

Aleksander Klosov, Łukasz Wolan

Państwowa Wyższa Szkoła Zawodowa im. Witelona w Legnicy

KOMPUTEROWA METODA GENEROWANIA TWARZY NA PODSTAWIE ANALIZY PCA ORAZ ALGORYTMU GENETYCZNEGO

1. Wstęp

Coraz częściej na stronach internetowych ich użytkownikom towarzyszą wirtualni asystenci internetowi, którzy udzielają informacji i służą rozmową. Dlaczego nie mogliby oni zmieniać wyglądu w zależności od indywidualnych upodobań każdego internauty lub mieć prezentację podobającą się wszystkim. Można oczywiście pozwolić użytkownikom na modyfikowanie wyglądu strony przez wybór „skórki” lub zmianę parametrów odpowiadających za wygląd. Użytkownik może również wstawiać własne zdjęcia asystenta. W dobie rozwiniętej technologii komputerowej pojawia się też inna koncepcja: komputer, poznając nasze preferencje, tworzy coś, co nam odpowiada. Preferencje można poznać, posługując się zestawem pytań, ale w tym przypadku nie jest to wygodne. Najłatwiej jest pokazać coś użytkownikowi i zapytać, czy mu się to podoba. I tu pojawia się koncepcja szacowania obiektów graficznych na stronie internetowej.

Pomysł polega na wykorzystaniu algorytmu genetycznego do generowania obrazów ocenianych przez użytkowników. Algorytmy genetyczne są znaną od kilkadziesiąt lat i sprawdzoną techniką poszukiwania optymalnych rozwiązań. W tym przypadku rozwiązania poszukujemy w przestrzeni obrazów. Kierunek poszukiwań określa użytkownik przez wystawianie ocen lub dokonanie jednoznacznych wyborów. Koncepcja ta wykorzystywana jest obecnie w programach tworzących portrety pamięciowe, z tym że działają one w środowisku jednorazowego użytkownika. Znane jest też wykorzystanie na stronie internetowej algorytmu ewolucyjnego do tworzenia twarzy, która zdaniem użytkowników jest uznawana, zgodnie z przyjętym kryterium oceny, za atrakcyjną lub młodą (rys. 1).



Rys. 1. Obraz stworzonej komputerowo twarzy, uznanej za najatrakcyjniejszą przez uczestników projektu dr V. Johnstona

Źródło: http://www-psych.nmsu.edu/~vic/faceprints/female_study.html.

Generowanie twarzy może przydać się nie tylko naukowcom, ale także mieć bardziej powszechne zastosowanie. Można tworzyć twarze do celów rozrywkowych, np. twarze bohaterów gier komputerowych, które będą się podobały przyszłym graczom, albo twarze internetowych chatbotów czy asystentów w portalach internetowych, ułatwiające poszukiwanie informacji przez prowadzenie dialogu z użytkownikiem w języku naturalnym.

Celem artykułu jest omówienie metody generowania twarzy ludzkiej na podstawie subiektywnych ocen wielu użytkowników z wykorzystaniem algorytmu genetycznego.

Opisane zastosowania nie ograniczają się wyłącznie do zagadnienia generowania twarzy ludzkiej, ale mogą być z powodzeniem wykorzystywane również do dopasowania wyglądu stron internetowych, a także w dziedzinach, takich jak projektowanie wnętrza oraz projektowanie karoserii samochodowych.

2. Przegląd istniejących rozwiązań

Generowanie twarzy z fragmentów

Najprostszą metodą generowania twarzy, jaka może każdemu przyjść na myśl, jest wykorzystanie gotowych komponentów, z których, jak z klocków, składa się dowolny obraz. Jeśli rozpatrzmy przykład rysowania portretu pamięciowego, to okazuje się, że wiele systemów korzysta właśnie z tej metody. Rysunek 2 pokazuje przykład wykorzystania systemu E-FIT. Innym popularnym systemem jest system

FACES. Twarz jest podzielona na wiele obszarów. Dla E-FIT są to: 1) włosy, 2) brwi, 3) głowa, 4) uszy, 5) oczy, 6) nos, 7) usta. W wypadku FACES jest znacznie więcej obszarów, czyli typów komponentów: 1) włosy, 2) głowa, 3) czoło, 4) brwi, 5) oczy, 6) nos, 7) usta, 8) szczęka, 9) podbródek, 10) linie ust, 11) linie uśmiechu, 12) linie oczu, 13) różne rodzaje zarostu.



Rys. 2. System tworzenia portretów pamięciowych E-FIT

Źródło: opracowanie własne.

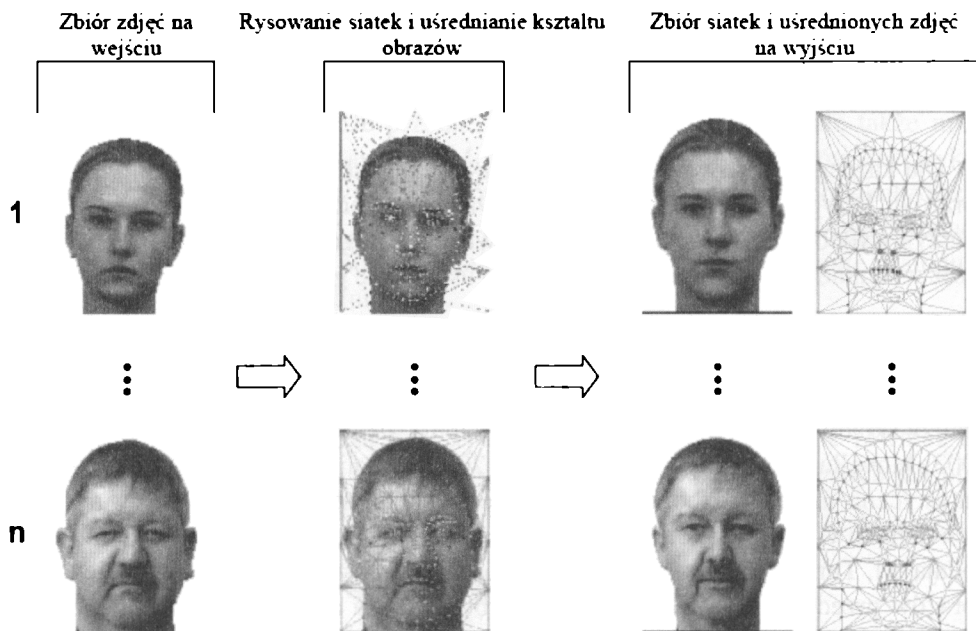
Dla każdego z typów fragmentów twarzy istnieje zbiór wielu instancji w postaci obrazów. Przygotowanie takiego zbioru wymaga stworzenia dużego zestawu zdjęć twarzy i wydobycia z każdego zdjęcia odpowiednich fragmentów. Twarz powstaje poprzez wybranie komponentu z każdego typu i ustawienie go w odpowiednim miejscu.

Generowanie obrazów z obiektów własnych

Naukowcy zajmujący się rozpoznawaniem obrazów, a konkretnie twarzy, w poszukiwaniu odpowiednich metod reprezentacji danych testowych sięgnęli do analizy składowych głównych – PCA (*Principal Component Analysis*) [Draper i in. 2003; Movellan 2003; Shlens 2003; Smith 2002]. Analiza składowych głównych to statystyczna metoda rozpoznawania wzorców w zbiorze danych i przedstawiania tych danych w sposób, który uwydatnia różnice i podobieństwa w tym zbiorze. Jak podają M. Turk i A. Pentland [1991], pomysł zastosowania PCA w kodowaniu obrazów przedstawili T. Sirovich i W. Kirby. Analiza dokonywana jest na zbiorze czarno-białych obrazów, dla których obliczana jest macierz kowariancji, a dla niej z kolei wartości i wektory własne. Wektory własne stanowią nową bazę w przestrzeni obrazów. Do tej nowej przestrzeni przekształca się obrazy za pomocą macierzy wektorów własnych. Składowe główne uzyskane w wyniku analizy (nazywane „obrazami własnymi” – *eigenpictures*) można uważać za zbiór cech, którymi różnią się obrazy. Oryginalne obrazy kodowane są przez kombinacje obrazów wła-

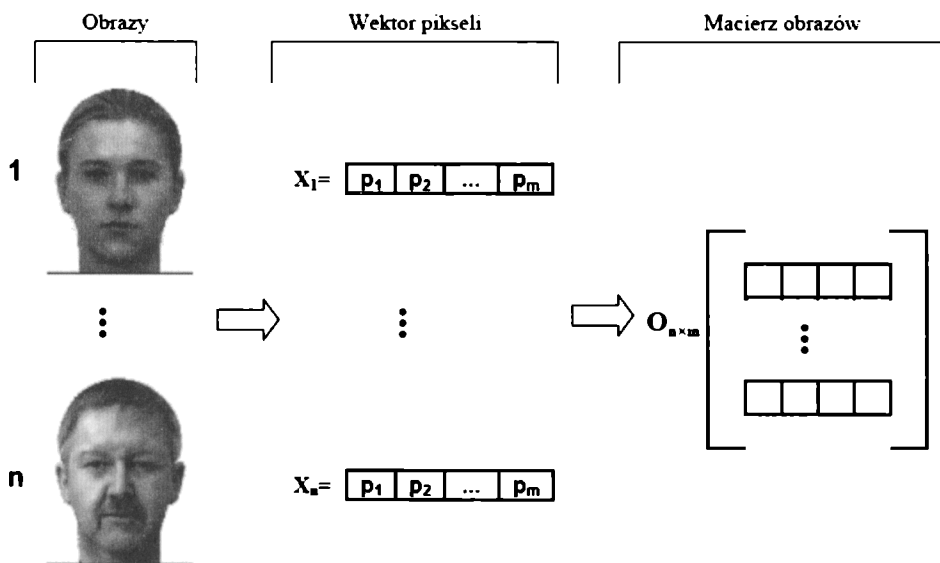
snych w odpowiednich proporcjach. Te proporcje tworzą pewien wektor współczynników. Jeśli chcemy rozpoznać jakiś obraz, należy go zapisać w postaci kombinacji liniowej obrazów własnych. Teraz, wśród oryginalnych obrazów, wystarczy znaleźć wektor współczynników, który jest najbardziej podobny do wektora rozpoznawanego obrazu. Podobieństwo wyraża się zazwyczaj zwykłą odległością euklidesową. Jeśli utworzymy liniową kombinację obrazów własnych w proporcjach innych niż dla oryginalnych obrazów, otrzymamy nowy obraz. Pomysł ten jest podstawą do generowania dowolnych obrazów. Opis jego wykorzystania można znaleźć w [Frowd i in. 2004; Gibson i in.; Hancock 2000; Hancock [www.stir.ac.uk/...](http://www.stir.ac.uk/)].

Aby analiza składowych głównych dała dobre efekty, należy wyodrębnić kształt i teksturę obrazu. W tym celu konieczne jest narysowanie siatki, która określi kształt i położenie ważniejszych fragmentów obrazu. Dla zbioru siatek można wyznaczyć średni (przeciętny) kształt obiektu na obrazie. Następnie przekształca się wszystkie obrazy za pomocą transformacji graficznych do obrazów o średnim kształcie. Obrazy te określa się też jako pozbawione kształtu – (*shape-free*). Rysunek 3 pokazuje ten proces.



Teraz analizie PCA można poddać osobno uśrednione obrazy oraz siatki kształtów. Wynikiem będzie otrzymanie modelu tekstur i kształtów, przy czym określe-

nie „tekstura” jest tu używane do określenia obrazu (wartości odcieni szarości pikseli), z którego usunięto kształt. Aby poddać dane dalszej analizie, należy je przedstawić w formie macierzowej. Na rysunku 4 przedstawiono sposób utworzenia macierzy obrazów. Kolejne wiersze lub kolumny pikseli ustawiane są w jednym wierszu. Każdy obraz tworzy jeden wiersz w macierzy. Podobnie tworzy się macierz dla siatek (rys. 5). Współrzędne każdego punktu siatki zapisuje się w kolejnych kolumnach wiersza odpowiadającego danej siatce.



Rys. 4. Utworzenie macierzy obrazów

Źródło: opracowanie własne.

Przed przystąpieniem do PCA od każdego wiersza macierzy trzeba odjąć średnią wszystkich wierszy, czyli przeciętną teksturę dla obrazów i przeciętny kształt dla siatek. Średnia tekstura i kształt będą później wykorzystywane przy tworzeniu obrazów. Macierz kowariancji można wyznaczyć jako iloczyn:

$$C = AA^T,$$

gdzie A jest macierzą obrazów lub siatek. Dla macierzy kowariancji wyznaczone są macierze wartości i wektorów własnych, które są posortowane względem malejących wartości własnych. Po wymnożeniu wektorów własnych tekstur i macierzy obrazu otrzymujemy macierz „tekstur własnych”, które dla twarzy nazywa się „twarzami własnymi” (*eigenfaces*). Analogicznie, dla kształtów tworzy się macierz „kształtów własnych” (*eigenshapes*). Jeśli przyjmniemy, że X jest macierzą tekstur lub siatek, a P macierzą odpowiadających im wektorów własnych macierzy kowariancji, to:

$$Y = PX$$

jest macierzą obiektów własnych (kształtów lub tekstur).

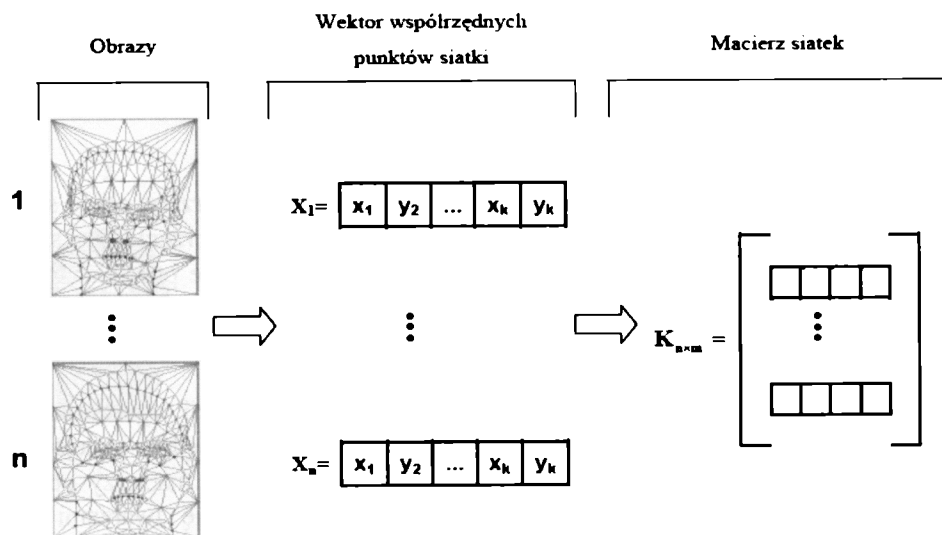
$$\text{cov}(X, Y) = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)}$$

$$O_{n \times m} = \begin{bmatrix} \text{---} \\ \vdots \\ \text{---} \end{bmatrix} \begin{matrix} X_1 \\ \vdots \\ X_n \end{matrix} \Rightarrow C_{n \times n} = \begin{bmatrix} \text{cov}(X_1, X_1) & \cdots & \text{cov}(X_1, X_n) \\ \vdots & \ddots & \vdots \\ \text{cov}(X_n, X_1) & \cdots & \text{cov}(X_n, X_n) \end{bmatrix}$$

Rys. 5. Wyznaczenie macierzy kowariancji dla macierzy obrazów

Źródło: opracowanie własne.

Utworzenie dowolnego obrazu polega na stworzeniu wektorów współczynników kształtu i tekstur, obliczeniu dla każdego z nich kombinacji liniowej odpowiednich obiektów własnych i dodaniu do niej średniej (kształtu i tekstury) obliczonej przed wykonaniem PCA. Otrzymujemy wtedy nowy obraz o średnim kształcie i nowej siatce. Teraz wystarczy obrazowi nadać kształt tej siatki.



Rys. 6. Przekształcenie siatek na macierz

Źródło: opracowanie własne.

Dowolna tekstura lub siatka **A** obliczana jest ze wzoru:

$$\mathbf{A} = \mathbf{W}\mathbf{Y} + \mathbf{X},$$

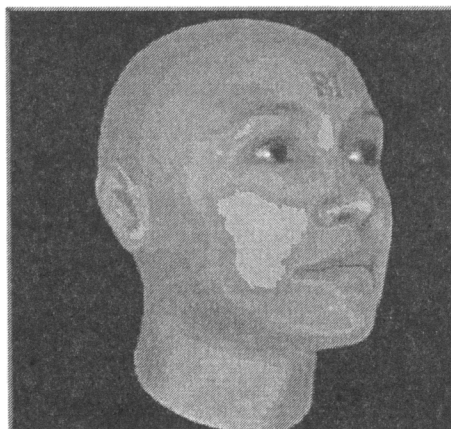
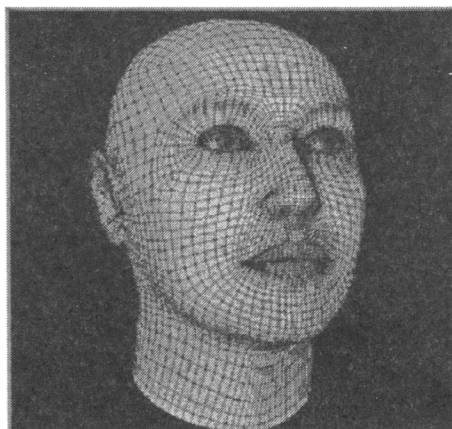
gdzie: **W** – wektor współczynników,

Y – macierz obiektów własnych,

X – średni wektor (obrazu lub siatki) obliczony na początku. Przy obliczaniu obrazu można ewentualnie pominąć obiekty własne, które nie kodują istotnych informacji, np. kąta nachylenia głowy w obrazach twarzy.

Renderowanie obrazów

Najbardziej skomplikowaną wśród opisywanych tu metod jest renderowanie obrazu. Wymaga ono stworzenia siatki obiektu i jego modelu (rys. 7).



Rys. 7. Przykład modelu oraz renderowania twarzy w programie FaceGen Modeller

Źródło: opracowanie własne.

Analizowany w pracy program FaceGen Modeller potrafi jednak stworzyć renderowany obraz głowy na podstawie zdjęć twarzy. Modyfikowanie cech twarzy, zarówno kształtu, jak i tekstury, możliwe jest za pomocą zmiany parametrów, takich jak np. wysunięcie podbródka czy uniesienie brwi.

3. Koncepcja wykorzystania algorytmu genetycznego

Genotyp

Jeśli chodzi o metodę tworzenia obrazu twarzy na podstawie jej fragmentów, osobnik mógłby się składać z trzech chromosomów:

1) komponentu – przechowującego numery komponentów, z których składa się obraz;

2) położenia – przechowującego współrzędne odpowiedniego komponentu lub przesunięcie względem domyślnego położenia;

3) rozmiaru – przechowującego wymiary komponentów lub ich skalę względem oryginalnego rozmiaru.

Przykład takiego osobnika zamieszczono na rys. 8.

Typ komponentu	Chromosom 1 (komponentu)	Chromosom 2 (położenia)	Chromosom 3 (rozmiaru)
oczy	120	5,6	1
nos	23	0.0	1
...	...	...	...
linie ust	5	1.-5	2

Rys. 8. Przykład osobnika złożonego z trzech chromosomów kodujących komponenty

Źródło: opracowanie własne.

Jeśli chodzi zaś o metodę obiektów własnych, osobnik może składać się z dwóch chromosomów. Pierwszy odpowiada za teksturę, a drugi za kształt. Chromosomy są wektorami współczynników dla obiektów własnych. Każdy chromosom ma stałą długość, określoną przez liczbę obiektów własnych, przy czym długość chromosomu tekstury i kształtu nie musi być równa. Każdy gen – współczynnik jednego obiektu własnego – jest tu liczbą rzeczywistą. Aby obrazy wyglądały w miarę realistycznie, odpowiednie wydaje się określenie minimalnej i maksymalnej wartości każdego z genów, które odpowiadają minimalnym i maksymalnym wartościom współczynników z bazowego zbioru obrazów.

Reprodukcja

Reprodukcja proporcjonalna (ruletkowa). W reprodukcji proporcjonalnej zmienną losową określającą prawdopodobieństwo reprodukcji każdego osobnika należy zdefiniować ze wzoru:

$$p_r(X) = \frac{\Phi(X)}{\sum_{Y \in P'} \Phi(X)}.$$

Prawdopodobieństwo wylosowania osobnika jest wprost proporcjonalne do jego przystosowania. Stosunek oczekiwanej liczby kopii dwóch osobników w populacji rodzicielskiej jest równy stosunkowi wartości ich przystosowania:

$$\frac{E(n(X))}{E(n(Y))} = \frac{p_r(X)}{p_r(Y)} = \frac{\Phi(X)}{\Phi(Y)}.$$

Reprodukcja rangowa. Prawdopodobieństwo reprodukcji każdego osobnika jest podane na podstawie rangi – liczby charakteryzującej jakość osobnika na tle populacji bazowej. Populację należy posortować według nierosnących wartości przystosowania osobników. Następnie należy przypisać osobnikom rangi, czyli kolejne liczby całkowite, poczynwszy od zera. Jeśli osobnicy mają taką samą wartość przystosowania, to nadaje się im tę samą rangę.

Prawdopodobieństwo należy obliczyć ze wzoru:

$$p_r(X) = a + k(r_{\max} - r(X))^b,$$

gdzie: r – ranga,

a, b, k – parametry dobrane tak, aby:

$$\sum_{i=1}^n p_r(X) = 1,$$

$$0 \leq p_r(X) \leq 1,$$

$$r(X) \geq r(Y) \Rightarrow p_r(X) \leq p_r(Y).$$

Reprodukcja rangowa pozwala na swobodne kształtowanie nacisku selektywnego [Arabas 2001]. Zwiększanie a powoduje globalne osłabienie nacisku selektywnego. Wzrost b zwiększa zróżnicowanie prawdopodobieństw reprodukcji lepszych i gorszych osobników. Wartości te należy ustalić w drodze testów na etapie dostrajania systemu.

Reprodukcja turniejowa. Z populacji bazowej wybiera się q osobników, następnie przeprowadza się między nimi turniej. Osobnik o największej wartości przystosowania trafia do populacji rodzicielskiej. Proces ten powtarza się aż do zapelnienia populacji rodzicielskiej.

Osobników do turnieju wybiera się po nadaniu im unikatowych rang, zgodnie ze wzorem:

$$p_r(X) = \frac{q}{n} \prod_{k=1}^{q-1} \frac{n - r(X) - k}{n - k}$$

lub

$$p_r(X) = \frac{q}{n} \left(\frac{n - r(X)}{n} \right)^{q-1},$$

gdzie: r – ranga,

n – licznosc populacji bazowej,

q – liczba osobników wybieranych do turnieju. Zastosowanie pierwszego zbioru sprawia, że $q - 1$ najgorszych osobników nie będzie reprodukować. Parametr q steruje intensywnością nacisku selektywnego i należy jego wartość ustalić w drodze testów na etapie dostrajania systemu.

Reprodukcja metodą wyboru losowego według reszt bez powtórzeń. Jest to kolejna warta uwagi metoda. Dla każdego osobnika oblicza się oczekiwaną liczbę kopii, która jest iloczynem liczności populacji i zwykłego prawdopodobieństwa reprodukcji:

$$E(n(X)) = n * p_r(X) = n \frac{\Phi(X)}{\sum_{Y \in P'} \Phi(X)},$$

gdzie n to liczność populacji bazowej.

Osobnik zostanie skopiowany tyle razy w populacji rodzicielskiej, ile wynosi część całkowita oczekiwanej liczby kopii. Części ułamkowe natomiast traktowane są jako prawdopodobieństwa sukcesu w kolejnych próbach Bernoullego, które wykonywane są do momentu zapelnienia populacji rodzicielskiej.

Sukcesja

Sukcesja z całkowitym zastępowaniem. Jest to najprostszy i najczęściej stosowany schemat. Nowa populacja jest tworzona przez całkowite zastąpienie populacji bazowej potomną.

Sukcesja elitarna. Nowa populacja tworzona jest częściowo z osobników populacji bazowej, a częściowo z populacji potomnej. Ogólny schemat polega na wybraniu z populacji bazowej h najlepszych osobników i połączeniu ich z populacją potomną. Z tej połączonej populacji wybieranych jest n najlepszych osobników, gdzie n to rozmiar nowej populacji.

W naszym projekcie ze względu na małą liczbę osobników h powinno być równe 1.

Krzyżowanie

Krzyżowanie wymieniające równomierne. Osobnicy potomni Y i Z tworzeni są przez wymianę każdego z genów osobników rodzicielskich $X1$ i $X2$, z zadanyim prawdopodobieństwem p^c .

$$Y_i = \begin{cases} X_i^1 \Rightarrow \xi_{U(0,1),i} < p_c \\ X_i^2 \Rightarrow \xi_{U(0,1),i} \geq p_c \end{cases},$$

$$Z_i = \begin{cases} X_i^2 \Rightarrow Y_i = X_i^1 \\ X_i^1 \Rightarrow Y_i = X_i^2 \end{cases}.$$

Funkcja przystosowania

Funkcja przystosowania na podstawie subiektywnych ocen użytkowników wyznacza wartość przystosowania osobników. Najprostszą metodą jest wyznaczenie średniej arytmetycznej. Użytkownicy mogą oceniać obrazy w skali punktowej, jak to jest pokazane na rys. 9, gdzie przykładowo 0 to najniższa ocena, a 5 najwyższa. Przystosowanie

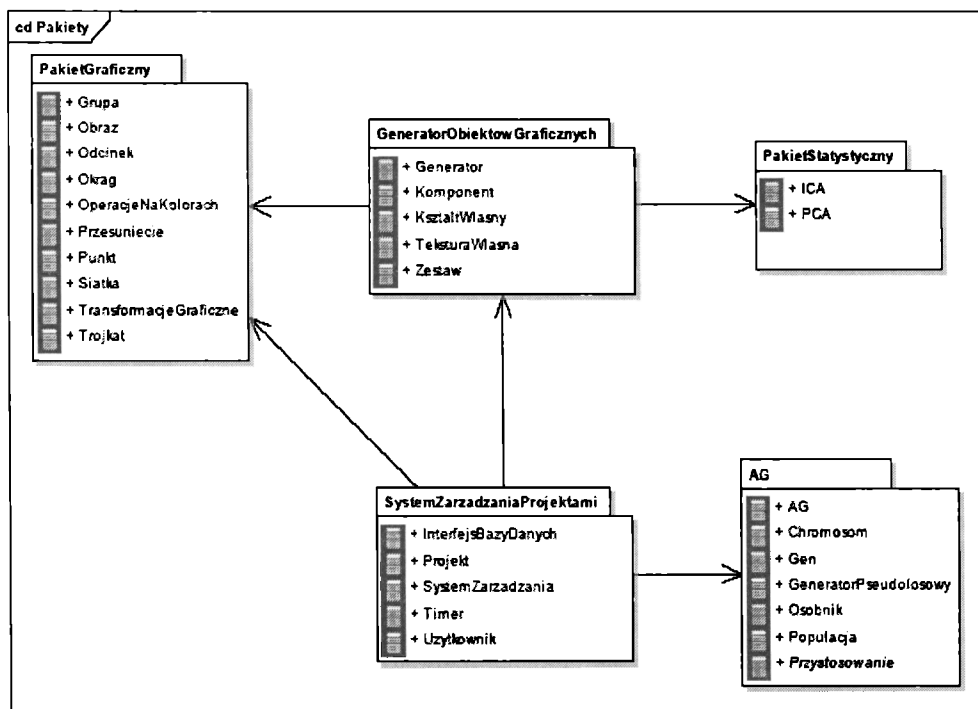
osobnika będzie średnią tych ocen. Jeśli przyjmiemy schemat oceniania pokazany na rys. 10, w którym użytkownik wybiera jeden lub kilka obrazów, można przyjąć, że wybrane obrazy otrzymują najwyższą ocenę, np. 5, a nie najniższą, np. 0.

Ponieważ w środowisku internetowym daną generację jest w stanie oceniać różna liczba użytkowników, można wprowadzić wagi ocen użytkowników proporcjonalne do liczby generacji, w których wzięli udział w aktualnym projekcie. Zatem osoby, które rzadko oceniają, miałyby mniejszy wpływ na wynikowy obraz.

4. Opis aplikacji wspomagającej testowanie

Architektura funkcjonalna aplikacji

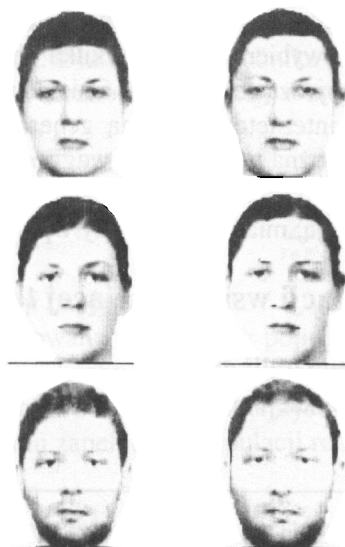
Wszystkie komponenty aplikacji zostały podzielone na 5 pakietów pokazanych na rys. 9.



Rys. 9. Architektura aplikacji komputerowej na potrzeby testowania

Źródło: opracowanie własne.

Aplikacje wykonano z użyciem języka Java. Interfejs użytkownika wykonano w postaci prostej strony internetowej. Fragment jednej ze stron testujących pokazano na rys. 10.



Rys. 10. Przykładowy sposób oceniania obrazów

Źródło: opracowanie własne.

Moduł AG

Diagram klas modułu implementacji algorytmu genetycznego przedstawiono na rys. 11.

System zarządzanie projektami

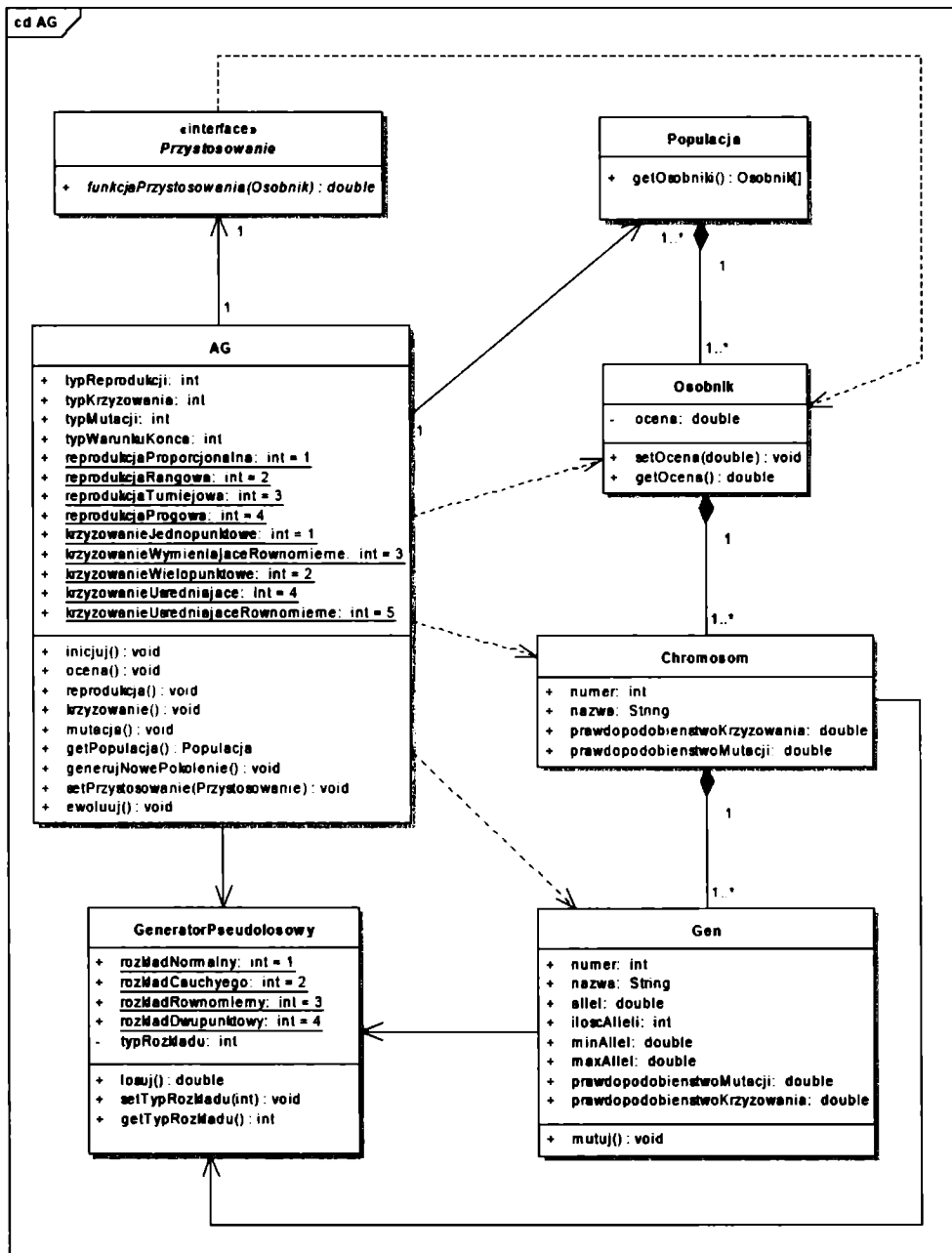
Głównym czynnikiem, od którego zależy strategia zarządzania, jest liczba użytkowników, którzy mogą brać udział w projekcie. Jeżeli tworzony jest projekt własny użytkownika, to tylko on będzie decydować o tym, kiedy ocenia obrazy, a tym samym – kiedy wygenerować następne pokolenie. Tenże użytkownik sam zadecyduje, kiedy zakończyć projekt, rezygnując z niego lub stwierdzając, że wygenerowany obraz go zadowala.

Sprawa komplikuje się, jeśli mamy do czynienia z dużą liczbą użytkowników. Tutaj trzeba zastosować specjalne mechanizmy kontrolujące moment wytworzenia nowej populacji oraz zakończenia projektu.

1. Generacja okresowa – nowa populacja tworzona jest co czas T niezależnie od tego, ile osób przyznało swoje oceny osobnikom z populacji bazowej.

2. Generacja progowa – nowa populacja tworzona jest w momencie, gdy k z n użytkowników oceni obrazy.

3. Generacja progowa z limitem czasu – jest to generacja progowa z określonym czasem na głosowanie. Jeśli minie czas T , pomimo że wymagana liczba użytkowników nie oddała głosów, tworzona jest nowa populacja.



Rys. 11. Diagram klas modułu implementacji algorytmu genetycznego

Źródło: opracowanie własne.

Alternatywne warianty zarządzania populacją. Wariant podstawowy zakłada, że w projekcie ogólnym jest jedna populacja oceniana przez wielu użytkowników. Można by jednak pokusić się o zaimplementowanie również innych ciekawych wariantów.

Wariant 1. Użytkownicy przyczyniają się do ewolucji swoich własnych, lokalnych populacji. Każdy użytkownik generuje jeden obraz, który mu najbardziej odpowiada. Z wygenerowanych obrazów tworzona jest globalna populacja. Każdy użytkownik ma jednego reprezentanta w tej populacji. Następnie populacje lokalne zastępowane są osobnikami z populacji globalnej. Powoduje to, że każdy użytkownik może wybierać jedynie z cech, które podobają się większości. Algorytm jest powtarzany. Odpowiednie operatory genetyczne mają zapobiec ponownemu osiągnięciu poprzedniego wyniku, np. krzyżowanie uśredniające działające na całym chromosomie, sukcesja elitarna itp. Implementacja powinna jednak wyglądać tak, aby na początku, gdy użytkownik po raz pierwszy generuje lokalnie twarz, nie ograniczać jego obszaru poszukiwań. Powinien to być etap eksploracji. Dopiero w kolejnych cyklach poszukiwania powinny być zawężone i odpowiadać etapowi eksploatacji.

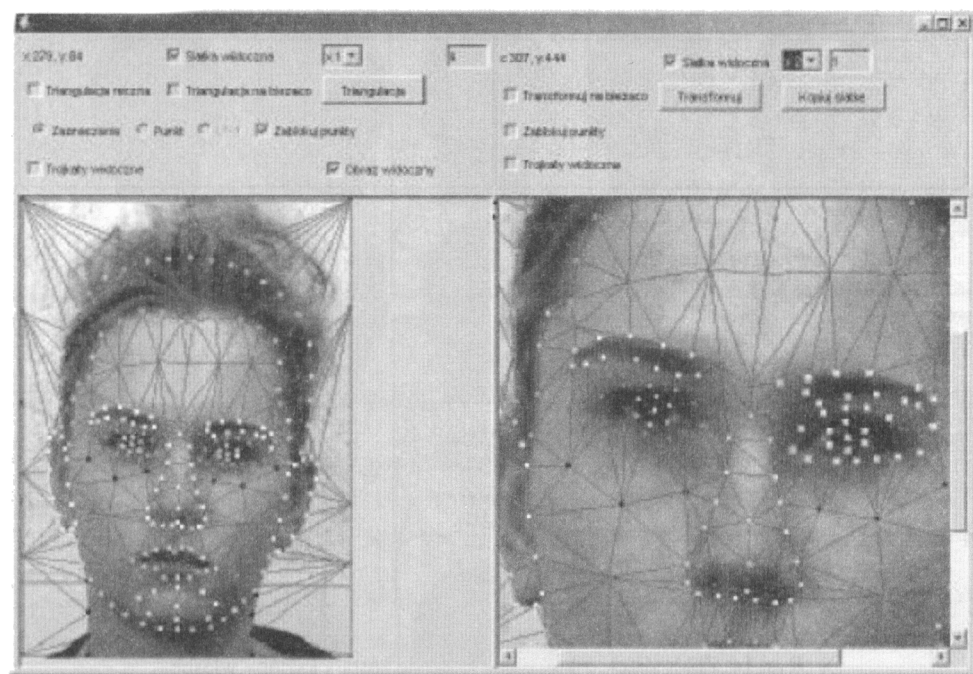
Liczba użytkowników nie może tu być duża, ponieważ na jednego użytkownika przypada jeden osobnik, a populacja powinna być mała.

Wariant 2. Użytkownicy generują najlepsze dla nich obrazy w projektach własnych. Z tych obrazów tworzony jest średni obraz.

Przygotowanie danych do systemu

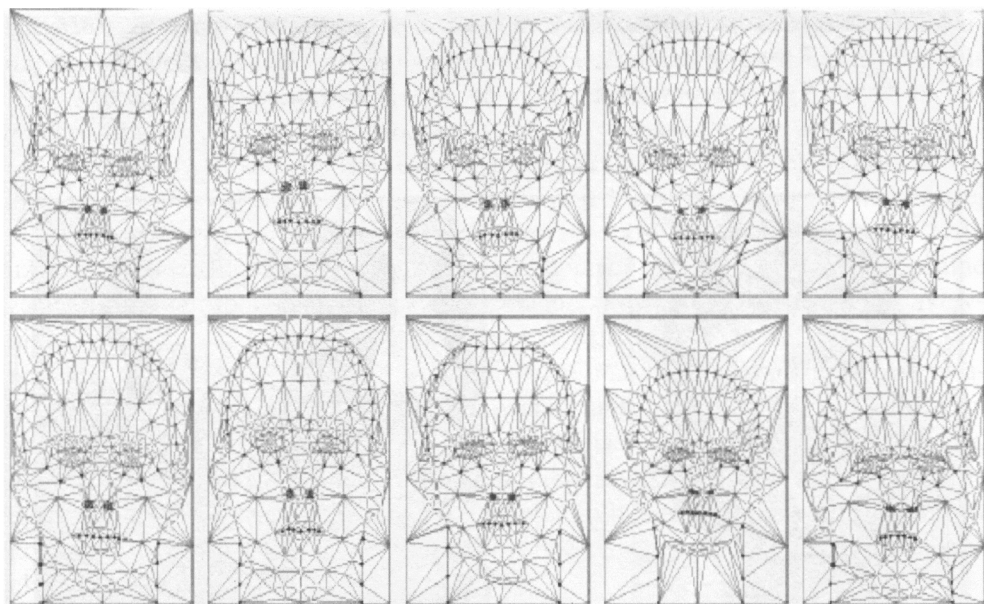
W tym punkcie pokazano, w jaki sposób można przygotować dane dla systemu, który będzie generował twarze z użyciem *eigenobjects*. Wykorzystywane tu zdjęcia twarzy pochodzą z bazy FaDab ze strony <http://icis.pcz.pl/~januszb/baza.htm>. Rysunek 12 pokazuje stworzoną w Javie aplikację, która pozwala na rysowanie siatek oraz transformowanie z ich użyciem obrazów. Do wytworzenia obiektów własnych zostało wykorzystane 10 zdjęć. Dla każdego z tych zdjęć została narysowana siatka (rys. 14). Tło z obrazów wraz z odstającymi kosmykami włosów zostało usunięte. Dla zestawu siatek wyznaczono wartość średnią (rys. 15). Każdej twarzy nadano średni kształt.

Twarz o przeciętnym kształcie i teksturze pokazuje rys. 17. Obrazy zostały poddane analizie PCA z użyciem środowiska MATLAB. Rysunek 18 pokazuje utworzone twarze własne (*eigenfaces*). Na rysunku 19 można zobaczyć, jaki wpływ na przeciętny obraz mają dodawane lub odejmowane w różnym stopniu twarze własne.



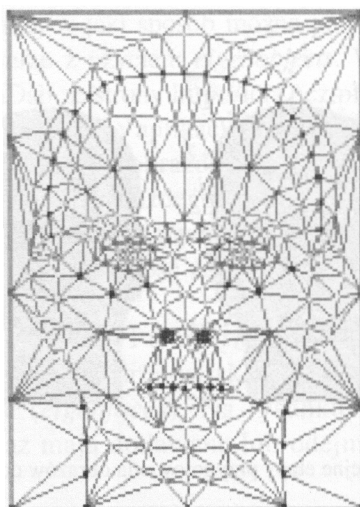
Rys. 13. Kolejne etapy przygotowania obrazów do analizy PCA

Źródło: opracowanie własne.



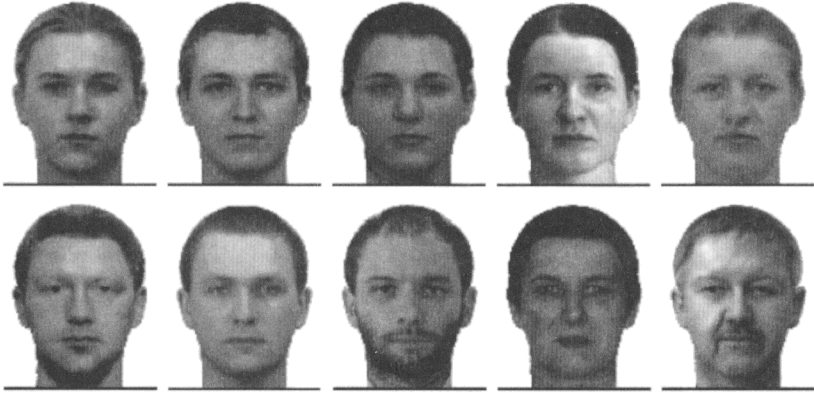
Rys. 14. Zbiór siatek, z których utworzone zostały kształty własne

Źródło: opracowanie własne.



Rys. 15. Siatka o średnim kształcie

Źródło: opracowanie własne.



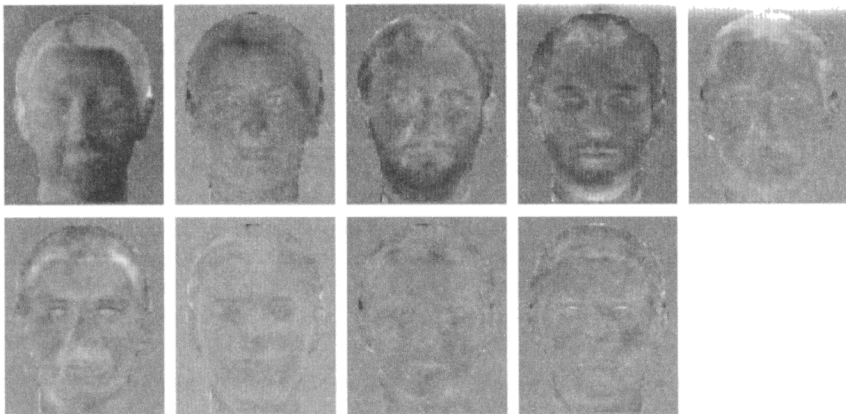
Rys. 16. Zestaw twarzy o przeciętnym kształcie, z których utworzone zostały tekstury własne

Źródło: opracowanie własne.



Rys. 17. Przeciętny obraz

Źródło: opracowanie własne.



Rys. 18. Zbiór utworzonych twarzy własnych (*eigenfaces*)

Źródło: opracowanie własne.



Rys. 19. Przykład wpływu dwóch pierwszych twarzy własnych. W środkowej kolumnie średnia tekstura, na lewo odejmowanie, na prawo dodawanie twarzy własnych

Źródło: opracowanie własne.

6. Podsumowanie

W pracy przedstawiono analizę metody generowania twarzy ludzkiej z wykorzystaniem algorytmu genetycznego. Generator działa na podstawie subiektywnych ocen użytkowników. Dostęp do systemu uzyskiwany jest przez Internet. Algorytm genetyczny generuje kolejne pokolenia osobników będących obrazem poszukiwanej twarzy. Pojedyncza twarz generowana jest w całości z wykorzystaniem analizy obrazów PCA. W odróżnieniu od tradycyjnych metod generowania twarzy z poszczególnych fragmentów opisana w pracy metoda nie wymaga utworzenia bazy danych takich fragmentów. Wykorzystywano populacje z maksymalną liczebnością 25 osobników.

Ciekawy wydaje się pomysł skrócenia liczby ocenianych obrazów przez jedną osobę, np. do dwóch. Wymagałoby to wykorzystania innej strategii ewolucyjnej. Stanowi to temat dalszych badań.

Literatura

- Arabas, J., *Wykłady z algorytmów ewolucyjnych*, Wydawnictwa Naukowo-Techniczne, Warszawa 2001.
- Draper B.A., Back K., Bartlett M.S., Beveridge J.R., *Recognizing Faces with PCA and ICA*, ComputerVision and Image Understanding 2003 nr 91, pp. 115-137.
- Frowd C.D., Hancock P.J.B., Carson D., *EvoFIT: A Holistic, Evolutionary Facial Imaging Technique for Creating Composites*, Department of Psychology, University of Stirling, UK, 2004.

- Frowd C.D., Hancock P.J.B., Carson D., *EvoFIT: A Holistic, Evolutionary Facial Imaging Technique for Creating Composites*, Department of Psychology, University of Stirling, UK, 2004.
- Goldberg D.E., *Algorytmy genetyczne i ich zastosowania*, WNT, Warszawa 2003.
- Ghanea-Hercock R., *Applied Evolutionary Algorithms in Java*, Springer-Verlag, 2003.
- Gibson S., Bejarano A.P., Solomon C., *Synthesis of Photographic Facial Composites using Evolutionary Algorithms*, School of Physical Sciences University of Kent.
http://www-psych.nmsu.edu/~vic/faceprints/female_study.html.
- Hancock P.J.B., *Evolving Faces from Principal Components*, "Behavior Research Methods, Instruments and Computers" 2000, vol. 32, nr 2, pp. 327-333.
- Hancock P.J.B., Bruce V., Burton A.M., *Testing Principal Component Representations for Faces*.
- Hancock P.J.B., Frowd C.D., *Evolutionary Generation of Faces*, In proceedings of AISB99, www.stir.ac.uk/staff/psychology/pjbh1/.
- Hyvarinen A., Oja E., *Independent Component Analysis: Algorithms and Applications*, "Neural Networks" 2000 nr 13 (4-5), s. 411-430.
- Michalewicz Z., *Algorytmy genetyczne + struktury danych = programy ewolucyjne*, Wydawnictwa Naukowo-Techniczne, Warszawa 2003.
- Movellan J.R., *Tutorial on Principal Component Analysis*, 2003.
- Shlens J., *A Tutorial on Principal Component Analysis – Derivation, Discussion and Singular Value Decomposition* 2003, www.sn1.salk.edu/~shlens/pub/notes/pca.pdf.
- Smith L.I., *A Tutorial on Principal Components Analysis* 2002, www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf.
- Turk M., Pentland A., *Eigenfaces for Recognition*, "Journal of Cognitive Neuroscience", vol. 3, nr 1, Massachusetts Institute of Technology, 1991.
- Wolan Ł., *Wykorzystanie algorytmu genetycznego do rejtingowania obiektów graficznych na stronie internetowej*, praca dyplomowa, prom. dr O. Klosow, Politechnika Wroclawska, Wydział Elektroniki, Wrocław 2005.

FACE GENERATING METHOD USING PCA AND GENETIC ALGORITHM

Summary

The human face image generated on websites by the use of multiuser environment can be applied in face recognition fields, virtual assistants or in computer game design. In the paper the proposition of using evolutionary algorithm and principal components analysis are described. During the evolutionary process users give the personal opinion about one or several faces. The method presents some advantages compared to the traditional face generating methods based on building faces from different parts of 7 to 13 types. Several strategies of users interaction are proposed.