

Metoda segmentacji odcinkowej obrazu rastrowego z wykorzystaniem strategii ewolucyjnych

Krzysztof MURAWSKI¹

STRESZCZENIE: Przedstawiono możliwości wykorzystania strategii ewolucyjnych w zadaniach segmentacji obrazu. Zaprezentowano algorytm ekstrakcji parametrów odcinków na obrazach rastrowych.

1. Wstęp

Segmentacja obrazu polega na wydzieleniu elementów składowych obrazu. Przykładem jest wydzielenie z obrazu np. linii prostych, odcinków, prostokątów. Segmentacja liniami prostymi polega na wyznaczeniu parametrów linii prostych znajdujących się w obrazie. Dobrze znane są metody ich poszukiwania np. za pomocą transformaty Hougha [23, 26 – 28], metody uczenia konkurencyjnego [25].

Podstawowymi problemami segmentacji obrazu liniami prostymi są: 1) dążenie do uzyskiwania możliwie dużej dokładności wyników; 2) redukcja zwykle dużej złożoności obliczeniowej w celu skrócenia czasu wykonywania segmentacji obrazu. Problem dokładności obliczeń oraz spożytkowania wyników transformacji Hougha przedstawiono w [23].

Główną przeszkodą w stosowaniu przekształcenia Hougha jest zależność jego efektywności od rozmiaru analizowanego obrazu. Bezpośrednie obliczenie transformaty Hougha jest efektywne jedynie do analizy obrazów o niewielkich wymiarach (o małej rozdzielczości). Uzyskiwana tą drogą dokładność wyznaczania parametrów rozpoznawanych obiektów (np. linii prostych) jest silnie zależna od wielkości obrazu

¹Zakład Systemów Komputerowych, Instytut Automatyki i Robotyki WAT, ul. Kaliskiego 2,
00 – 908 Warszawa.

i czasu obliczeń. Mała dokładność wyników może prowadzić do błędnego rozpoznania obiektów w systemie widzenia komputerowego. Szybka realizacja transformaty Hougha (szczególnie, gdy rozważana jest duża liczba parametrów) wymaga znacznej mocy obliczeniowej. To wymaganie często dyskwalifikuje transformatę Hougha z wielu zastosowań.

Rozwiązanie problemu segmentacji obrazu przy pomocy sieci neuronowych także jest problematyczne. Sekwencyjne poszukiwanie obiektów na analizowanym obrazie nie zapewnia wymaganej szybkości działania. Przedstawiona w [24, 25] propozycja wykorzystania sztucznych sieci neuronowych do segmentacji obrazów rastrowych również ujawnia trudności z ich zastosowaniem. Prezentowane w [25] wykrywanie parametrów linii prostych na obrazach rastrowych jakkolwiek skuteczne, wymaga sekwencyjnego poszukiwania tychże parametrów.

Alternatywnym podejściem do problemu segmentacji obrazów rastrowych jest zastosowanie strategii ewolucyjnych. Wykorzystanie własności strategii ewolucyjnych do rozwiązania problemu segmentacji obrazów rastrowych wydaje się być obiecującym. Próby zastosowania strategii ewolucyjnych do segmentacji obrazów rastrowych są uzasadnione następującymi oczekiwaniami:

1. oczekiwanie 1 – sposób poszukiwania ekstremum jest podobny do poszukiwania losowego, co powinno zapobiec zjawisku uzyskiwania pojedynczych rozwiązań lokalnych;
2. oczekiwanie 2 – nie występują problemy obliczeniowe związane z wyznaczaniem wielu ekstremów lokalnych, istotne np. przy wykorzystywaniu transformaty Hougha [25];
3. oczekiwanie 3 – szybszy – w porównaniu z poszukiwaniem losowym – proces uzyskiwania rozwiązań dzięki mechanizmowi dziedziczenia cech wartościowych;
4. oczekiwanie 4 – możliwość uzyskania dużej dokładności rezultatów;

2. Strategie ewolucyjne

Strategie ewolucyjne oraz algorytmy genetyczne to jedne z głównych kierunków rozwoju symulacji ewolucji. Różnice i podobieństwa pomiędzy wymienionymi kierunkami przedstawiono w tabeli 1.

Tabela 1. Strategie ewolucyjne, algorytmy genetyczne - różnice i podobieństwa

Algorytmy genetyczne	Strategie ewolucyjne
Cechy wspólne algorytmów genetycznych i strategii ewolucyjnych:	
<ol style="list-style-type: none"> 1. Operują na populacjach potencjalnych rozwiązań. 2. Korzystają z zasady selekcji i przetwarzania osobników najlepiej przystosowanych. 	
Osobnik:	Osobnik:
Wektor binarny	Wektor liczb zmiennoprzecinkowych
Proces selekcji:	Proces selekcji:
Populacja pośrednia tworzona jest ze wszystkich rodziców i pewnej liczby potomków. Selekcja polega na wyborze takiej samej liczby osobników jak liczebność populacji rodzicielskiej. Szanse wyboru zależą od przystosowania osobników.	Populacja pośrednia tworzona jest z populacji rodziców i potomków. Selekcja polega na redukcji rozmiaru populacji pośredniej do rozmiaru populacji rodzicielskiej przez usunięcie osobników najgorzej przystosowanych.
Procedura selekcji:	Procedura selekcji:
Losowa	Deterministyczna
Wynik metody selekcji:	Wynik metody selekcji:
Lepiej przystosowane osobniki mogą być wybrane kilkakrotnie. W populacji wynikowej mogą pojawić się osobniki najslabsze.	Wybrane są osobniki najlepiej przystosowane. Osobniki wybierane są bez powtórzeń.
Kolejność procedur selekcji i rekombinacji:	Kolejność procedur selekcji i rekombinacji:
Najpierw dokonywana jest selekcja genotypów a potem ich rekombinacja.	Najpierw dokonywana jest rekombinacja genotypów a potem ich selekcja.
Zmienność parametrów operatorów genetycznych:	Zmienność parametrów operatorów genetycznych:
Prawdopodobieństwo mutacji i krzyżowania pozostaje stałe w trakcie całego procesu ewolucji.	Prawdopodobieństwo mutacji i krzyżowania ulega ciągłej zmianie

Strategie ewolucyjne reprezentują nową ideę rozwiązywania zadań optymalizacyjnych. Podstawą ich działania jest naśladowanie, symulacja, naturalnych

procesów ewolucji. Fundamentalna prawidłowość, polegająca na tym, że w procesie ewolucji przeżywają osobniki najlepiej przystosowane, zostaje wykorzystana do selekcji najlepszych rozwiązań w procesie poszukiwania optimum.

Pojęcia stosowane w genetyce komputerowej są w większości odpowiednikami funkcjonujących w genetyce naturalnej. Wyróżnić tu należy następujące z nich:

- ◆ gen – pojedynczy element chromosomu;
- ◆ chromosom – łańcuch lub ciąg kodowy, uporządkowany ciąg genów;
- ◆ genotyp – struktura złożona z chromosomów właściwa dla danego osobnika;
- ◆ osobnik – zakodowany w postaci genotypu zbiór parametrów zadania;
- ◆ populacja – zbiór osobników (o określonej liczebności);
- ◆ allel – wartość danego genu;
- ◆ locus – miejsce położenia genu w genotypie (w szczególnym przypadku w chromosomie);
- ◆ fenotyp – zespół wartości odpowiadający danemu genotypowi.

Do ważnych pojęć stosowanych w genetyce komputerowej należy również zaliczyć, nie określoną w formalny sposób w genetyce naturalnej, *funkcję przystosowania (oceny)*. Funkcja ta pozwala ocenić stopień adaptacji poszczególnych osobników. Wyznacza wartość wskaźnika stanowiącego podstawę wyboru osobników najbardziej wartościowych. Przyjmuje się, że osobniki najbardziej wartościowe generują największe wartości funkcji przystosowania. Wartość funkcji przystosowania jest maksymalizowana w trakcie całego procesu ewolucji.

3. Sformułowanie problemu

Segmentacja obrazów rastrowych odgrywa ważną rolę w systemie widzenia komputerowego. Zadaniem segmentacji obrazów jest wydzielenie z analizowanego kadru jego podstawowych składowych (segmentów), najczęściej prymitywów graficznych (odcinków, prostych, okręgów, prostokątów). Ekstrahowane segmenty są podstawą do analizy treści obrazu. W niniejszym artykule problem segmentacji rozważono na przykładzie wyodrębniania odcinków w analizowanym obrazie.

W dalszej części artykułu rozważono segmentację obrazów rastrowych wyrażonych funkcją:

$$(1) \quad \mathbf{B} : \mathbf{D} \rightarrow \mathbf{C}$$

gdzie:

$$(2) \quad \mathbf{D} = [1, N] \times [1, M] \subset \mathbf{N}^2$$

\mathbf{C} – zbiór kolorów,

$[1, N]$ – przedział w zbiorze liczb naturalnych,

N – liczba wierszy obrazu,

M – liczba kolumn obrazu.

Przyjmuje się, że analizowany jest obraz konturowy wytworzony z wejściowego obrazu rastrowego. Segmentacja odcinkowa obrazu rastrowego polega na wyodrębnieniu odcinków. Każdy odcinek jest opisany przez jego dwa punkty końcowe $P_1, P_2 \in \mathbf{R}^2$. Tak np. odcinek $s(P_1, P_2)$ o końcach $P_1, P_2 \in \mathbf{R}^2$ jest rozumiany następująco:

$$(3) \quad s(P_1, P_2) = \left\{ x \in \mathbf{R}^2 : \bigvee_{t \in [0,1]} x = t \cdot P_1 + (1-t) \cdot P_2 \right\}$$

Rozwiązanie zadania polega na:

- wykryciu na odrzynie odcinków o wyróżnionym kolorze „c”,
- wyznaczeniu parametrów P_1, P_2 tych odcinków,

Proponowana w artykule metoda rozwiązania problemu oparta jest na wykorzystaniu właściwości strategii ewolucyjnych. Wyznaczane parametry odcinków: P_1, P_2 są kodowane, a następnie przetwarzane w postaci chromosomów. Zgodność położenia wyznaczonego odcinka w stosunku do odcinka znajdującego się na analizowanym obrazie określa wartość funkcji przystosowania $F(P_1, P_2)$.

Wyróżniono dwie metody obliczania funkcji przystosowania wygenerowanego odcinka do wzorca:

- a. Metoda 1 – obliczanie funkcji przystosowania polega na zliczaniu tych pikseli o wyróżnionym kolorze, które leżą na odcinku $s(P_1, P_2)$. Wyznaczenie liczebności zbioru pikseli będącego częścią wspólną wygenerowanego odcinka oraz punktów obrazu można wyrazić wzorem:

$$(4) \quad F(P_1, P_2) = \| b(s_\varepsilon(P_1, P_2)) \cap \mathbf{B}^{-1}(c) \|$$

przy czym:

- funkcja b dla danego podzbioru \mathbf{A} płaszczyzny \mathbf{R}^2 ustala zbiór $b(\mathbf{A}) \subset \mathbf{D}$ pikseli reprezentujących zbiór \mathbf{A} na obrazie;

$$(5) \quad b(\mathbf{A}) = \{p \in \mathbf{D} : d(p) \in \mathbf{A}\}, \quad \mathbf{A} \subset \mathbf{R}^2$$

- $s_\varepsilon(P_1, P_2)$ oznacza zbiór punktów płaszczyzny położonych w odległości nie większej niż ε od odcinka $s(P_1, P_2)$;

$$(6) \quad s_\varepsilon(P_1, P_2) = \left\{ x \in \mathbf{R}^2 : \bigvee_{y \in s(P_1, P_2)} \|x - y\| \leq \varepsilon \right\}$$

- zbiór $\mathbf{B}^{-1}(c)$ składa się ze wszystkich punktów obrazu o kolorze c

$$(7) \quad \mathbf{B}^{-1}(c) = \{p \in \mathbf{D} : \mathbf{B}(p) = c\}$$

- wartość funkcji $d(p)$ określa współrzędne piksela $p = (w, k)$ na płaszczyźnie

$$(8) \quad d(p) = (w \cdot \sigma_1, k \cdot \sigma_2) \in \mathbf{R}^2$$

gdzie:

σ_1, σ_2 – współczynniki skali względem współrzędnych w, k

$\| \mathbf{B} \|$ oznacza liczebność zbioru $\mathbf{B} \subset \mathbf{D}$

- b. Metoda 2 – obliczanie funkcji przystosowania polega na wyliczeniu stopnia pokrycia wygenerowanego odcinka $s(P_1, P_2)$ pikselami o wyróżnionym kolorze. Wyznaczenie stopnia pokrycia wygenerowanego odcinka pikselami o wyróżnionym kolorze c można wyrazić wzorem:

$$(9) \quad F(P_1, P_2) = \frac{L_c(P_1, P_2)}{L_s(P_1, P_2)} \cdot 100\%$$

gdzie:

- $L_c(P_1, P_2)$ liczebność zbioru pikseli będącego częścią wspólną wygenerowanego odcinka oraz wyróżnionych kolorem c punktów obrazu:

$$(10) \quad L_c(P_1, P_2) = \| b(s_\varepsilon(P_1, P_2)) \cap \mathbf{B}^{-1}(c) \|$$

– $L_s(P_1, P_2)$ liczebność zbioru pikseli należących do odcinka $s(P_1, P_2)$:

$$(11) \quad L_s(P_1, P_2) = \| b(s_\varepsilon(P_1, P_2)) \|$$

4. Metoda estymacji parametrów odcinków

Idea procesu poszukiwania rozwiązania jest następująca. Na wstępie generowanych jest losowo μ odcinków. Parametry wygenerowanych odcinków są następnie kodowane w chromosomach. Wygenerowane w ten sposób chromosomy tworzą tzw. populację rodzicielską. Populacja rodzicielska stanowi bazę do wygenerowania populacji pośredniej (nowych odcinków) – potomstwa (offspring). Populacja dzieci uzyskiwana jest w wyniku działania operatorów genetycznych (np. mutacji, krzyżowania, inwersji) na populację rodziców. Z wytworzonej populacji potomstwa – zwykle ilościowo większej od zainicjowanej pierwotnie populacji – uzyskiwana jest kolejna populacja rodziców (kolejne odcinki). W każdym etapie algorytmu wszystkie wygenerowane odcinki rywalizują ze sobą dążąc do uzyskania maksymalnej wartości funkcji przystosowania. W rezultacie działania strategii ewolucyjnych w kolejnych etapach otrzymujemy chromosomy bardziej przystosowane. Otrzymywane rezultaty zależą od przyjętej metody kontroli populacji genetycznej [2 – 4, 6, 17, 20, 21]. Proces poszukiwania segmentów odcinkowych oprzemy na założeniu, że końce odcinka są reprezentowane przez wyróżnione, tzn. o kolorze „c”, piksele obrazu. Niech W_n, W_m oznaczają wybrane losowo ze zbioru \mathbf{D} punkty krańcowe odcinka, takie że $W_n \neq W_m$. Niech $P_n = d(W_n)$, $P_m = d(W_m)$. Wtedy $\bar{F}(W_n, W_m) = F(d(W_n), d(W_m))$ oznacza wartość funkcji przystosowania odcinka o końcach $W_n, W_m \in \mathbf{D}$. Modyfikację parametrów P_n, P_m dokonuje się w ten sposób, by w kolejnych iteracjach algorytmu segmentacji obrazu maksymalizować wartość funkcji przystosowania.

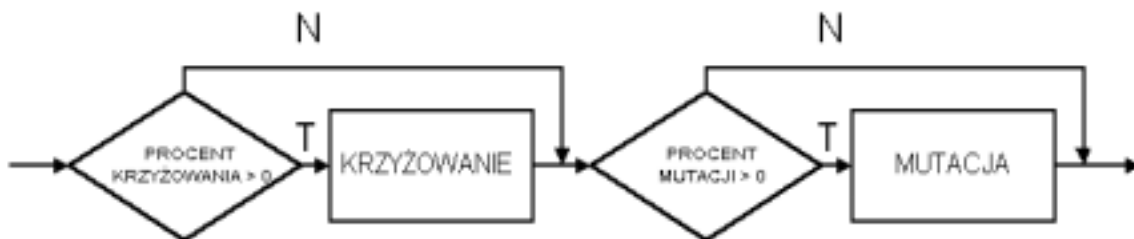
Proponowane rozwiązanie polega na:

1. losowym wyborze μ rodziców (odcinków określanych punktami $W_n, W_m \in \mathbf{D}$), gdzie μ określa wielkość populacji rodzicielskiej;

2. obliczeniu funkcji przystosowania $\bar{F}(W_n, W_m) = F(d(W_n), d(W_m))$ dla μ osobników populacji rodzicielskiej;
3. wytworzeniu za pomocą operatorów genetycznych λ dzieci dla każdego z μ rodziców;
4. obliczeniu funkcji przystosowania $\bar{F}(W_n, W_m)$ dla osobników wytworzonej populacji genetycznej;
5. redukcji rozmiaru wytworzonej populacji, $(\mu + \mu * \lambda)$, do rozmiaru populacji rodzicielskiej μ zgodnie z przyjętą metodą kontroli populacji genetycznej $(\mu + \lambda)$ - wytworzenie nowej populacji rodzicielskiej;
6. sprawdzeniu czy otrzymane rezultaty należą do zadowalających, bądź czy nie została wyczerpana założona liczba N iteracji.

Obliczana w punkcie drugim algorytmu funkcja przystosowania pokazuje stopień atrakcyjności populacji początkowej. Większe wartości funkcji pozwalają prognozować szybsze otrzymanie akceptowalnych wyników.

W punkcie trzecim algorytmu tworzona jest populacja potomków (dzieci). Sposób tworzenia dzieci zależy od wartości progów: mutacji i krzyżowania (dopuszcza się także adaptację tych parametrów podczas działania algorytmu). Wpływ stopnia mutacji oraz krzyżowania na tworzenie nowego potomka przedstawia rys. 1.



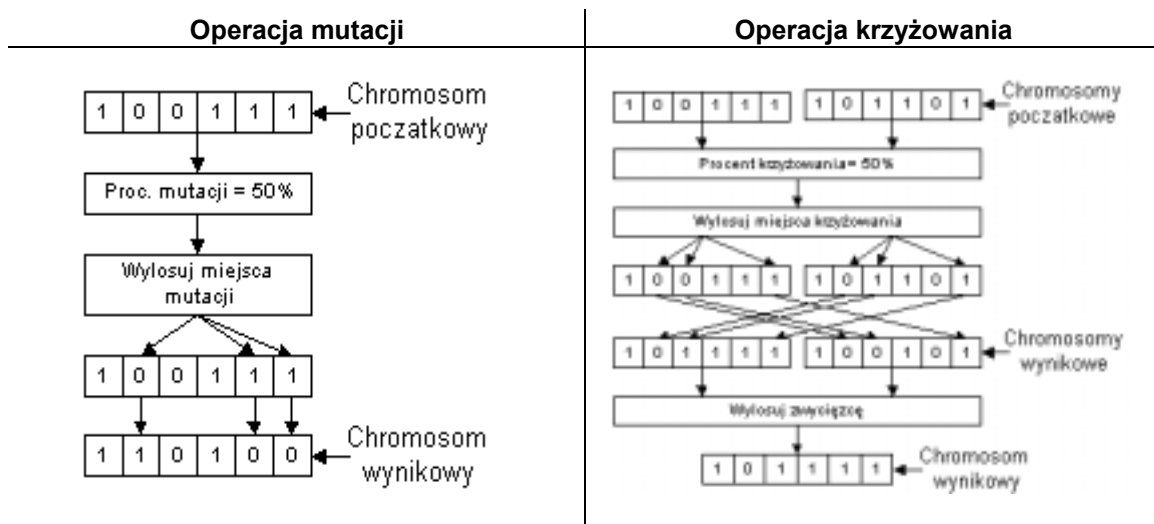
Rys. 1. Wpływ stopnia mutacji oraz krzyżowania na tworzenie nowego potomka

Ideę działania operatorów mutacji i krzyżowania przy binarnym kodowaniu chromosomu (właściwym dla algorytmów genetycznych) przedstawia rys. 2.

Po obliczeniu w punkcie czwartym dla nowo wytworzonych osobników wartości funkcji przystosowania, następuje modyfikacja populacji rodzicielskiej – punkt piąty.

Czynność ta przy założonej wcześniej metodzie kontroli populacji, określanej umownie $(\mu + \lambda)$, dokonywana jest według następującego algorytmu:

1. zbuduj populację przejściową z połączenia populacji: rodzicielskiej i wytworzonej populacji dzieci,
2. posortuj populację przejściową:
 - malejąco – w przypadku poszukiwania maksimum,
 - rosnąco – w przypadku poszukiwania minimum.
3. przyjmij za populację rodzicielską μ pierwszych osobników populacji przejściowej,
4. usuń populację przejściową.



Rys. 2. Idea działania operatorów mutacji i krzyżowania

5. Segmentacja odcinkami obrazów rastrowych

Pokażemy, przykłady segmentacji obrazu oparte na strategiach ewolucyjnych.

5.1. Segmentacja obrazów zawierających tylko odcinki

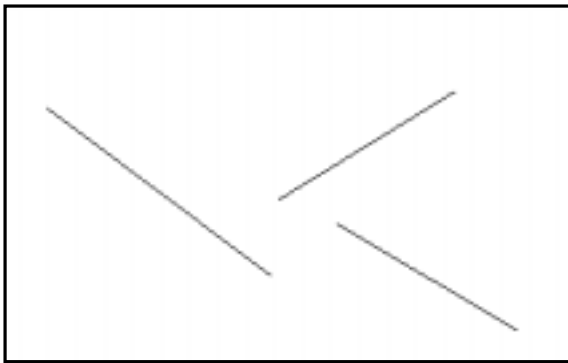
Zastosowano następujące wartości parametrów zaproponowanego algorytmu segmentacji obrazu:

- liczba poszukiwanych obiektów: 3;
- liczba generacji: 200;
- licznosc populacji rodzicielskiej: 5;

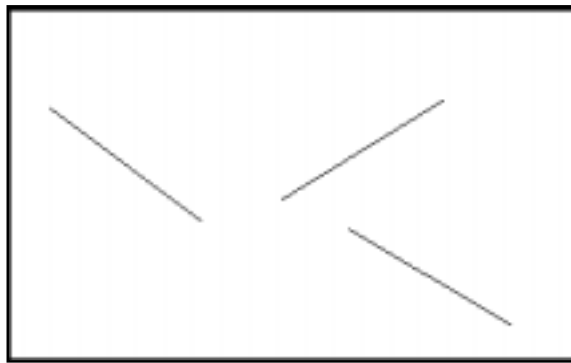
- liczba potomków przypadających na jednego rodzica: 10;
- rodzaje operatorów genetycznych: mutacja;
- wielkość progu mutacji: generowana losowo z zbioru {0%, 50%, 100%};
- metoda kontroli populacji genetycznej: $(\mu + \lambda)$;
- metoda obliczania funkcji przystosowania: adaptacyjna – pkt. 3 – metoda 2;
- minimalna długość generowanych odcinków: 50 pikseli;
- maksymalna długość generowanych odcinków: losowa, większa od 50 pikseli;
- minimalna zgodność dla odcinków uznawanych za akceptowalne: 50%.

Rys. 3a przedstawia obraz poddawany segmentacji. W wyniku działania algorytmu oszacowano parametry poszukiwanych odcinków, tabela 2. Graficzną reprezentację wyników przedstawia rys. 3b.

3a



3b



Rys. 3. Obraz poddawany i uzyskany po segmentacji obrazu

Tabela 2. Parametry odcinków uzyskane w wyniku segmentacji

Lp.	P_1		P_2		Fitness value
	X	Y	X	Y	
1.	23	143	109	78	100.00
2.	192	74	284	19	100.00
3.	154	91	246	148	100.00

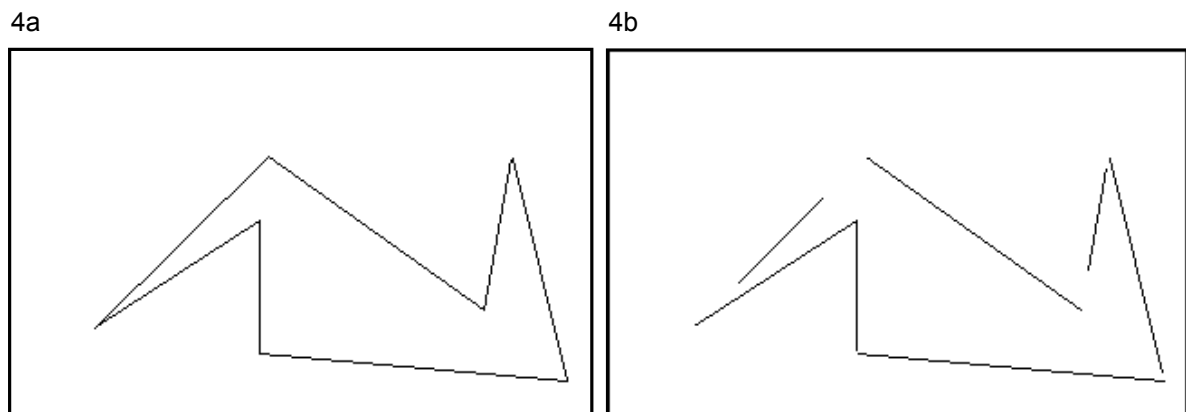
5.2. Segmentacja obrazów konturowych – złożonych z połączonych ze sobą odcinków

Eksperyment 1

Zastosowano następujące wartości parametrów zaproponowanego algorytmu segmentacji obrazu:

- liczba poszukiwanych obiektów: 7;
- liczba generacji: 200;
- licznosc populacji rodzicielskiej: 5;
- liczba potomków przypadających na jednego rodzica: 10;
- rodzaje operatorów genetycznych: mutacja;
- wielkość progu mutacji: generowana losowo ze zbioru {0%, 50%, 100%};
- metoda kontroli populacji genetycznej: $(\mu + \lambda)$;
- metoda obliczania funkcji przystosowania: adaptacyjna – pkt. 3 – metoda 2;
- minimalna długość generowanych odcinków: 50 pikseli;
- maksymalna długość generowanych odcinków: 200 pikseli;
- minimalna zgodność dla odcinków uznawanych za akceptowalne: 50%.

Obraz wejściowy (4a) i uzyskany po segmentacji (4b) przedstawia rys. 4. Wynik działania algorytmu zawiera tabela 3.



Rys. 4. Obraz poddawany i uzyskany po segmentacji obrazu

Tabela 3. Parametry odcinków uzyskane podczas segmentacji

Lp.	P_1		P_2		Fitness value
	X	Y	X	Y	
1.	307	18	137	33	100,00
2.	261	57	142	141	100,00
3.	118	118	71	71	100,00
4.	277	140	306	22	100,00
5.	137	106	47	48	100,00
6.	265	79	275	135	98,25
7.	137	105	137	34	100,00

Eksperyment 2

Zastosowano następujące wartości parametrów zaproponowanego algorytmu segmentacji obrazu:

- liczba poszukiwanych obiektów: 10;
- liczba generacji: 200;
- licznosc populacji rodzicielskiej: 5;
- liczba potomków przypadających na jednego rodzica: 10;
- rodzaje operatorów genetycznych: mutacja;
- wielkość progu mutacji: generowana losowo ze zbioru {0%, 50%, 100%};
- metoda kontroli populacji genetycznej: $(\mu + \lambda)$;
- metoda obliczania funkcji przystosowania: adaptacyjna – pkt. 3 – metoda 2;
- minimalna długość generowanych odcinków: 50 pikseli;
- maksymalna długość generowanych odcinków: 200 pikseli;
- minimalna zgodność dla odcinków uznawanych za akceptowalne: 50%.

Obraz wejściowy (5a) i uzyskany po segmentacji (5b) przedstawia rys. 5.

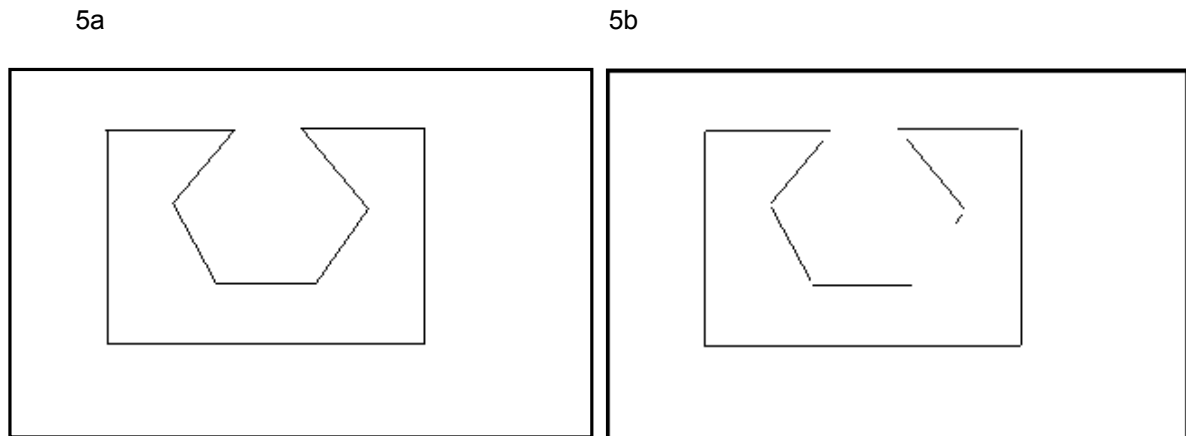
Eksperyment 3

Zastosowano następujące wartości parametrów zaproponowanego algorytmu segmentacji obrazu:

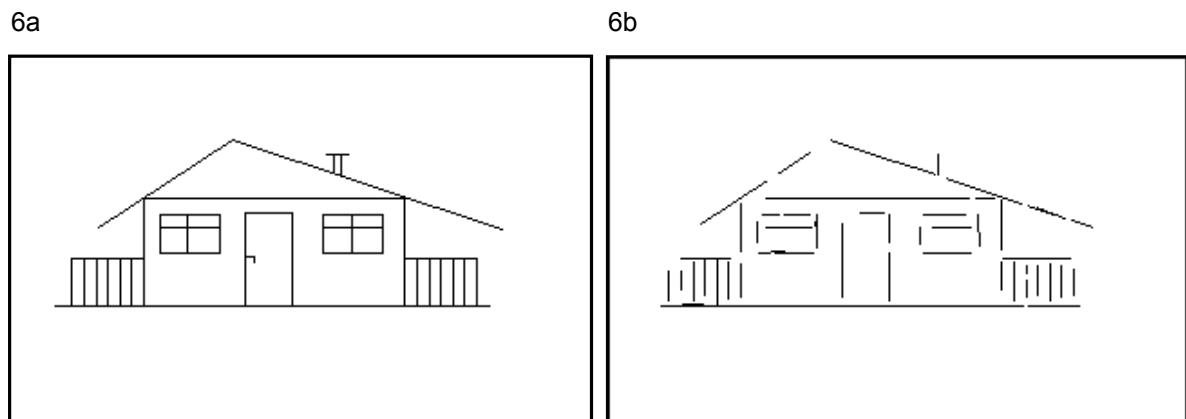
- liczba poszukiwanych obiektów: 60;

- liczba generacji: 5;
- licznosc populacji rodzicielskiej: 5;
- liczba potomków przypadających na jednego rodzica: 10;
- rodzaje operatorów genetycznych: mutacja;
- wielkość progu mutacji: generowana losowo ze zbioru {0%, 50%, 100%};
- metoda kontroli populacji genetycznej: $(\mu + \lambda)$;
- metoda obliczania funkcji przystosowania: adaptacyjna – pkt. 3 – metoda 2;
- minimalna długość generowanych odcinków: 10 pikseli;
- maksymalna długość generowanych odcinków: 200 pikseli;
- minimalna zgodność dla odcinków uznawanych za akceptowalne: 50%.

Obraz wejściowy (6a) i uzyskany po segmentacji (6b) przedstawia rys. 6.



Rys. 5. Obraz wejściowy i uzyskany po segmentacji obrazu



Rys. 6. Obraz poddawany i uzyskany po segmentacji obrazu

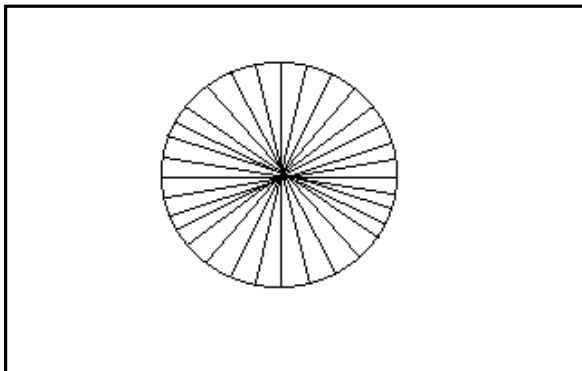
Eksperyment 4

Zastosowano następujące wartości parametrów zaproponowanego algorytmu segmentacji obrazu:

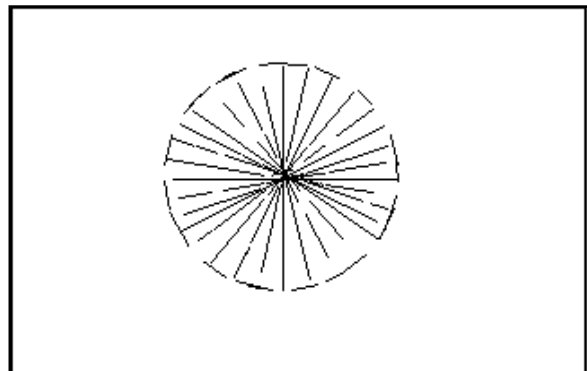
- liczba poszukiwanych obiektów: 100;
- liczba generacji: 20;
- licznosc populacji rodzicielskiej: 5;
- liczba potomków przypadających na jednego rodzica: 10;
- rodzaje operatorów genetycznych: mutacja;
- wielkość progu mutacji: generowana losowo ze zbioru {0%, 50%, 100%};
- metoda kontroli populacji genetycznej: $(\mu + \lambda)$;
- metoda obliczania funkcji przystosowania: adaptacyjna – pkt. 3 – metoda 2;
- minimalna długość generowanych odcinków: 2 piksele;
- maksymalna długość generowanych odcinków: 200 pikseli;
- minimalna zgodność dla odcinków uznawanych za akceptowalne: 50%.

Obraz wejściowy (7a) i uzyskany po segmentacji (7b) przedstawia rys. 7.

7a



7b



Rys. 7. Obraz poddawany i uzyskany po segmentacji obrazu

5.3. Segmentacja obrazów rzeczywistych

Zastosowano następujące wartości parametrów zaproponowanego algorytmu segmentacji obrazu:

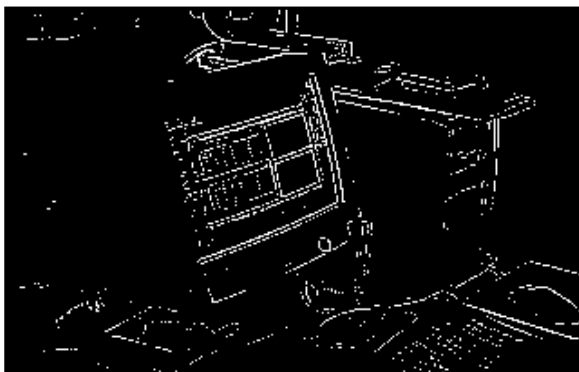
- liczba poszukiwanych obiektów: 300;
- liczba generacji: 200;

- liczność populacji rodzicielskiej: 5;
- liczba potomków przypadających na jednego rodzica: 10;
- rodzaje operatorów genetycznych: mutacja;
- wielkość progu mutacji: generowana losowo ze zbioru {0%, 50%, 100%};
- metoda kontroli populacji genetycznej: $(\mu + \lambda)$;
- metoda obliczania funkcji przystosowania: adaptacyjna – pkt. 3 – metoda 2;
- minimalna długość generowanych odcinków: 2 piksele;
- maksymalna długość generowanych odcinków: 200 pikseli;
- minimalna zgodność dla odcinków uznawanych za akceptowalne: 50%.

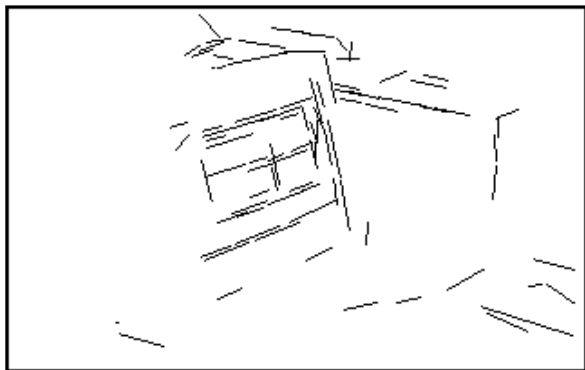
8a



8b



8c



8d



Rys. 8. Segmentacja obrazu pozyskiwanego z kamery

Rys. 8a przedstawia obraz pozyskany z kamery. Obraz ten poddany został wstępnej obróbce obrazu. Do wstępnej obróbki obrazu zaliczono:

- zmianę reprezentacji kolorów z RGB do Gray Scale,
- zastosowanie filtracji obrazu wykrywającej kontury – filtracji Sobela,
- zastosowanie mechanizmu progowania (binaryzacji).

Uzyskany po obróbce wstępnej obraz wejściowy 8b poddano segmentacji. Graficzną reprezentację uzyskanych wyników przedstawia rys. 8c. Ostateczne złożenie wydzielonych segmentów z obrazem wejściowym (o zmniejszonej jasności w celu uwydatnienia wyznaczonych odcinków) przedstawia rys. 8d.

6. Uwagi końcowe

1. Uzyskane podczas eksperymentów rezultaty są w większości zgodne z analizowanymi obrazami. W niektórych przypadkach proponowana metoda nie znalazła dokładnie wszystkich zawartych w obrazie odcinków. Procent nie wykrytych odcinków jest jednak znikomo mały. Nie wpływa na pozytywną ocenę przydatności opracowanej metody do segmentacji obrazów. Niewykrycie wszystkich zawartych w obrazie odcinków jest konsekwencją ograniczenia czasu działania algorytmu segmentacji obrazu rastrowego.
2. Segmentacja odcinkowa w porównaniu z segmentacją liniami prostymi umożliwia uzyskanie wierniejszego opisu wyróżnianych segmentów. W przypadku segmentacji liniami prostymi często występuje konieczność wyznaczania odcinków drogą porównywania wyznaczonych linii prostych z obrazem.

Literatura

- [1] Arbushi V., Cantoni V., Musso G.: *Recognition and Location of Mechanical Parts Using Hough Transform*. W: Levialdi (ed.): *Image Analysis*. Pitman, London, 1983.
- [2] Agapie A.: *Genetic algorithms: Theory and applications*. Journal of Computing Anticipatory Systems CHAOS ASBL, 2000.
- [3] Banzhaf W., Daida J., Eiben E. A., Garzon H. M., Honavar V., Jakiela M., and Smith E. R. (ed.): *Proc. Genetic and Evolutionary Computation Conf. (GECCO'99)*, Orlando FL, July 13-17, 1999. Morgan Kaufmann, San Francisco CA.
- [4] Beyer Hans – Georg.: *Some aspects of the "evolution strategy" for solving tsp-like optimization problems appearing at the design studies of a 0.5teV - linear collider*. In R. Männer and B. Manderick (ed.): *Parallel Problem Solving from Nature - Proc. Second Conf. PPSN*, pages 361-370, Free University of Brussels, September 28-30, 1992. Elsevier, Amsterdam.
- [5] Bäck Th.: *Optimization by means of genetic algorithms*. In Köhler E. (ed.): *36th Int'l Scientific Colloquium*, pages 163-169, Technical University of Ilmenau, October 21-24, 1991.
- [6] Bäck Th., Fogel D. B., and Michalewicz Z. (ed.): *Handbook of Evolutionary Computation*. Oxford University Press, New York, and Institute of Physics Publishing, Bristol, 1997.

- [7] Bäck Th.: *Adaptive learning by evolutionary algorithms within search spaces of different complexity*. In Bock P., Loew M., Radermacher F. J., and Richter M. M. (ed.): *Proc. Third Int'l FAW Workshop Adaptive Learning and Neural Networks*, pages 1-17, Schloss Reisingburg, Günzburg, July 2-7, 1991. Research Institute for Applied Knowledge Engineering (FAW), Ulm, 1992.
- [8] Chakraborty S., De S., and Deb K.: *Model-based object recognition from a complex binary imagery using genetic algorithm*. In Poli R., Voigt H. M., Cagnoni S., Corne D., Smith G. D., and Fogarty T. C. (ed.): *Evolutionary Image Analysis, Signal Processing, and Telecommunications (EvoIASP'99 and EuroEcTel'99) Proc.*, volume 1596 of Lecture Notes in Computer Science, pages 150-161, Gothenburg, Sweden, May 26-27, 1999. Springer, Berlin.
- [9] Claus V., Hopf J., and Schwefel H. P. (ed.): *Evolutionary Algorithms and their Application*, Dagstuhl-Seminar-Report No. 140, March 25-29, 1996. IBFI GmbH, Schloss Dagstuhl, Wadern.
- [10] Davidor Y., and Schwefel H. P.: *An introduction to adaptive optimization algorithms based on principles of natural evolution*. In B. Souek and The IRIS Group (ed.): *Dynamic, Genetic, and Chaotic Programming*, pages 183-202. Wiley Interscience, New York, 1992.
- [11] Davis L. S.: *Hierarchical Generalized Hough Transform and Line B Segment Based Generalized Hough Transforms*. *Pattern Recognition*, 15, 1982, pp. 277-285.
- [12] Deb K.: *Evolutionary algorithms for multi – criterion optimization in engineering design*. In K. Miettinen, M. M. Mäkelä, P. Neittaanmäki, and J. Periaux (ed.): *Evolutionary Algorithms in Engineering and Computer Science*, pages 135-161. Wiley, Chichester, 1999.
- [13] Deb K.: *An introduction to genetic algorithms*. *Sadhan a Journal*, 24(4 & 5):293-315, 1999.
- [14] Dirk V. A.: *Evolution strategies in noisy environments - A survey of existing work*. In Proc. Second EvoNet Summer School on Theoretical Aspects of Evolutionary Computing, Antwerp, September 1-7, 1999.
- [15] Droste S., Wiesmann D.: *On representation and genetic operators in evolutionary algorithms*. Technical Report of the Collaborative Research Center 531 "Computational Intelligence" CI-41/98, University of Dortmund, July 1998.
- [16] Eiben E. A., Bäck Th.: *An empirical investigation of multiparent recombination operators in evolution strategies*. *Evolutionary Computation*, 5(3):347-365, 1997.
- [17] Eiben E. A., Rudolph G.: *Special Issue on the Theory of Evolutionary Computation - Preface*. *Theoretical Computer Science*, 229:1-2, 1999.
- [18] Fogel J. L., Angeline J. P., and Bäck Th. (ed.): *Evolutionary Programming V – Proc. Fifth Annual Conf. Evolutionary Programming (EP'96)*, San Diego CA, February 29 - March 3, 1996. The MIT Press, Cambridge MA.
- [19] Graf J., Banzhaf W.: *Interactive evolution of images*. In J. R. McDonnell, R. G. Reynolds, and D. B. Fogel (ed.): *Evolutionary Programming IV - Proc. Fourth Annual Conf. Evolutionary Programming (EP'95)*, pages 53-65, San Diego CA, March 1-4, 1995. The MIT Press, Cambridge MA.
- [20] Grünz L., Beyer G. H.: *Some observations on the interaction of recombination and self-adaptation in evolution strategies*. In P. J. Angeline and V. W. Porto (ed.): *Proc. 1999 Congress on Evolutionary Computation (CEC'99)*, volume 1, pages 639-645, Washington D.C., July 6-9, 1999. IEEE Press, Piscataway NJ.
- [21] Hoffmeister F., Bäck Th.: *Genetic algorithms and evolution strategies: similarities and differences*. Technical Report "Grüne Reihe" No. 365, Department of Computer Science, University of Dortmund, November 1990.

- [22] Hough P. C. V.: *Method and Means for Recognition Complex Patterns*. U. S. Patent 3069654, 1962.
- [23] Kwiatkowski W.: *Defuzyfikacja transformaty Hougha*. Biuletyn IAIr WAT, 5, 1996.
- [24] Malinowski T.: *Sztuczne sieci neuronowe w zadaniach segmentacji obrazów na podstawie tekstur*. Biuletyn IAIr WAT, 8, 1997.
- [25] Wojdyło W, Kwiatkowski W.: *Wykrywanie i pomiar parametrów linii prostych na obrazach rastrowych metodą uczenia konkurencyjnego*. Biuletyn IAIr WAT, 5, 1996.
- [26] Żorski W.: *Application of the Hough transform with a clustering technique to segmentation of digital images*. Proceedings of the GKPO'96 Conference. Machine Graphics & Vision, Vol.5, Nos. 1/2, 1996.
- [27] Żorski W.: *Segmentacja obrazów rastrowych w oparciu o transformatę Hougha z wykorzystaniem techniki grupowania*. Biuletyn IAIr WAT, 5, 1996.
- [28] Żorski W.: *Rozpoznawanie znaków alfanumerycznych na podstawie transformaty Hougha z wykorzystaniem techniki grupowania*. Biuletyn IAIr WAT, 8, 1997.

Recenzent: dr hab. inż. Włodzimierz Kwiatkowski, prof. WAT
Praca wpłynęła do redakcji: 30.09.2000