

対話型進化計算手法を用いた フォント自動生成システム

中川雄太[†] 吉田香[†]

DTP を行うにはフォントの選択が非常に重要であるが、PC に予め用意されているフォントの種類は限られている。本研究では、ユーザの主観的評価を随時計算に取り込み最適化を行う対話型進化計算手法を用いて、ユーザの感性に応じたフォントを生成するシステムの構築を行う。その際、欧文書体のエレメントごとにフォントを変異させることにより、デザインの一貫性を保ちながら新しいフォントを生成する手法を提案する。一方、IEC は計算結果が初期集団に大きく依存することや、ユーザの評価による負担、解が収束しないといった問題がある。そこで、本研究ではこれらを解決するための初期集団の生成方法や、設計変数の指定方法を提案する。最後に、提案システムを実装し、動作確認を行った。

Font Generating System using Interactive Evolutionary Computation

Yuta Nakagawa[†] Kaori Yoshida[†]

Font Generating System "PALETTE" is designed to emerge various fonts based on user's Kansei without hand drawing. It enables to generate a new font by selecting user's favorite fonts from displayed candidates, save font information based on user's selection, and reuse saved font to the next font creation. To design PALETTE we focused on Interactive Evolutionary Computation (IEC) technique to take user's subjective desire on font shape into the system. IEC is well known as a method to use user's evaluation interactively. PALETTE has the following modules; (i) create font candidate using basic font shape, for example "century", or saved font information, (ii) generate new fonts using genetic algorithm which can modify each element of the selected fonts, and (iii) export letter with the selected font and save font information. Furthermore we implemented the tool to export font information represented by XML format, because font information should be defined at first. In this paper, we explain about PALETTE in detail and show the experimental results. PALETTE provides not only generating new font but each user's subjective desire on font shape.

1. はじめに

近年、PC の普及や性能の向上により、個人でも DTP や HP 制作など、文字やイラストを用いたデザインを行うことが多くなってきた。DTP を行うにはフォントの選択が非常に重要である[1]が、PC に予め用意されているフォントの種類は限られている。また、フォントエディタにより利用者の好みのフォントを作成することができるが、一文字ずつ文字を構成する輪郭を定義するのは、利用者にとって大きな負担であるといえる。そこで、利用者に負担をかけることなく、利用者の求めるフォントを自動生成するシステムが求められている。利用者の好むデザインは利用者の主観的な判断に依存するため、一般的に定義することは困難である。そのため、利用者の主観的評価を随時計算にとりこむことによって、効率的に利用者の感性を反映する方法が望ましいと考える。

本研究では、利用者の感性に応じたフォントを自動生成するシステムの構築を目的とする。フォントの生成には対話型進化計算手法(Interactive Evolutionary Computation: IEC)[2]を用い、アウトラインフォントを構成する文字の輪郭の座標を動的に変化させることで、利用者の感性に応じたフォントを生成する。進化計算には、生物の進化のメカニズムを模倣したアルゴリズムである遺伝的アルゴリズム(Genetic Algorithm: GA)[3]を用いる。GA は、解の候補をビットコーディングした遺伝子で表現し、適応度の高い個体を優先的に選択して、交叉(組み換え)・突然変異などの遺伝的操作を繰り返しながらよりよい解を探索する手法である。IEC では、GA の評価系を人為的な評価に置き換えることで、利用者が所望する解を見つける。IEC により利用者には複数のフォントが提示されるため、利用者自身の好むフォントや利用者が意図しなかった新規性のあるフォントが作成できると期待される。一方、欧文書体には、エレメントと呼ばれるフォントの造形的特徴をもった部分を構成する要素がある[4]。DTP では欧文書体のエレメントごとにデザインを行い、書体を形成する。本研究では、このエレメントごとに進化計算を行うことで、デザインの一貫性を保ちつつ、新しいフォント生成を行うことを試みる。このため、本研究ではエレメント情報を付与した、拡張可能な書体を XML 形式で定義する。

2. 関連研究

感性工学では、利用者の好みや感性といった非線形な特性を扱うために、ニューラルネットワーク (Neural Network: NN) や GA などの手法が用いられている。奥野らは、強調フィルタリングを用いた感性情報処理フレームワークを提案し、その適用例として漢

[†]九州工業大学大学院情報工学府
Kyushu Institute of Technology Graduated School of Computer Science and Systems Engineering

字のフォント自動作成システムの構築を行っている[5]。NN とは、人間の脳の仕組みを模倣したアルゴリズムであり、定式化が困難な問題に対して既知の学習結果から解を予測する手法である。奥野らのシステムでは、NN により「古い」や「洗練された」など 12 種類の感性語から、利用者の感性に応じたフォントの形状を推測している。その際、過去に蓄積されたユーザ自身のデータと、他のユーザのデータを利用し、協調フィルタリングにより、自身と自身に似ているユーザのデータに重みづけをおこなうことで、ユーザの感性を反映する。また、出力結果をユーザが評価し、教師信号として感性反映部に入力することにより、利用者に意識させずに学習を行わせる。一方で、NN を用いた推測を行うには予めそれぞれの感性語に対するユーザと他のユーザのデータが十分に必要であり、これらのデータを収集することが課題となる。また、感性語に対する印象は人それぞれとらえ方が異なるため、予め用意された 12 種類の感性語の組み合わせから、利用者の作成したい形状を予測するのは困難である。よって、利用者の学習データが少ない場合でも、利用者の感性に応じたフォントを対話的に作成できる機構や、解の多峰性を考慮した特徴量の保存・反映方法を考える必要がある。Michael Schmitz らによりウェブサービスとして公開されている *genotyp* では、GA を用いることで利用者の学習なしに欧文フォントを自動生成することができる[6]。このシステムでは、フォントの形状を表す特徴量として、*extents* (文字の骨格)、*line strength* (文字の太さ)、*serif* (ストロークの端についた小さな飾り) が用いられている。しかし、フォントパラメータが少ないため、文字の形を柔軟に変えることや、文字の一部のみを変えることは困難である。そのため、利用者の細かな感性に応じたフォントを生成することは難しい。

利用者の感性を扱うために、NN や GA では、人間の評価系の代替モデルを作り、最適化システムに組み込む必要がある。しかし、人間の感性を定式化することは非常に困難であるため、人間の評価そのものを評価系に組み込む手法が考えられている[7]。このような手法のうち、EC 技術に GA を用いる方法に対話型遺伝的アルゴリズム (Interactive Genetic Algorithm: IGA) という。IGA は利用者の感性を反映できるという利点から、利用者の感性に応じたコンテンツの検索や作成に用いられている。一般的に GA では多数の解から評価値の高い個体を絞っていき、最適解を 1 つ求める手法である。しかし、IGA では人間の評価がゆらぐため、多様な個体を生成する必要がある。そのため、利用者が予想をしておらず、評価が高い個体を見つけるような、発想支援として活用することができる[8]。しかし、利用者が解を絞り込む目的で評価した場合、利用者の意図していない個体が出ることは望ましくない。また、EC による探索である程度解が絞り込まれると、利用者は手動で解を修正した方が、効率が良い場合もある。そこで濱田らは、このような問題を発想支援、個体の探索、個体の修正の 3 つに切り分けて問題解決を行っている[9]。個体の修正では、利用者自ら手動で個体を修正することにより、確実に目的の個体を生成することができる。しかし、この機能の有

用性は利用者により大きく異なるという報告がなされている。また、岩切らはこれらの問題に対して、すでに収束したパラメータに対して再度発散させるのは不相当だとし、利用者がパラメータに対して収束値を固定できる機能を設けている。また、濱田らと同様に、コンテンツの細かなパラメータを編集できる機能を有し、IEC では表現できない部分を補完している。一方 IGA の別の問題点として、ユーザの評価による負担がある。岩切らが提案する手法のように、一般的に IGA では提示される全ての個体に 5 段階などの利用者の評価を行う必要がある。しかし、全個体へ評価を行う場合、提示個体数や評価世代数の増加によりユーザの負担は増加する。そこで、伊藤らはユーザが個体を選択したか否かの 2 値で評価を行い、選ばれた個体と類似度の高い個体の評価を高くすることにより、ユーザの負担を軽減している[10]。IGA では利用者の感性を反映して集団を生成できるが、細かなパラメータの調整には限度があり、最終的には利用者が手動で修正を加える必要があることがわかる。また、少ない評価で世代交代をさせる仕組みが必要となる。

フォントを生成する研究では、用いる手法に応じて書体を定義する方法が一般的である。野沢らは、漢字のエレメントに着目し、各エレメントの輪郭を、XML を用いて独自に定義を行っている[11]。XML にはエレメントの輪郭を表す輪郭データが記述されており、別に用意した骨格データに肉付けを行うことで文字を形成している。16 種の漢字のエレメントを組み合わせることで、漢字を形成する。フォントの形状を表す特徴量として *Weight*, *Uroko*, *Round*, *Slope*, *SkeltonNoise*, *OutlineNoise* を用いている。同様に、漢字のエレメントごとにデザインを行うシステムとして、KAGE システム (Kanji-glyph Automatic Generating Engine)[12]を用いたグリフウィキ (Glyph Wiki) がある[13]。GlyphWiki では、一字ずつ作成する必要があるが、エレメントを構成する線の形や線の種類など細かな設定ができるため、柔軟なフォントの生成が行える。

3. 欧文書体のエレメント

欧文書体のデザインでは、各部のデザインが決まれば、他の同様の部分も概ね定義できるとされている[14]。このような欧文の造形的特徴をもった部分を表す要素を、欧文書体のエレメントと呼ぶ。例として、大文字の L と R には含まれるエレメントを図 1 に示す。これらは JIS X 4163[15]を引用して作成された標準情報 (TR) TR X 0003:2000[a]に記述されている。例えば、文字の骨格を表す直線部分は *stem* であり、また、嘴のような形状を持つ文字の終端部分は *beak* である。欧文のエレメントは、国や機関・書籍などによってエレメントの名称が異なるため、本研究では標準情報 (TR)

a) 標準情報 (TR) は、TR X 0003:1996 フォント情報処理用語の内容に関して、財団法人日本規格協会情報技術標準化センター (INSTAC) の電子出版技術調査委員会において改正原案を作成し、日本工業標準調査会の審議を経て、タイプ II の TR X 0003:2000 (http://www.y-adagio.com/public/standards/tr_fnttrm/main.htm) として公表するものである。

TR X 0003:2000 に定義されている名称を用いている。

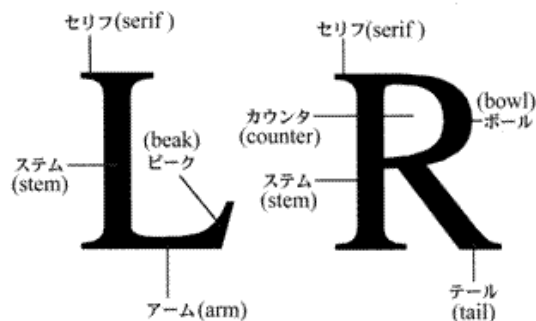


図 1 欧文書体のエレメント[a]

4. 提案システム

本システムは、定義された欧文書体の輪郭の座標を動的に変化させることで、利用者の感性に応じたフォントの生成を行う。欧文書体は、書体定義作成者により、書体定義にしたがって XML で記述される。フォント作成者は、フォント作成アプリケーション(Personal-Adapted LETTER: PALETTE)に生成したい文字列を入力する。IEC により提示されたフォントに対して、選択・評価を行うだけで利用者の好むフォントが生成できると期待される。システムの概要を図 2 に示す。次に、フォントの作成の流れを以下に示す。

4.1 書体の定義

PC 上で扱うさまざまなフォントは、輪郭をデザインすることで、質の高いフォントを作成している。本研究では、これらの既存のフォントの輪郭を利用することで、高品質かつ多様なフォントから書体の定義を行う。また書体の定義は、アルファベットごとに行う。拡張可能な設計にするため、XML 形式で定義する。定義した書体には、その XML が表すアルファベットや参考とした書体名を記述する書体情報、欧文のエレメントを表すエレメント情報、エレメントの輪郭を表す輪郭情報、輪郭情報に記述された座標の位置を明示的に表す位置情報を記述している。図 3 に XML 形式による書体定義の一例を示す。書体の定義の流れを以下に述べる。

まず、書体定義作成者は Inkscape[b]などのベクトル画像編集ソフトウェアを用いて、好みのフォントとでアルファベットを記述した PostScript ファイルを用意する。その後、書体定義支援ツールを用いて、XML ファイルに変換する。最後に、XML Nodepad

b) Inkscape (<http://www.inkscape.org/>)

2007[c]などの XML エディタを用いて、エレメント情報と位置情報を付加することで書体の定義を行う。

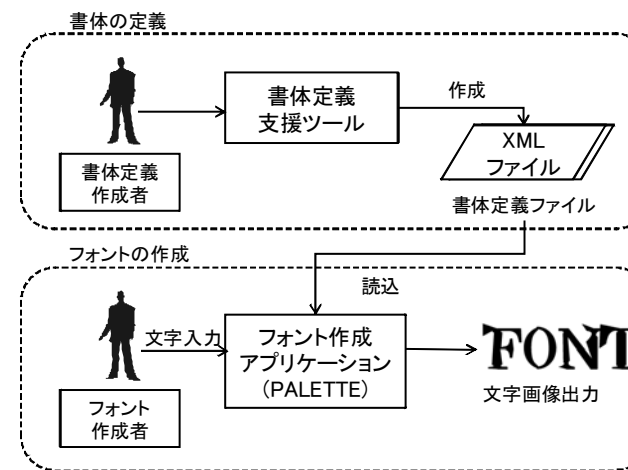


図 2 システム概要

```
<?xml version="1.0" encoding="utf-8"?>
<font char="A" type="century">
  <Stem>
    <vline fromX="195" fromY="131" toX="324" toY="462" position="URP" />
    <hline fromX="324" fromY="462" toX="298" toY="460" position="LRP" />
    <vline fromX="298" fromY="460" toX="241" toY="381" position="LBP" />
    <hline fromX="171" fromY="201" toX="195" toY="131" position="ULP" />
  </Stem>
  <HRB_Serif>
    <curve fromX="324" fromY="462" toX="356" toY="482"
    ctr 1X="329" ctr1Y="475" ctr2X="340" ctr2Y="482"/>
    <hline fromX="356" fromY="482" toX="367" toY="482" />
    ...
  </HRB_Serif>
</font>
```

図 3 XML 形式による書体定義の例

c) XML Nodepad 2007
<http://www.microsoft.com/downloads/details.aspx?familyid=72d6aa49-787d-4118-ba5f-4f30fe913628&displaylang=en>

4.1.1 エレメント情報

標準情報で定義されているエレメントをもとに、造形的特徴のない部分を独自に定義したエレメントを用いて欧文を構成する。標準情報において、欧文の大文字に定義されているエレメントは 14 種であり、さらに本手法では 5 種類を追加して 19 種とする。また、これらのエレメントを向きによる違いを区別することで、31 のエレメントに分けて定義する。例えば、図 4 に示すように、Z にはくちばしのような形状のターミナルである Beak が先端と末端に定義されている。これらの Beak は形が同じであるが向きが異なるため、文字の左上に位置する Beak を LT_Beak、右下に位置する Beak を RB_Beak と区別できるように定義する。定義したエレメント情報については、付録に掲載する。

4.1.2 輪郭情報

各文字は、500×500 のサイズの空間上にフォントサイズ 500 で描いたときの座標データを用いる。エレメント情報を付与した文字の輪郭は図 5 のように表される。各エレメントの輪郭は、直線と 3 次ベジエ曲線によってあらわされる閉曲線として表現する。また、要素名を線の種類とし、直線は line、曲線は curve で記述する。さらに、属性に始点、終点と 2 つの制御点、値にそれぞれの座標値を記述する。



図 4 エレメントの区別

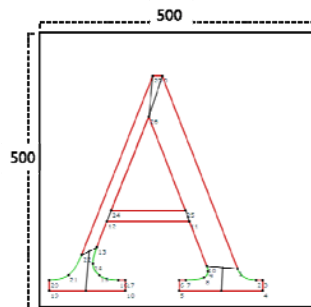


図 5 輪郭データの例

4.1.3 位置情報

本手法では、フォントの輪郭上の決められた点の座標を動かすことで、フォントの輪郭を変更する。位置データは、これらの変更の対象となる座標に付与するデータであり、エレメント内に出現する座標の位置を明示的に表している。位置情報は position 属性で記述され、例えば図 3 の 4 行目に記述している position="URP" は右上(Upper Right Point)を表す。

4.1.4 書体定義支援ツール

前節までに述べた書体定義は XML 形式で表され、テキストエディタなどで作成可能である。本研究では、図 6 に示すような書体定義支援ツールを実装した。このツールでは文字が書かれた PostScript ファイルを、XML に変換することができる。出力された XML にエレメント情報と位置情報を付与することで、書体の定義を行うことができる。

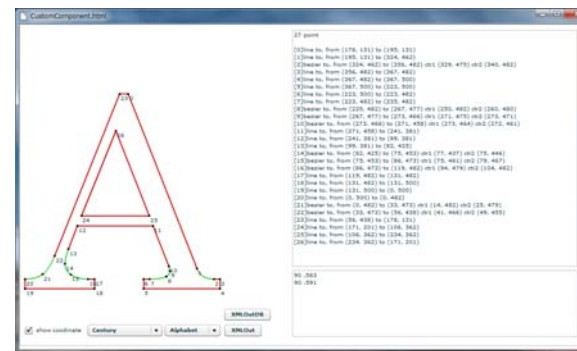


図 6 書体定義作成支援ツール

4.2 フォントの作成

PALETTE では、人工生命の手法や技法を応用し、GA を用いて書体定義を進化させることで、新しいフォントの作成を行う。一般的な IGA では、全ての個体に評価を行う必要があるが、本研究ではユーザが満足する個体をもとに次世代の個体を生成することで、ユーザの選択による負担を軽減する。また、初期集団の生成にフォント名やラベルを選択できるようにすることで、ユーザの求める個体や多様な個体の生成を行う。フォント作成アプリケーションの概要を図 7 に示す。フォント作成の流れを以下に述べる。

まず、フォント作成者は PALETTE に生成したい文字列を入力する。次に、PALETTE によって生成され、提示されたフォントに対して評価を行う。その際、利用者の目標となるフォントが提示された場合、そのフォントを用いて描画した文字列の画像を出力して終了となる。目標となるフォントが提示されなかった場合、提示されたフォントの中から、気に入ったものを 1 つまたは 2 つ選ぶ。なければ、選ばなくてもよい。PALETTE は選ばれた個体数や文字列に応じて、突然変異、交叉、エリート選択などの GA 操作を行う。その後、次世代の集団を生成し、再度利用者に提示する。これを、利用者の目標となるフォントが生成されるまで繰り返す。また、PALETTE では利用者の好みのフォントが生成されたとき、そのときの文字の形を表す特徴量にラベルを付

与して保存することができる。保存した特徴量は、次回の初期集団の生成ときにラベルから参照することができ、いつでも保存したフォントを利用することができる。

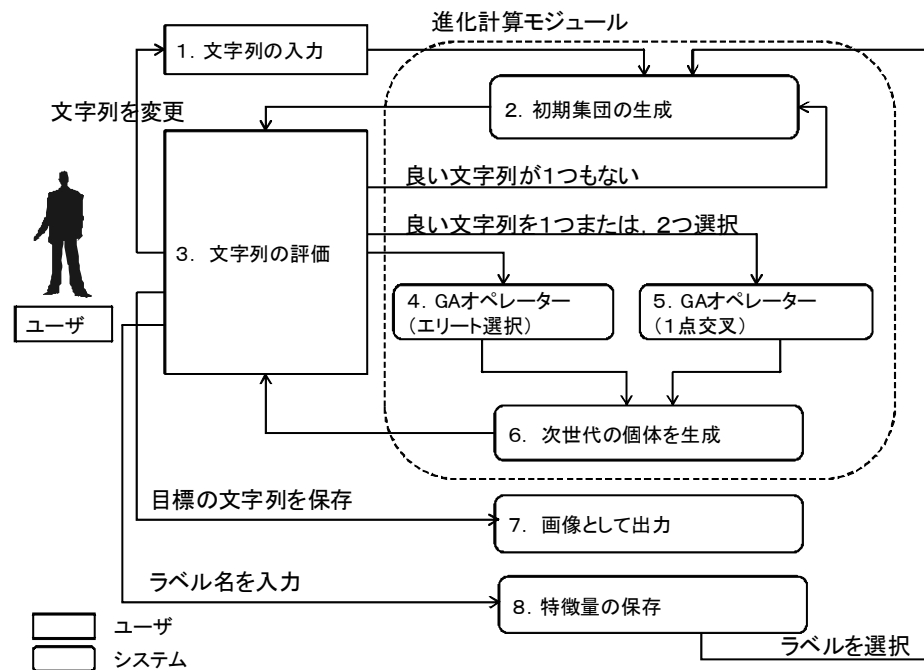


図7 PALETTEの概要

4.2.1 エレメントの輪郭の変異

提案手法では、エレメント毎に輪郭の座標を変えることで、デザインの統一性を保ちつつ、新しいフォントの生成を試みる。これらのエレメントを変異させる方法として、エレメントの輪郭を変異させる操作関数をあらかじめ複数定義する。例として、century 体の G に対して、Spur と呼ばれるひげ根のようなエレメントをシャープな形にする、Stress と呼ばれる曲線部分をふわふわした形にする、Horn と呼ばれる角のような部分を大きくするという操作を与える。これら3つの操作を与えて変異した例を図8に示す。エレメントの操作関数についてはエレメント毎に異なるが、変異させる28エレメントに対して、総数で101の操作関数を定義する。その一例として、Horn, Stress, Spur のエレメントを表1に示す。これらの操作関数を組み合わせて、利用者の求めるフォントを生成する。また、進化計算において、変異する際に使用されるエ

レメントの操作関数はランダムで決定する。

表1 エレメントの操作関数の一例

エレメント	関数	概要
R_Horn	enlargeR_Horn	R_Horn を大きくする
	shrinkR_Horn	R_Horn を小さくする
	mutateR_Horn	R_Horn の形をランダムに変える
Spur	sharpenSpur	Spur をシャープな形にする
	enlargeSpur	Spur を大きくする
	mutateSpur	Spur の形をランダムに変える
Stress	fluffStress	Stress をふわふわした感じにする
	jagStress	Stress をギザギザした感じにする
	mutateStress	Stress を手書き風(ガタガタした感じ)にする

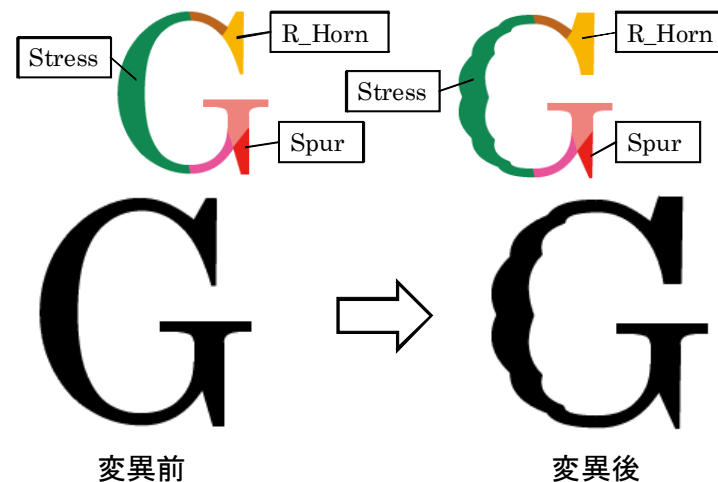


図8 エレメントの輪郭の変異

4.2.2 遺伝子型の定義

提案手法では、GAにおける遺伝子型(染色体)を、前節で述べたエレメントの操作関数の履歴とする。この履歴は、書体定義により作成されたXML形式のフォントから、世代交代により変異したエレメントや変異方法、変異の度合いを記述したもの

である。同様に図 2 における 9. 特徴量の保存では、この遺伝子型が XML で保存され、初期集団の生成において遺伝子型からフォントを復元する操作を行う。

4.2.3 進化計算モジュール

本手法では、フォント作成者の選択による負担を軽減するため、提示された個体を選択したか否かの 2 値で評価を行う。フォント作成者が提示された個体から嗜好に合った個体を親とし、GA 操作を行う。残りの個体に対しては類似度により評価値を計算する手法もあるが、類似度の計算方法と利用者の嗜好が必ずしも一致しないため、選ばれなかった個体は全て死滅させる手法を提案する。以下に、それぞれの GA 操作を述べる。

● 初期集団の生成

GA では、初期集団を発生させ、適応度の高い個体から次世代の集団を作るため、次世代の集団は初期集団に大きく依存するという性質がある。そこで、初期集団には多様な個体を生成し、提示する必要がある。そこで、PALETTE は入力された文字列と指定されたフォント名をもとに書体定義に定義されている XML を取得し、複数回変異させることにより、個体の多様性を創出する。変異する回数は、入力された文字列に含まれるエレメントに応じて変化するが、ユーザの好みの変異量を反映できるようにインタフェースで設定できるようにする。

一方、生成したいフォントの形のイメージを利用者が予め持っている場合、そのイメージから初期個体を生成することが望ましい。そこで、PALETTE ではフォントを作成した際、そのときの遺伝子型を任意のラベルとともに保存できる機構を設ける。これにより、作成したいフォントに似ているフォントを以前生成したことがある場合、その時の遺伝子型をもとに初期集団を生成することにより、フォントを作成する手間を大きく省くことができる。具体的には、選択されたラベルと対になっている遺伝子型を参照する。ある個体をその遺伝子をもとに生成し、残りの個体はその個体の複製に置き換える。その後、複製した個体を複数回変異させることにより、初期集団としての多様性を創出する。

● エリート選択

ユーザの選択において、選択された個体が 1 つである場合、エリート選択を行う。対話型進化計算では、遺伝的アルゴリズムにおける評価関数がユーザの選択そのものとなる。そこで、選ばれた個体は最も評価が高い個体であるとし、残りの個体を選ばれた個体の複製に置き換える。その後、置き換えた個体を複数回変異させ、多様な個体群を生成する。最後に、選んだ個体と置き換えた個体を次世代の個体としてユーザに提示する。

● 1 点交差

ユーザの選択において、選択された個体が 2 つである場合、1 点交差を行う。選ばれた個体は評価が高い個体であるとし、次世代の親とする。次に、親となる 2

つの個体の遺伝子から、交叉点を 1 点ランダムに選ぶ。選んだ交叉点の場所をもとに遺伝子配列を入れ替え、次世代の遺伝子を 2 つ生成する。次に、生成された遺伝子をもとに個体を生成し、全個体をそれに置き換える。その後、全ての個体に複数回変異を与えることにより、個体の多様性を創出する。最後に、個体をユーザに提示する。

● 初期化

ユーザの選択において、選択された個体がない場合、初期化を行う。全ての個体を死滅させ、初期集団の生成と同様に、書体定義に定義した XML から複数回変異を与えることで、再度多様な個体を生成し、ユーザに提示する。

4.2.4 進化の収束

利用者によっては、世代交代がある程度進むと、目標とするフォントに形が近くなったため、これ以上変異させたくないエレメントが存在する場合あると考えられる。しかし、EC では評価系をユーザの選択に置き換えているため、本研究のような、解が無数にあるような問題に対して、最適解に近いかどうかはユーザにしか分からない。そのため、システムは多様性を創出するための進化を繰り返し、解が発散してしまう可能性がある。ユーザの選択履歴から、最適な特徴量を推測することは可能であるが、それが必ずしも適切でなく、また特徴量の多い問題に対しては、ユーザは多くの個体を何世代にもわたって評価しなければならないため、ユーザに負担がかかるという問題が発生する。そこで、PALETTE では、変異させるエレメントをユーザが指定できるようにすることで、ユーザの任意のタイミングで解を収束できるようにする。具体的には、初期集団を生成した際、入力された文字列に含まれるエレメントのチェックボックスが有効になり、変異させたくないエレメントのチェックを外すことにより、解の不本意な発散を防ぐ。

4.3 フォント作成アプリケーション (PALETTE)

提案手法を実装したアプリケーションである PALETTE を図 9 に示す。PALETTE のユーザインタフェースは、個体表示エリア、進化計算操作エリア、設計変数設定エリアに分けられる。個体表示エリアでは、8 個のパネルに、生成したフォントで入力文字列が描画される。それぞれのパネルに、IEC の選択に使用するチェックボックスと、詳細画面を表示するボタンを用意している。詳細画面を図 10 に示す。詳細画面では、生成したフォントを構成するエレメントの確認や、画像としての保存、ラベルの保存が行える。進化計算操作エリアでは、文字列を入力するエリアや進化させるボタン、初期化を行うボタンがある。設計変数設定エリアでは、GA における突然変異率や変異強度などの値の設定、初期個体を生成する際に用いるフォントの選択、初期個体を生成する際のラベルの選択、世代交代において変異させるエレメントの設定を行うことができる。

今後は評価実験を行い、PALETTEの有用性の評価を行う。評価指標としては、生成されるフォントの多様性、目標のフォントが生成できることの実用性、利用者の負担の軽減やPALETTEの使い勝手などの使用性を予定している。

PALETTEはIECにより、利用者の好みのフォントや思いもよらない新規性のあるフォントが生成できることが期待できる。しかし一方で、遺伝的アルゴリズムの初期個体に大きく依存するという性質を受け継いでいる。そこで、初期個体を生成する際に用いる書体定義を増やすことや操作関数を増やすことがあげられる。また、GAは最適解の一つを見つける方法であるため、世代交代を繰り返すに当たり似たような個体ばかり残るといった問題点がある。しかし、利用者の感性というものは曖昧であるため、目標となる個体が複数存在する場合がある。そこで、選択を行う際に利用者が意図的にグループ分けを行えるようにすることで、嗜好の多峰性に対応可能になると考える。

謝辞

本研究は日本学術振興会科学研究費補助金(18100001)の助成による研究成果である。

参考文献

- [1] Robin Williams, The Non-Designer's Design Book Second Edition, 毎日コミュニケーションズ(2004)
- [2] 山川望, 廣安知之, 三木光範, 対話型遺伝的アルゴリズムの基礎, ISDL Report No.20050916001(2005)
- [3] 伊庭齊志, 進化論的計算手法, オーム社(2005)
- [4] 亀尾敦, 大森裕二, Design Basic Book: はじめて学ぶ, デザインの法則, BNN 新社(2007)
- [5] 奥野陽, 萩原将文, 協調フィルタリングを用いた感性情報処理フレームワーク, 日本感性工学大会予稿集第9th 頁:B61(2007)
- [6] Michael Schmitz, genoTyp, University of Arts Berlin(2004)
<http://www.genoTyp.com>
- [7] 高木英行, 畝見達夫, 寺野隆雄, 対話型進化計算手法の研究動向, 人工知能学会誌 13(5), pp.692-703(1998)
- [8] 西野浩明, 高木英行, 宇津宮孝一, 対話型進化計算を用いた創作支援型3次元モデラ, 電子情報通信学会論文誌, Vol.J,85-D-II, No9, pp.1473-1483(2002)
- [9] 濱田悠介, 狩野均, 発送支援機能を有する対話型進化計算によるWebページのデザイン生成システム, 知能システムシンポジウム資料 35th, pp160-165(2008)
- [10] 伊藤冬子, 山川望, 廣安知之, 三木光範, 佐々木康成, 対話型遺伝的アルゴリズムによるユーザの嗜好に基づいた商品選択支援, 電子情報通信学会第11回 Web インテリジェンスとインタラクション研究会(WI2)(2008)
- [11] 野沢貴, 堀田創, 萩原将文, パラメータ拡張可能なフォント自動生成システムへの応用, 電子情報通信学会論文誌, Vol.J90-D No.10, pp2755-2763(2007)
- [12] 上地宏一, KAGEシステムによる漢字フォント政策支援, CHISE Conference2005 報告書, pp21-26(2005)
- [13] 上地宏一, Glyph Wiki, <http://fonts.jp/glyphwiki/>

[14] 小林章, 欧文書体—その背景と使い方(新デザインガイド), 美術出版社(2005)

[15] Information technology - Font information interchange Part 3: Glyph shape representation JIS X

付録

付録 A.1 定義したエレメント

本研究で定義したエレメントを以下の図に示す。独自に定義したエレメントは、Horn, Nose, Spur, Tongueの4エレメントである。また、変異に用いないエレメントはここでは紹介していない。

