

Paweł Gryboś*, Sławomir Mikrut*

Analiza możliwości wykorzystania funkcji biblioteki „OpenCV 1.0” do automatycznego dopasowania zdjęć lotniczych**

1. Wprowadzenie

Zdjęcia lotnicze są coraz częściej wykorzystywane jako materiały do tworzenia dokumentacji geodezyjnej i kartograficznej, a także do tworzenia modeli 3D powierzchni. W procesach produkcyjnych map, modeli numerycznych terenu lub powierzchni, wykorzystuje się coraz częściej matching, czyli metody automatycznej korelacji zdjęć. Pozwalają one na automatyczny i szybki pomiar tych samych punktów na dwóch lub więcej zdjęciach jednocześnie. W niniejszej pracy zaprezentowano przykład aplikacji z wykorzystaniem funkcji ogólnodostępnej biblioteki do obróbki obrazów napisanej w języku C++, a następnie dokonano jej pewnych modyfikacji, celem podniesienia efektywności.

Biblioteka z rodziny tzw. *open source* o nazwie „OpenCV” została stworzona przez pracowników firmy Intel. Oprogramowano w niej bardzo dużą ilość funkcji i struktur możliwą do wykorzystania w procesie przetwarzania obrazów. Znajdują się tam gotowe funkcje wykonujące przekształcenia geometryczne, arytmetyczne i morfologiczne, pozwalające wczytywać, tworzyć i zapisywać obrazy w wielu formatach. Jedną grupę stanowią funkcje dopasowujące obrazy.

Na wstępie omówiono zagadnienia korelacji obrazów cyfrowych na przykładzie zdjęć lotniczych. Następnie przedstawiono algorytm oraz funkcje dostępne w bibliotece OpenCV. Testy przeprowadzono na rzeczywistych obrazach Bytomia i Krakowa.

2. Dopasowanie w przypadku zdjęć lotniczych

Podstawy teoretyczne zagadnienia dopasowania obrazów zostały szeroko opisane w literaturze [1, 2, 5–8]. Podsumowując, można stwierdzić, że dopasowanie dwóch obrazów (*matching*) – w artykule umownie nazywanych „lewy” i „prawy” – odbywa się po-

* Zakład Fotogrametrii i Informatyki Teledetekcyjnej, Akademia Górniczo-Hutnicza w Krakowie; pawelgrybos@interia.pl, smikrut@agh.edu.pl

** Praca finansowana z grantu Komitetu Badań Naukowych nr 4T12E 001 27

przez porównanie wartości jasności pikseli (*area-based matching*) lub cech takich jak krawędzie lub naroża obiektów (*feature-based matching*) występujących na danym fragmencie obrazu. Druga z metod wymaga wstępnego przetworzenia obrazów, w celu wydobycia interesujących nas cech. Ponadto dokładność dopasowania nie przekracza jednego piksela. Dopasowanie poprzez porównanie wartości jasności pikseli jest oparte na obliczaniu odpowiedniego współczynnika mówiącego o „jakości” dopasowania. W zależności od algorytmu jako miarę wykorzystuje się różnicę wartości jasności pikseli lub najczęściej współczynnik korelacji dwóch obrazów.

Przy obliczaniu dopasowania, w przypadku zdjęć lotniczych, pojawiają się problemy związane z faktem wzajemnego skręcenia obrazów, niejednakowym oświetleniem scen w chwilach fotografowania oraz deniwelacjami terenu. Skręcenie obrazów powoduje, iż w dopasowanym fragmencie obrazów, piksele z lewego zdjęcia nie będą idealnie pokrywać się z pikselami ze zdjęcia prawego – pomijając wpływ oświetlenia i deniwelacji. Różnica będzie tym większa im dalej będzie znajdował się piksel od środka obrazu. Skręcenie ma znaczenie przy wyliczaniu zgrubnego położenia szukanego punktu na lewym zdjęciu z wykorzystaniem przesunięcia między obrazami.

Wpływ niejednakowego oświetlenia można wyeliminować, stosując algorytm oparty na współczynniku kowariancji dwóch sygnałów dwuwymiarowych (obrazów).

Deniwelacje terenu są problemem zarówno przy obliczaniu zgrubnego położenia szukanego punktu, jak i przy obliczaniu współczynnika dopasowania dla danego fragmentu obrazu. Pod tym pojęciem możemy zaklasyfikować także dachy budynków, a w szczególności tych wysokich. Wykonanie zdjęcia budynku kilkunastopiętrowego z wysokości 1000 m powoduje, iż dach na zdjęciu lewym znajduje się w innym miejscu niż na zdjęciu prawym w odniesieniu do tych samych punktów terenu. Otoczenie dla punktu znajdującego się w pobliżu dachu na zdjęciu lewym, będzie całkiem inne na zdjęciu prawym.

Na końcu należy wspomnieć także o jeszcze jednym ważnym czynnikiem określającym obraz. Chodzi tutaj o to, co przedstawia dane zdjęcie, czyli strukturę obrazu. Korelacja na obrazach przedstawiających lasy, krzewy lub zawierających jednolite fragmenty takie jak wody stojące, łąki jest obarczona dużym prawdopodobieństwem wystąpienia błędnego dopasowania pomimo wysokiego współczynnika.

Najlepsze do korelacji wydają się być tereny o zróżnicowanej strukturze, zawierające elementy liniowe wzajemnie przecinające się (drogi, chodniki, ścieżki, granice pól itp.), elementy punktowe wyróżniające się jasnością w stosunku do tła oraz elementy powierzchniowe o urozmaiconej budowie. Jednakże, jak można wnioskować z poprzedniego akapitu, obraz przedstawiający blokowisko, pomimo że zawiera dużo elementów liniowych i punktowych nie da wyników, jakich spodziewalibyśmy się, co zostanie poparte wynikami testów w dalszej części artykułu.

3. Funkcje biblioteki OpenCV wykorzystane w algorytmie korelacji

Podstawową funkcją jaka została użyta jest *cvMatchTemplate()*, dokonuje porównania dwóch obrazów, a wynik zapisuje w postaci innego obrazu (mapy), w którym wartości piksela równają się wartości współczynnika dopasowania między tymi obrazami.

Składnia funkcji jest następująca:

```
void cvMatchTemplate (const CvArr* image,
                    const CvArr* templ, CvArr* result, int method)
```

gdzie:

image – obraz przeszukiwany,
templ – obraz dopasowywany,
result – mapa dopasowania,
method – algorytm obliczania współczynnika dopasowania.

Możliwe wartości parametru *method* określają następujące algorytmy:

- *method* = CV_TM_SQDIFF (0)

$$R(x, y) = \sum_{x', y'} (T(x', y') - I(x + x', y + y'))^2 \quad (1)$$

- *method* = CV_TM_SQDIFF_NORMED (1)

$$R(x, y) = \frac{\sum_{x', y'} (T(x', y') - I(x + x', y + y'))^2}{\sqrt{\sum_{x', y'} T(x', y')^2 \cdot \sum_{x', y'} I(x + x', y + y')^2}} \quad (2)$$

- *method* = CV_TM_CCORR (2)

$$R(x, y) = \sum_{x', y'} (T(x', y') \cdot I(x + x', y + y')) \quad (3)$$

- *method* = CV_TM_CCORR_NORMED (3)

$$R(x, y) = \frac{\sum_{x', y'} (T(x', y') \cdot I(x + x', y + y'))}{\sqrt{\sum_{x', y'} T(x', y')^2 \cdot \sum_{x', y'} I(x + x', y + y')^2}} \quad (4)$$

- *method* = CV_TM_CCOEFF (4):

$$R(x, y) = \sum_{x', y'} (T'(x', y') \cdot I'(x + x', y + y')) \quad (5a)$$

$$T'(x', y') = T(x', y') - \frac{1}{(w \cdot h)} \sum T(x'', y'') \quad (5b)$$

$$I'(x + x', y + y') = I(x + x', y + y') - \frac{1}{(w \cdot h)} \sum I(x + x'', y + y'') \quad (5c)$$

– *method* = CV_TM_CCOEFF_NORMED

$$R(x, y) = \frac{\sum_{x', y'} (T'(x', y') - I'(x + x', y + y'))}{\sqrt{\sum_{x', y'} T'(x', y')^2 \cdot \sum_{x', y'} I'(x + x', y + y')^2}} \quad (6)$$

gdzie:

I – *image*,

T – *template* (dopasowywany obraz),

R – wynik, czyli mapa dopasowania; $x' = 0 \dots w - 1$, $y' = 0 \dots h - 1$ (w – szerokość obrazu dopasowywanego, h – jego wysokość).

Funkcja działa w taki sposób, iż po obrazie przeszukiwanym (*I*) przesuwany jest obraz dopasowywany (*T*) i dla każdego położenia obliczana jest wartość wynikająca z algorytmu, a następnie ta wartość jest zapisywana na obrazie wynikowym (*R*).

Obraz wynikowy jest więc mapą, która przedstawia „jakość” dopasowania. Do znalezienia minimum (*method* = 0,1) lub maksimum (*method* = 2..5) wartości służy funkcja:

```
void cvMinMaxLoc( const CvArr* arr, double* min_val,
double* max_val, CvPoint* min_loc, CvPoint* max_loc,
const CvArr* mask ),
```

która znajduje globalne maksimum i minimum na obrazie, zwracając w wyniku działania współrzędne obydwu ekstremów i ich wartości. Kolejnymi funkcjami, które zostały użyte w algorytmie są funkcje do tworzenia piramidy obrazów:

```
void cvPyrDown( const CvArr* src, CvArr* dst, int filter=CV_GAUSSIAN_5x5 ),
void cvPyrUp( const CvArr* src, CvArr* dst, int filter=CV_GAUSSIAN_5x5 ).
```

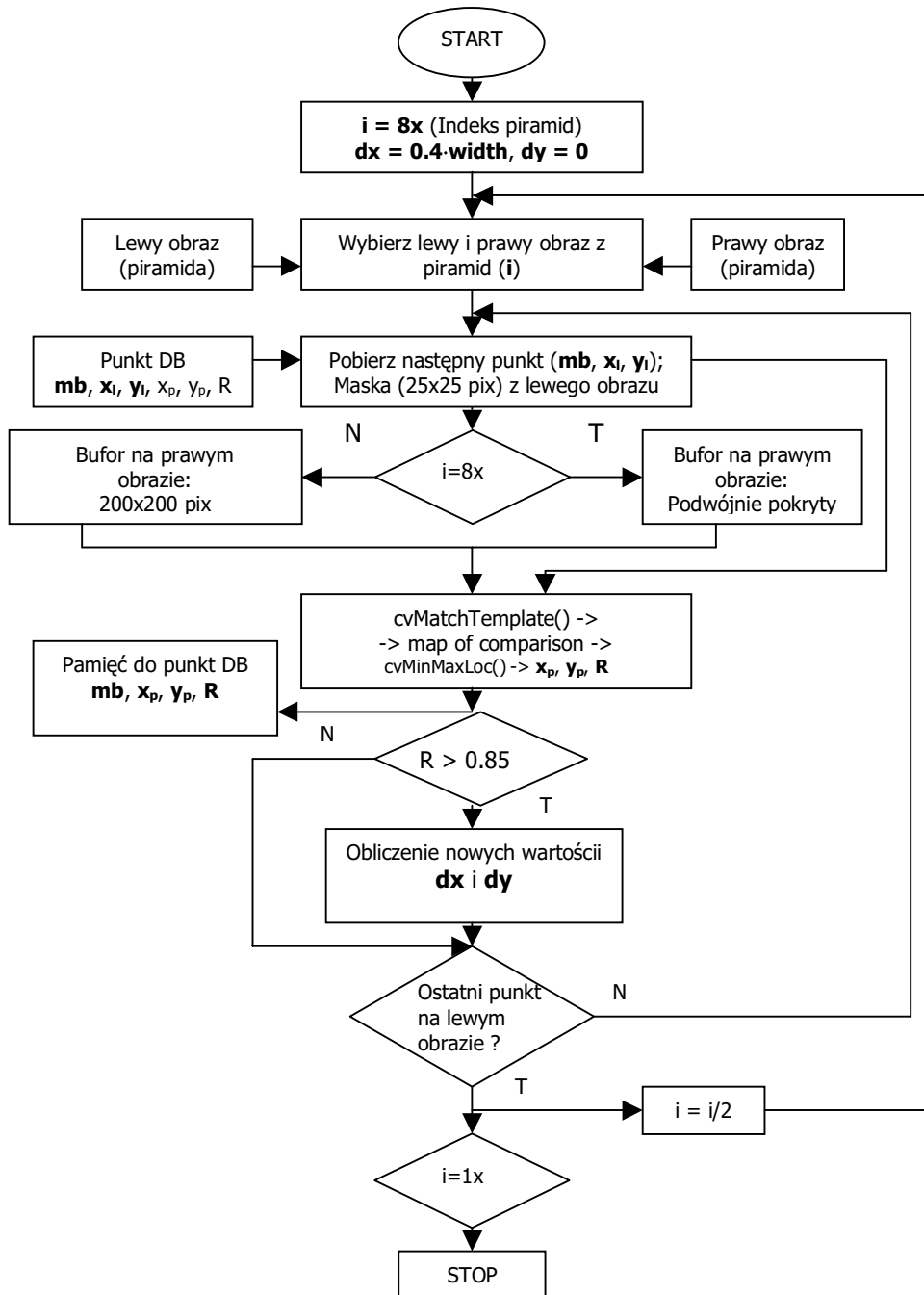
Pierwsza z nich pozwala na zmniejszenie dwukrotnie obrazu, a druga na operację odwrotną. Nowa wartość pikseli jest obliczana z wykorzystaniem filtru Gaussa na podstawie pierwotnych wartości pikseli. Aby uzyskać obraz 16-krotnie mniejszy, należy użyć czterokrotnie funkcji *cvPyrDown*.

Osobną grupę stanowią funkcje do tworzenia (*cvCreateImage()*) i kasowania (*cvReleaseImage()*) obrazów w pamięci komputera oraz do wczytywania (*cvLoadImage()*) i zapisywania (*cvSaveImage()*) obrazów do plików. Pozwalają one na obsługę zarówno wielu popularnych formatów plików (BMP, TIFF, JPEG i inne), jak i obrazów kolorowych.

Szerszy opis biblioteki można znaleźć w literaturze [3, 4].

4. Algorytm dopasowania obrazów zaimplementowany z wykorzystaniem funkcji bibliotecznych

Autorski schemat algorytmu przedstawiono na rysunku 1. Ogólna zasada polega na dopasowywaniu punktów na obu zdjęciach, przechodząc przez wszystkie poziomy piramidy od najmniejszej rozdzielczości do największej.



Rys. 1. Algorytm aplikacji

W pierwszej iteracji korelowane są obrazy ośmiokrotnie mniejsze od oryginału. Dla każdego punktu ze zdjęcia lewego szukany jest punkt na zdjęciu prawym na całym obszarze podwójnego pokrycia. Wynik matchingu zapisywany jest w bazie w postaci numeru oraz współrzędnych punktów i wartości współczynnika dopasowania.

Następna iteracja, na obrazie dwukrotnie większym, przebiega w podobny sposób, ale obszar przeszukiwania na zdjęciu prawym jest zdeterminowany przez bufor (przyjęto 200×200 pikseli) o punkcie centralnym odpowiadającym punktowi wyznaczonemu z poprzedniej iteracji w przypadku, gdy współczynnik dopasowania jest większy od 0,80. Jeżeli współczynnik jest mniejszy niż 0,80, to punkt środkowy bufora jest wyznaczany na podstawie współrzędnych punktu ze zdjęcia lewego i wartości dx i dy , które są uaktualniane przez cały czas. Ostatnia iteracja jest wykonywana na obrazie oryginalnym. Końcowe wyniki są zapisywane do pliku tekstowego i mogą służyć jako dane do programu wyliczającego parametry orientacji wzajemnej.

5. Testowanie działania algorytmu

Celem oceny poprawności działania algorytmu oraz określenia progu współczynnika dopasowania, zostały przeprowadzone testy na dwóch wybranych stereogramach:

- 1) BYTOM_L.tif i BYTOM_P.tif – zdjęcia Bytomia,
- 2) 4671_600_8_L.tif i 4671_600_8_P.tif – zdjęcia okolic Krakowa.

Zdjęcia Bytomia miały wielkość 5448×5424 pikseli, oraz rozdzielczość terenową 36 cm/piksel, a Krakowa nieco mniejsze 5433×5365 i 5433×5478 , o tej samej rozdzielczości terenowej.

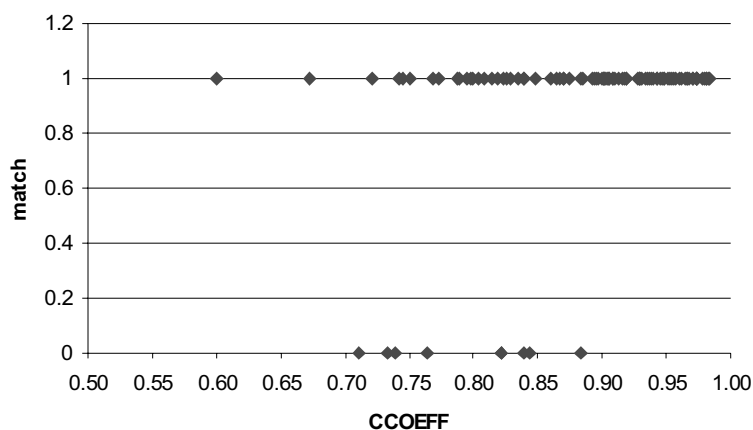
Zobrazowania Bytomia charakteryzują się gęstą zabudową miejską, z dużą ilością wysokich bloków. Natomiast zdjęcia z Krakowa obejmują tereny podmiejskie z polami uprawnymi i stawami. Oba stereogramy były zdjęciami o ośmiobitowej skali szarości.

Pierwszy test został przeprowadzony na zobrazowaniach Bytomia. W pierwszej fazie zostało wybranych manualnie, na zdjęciu lewym, 100 punktów, które wg operatora były dobrymi punktami do identyfikacji. Tak wybrane punkty zostały wczytane do aplikacji i „znajdzone” (dopasowane) na zdjęciu prawym, z algorytmem obliczania współczynnika $CV_TM_CCOEFF_NORMED$ (kowariancja dwóch sygnałów). Bufor przeszukiwania był ustawiony na 200×200 pikseli, a okno template'u na 25 pikseli (dobre na podstawie literatury). Dokonano oceny poprawności dopasowania oraz zbadano, jakie wartości przyjmuje współczynnik dopasowania dla punktów poprawnych i źle skorelowanych.

Na rysunku 2 przedstawiono zależność współczynnika CCOEFF od poprawności dopasowania.

Poprawność punktów została określona na podstawie wzrokowej oceny znalezionych punktów, przez dwie osoby, oraz dodatkowo wszystkie punkty zostały pomierzone na autografie cyfrowym VSD.

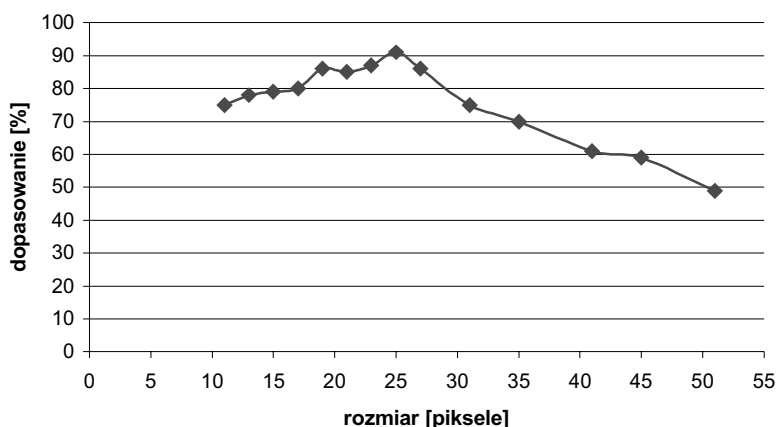
Jak widać na rysunku 2, dla większości punktów dobrze dopasowanych wartość współczynnika korelacji jest większa od 0,90, a dla punktów błędnie dopasowanych jest mniejsza od tej wartości.



Rys. 2. Wykres zależności poprawności dopasowania od współczynnika korelacji

Jeżeliby brać pod uwagę wartość graniczną równą 0,8 (wartość graniczna dla programów komercyjnych), tylko 4 punkty źle dopasowane mają wartości CCOEFF większe – różnica między tymi punktami, a punktami rzeczywistymi nie przekraczała jednego piksela. Wzrasta natomiast znacznie liczba punktów dobrze dopasowanych, które byłyby zaklasyfikowane przez automat jako niepoprawne. Skuteczność dopasowania wyniosła 91%. Procent skuteczności określony na podstawie współczynnika korelacji wyniósł 83%.

Punkty te posłużyły także do wyznaczenia optymalnego rozmiaru okna dopasowywanego (*template*). W kolejnym teście został przeprowadzony matching dla rozmiarów okna dopasowywanego od 11×11 do 51×51 pikseli. Wykres zależności skuteczności dopasowania od rozmiaru okna przedstawiony jest na rysunku 3.

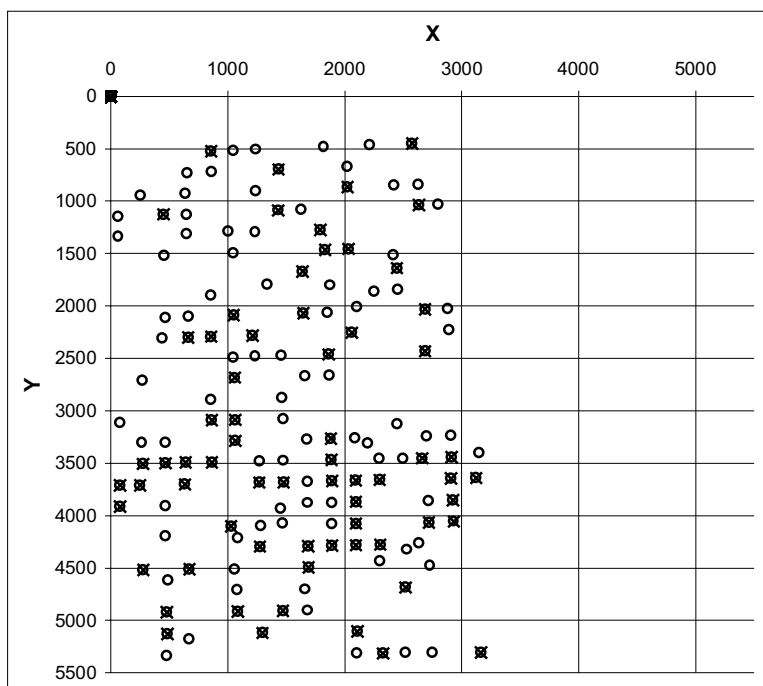


Rys. 3. Wykres zależności skuteczności dopasowania od rozmiaru okna dopasowywanego

Z rysunku 3 można wywnioskować, iż optymalnym rozmiarem okna jest 25×25 pikseli. Zarówno po zwiększeniu, jak i zmniejszeniu okna, spada skuteczność dopasowania punktów. Należy tu jednak zaznaczyć, że testy te zostały przeprowadzone na obrazowaniach miasta Bytom, przedstawiającym zwartą zabudowę miejską. Należałoby jeszcze sprawdzić, jak przedstawia się taki wykres dla zdjęć terenów podmiejskich i wiejskich. Przeprowadzono również testy z wykonaniem matchingu siatki punktów dla zobrazowania Bytomia i Krakowa. W przypadku Bytomia oczko siatki miało rozmiar 200 pikseli, a poprawność dopasowania określana na podstawie współczynnika korelacji przedstawia się zgodnie z tabelą 1.

Tabela 1
Wyniki korelacji dla wybranych obrazów testowych

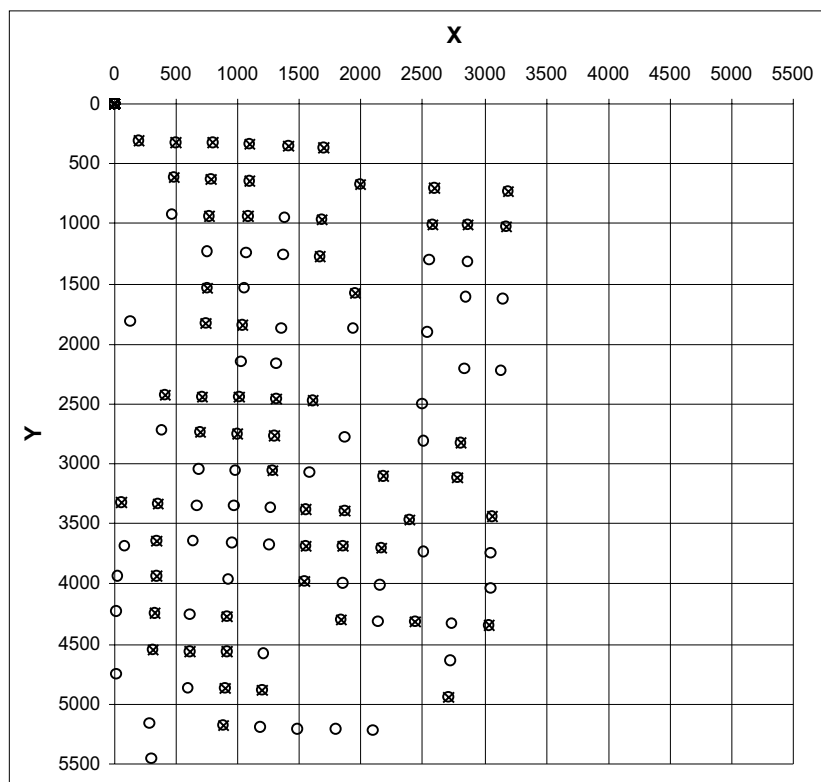
		CCOEFF>0,80	CCOEFF>0,85	CCOEFF>0,90
Bytom	Punkty poprawne	150	107	67
	Procent	37,5	26,8	16,8
Kraków	Punkty poprawne	112	87	59
	Procent	56,6	43,9	29,8



Rys. 4. Rozmieszczenie dopasowanych punktów na prawym zdjęciu zobrazowania Bytomia
Objaśnienia w tekście

Przy zaostrzeniu kryterium oceny poprawności dopasowania na podstawie współczynnika korelacji procent skuteczności spada do około 17%. W przypadku zobrazowania terenów podmiejskich Krakowa sytuacja przedstawia się lepiej. Skuteczność dopasowania dochodzi do 60% dla $CCOEFF > 0,8$ i 30% dla $CCOEFF > 0,9$.

Warto w tych dwóch przypadkach przyjrzeć się rozmieszczeniu dopasowanych punktów na zdjęciu. Na rysunkach 4 i 5 przedstawione są kolejno wykresy z zaznaczonymi punktami dla zobrazowania Bytomia i Krakowa. Na każdym z wykresów znajdują się dwa rodzaje punktów: znacznikami w postaci okręgów zaznaczone są punkty, które zostały dopasowane ze współczynnikiem większym od 0,8, a znacznikami w postaci krzyży, zaznaczone są punkty, które zostały dopasowane ze współczynnikiem większym od 0,9.



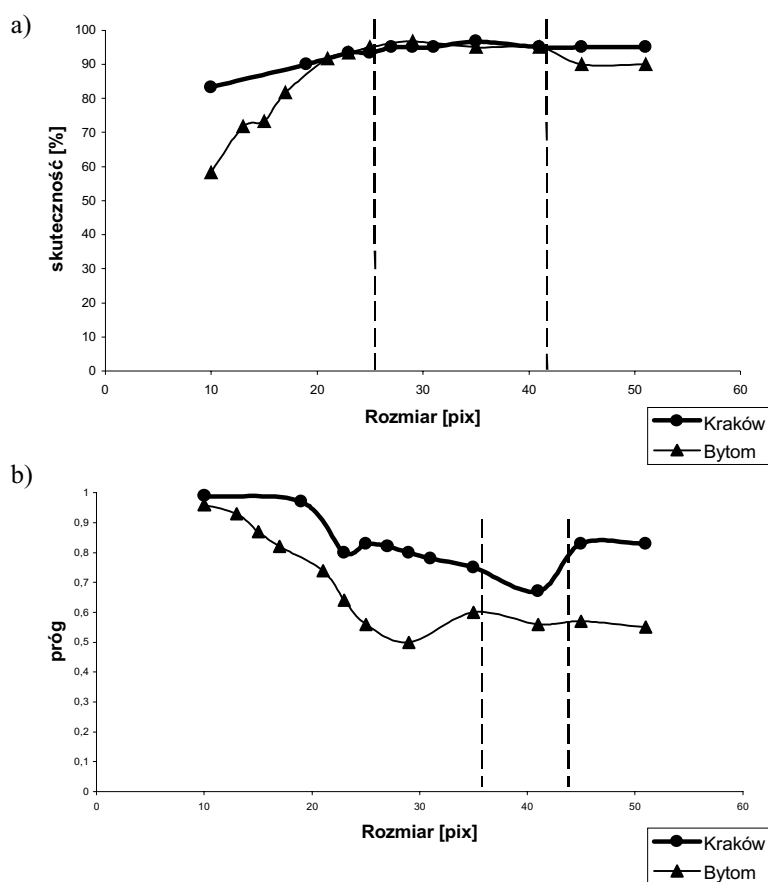
Rys. 5. Rozmieszczenie dopasowanych punktów na prawym zdjęciu zobrazowania Krakowa
Objaśnienia w tekście

Jak widać na wykresach, rozmieszczenie tych punktów układa się w siatkę, co potwierdza, iż punkty zostały dobrze dopasowane. Również rozmieszczenie punktów na zdjęciu pozwala uznać te punkty jako dobre do wyznaczenia parametrów orientacji wzajemnej zdjęć.

5.1. Sprawdzenie optymalnego rozmiaru maski (*template*) dla każdego poziomu piramidy

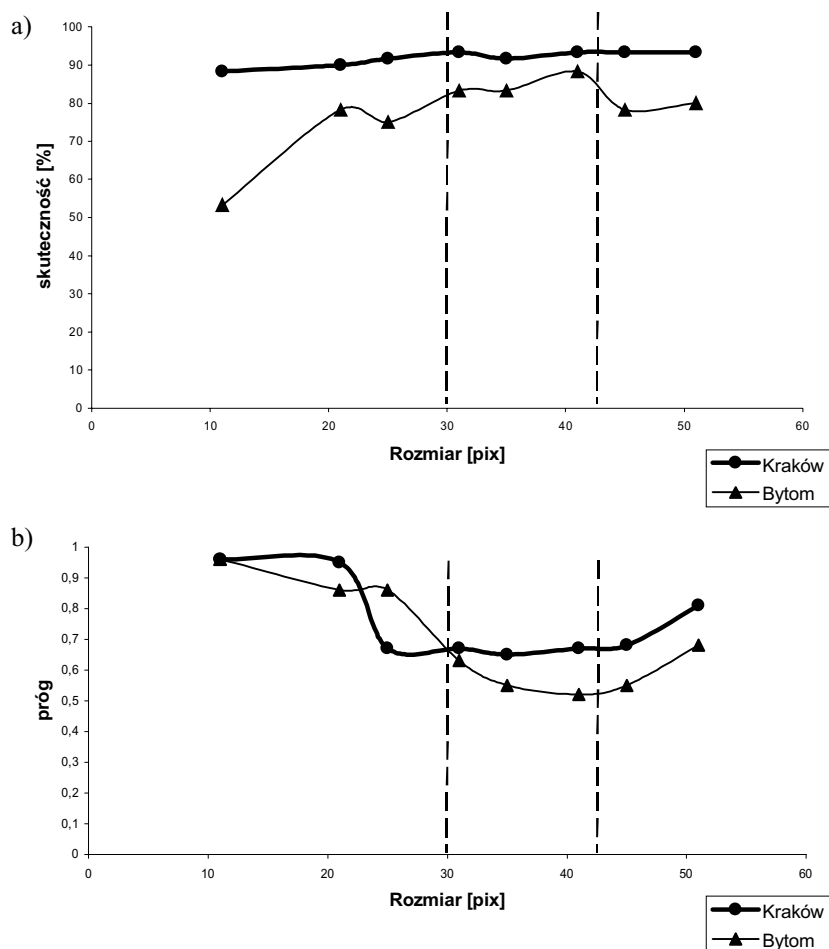
Kolejne testy miały za zadanie określić optymalny rozmiar maski (*template*) dla każdego poziomu piramidy, ponieważ każdy z nich różni się stopniem generalizacji, co powinno wpływać na skuteczność dopasowania i wiązać się z rozmiarem dopasowywanego wycinka lewego obrazu.

Testy zostały przeprowadzone na zdjęciach Bytomia i Krakowa. Dopasowywana była siatka złożona z 60 punktów, o rozmiarze oczka 500 pikseli. Badane były obrazy ośmio-, cztero- i dwukrotnie mniejsze od oryginału. Oprócz skuteczności dopasowania w zależności od rozmiaru, brana była pod uwagę wartość progu, czyli liczba, jaka mogła być przyjęta w algorytmie do automatycznej oceny poprawności dopasowania. Przetestowano rozmiary od 11×11 do 51×51 . Dla każdego poziomu piramidy zestawiono dwa wykresy przedstawiające zależność skuteczności i progu od rozmiaru maski (rys. 6–8).



Rys. 6. Wykresy skuteczności dopasowania (a) i progu automatycznej oceny poprawności poprawnego dopasowania (b) w funkcji rozmiaru maski (*template*) dla obrazu ośmiokrotnie mniejszego

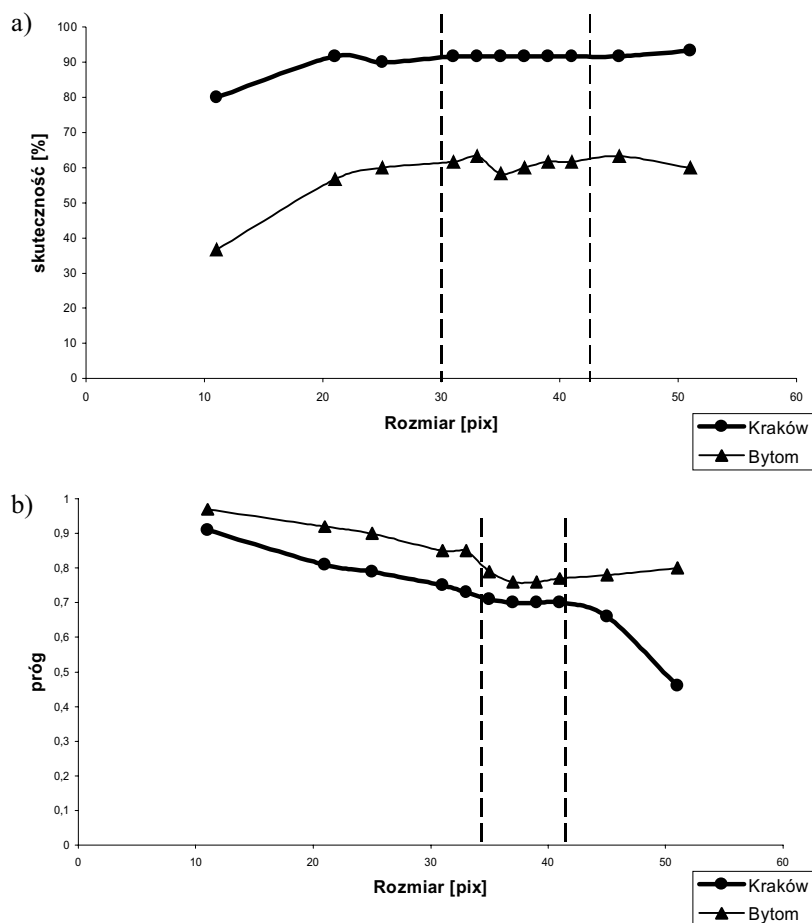
Objaśnienia w tekście



Rys. 7. Wykresy skuteczności dopasowania (a) i progu automatycznej oceny poprawności poprawnego dopasowania (b) w funkcji rozmiaru maski (*template*) dla obrazu czterokrotnie mniejszego
Objaśnienia w tekście

Na każdym z wykresów zaznaczony jest linią przerywaną przedział reprezentujący optymalny zakres rozmiaru dopasowywanego okna. Końcowy zakres jest iloczynem przedziałów dla wykresów skuteczności dopasowania oraz progu.

Wcześniejsze testy wykazały, iż optymalny rozmiar maski (*template*) dla obrazu oryginalnego wynosi 25 pikseli. W przypadku obrazów pomniejszonych (zdegradowanych) wartość optymalnego rozmiaru wzrasta i może być dobrana z przedziału od ponad 30 do ponad 40, średnio dla wszystkich poziomów piramidy. Dla obrazów najmniejszych, w przedziale 29÷51, skuteczność praktycznie się nie zmienia i jest tak samo wysoka dla obydwu scen.



Rys. 8. Wykresy skuteczności dopasowania (a) i progu automatycznej oceny poprawności dobrego dopasowania (b) w funkcji rozmiaru maski (*template*) dla obrazu dwukrotnie mniejszego

Wynika to z faktu, iż degradacja na obrazie Bytomia jest już tak znaczna, że zredukowany jest wpływ wysokich budynków i gęstej zabudowy miejskiej na skuteczność dopasowania. Wykres progu przedstawia w jeszcze lepszym świetle zobrazowanie Bytomia, gdyż w zaznaczonym przedziale wartość progu oscyluje w okolicy 0,60, przy 0,70 dla Krakowa. Na wyższym poziomie piramidy uwidacznia się już wpływ różnorodności terenów przedstawionych przez testowanie sceny, skuteczność dopasowania dla Krakowa jest wyższa, ale próg jest niższy dla Bytomia. Optymalny rozmiar maski (*template*) mieści się w przedziale do 30 do 45. Próg w przypadku Krakowa waha się w pobliżu 0,70, a dla Bytomia jest to wartość od około 0,60 do nawet 0,55.

Dla obrazów dwukrotnie mniejszych optymalny przedział jest podobny jak dla obrazów czterokrotnie mniejszych. Wzrasta natomiast wartość progu: Kraków – 0,70 i Bytom –

0,80±0,75. Uwidacznia się także w jeszcze większym stopniu, różnica w skuteczności dopasowania.

Na podstawie tych wyników możemy określić optymalne parametry dla algorytmu dopasowania, które pozwolą nam jak najlepiej dopasowywać punktu począwszy od najmniejszego obrazu piramidy.

6. Podsumowanie i wnioski

W ramach badań mających na celu opracowanie systemu do korelacji obrazów, zaimplementowano wybrane funkcje ogólnie dostępnej (*open source*) biblioteki OpenCV. Oprogramowano i przetestowano powstały system. Wstępne wyniki badań ukazują celność obranej drogi. Do przetestowania wybrano rzeczywiste obrazy zdjęć lotniczych z dwóch odmiennych rejonów, tj. Bytomia i Krakowa (różna skala i tekstura obrazu).

W trakcie badań został ustalony optymalny rozmiar okna dopasowywanego oraz wartość współczynnika korelacji decydująca o poprawności dopasowania. Wyniki dopasowania zdjęć lotniczych są zbliżone do wartości uzyskiwanych w programach komercyjnych. Również zastosowanie opcji z wygenerowaną wcześniej siatką punktów przyniosło spodziewane efekty.

Dopasowane punkty pokrywają obszar stereogramu i są rozmieszczone w miarę równomiernie, co jest niezbędne do wykonania procesu orientacji wzajemnej zdjęć lotniczych.

W pracy dokonano również badania optymalnego doboru wielkości maski (*template*) do rozmiaru piramidy. Zauważono, że wielkość rozmiaru maski wzrasta wraz ze zmniejszaniem się rozdzielczości obrazu. Stwierdzono, że optymalny rozmiar maski mieści się w przedziale 30±45.

Kolejnym krokiem w planowanych badaniach będzie zastosowanie dopasowania obrazów na podstawie ich cech.

Literatura

- [1] Kraus K.: *Photogrammetry*. Bonn, Dummler 1997
- [2] Praca zbiorowa, *Manual of Photogrammetry. 5th ed.* Bethesda, ASPRS, 2004
- [3] <http://www.intel.com/technology/computing/opencv/>
- [4] <http://opencvlibrary.sourceforge.net/>
- [5] Chwastek T., Mikrut S.: *Prezentacja autorskiego programu automatycznego pomiaru znaczków tłowych na obrazach zdjęć lotniczych*. Olsztyn, Archiwum Fotogrametrii, Kartografii i Teledetekcji 2006
- [6] Cyganek B.: *Komputerowe przetwarzanie obrazów trójwymiarowych*. Warszawa, Akademicka Oficyna Wydawnicza EXIT 2002
- [7] Sawicki P., Ostrowski B.: *Badanie wybranych metod matchingu do pomiaru punktów na cyfrowych obrazach bliskiego zasięgu*. Roczniki Geomatyki, t. III, z. 2, 2005
- [8] Zieliński J.: *Strategia automatyzacji pomiarów na stereogramach cyfrowych z zastosowaniem metod korelacyjnych*. Kraków, Archiwum Fotogrametrii, Kartografii i Teledetekcji 1998