない。このために人間にとって理解しやすい手段として、文字列・数値データよりもグラフィックスを利用して加工された図形データを用いることが望ましいことは明かである。ここでは、グラフィックスを利用したユーザインターフェイスを有効に活用して光学系の基本設計を行った過程を述べる。

2. 要求仕様とその実現例提示

与えられた要求仕様は次のようなものである。即ち「方向を自在に制御でき、しかも光学特性を変化させないリレー光学系の実現」である。問題は二つあり、一つはリレー光学系の実現であり、他の一つは方向可変性である。

2.1 基本処理方針

本設計過程は、リレー光学系の基本モデル及びさらに方向可変性を追加したモデルを Prolog言語の持つ推論機能を利用してその実現例として求め、その光学特性の動的機能をレイ・トレーシングで確認しその結果を3次元ディスプレイで表示し、最適な実現例をユーザが判断してこれを基本設計とすることにした。これらの作業を対話処理で行うために推論、レイ・トレーシング、表示の3種のプログラムを同時に実行させる協調処理方式とした(図1参照)。

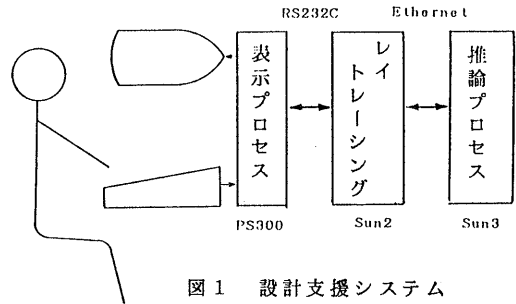


図1 設計支援システム

2.2 リレーレンズ系の述語表現

光学測定機器にリレーレンズ系と呼ばれる光学系が使われることがあるがリレーレンズ系とは、レンズを用いて光学特性を空間的に2つの基準面の間でリレー(中継)するだけでなら特性に変化を与えない光学系である。このリレーレンズ系の条件を述語論理で表現すれば推論で実現例を得ることができる。リレーレンズ系の成立条件である光学特性を変化させないことを述語論理で表現するために二つの典型的な例で代表させて記述した。即ち入射基準面に入る平行な入射光線(平面波)は射出基準面から平行で出てゆき、一方入射基準面内の一点に焦点を結ぶような光線(球面波)は、射出基準面内の一点に集中した後拡散して出てゆく。これを図示したものが図2で、Prolog言語による記述例を図3に示す。同図ではまずリレーレンズ系を構成するレンズの特性を記述し、その後適宜レンズを設置したときの光線の伝播の仕方を記述し、それを利用してリレー系の成立条件を記述している。ここで球面波は、波面の曲率中心位置で記述すれば簡潔となるのでこれを用いる

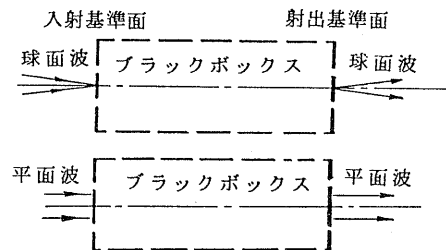


図2 リレーレンズ系の条件

```

lens_character(Focal_pos1,Lens,Lens_pos1,Focal_next):-
    Focal_pos1 < 9999.0, lense(Lens,F),
    A is Lens_pos1 - Focal_pos1,
    A =Y= F,
    B is A*F/(A-F),
    Focal_next is Lens_pos1 + B.

lens_character(Focal_pos1,Lens,Lens_pos1,Focal_next):-
    Focal_pos1 >= 9999.0, lense(Lens,F),
    Focal_next is Lens_pos1 + F.

lens_character(Focal_pos1,Lens,Lens_pos1,99999.0):-
    Focal_pos1<99999.0,lense(Lens,F),
    F is Lens_pos1 - Focal_pos1.

lense(lense1,100.0).
lense(lense2,50.0).

try_locate_lense(S,P,F,Y):- S < 9999.0, Y is S + 3.0*F.
try_locate_lense(S,P,F,Y):- S < 9999.0, Y is S + 2.0*F.
try_locate_lense(S,P,F,Y):- S < 9999.0, Y is S + F.
try_locate_lense(S,P,F,Y):- S < 9999.0, Y is S.
try_locate_lense(S,P,F,Y):- S >= 9999.0, Y is P + 3.0*F.
try_locate_lense(S,P,F,Y):- S >= 9999.0, Y is P + 2.0*F.
try_locate_lense(S,P,F,Y):- S >= 9999.0, Y is P + F.
try_locate_lense(S,P,F,Y):- S >= 9999.0, Y is P.

reference_point(S,P,F,Y):- S < 9999.0, Y is S.
reference_point(S,P,F,Y):- S >= 9999.0, Y is P + F.

ray_passing(Focal_pos1,Pos1_in,Lens,Lens_pos1,
            Focal_nxt,Pos1_nxt):-
    lense(Lens,Focus),
    try_locate_lense(Focal_pos1,Pos1_in,Focus,Lens_pos1),
    lens_character(Focal_pos1,Lens,Lens_pos1,Focal_nxt),
    reference_point(Focal_nxt,Lens_pos1,Focus,Pos1_nxt),
    Pos1_nxt>Pos1_in.

ray_transfer(Limit,((Lens,Lens_pos1),Pos1_out),
            Focal_pos1_in,Pos1_in,Focal_pos1_out,
            Pos1_out,(Focal_pos1_out)):-
    ray_passing(Focal_pos1_in,Pos1_in,
                Lens,Lens_pos1,Focal_pos1_out,Pos1_out),
    Limit < Pos1_out.

ray_transfer(Limit,((Lens,Lens_pos1):List),
            Focal_pos1_in,Pos1_in,Focal_pos1_out,Pos1_out,
            (Focal_nxt:Fcs)):-
    ray_passing(Focal_pos1_in,Pos1_in,
                Lens,Lens_pos1,Focal_nxt,Pos1_nxt),
    Pos1_nxt < Limit,
    ray_transfer(Limit,List,
                Focal_nxt,Pos1_nxt,Focal_pos1_out,Pos1_out,Fcs).

relay(Limit,Plan,Fcs):-
    ray_transfer(Limit,Plan,99999.0,0.0,99999.0,Pos1,Fcs1),
    ray_transfer(Limit,Plan,0.0,0.0,Focal_pos1,Pos1,Fcs2),
    append(Fcs1,Fcs2,Fcs3),
    extract(Fcs3,Pos1,Fcs3).

```

図3 述語表現によるリレーレンズ系

が、この場合平面波が表現できないので、平面波は曲率中心位置を仮に 99999.0としている。またリレーレンズ系を薄凸レンズで構成させることにし、薄凸レンズの条件にGaussの公式を用いた。しかしこの場合も平行光の焦点位置が表現できないため、9999.0以上の値の場合として分離して記述した。リレーレンズ系の成立条件として上記の2つを用いたがこれらは必要条件であるが十分条件ではない。実際2つの条件だけで求めた提示例の中には光学系としての倍率が1でない場合があった。これらの場合はグラフィック出力で直ちに識別できるので、応答性を重視して条件数を増やさなかった。

2.3 方向可変性の条件

光線の方向を変化させるためには、回転する平面鏡による方法や超音波回折格子による方法が考えられるが、ここでは回転鏡を前提として用いることとした。問題はその挿入位置であるが直感的に考えて、入射基準面上の点光源からの光が再び焦点を結ぶ位置（一般に複数箇所ある）または入射基準面にはいる平行な光が焦点を結ぶ位置（これも複数箇所）のいずれかにすることにした。これらの位置は前節に於てリレーレンズ系の構成時に条件を追加することにより求めることができる。

3. グラフィック支援設計過程

前章で示した条件の基本設計解の動作時の特性変化を確認するためにレイ・トレーシングを行い、その結果を3次元表示して視覚的に容易に検討できるようにした。レイ・トレーシング部分については、既に報告済みであるため¹⁾ここでは特性変化についての確認調査について述べる。要求仕様にある方向可変性は、リレー系の射出基準面内の一点を中心として方向を変化させることが望ましい。そのため前章で提案された設計例の回転平面鏡を回転（揺動）させた状態での光線追跡を行った。提示例の中で挿入位置の違いにより射出光線が回転（揺動）しなくて平行移動するものがみられた。また、前述したように与えたリレー系の条件は満たすが倍率が1でないものがあった。この様な所期の意図に反するものはレイ・トレーシングとその結果のグラフィック出力によって直ちに判定出来ることがわかり、グラフィックスの有効性が確認できた。以下グラフィック画面を用いて設計の経過を説明する。図4にリレーレンズ系の実現例を数例示す。これは焦点距離が倍異なる2つのレンズを用いて与えられた距離以上の間隔でリレーレンズ系の例を提示したものであり、右方の原点と左方の四角の枠の位置までがリレー系となっている。直径が大きい方が焦点距離の長いレンズである。原点から点

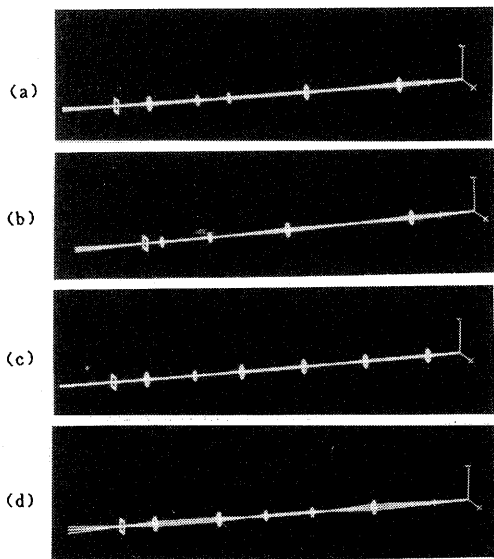


図4 リレーレンズ系の実現例

光源と平行光の2つの光線を与えたときの屈折状態をグラフィック出力したものである。その中で最も単純な部品点数の少ない (b) を最良候補とした。次にこの例について方向可変性を付加した場合に実現例が幾通りか提示されるが、その中で上に述べたように正しく最終光線が振れる場合を取り出したのが図5に示す2例である。この2例について平面鏡をわずかに回転させた例を図6に示す。この後のアイデアは設計者が以上の提示例よりヒントを得てすぐに思い付いたことであるが、図7に示すように図6の2例を同時に適用し、

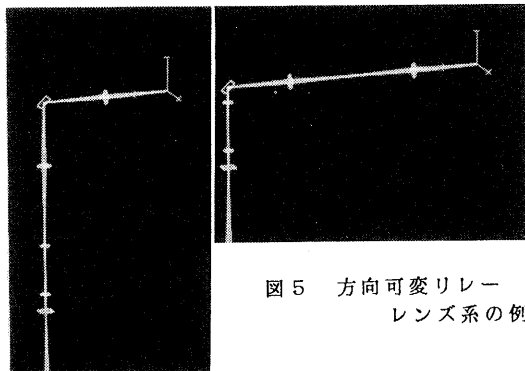


図5 方向可変リレーレンズ系の例

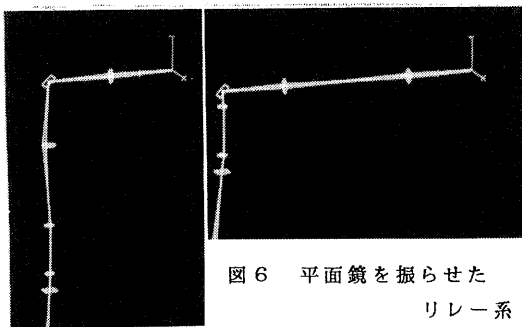


図6 平面鏡を振らせたリレー系

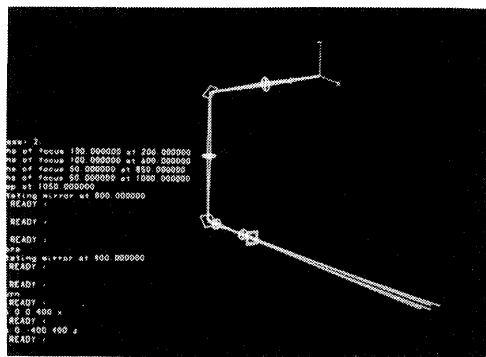


図7 2つの平面鏡を用いたリレー系

```

mm: 2
#e of focus 100 000000 at 200 000000
#e of focus 100 000000 at 400 000000
#e of focus 50 000000 at 800 000000
#e of focus 50 000000 at 1000 000000
#e at 1250 000000
setting mirror at 800 000000
READY:
READY:
+
setting mirror at 400 000000
READY:
READY:
#m
#e of focus 100 400 x
READY:
#e of focus 100 400 #
READY:

```

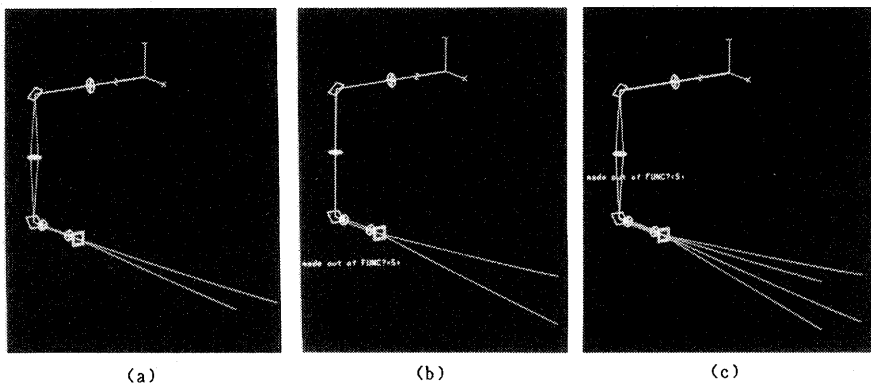


図8 2平面鏡を振らせたリレー系

2つの平面鏡を使って各々の平面鏡に1自由度の回転を与えるだけで、射出光線を2自由度で振らせることが出来る。入射面である原点からの光線を最初の鏡だけで振らせた場合が図8(a)で2番目の鏡だけを振らせた場合が同図(b)で、両方の鏡を振らせると同図(c)の様になる。更にこのままでは光線の振れ角が小さく実用にならないので、口径比を大きく取らなければならないことが判り、レンズを球面鏡におきかえることにした(図9参照)。

球面鏡を使うことによる非点収差を抑えることと光学部品同志の干渉をさけることが相反し、妥協点の検討はレイトレーシングで数値とグラフィックスと両方で確認して、許容誤差範囲に収まるように光学部品の特性や配置を調整した。この作業はレイトレーシングプログラムで対話的にグラフィック出力で確認しながら行った。最終の部品配置は図10に示す通りである。

4. 問題点と今後の方向

一般によく言われていることだが、要求仕様を計算機の内部表現に変換する部分が困難で、本過程では手作業のプログラミングになってしまい今後の検討が必要である。Prolog言語の推論部分、FORTRANによるレイ・トレーシングの数値計算、C言語によるプロセス間通信用インターフェイスプログラム、データフロー型専用言語による3次元ディスプレイプログラム等、各言語の長所を有効に利用するために独立したプログラムを複数実行させて、通信によって効率よく全体としてまとめた仕事を実行することが出来た。今後ネットワークの拡充と共にこの様な分散処理による効率的な方法が有効になるものと予想される。

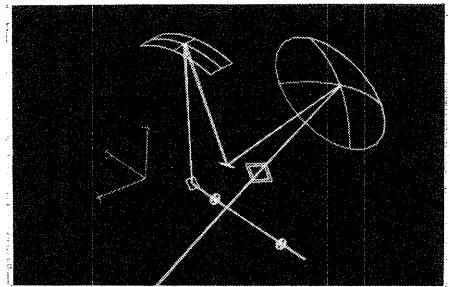


図9 凹面鏡を用いたリレー系

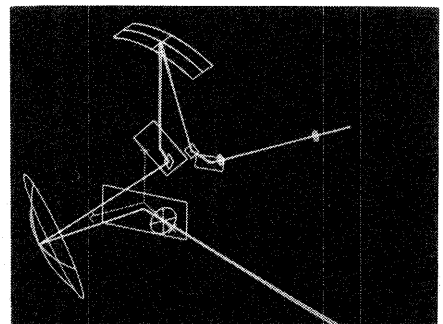


図10 最終配置のリレー系

5. おわりに

本報告ではアイデアの展開の段階で計算機支援を図ることを目的に、条件を与えれば設計案をグラフィック出力する一連のプログラム群を作成し、概要を紹介した。提示された案の一つを更に改良して実用化したが、様々なケースを検討したり、新しいアイデアを生むきっかけにグラフィックインターフェイスは設計者との対話手段に欠かせないことが実感として感じられた。

6. 参考文献

- 1) 福井幸男: オブジェクト指向レイ・トレーシングと応用例、グラフィックスとCAD研究会資料、22、1 (1986)

討 論

13. 設計課程におけるグラフィックインタフェース

福井（製科研）

川合：光源は点光源ですか。

福井：実際に使うときは平行な2本のビームです。最後にやはり平行な光線が出て来ることが条件になります。設計段階では、点光源と平行なビーム（面光源とも考えられる）を使っています。

川合：先ほどの解では、平面鏡はビームが軸に収束したところに置かれていたような気がしますが。

福井：平面鏡をどこに置くかが問題で、これは条件として与えます。ビームが交わる位置が一番ミラーの面積が小さくてすむので、その位置の条件を与えました。

辻堂：収差は考慮していますか。

福井：収差は当然出てきます。収差を最小に押さえるためにレイアウトをうまく考えます。あとは誤差を統計処理し正しい値を推定して使うようにしています。そうすることで1万分の1ぐらいの精度になります。

坂下：リレーレンズ系をPrologの式で記述するのは大変ではないですか。

福井：簡単です。Prologで記述するのはミラーの位置だけで、実際のレイアウトは全部レイトレーシングのプログラムがやっています。

近藤（東芝）：ユーザがミラーを回転させるときにはPrologのプログラムは動かないのですか。

福井：そうです。レイトレーシングのプログラムがユーザに3次元的な場面を与えて、ユーザはインタラクティブにミラーの位置や向きを変えます。Prologの方は要求が来るまで待っています。

近藤（東芝）：応答速度はどれぐらいですか。

福井：現在Sun-2を使っていて、PS-300との間をRS-232Cの9600ボーでつないでいるので、データのやり取りに少し時間がかかりますが、計算自体はほとんど無視できる時間です。

関根：2次元の表示を見て3次元的な光の軌跡を把握できるのですか。

福井：それはPS-300がハードで3次元の表示機能を持っているので、視線方向を変えることによって即座に分かります。

辻堂：PS-300の機能はどの程度使っていますか。

福井：ここで使っているPS-300の機能は、3次元のデータを画面に表示してそれを回転するだけの機能です。高級な機能を使うと移植するときに大変なので使っていません。

川合：リレーレンズ系以外にもこのような仕事はなさっていますか。

福井：少しやっています。