

POLITECHNIKA WROCŁAWSKA
WYDZIAŁ ELEKTRONIKI

KIERUNEK: Informatyka (INF)
SPECJALNOŚĆ: Inżynieria systemów informatycznych (INS)

**PRACA DYPLOMOWA
MAGISTERSKA**

Wybór cech do rozpoznawania
złożonych form przestrzennych

Selection of features
for recognizing complicated shapes

AUTOR:
inż. Jan Słowik

PROWADZĄCY PRACĘ:
Prof. dr hab. inż. Ewaryst Rafajłowicz, I-6

OCENA PRACY:

Dedykuję moim rodzicom.

Spis treści

1 Wstęp	3
I Teoria	4
2 Przegląd algorytmów wyszukujących cechy obrazów	5
2.1 Algorytm SIFT	7
2.1.1 Detekcja skalo-przestrzennych ekstremów	7
2.1.2 Selekcja punktów charakterystycznych	10
2.1.3 Określenie orientacji punktów charakterystycznych	11
2.1.4 Budowa deskryptora	12
2.2 Algorytm SURF	14
2.2.1 Obliczenie obrazów całkowych	15
2.2.2 Lokalizowanie punktów charakterystycznych	15
2.2.3 Określenie orientacji punktów charakterystycznych	19
2.2.4 Budowa deskryptora	21
2.3 Algorytm STAR	24
2.3.1 Obliczenie pochyłych obrazów całkowych	24
2.3.2 Lokalizowanie punktów charakterystycznych	24
2.3.3 Selekcja punktów charakterystycznych	26
2.3.4 Budowa deskryptora	26
2.4 Algorytm FAST	27
2.4.1 Zasady egzaminowania	27
2.4.2 Uczenie maszynowe	27
2.4.3 Selekcja punktów charakterystycznych	29
2.4.4 Budowa deskryptora	30
2.5 Algorytm BRIF	31
2.5.1 Zasada działania	31
2.5.2 Wstępne rozmycie obrazu	32
2.5.3 Wybór testowanych par punktów	32
2.6 Algorytm ORB	35
2.6.1 Lokalizowanie punktów charakterystycznych	35
2.6.2 Budowa deskryptora	39
II Badania	42
3 Opis badań	43
3.1 Badane algorytmy	43

3.2	Kryteria badań	44
3.3	Zbiory testowe	44
4	Zbiór testowy Mikołajczyka	45
4.1	Rozmycie obrazu	46
4.1.1	BIKES	46
4.1.2	TREES	50
4.1.3	Dyskusja wyników	54
4.2	Zmiana położenia obserwatora	55
4.2.1	GRAFFITI	55
4.2.2	WALL	59
4.2.3	Dyskusja wyników	64
4.3	Rotacja i zmiana skali	64
4.3.1	BARK	64
4.3.2	BOAT	69
4.3.3	Dyskusja wyników	73
4.4	Zmiana oświetlenia	73
4.4.1	CARS	73
4.4.2	Dyskusja wyników	78
5	Zbiór testowy obrazów skalnych	79
5.1	Rotacja	80
5.1.1	R1	80
5.1.2	R2	84
5.1.3	R3	89
5.1.4	Dyskusja wyników	94
5.2	Zmiana położenia obserwatora	95
5.2.1	V1	95
5.2.2	V2	99
5.2.3	Dyskusja wyników	99
5.3	Zmiana skali	104
5.3.1	Z1	104
5.3.2	Dyskusja wyników	109
6	Podsumowanie	110
A	Opis aplikacji testowej	111
B	Zawartość płyty DVD	113
	Bibliografia	113

Rozdział 1

Wstęp

Widzenie maszynowe jest podzikiem szerokiej dziedziny informatyki jaką jest przetwarzanie obrazów. W ostatnich latach nauka ta dynamicznie się rozwija wraz ze wzrostem mocy obliczeniowej. W ramach widzenia maszynowego niezwykle ciekawym zagadnieniem jest rozpoznawanie obrazów.

Samo zadanie rozpoznawania obrazów, nie jest z punktu widzenia sztucznej inteligencji dużym problem. Aby móc rozpoznawać obrazy należy posiadać zbiór cech według których będziemy klasyfikować. Przetwarzanie obrazów rozumiane jako filtracja i analiza, mające na celu opracowanie zbioru cech obrazu jest najbardziej złożonym i skomplikowanym procesem.

Początki widzenia maszynowego znajdują się w przemyśle, gdzie systemy wizyjne były i są wykorzystywane do nadzorowania procesów produkcyjnych. Odbywa się to w ścisłe określonych i zoptymalizowanych dla konkretnego systemu warunkach działania. W raz rozwojem innych dziedzin informatyki takich jak np. rozszerzona rzeczywistość czy robotyka, widzeniu maszynowemu stawiane są coraz to nowe wyzwania. Są to np. jak praca w zmiennej orientacji, w zróżnicowanym środowisku w dowolnych warunkach. Wymusza to poszukiwanie nowych rozwiązań wykrywających cechy charakterystyczne obiektów. Cechy takie powinny być możliwie jak najmniej zależne od transformacji takich jak skalowanie i rotacja obrazu, lokalizacji kamery oraz odporne na zmienne oświetlenie obiektu.

Celem badawczym niniejszej pracy było przebadanie 6 algorytmów służących do ekstrakcji i opisu cech obrazów. Algorytmy te zostały przetestowane pod kątem skuteczności jak i szybkości działania. Ponadto podjęto prace mające na celu opracowanie hybrydowego algorytmu będącego syntezą wcześniej opisanych metod metod. Jako przykład skomplikowanych form przestrzennych zdecydowano się część testów wykonać na obrazach przedstawiających formacje skalne.

Część I

Teoria

Rozdział 2

Przegląd algorytmów wyszukujących cechy obrazów

Poszukując algorytmów służących do lokalizowania cech charakterystycznych obrazów posłużono się badaniami [10], [9]. Autor w oparciu o prace [20] dokonał zestawienia popularnych metod zaimplementowanych w służących do przetwarzania obrazów bibliotece OpenCV (wersje 2.2.0 i 2.3.1). Algorytmy były badane na testowym zbiorze obrazów przedstawionym na rysunku 2.1 poddanego przekształceniom programowym. Wyniki badań prezentują rezultaty od 2.2, do 2.4.



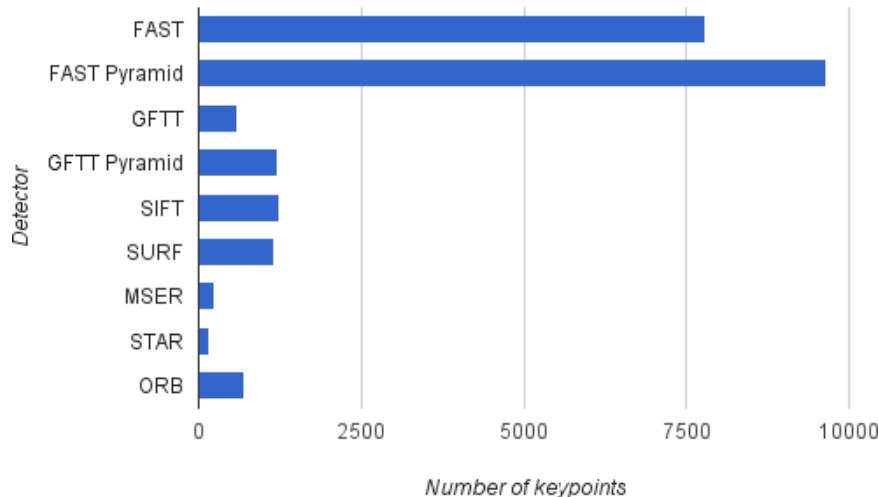
(a) Barbara

(b) Lena

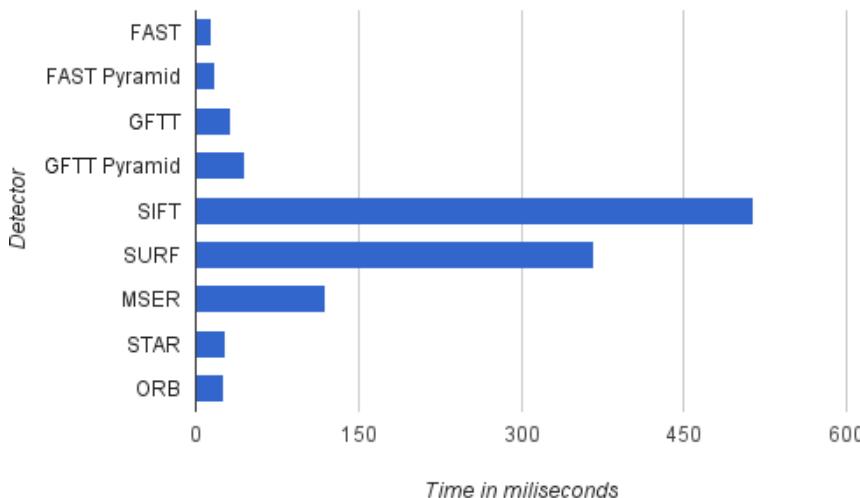
(c) Peppers

(d) Mandril

Rysunek 2.1 Zestaw obrazów testowych do porównania algorytmów

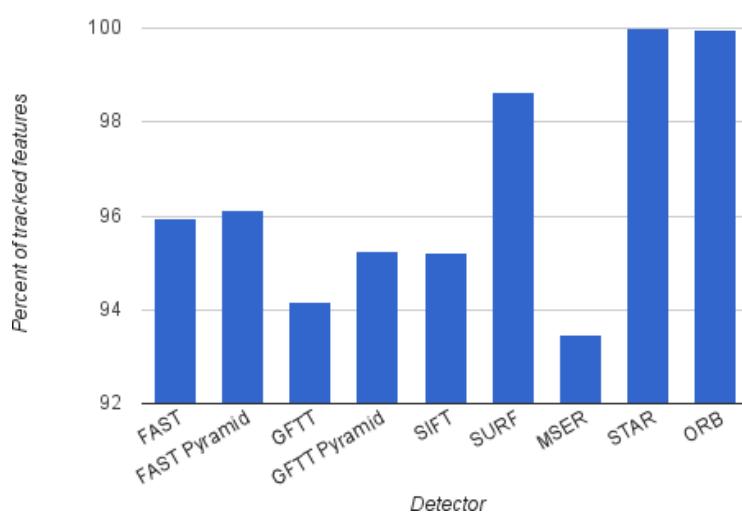


Rysunek 2.2 Liczba wykrytych cech



Rysunek 2.3 czas potrzebny na wykrycie cechy

{fig:Detector}



Rysunek 2.4 Skuteczność śledzenia wyszukanych cech

{fig:Perce}

Na podstawie zaprezentowanych wyników wszystkie algorytmy możemy zaklasyfikować jako dobre, gdyż pozwalają śledzić zlokalizowane cechy z ponad 90 % skutecznością. Niemniej jednak poszczególne metody różnią się szybkością i ergonomią działania. Algorytm **FAST** dostarcza bardzo dużą liczbę punktów charakterystycznych w krótkim czasie, ale ze względu na ich stan nie zawsze można je wykorzystać. Algorytmy **SURF**, **STAR** i **ORB** lokalizują mniej cech ale znacznie wyższej klasy. Pośród tych algorytmów przeciętnie wypada algorytm **SIFT**, który jednak jest jednym z najpopularniejszych i najlepiej udokumentowanych. Należy jednak pamiętać, że zbiory testowe zostały zbudowane na podstawie jednego zdjęcia, poddanego obróbce programowej. W rzeczywistym środowisku działania algorytmy nie operują na tak "wyidealizowanych".

Obrazy skalne są obrazami specyficznymi ze względu na wprowadzającą dużo informacji fakturę skały. Właściwość ta może być przydatna w rozpoznawaniu obiektów jak i stanowić rodzaj szumu zakłócającego pracę algorytmu. Przed dokonaniem eksperymentów nie sposób tego jednoznacznie stwierdzić, a co za tym idzie wybrać optymalną metodę. W związku z powyższym w niniejszej pracy zdecydowano się szczegółowo opisać i przebadać 5 wspomnianych algorytmów reprezentujących różne podejścia oraz algorytm opisu cech - **BRIEF**.

2.1 Algorytm SIFT

Algorytm SIFT (skrót od Scale Invariant Feature Transform) został opublikowany w 1999 roku przez Davida G. Lowe [11]. Najpełniejszy jego opis można znaleźć w pracy [12]. Jest to jeden z najefektywniejszych i szeroko stosowanych algorytmów wyszukiwania charakterystycznych punktów obrazu. Pozwala on lokalizować i opisywać specyficzne cechy obrazu, które są niezależne od transformacji takich jak skalowanie i rotacja, a także w dużym stopniu odporne na zmienne oświetlenie i położenie kamery. Wyszukane punkty kluczowe są dobrze rozpoznawalne zarówno w dziedzinie przestrzennej jak częstotliwościowej. Algorytm ten jest chroniony prawem patentowym.

W działaniu algorytmu możemy wyróżnić 4 etapy przetwarzania:

1. Detekcja skalo-przestrzennych ekstremów niezależnych od skali i orientacji obrazu.
2. Selekcja punktów charakterystycznych.
3. Przypisanie orientacji punktom charakterystycznym.
4. Budowa deskryptorów punktów charakterystycznych.

2.1.1 Detekcja skalo-przestrzennych ekstremów

Rozmycie Gaussowskie

Pierwszym krokiem służącym pozyskaniu punktów charakterystycznych obrazu w jest wygenerowanie jego reprezentacji skalo-przestrzennej (ang. scale-space). Oryginalny obraz $I(x,y)$ jest poddawany progresywnemu rozmyciu poprzez operację splotu ze jądrem rozmycia gaussowskiego $G(x, y, \sigma)$. W wyniku działania tej operacji otrzymujemy rozmyty obraz $L(x, y, \sigma)$.

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (2.1)$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.2)$$

Jądro rozmycia jest skalowane poprzez współczynnik k . Zwiększając wartość współczynnika k otrzymujemy coraz bardziej rozmyte obrazy.

$$G(x, y, k\sigma) = \frac{1}{2\pi(k\sigma)^2} e^{-\frac{x^2+y^2}{2(k\sigma)^2}} \quad (2.3)$$

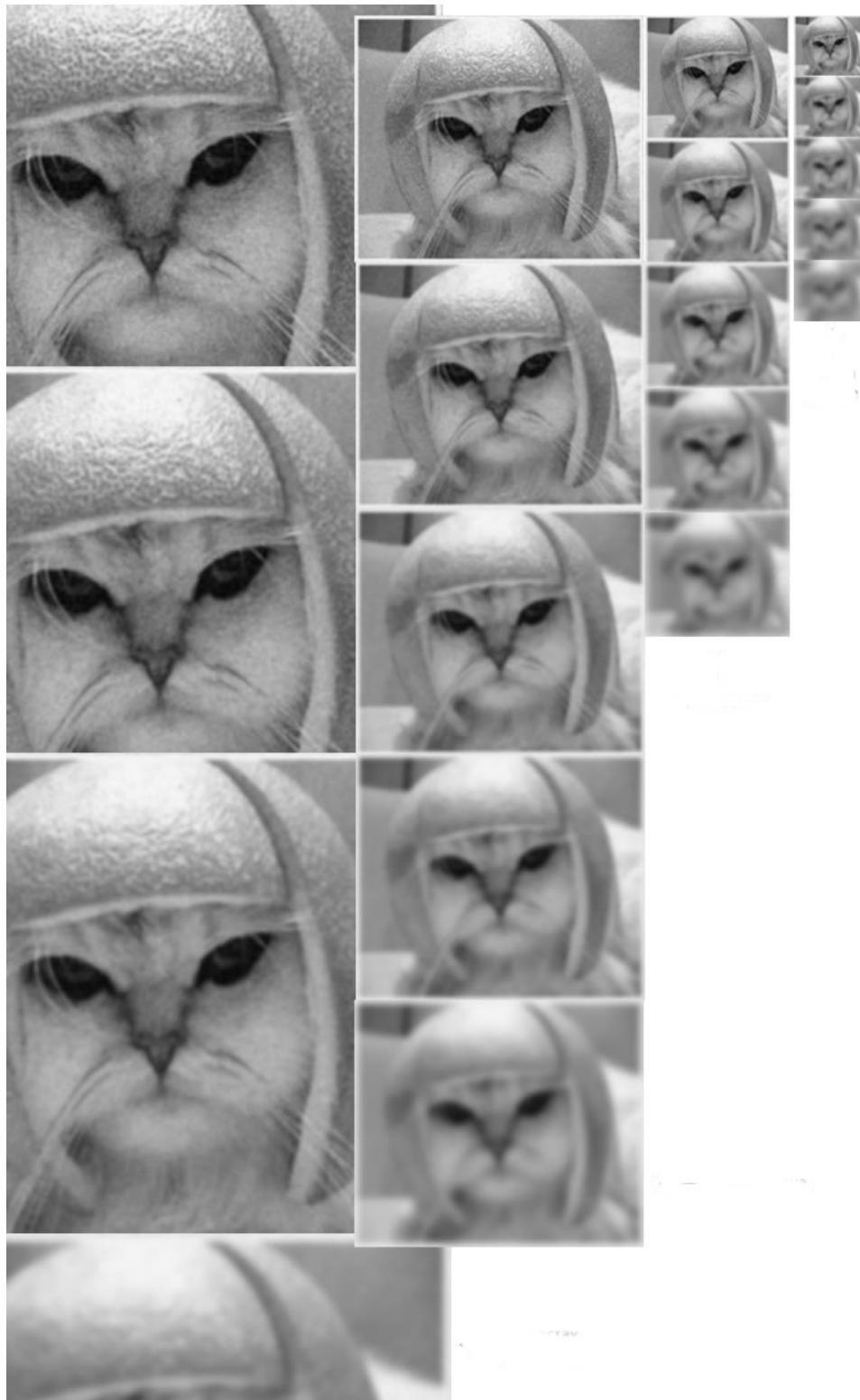
Obrazy rozmyte i oryginalny są grupowane w oktawę. Następnie obraz oryginalny jest próbkowany z dwa razy mniejszą częstotliwością, skutkiem czego otrzymujemy przeskaloną, pomniejszoną kopię. Kopia ta podobnie jak oryginał jest poddawana serii progresywnych rozmyć, w wyniku czego otrzymujemy drugą oktawę. Zabiegi te są powtarzane kilkukrotnie. Twórca algorytmu SIFT, David Lowe zaleca utworzenie 4 oktaw, składających się z 5 obrazów.

Różnica Gaussianów

Kolejnym krokiem jest odjęcie od siebie sąsiadujących obrazów w oktawie. Uzyskane w ten sposób obrazy nazywamy różnicą Gausiannów (ang. DoG – Difference of Gaussian). Z każdej 5 obrazowej oktawy otrzymujemy 4 obrazy różnicowe.

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \quad (2.4)$$

$$= L(x, y, k\sigma) - L(x, y, \sigma) \quad (2.5)$$



Rysunek 2.5 Zestawienie kolejnych oktaw badanego obrazu.

{fig:sift_a}

Różnica Gaussianów jest efektywną aproksymacją innej operacji służącej do lokalizowania punktów charakterystycznych – skalowo znormalizowanego Laplasianu Gausjanów $\sigma^2 \nabla^2 G$ (ang. LoG – Laplacian of Gaussian). Ekstrema LoG dostarczają znacznie bardziej stabilnych punktów charakterystycznych niż detektory takie jak Hessian czy funkcja

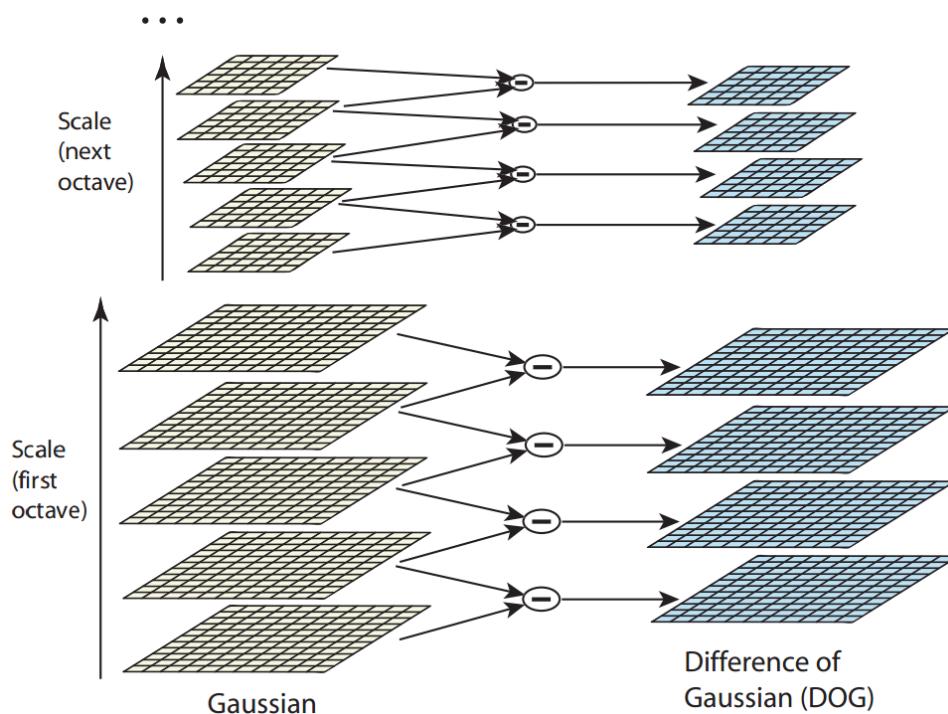
Harrisa.

$$\frac{\partial G}{\partial \sigma} = \sigma \nabla^2 G \quad (2.6)$$

$$\sigma \nabla^2 G = \frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{\sigma(k-1)} \quad (2.7)$$

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k-1)\sigma^2 \nabla^2 G \quad (2.8)$$

Aby obliczyć Laplasian Gausjanów należy obraz poddać rozmyciu, a następnie policzyć jego pochodne drugiego stopnia, aby znaleźć lokalne ekstrema. Rozmycie ma na celu redukcję szumów z obrazu, na które bardzo wrażliwe są pochodne drugiego stopnia. W efekcie otrzymujemy stabilne punkty reprezentujące krawędzie i narożniki w obrazie. Mankamentem tej metody jest złożoność obliczeniowa otrzymywania pochodnych drugiego stopnia. Stosując podejście zaproponowane w algorytmie SIFT, duży narzut operacji został zredukowany do prostej subtrakcji obrazów.

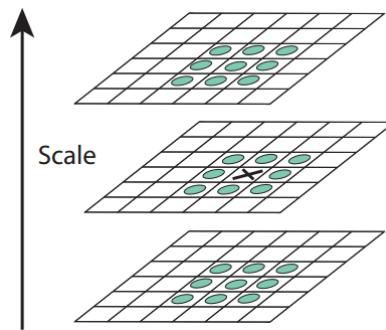


Rysunek 2.6 Schemat generowania obrazów różnicy Gaussianów

{fig:sift_0}

Ekstrema jako potencjalne punkty charakterystyczne

Aby zlokalizować lokalne ekstremum należy dokonać badania sąsiedztwa potencjalnego punktu charakterystycznego. W tym celu dokonujemy iteracji po wszystkich pikselach obrazu porównując je z otoczeniem "poziomym" (w obrębie jednego obrazu) jak i "pionowym" (czyli z otaczającymi pikselami z szeregu sąsiadujących rozmytych obrazów). Przedstawia to rysunek 2.7. W związku, że skrajne obrazy w oktawie mają tylko po jednym sąsiedzie, możemy je pominąć przy analizie.



Rysunek 2.7 Porównywanie otoczenia piksela w celu znalezienia lokalnego ekstremum

{fig:sift_01}

Subpixselowa lokalizacja punktów charakterystycznych

W związku, z tym, ekstrema często wypadają między sąsiadującymi pikselami, dokładną pozycję ekstremum uzyskuje się korzystając z rozwinięcia Taylora.

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \quad (2.9)$$

$$\hat{x} = -\frac{\partial^2 D}{\partial x^2}^{-1} \frac{\partial D}{\partial x} \quad (2.10)$$

2.1.2 Selekcja punktów charakterystycznych

W wyniku działania wcześniej opisanych kroków algorytm generuje ogromną liczbę punktów charakterystycznych. Niestety znaczna część punktów jest słabej jakości, zbyt mało kontrastowa lub leży na krawędziach. Nie pozwala to na ich praktyczne wykorzystanie i takich punktów należy się pozbyć.

Usuwanie cech mało kontrastowych

Usuwanie cech mało kontrastowych odbywa się poprzez progowanie. Jeżeli wartość bezwzględna piksela obrazu DoG jest mniejsza niż heurystycznie dobrany próg piksel taki jest odrzucany. W pracy [12] autor algorytmu zakładał próg rzędu 3 %. W przypadku gdy punktem charakterystycznym jest subpixsel, jego wartość podobnie jak pozycje wyliczamy z szeregu Taylora.

$$D(\hat{x}) = D + \frac{1}{2} \frac{\partial D}{\partial x} \hat{x} \quad (2.11)$$

Usuwanie punktów leżących na krawędziach

Różnica Gaussianów generuje silną odpowiedź na krawędziach. Powoduje to pewnego rodzaju przekłamania, jak np lokalizowanie punktów charakterystycznych w miejscach gdzie są one słabo określone i bardzo podatne na zakłóczenia. W przypadku krawędzi, gradient prostopadły do niej jest duży, a równoległy mały. Warunki te nie są wystarczające do ulokowania stabilnego punktu charakterystycznego. Najlepsze punkty charakterystyczne znajdują się w rogach, ze względu na duże wartości gradientów w obu kierunkach. W selekcji punktów charakterystycznych wykorzystuje się dwuwymiarową macierz Hesjanu \mathbf{H} , która jest liczona w potencjalnym punkcie charakterystycznym i w skali jemu odpowiadającej.

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (2.12)$$

Pochodne D_{xx}, D_{yy}, D_{xy} są liczone poprzez odejmowanie pikseli z sąsiedztwa badanego punktu.

Wartości własne macierzy są proporcjonalne do głównej krzywizny obrazu różnicowego D. Na podstawie podejścia zaproponowanego przez Harrisa i Stephensona w 1988 roku [8], obliczanie wartości własnych macierzy możemy zastąpić poprzez obliczenie ich stosunku.

Niech α i β będą wartościami własnymi macierzy. Ich sumę i iloczyn możemy obliczyć ze śladu i wyznacznika macierzy Hesjanu.

$$Tr(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta \quad (2.13)$$

$$Det(\mathbf{H}) = D_{xx}D_{yy} - D_{xy}^2 = \alpha\beta \quad (2.14)$$

Gdy wyznacznik macierzy \mathbf{H} jest ujemny badany punkt jest odrzucany. W sytuacji gdy wyznacznik jest dodatni stosujemy podstawienie $\alpha = r\beta$ co pozwala nam zapisać stosunek między śladem i wyznacznikiem macierzy za pomocą samego współczynnika r

$$\frac{Tr(\mathbf{H})^2}{Det(\mathbf{H})} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(r\beta + \beta)^2}{r\beta^2} = \frac{(r+1)^2}{r} \quad (2.15)$$

Aby wyeliminować odpowiedź krawędziową należy porównać stosunek głównych krzywizn badanego punktu i dokonać prostego progowania z użyciem współczynnika r .

$$\frac{Tr(\mathbf{H})^2}{Det(\mathbf{H})} < \frac{(r+1)^2}{r} \quad (2.16) \quad \text{\{eqn:ratio_rule\}}$$

Przedstawiona metoda jest bardzo efektywna obliczeniowo. Do przetestowania pojedynczego potencjalnego punktu charakterystycznego potrzebuje mniej niż 20 operacji zmiennoprzecinkowych. W badaniach [12] sugerowany współczynnik r wynosi 10.

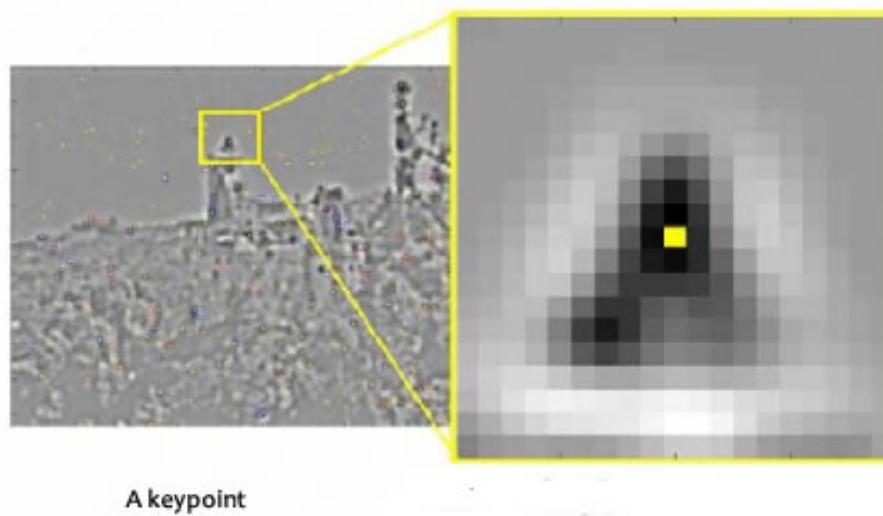
2.1.3 Określenie orientacji punktów charakterystycznych

Wyselekcjonowanego zbioru punktów charakterystycznych należy przypisać orientacje. Zadanie to jest kluczowe w działaniu algorytmu, gdyż pozwala uniezależnić punkty charakterystyczne od rotacji obrazu. Na podstawie obrazu poddanemu rozmyciu Gaussa w skali rozmycia w jakiej dany punkt charakterystyczny został znaleziony wylicza się wartości i orientacje gradientów w otoczeniu punktu charakterystycznego:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (2.17)$$

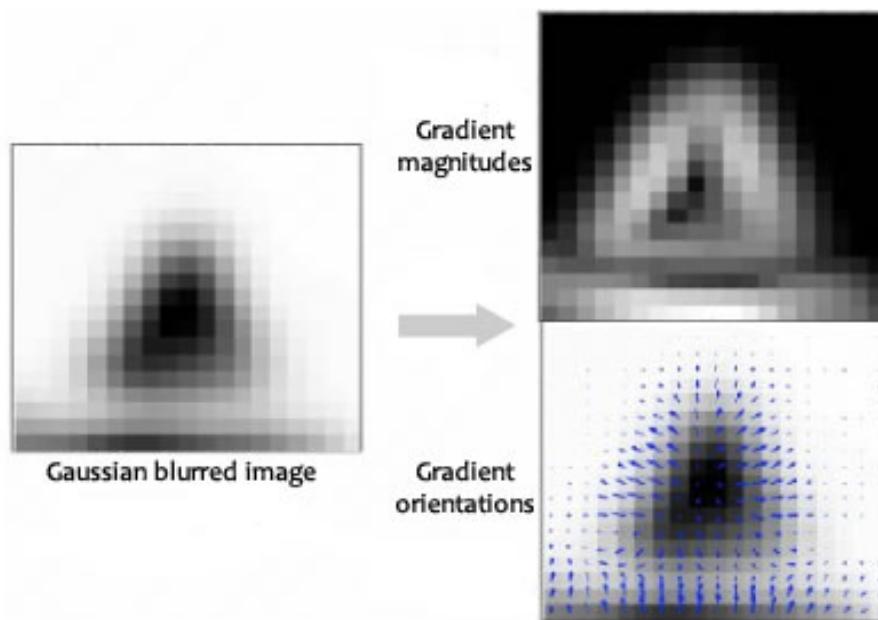
$$\theta(x, y) = atan\left(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)}\right) \quad (2.18)$$

W oparciu o rozkład gradientów wokół punktu charakterystycznego budowany jest histogram reprezentujący orientacje gradientów. Histogram ten jest podzielony 36 słupków o szerokości 10° pokrywający pełny zakres 360° . Najwyższy pik wykresu odpowiada orientacji jaka zostanie przypisana punktowi charakterystycznemu. W sytuacji gdy inny słupek osiągnął wartość powyżej 80% wartości maksymalnej, jego orientacja również zostaje odnotowana poprzez utworzenie dodatkowego bliźniaczego punktu, różniącego się orientacją. Według badań [12] tylko około 15% posiada wielokierunkową orientację, aczkolwiek jej uwzględnienie znaczaco poprawia jakość dopasowań.



Rysunek 2.8 Badanie otoczenia punktu charakterystycznego - rozmycie

{fig:sift_a}



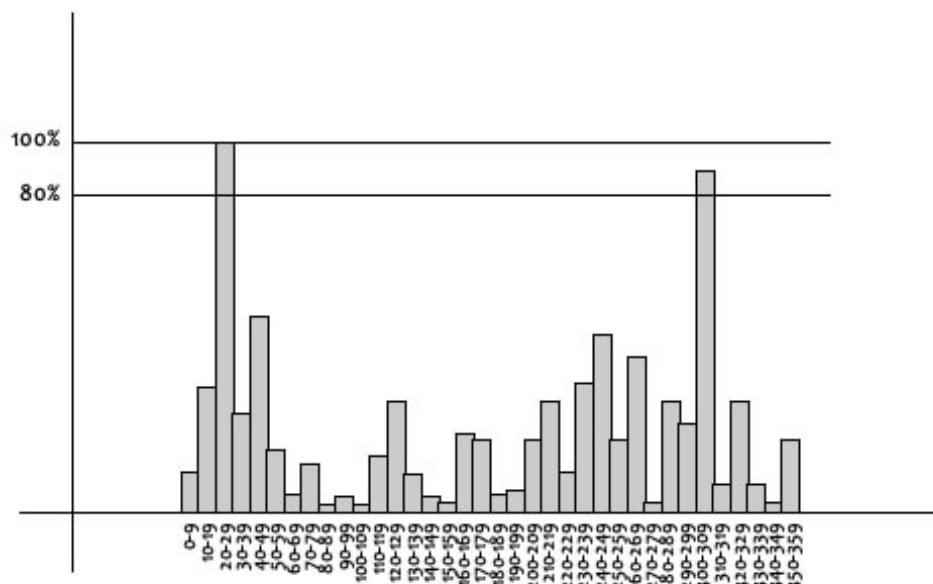
Rysunek 2.9 Badanie otoczenia punktu charakterystycznego - badanie gradientów

{fig:sift_a}

2.1.4 Budowa deskryptora

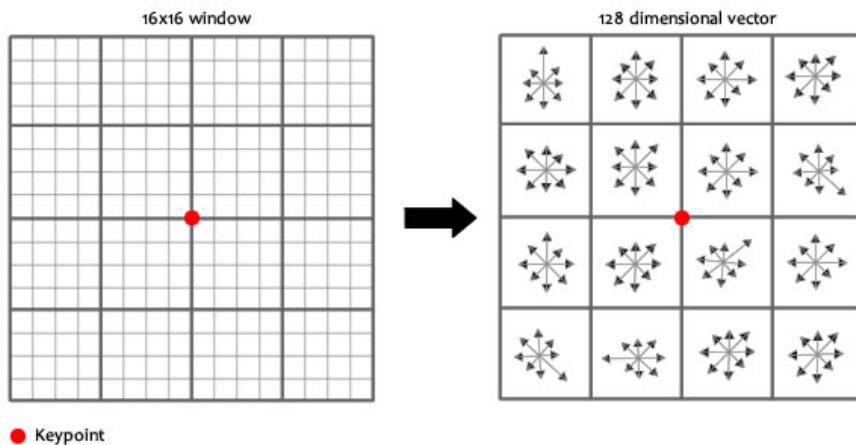
Finalny etapem algorytmu SIFT jest wyliczenie unikalnego deskryptora punktu charakterystycznego. Wokół punktu charakterystycznego badane jest okno o rozmiarze 16 na 16 pikseli podzielone na 16 sektorów. Sektor stanowi kwadrat o boku 4 pikseli. Dla każdego pojedynczego sektora obliczane są wielkości i kierunki gradientów, które zostają zapisane za pomocą 8 wartościowego histogramu. Orientacje gradientów wyrażane są jako wielokrotności kąta 45° .

Po obliczeniu 16 histogramów, cała tablica sektorów zostaje przeskalowana. Spowodowane to jest tym, że okna leżące bliżej punktu charakterystycznego są istotniejsze i należy im przypisać większą wagę. Analogicznie zmniejszany jest wpływ wartości znajdujących



Rysunek 2.10 Histogram reprezentujący orientacje gradientów wokół punktu charakterystycznego

{fig:sift_a}



Rysunek 2.11 Tworzenie deskryptora punktu charakterystycznego - lokalna siatka histogramów

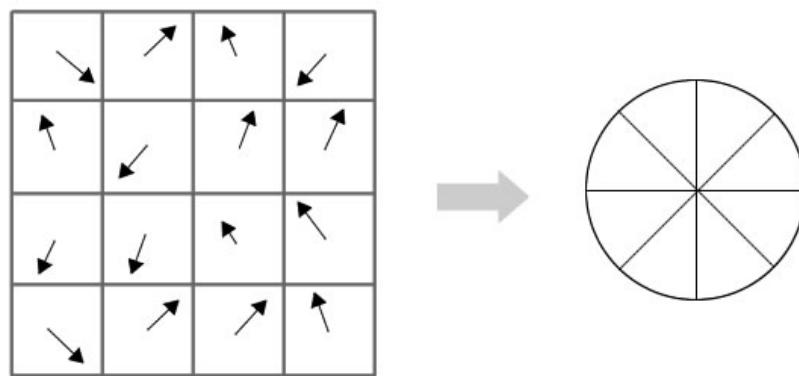
{fig:sift_a}

się brzegach badanego obszaru.

W wyniku powyższych operacji otrzymujemy wektor 128 liczb (8-kierunkowe histogramy zgrupowane w 4 wierszach po 4 kolumny). Wektor ten jest poddawany normalizacji.

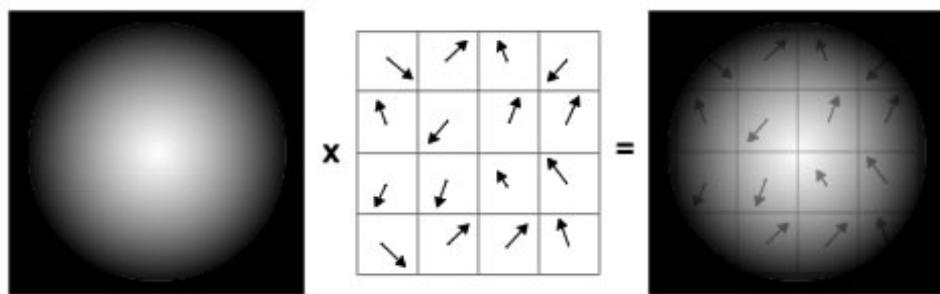
Następnie aby uzyskać niezależność od rotacji, od wektora jest odejmowana orientacja punktu charakterystycznego obliczona w poprzednim etapie. Aby uzyskać niezależność od oświetlenia współczynniki znormalizowanego wektora, są poddawane półprogowaniu z parametrem 0,2. W sytuacji gdy jakiś współczynnik jest większy od zadanego progu, przypisywana mu jest wartość 0,2 po czym cały wektor poddawany jest ponownej normalizacji.

W wyniku całego procesu otrzymujemy 128 wymiarowy deskryptor punktu charakterystycznego.



Rysunek 2.12 Tworzenie deskryptora punktu charakterystycznego - lokalna siatka histogramów

{fig:sift_a}



Rysunek 2.13 Siatka histogramów uwzględniająca odległość od punktu charakterystycznego

{fig:sift_a}

2.2 Algorytm SURF

Algorytm SURF (skrót od Speeded Up Robust Features) został po raz pierwszy zaprezentowany w 2006 roku w pracy doktorskiej Herberta Baya [3]. Pełny jego opis można znaleźć również w pracy[4]. Wykorzystywany jest on w zadaniach rozpoznawania obrazów oraz w rekonstrukcji obiektów trójwymiarowych. Algorytm częściowo jest inspirowany algorytmem SIFT i podobnie jak on aby usprawnić obliczenia korzysta z aproksymacji skomplikowanych operatorów.

W jego działaniu możemy wyróżnić następujące etapy:

1. Obliczenie obrazów całkowych.
2. Lokalizowanie punktów charakterystycznych.
3. Konstruowanie 64 wymiarowego deskryptora punktu charakterystycznego.

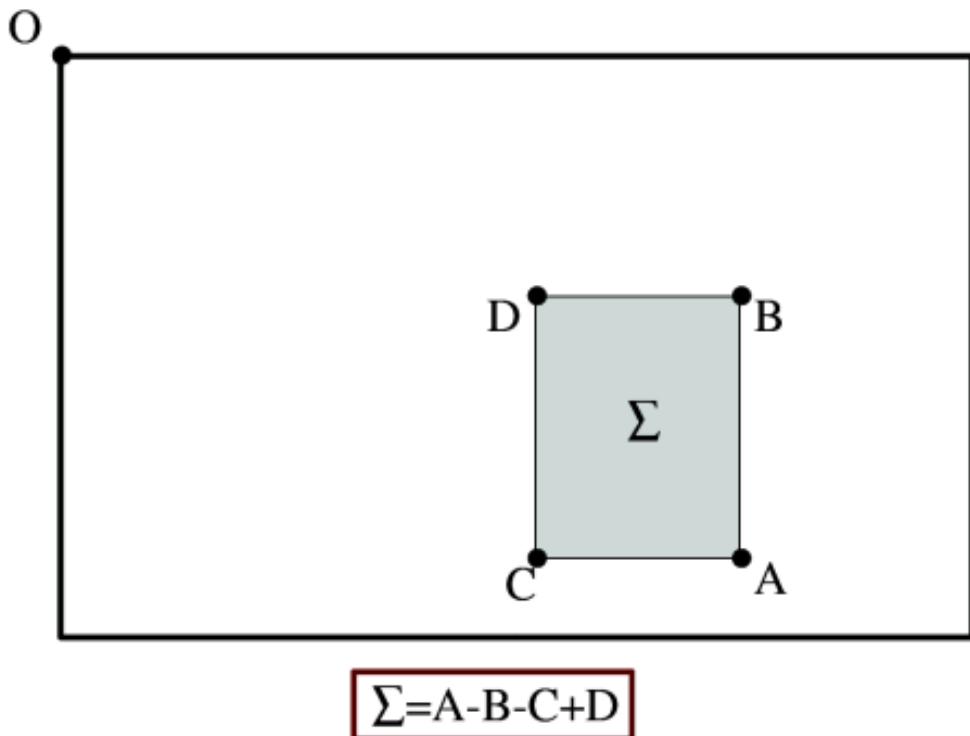
SURF podobnie jak algorytm SIFT jest chroniony patentem amerykańskim.

2.2.1 Obliczenie obrazów całkowych

Działanie algorytmu SURF rozpoczyna się od przygotowania obrazów całkowych. Są one wyliczane wg. wzoru 2.19.

$$I_{\sum}(x, y) = \sum_{x'=0}^x \sum_{y'=0}^y I(x', y') \quad (2.19) \quad \text{eqn:surf_i}$$

Przygotowanie reprezentacji badanego obrazu w postaci całkowej pozwala, zredukować obliczenia intensywności dowolnego fragmentu obrazu do trzech prostych operacji arytmetycznych o stałym czasie wykonania. Ma to kluczowe znaczenie w przypadku analizy dużych fragmentów obrazów.



Rysunek 2.14 Obliczanie intensywności wybranego obszaru w oparciu o obrazy całkowe {fig:surf_b}

2.2.2 Lokalizowanie punktów charakterystycznych

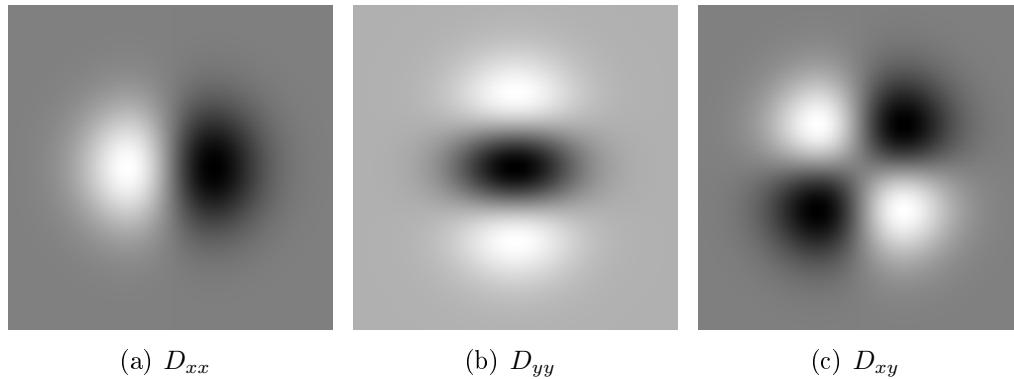
Wykrywanie punktów charakterystycznych za pomocą macierzy Hesjanu

Algorytm do wykrywania obszarów charakterystycznych wykorzystuje macierz Hesjanu. Obszary te są lokowane w miejscach, w których wyznacznik macierzy osiąga lokalne maksimum.

$$\mathbf{H}(x, y, \sigma) = \begin{bmatrix} L_{xx}(x, y, \sigma) & L_{xy}(x, y, \sigma) \\ L_{xy}(x, y, \sigma) & L_{yy}(y, y, \sigma) \end{bmatrix} \quad (2.20)$$

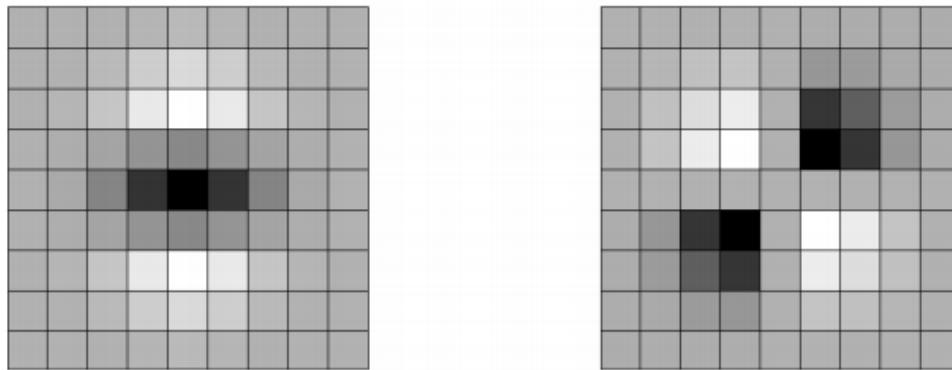
Wyrażenie $L_{xx}(x, y, \sigma)$ to obraz splotu pochodnej cząstkowej drugiego stopnia funkcji rozmycia Gaussa ($D_{xx} = \frac{\partial^2}{\partial x^2} \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$) i obrazu I w punkcie (x,y). Analogicznie $L_{yy}(x, y, \sigma)$ i $L_{xz}(x, y, \sigma)$. Przedstawia je rysunek 2.15

W praktyce stosuję się powyższe jądra w dyskretnej formie co przedstawia rysunek 2.16. Skutkuje to pogorszeniem jakości pracy operatorów w sytuacji gdy obraz jest obrócony o nieparzystą wielokrotność kąta 45° i może prowadzić do pojawienia się artefaktów.



Rysunek 2.15 Pochodne cząstkowe drugiego stopnia funkcji rozmycia Gaussa

{pict:surf_2}

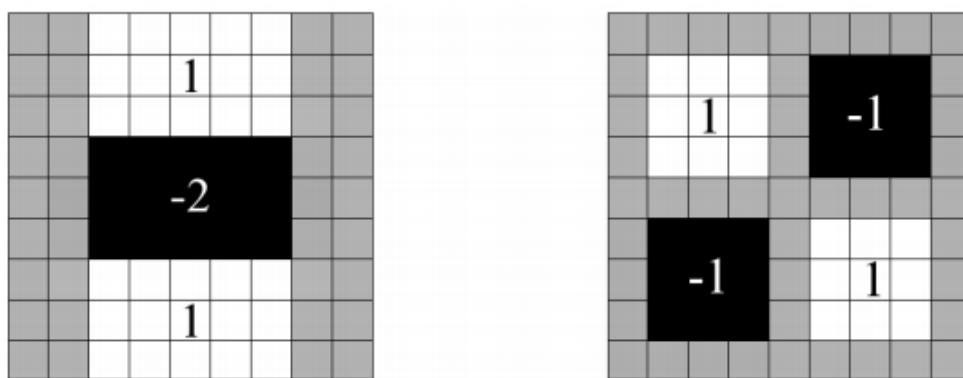


Rysunek 2.16 Dyskretne pochodne cząstkowe drugiego stopnia Gaussianów

{fig:surf_b}

Aproksymacja jąder rozmycia

Bay podobnie jak Lowe w algorytmie SIFT zastosował podejście polegające na dużym uproszczeniu realizowanych operacji. Jądra rozmycia zostały w radykalny sposób aproksymowane [rysunek 2.17].



Rysunek 2.17 Aproksymacje pochodnych cząstkowych drugiego stopnia Gaussianów

{fig:surf_k}

W związku z zastosowaniem aproksymowanych okien rozmycia wyrażenie na wyznacznik Hesjanu przyjmuje następującą postać:

$$\det(\mathbf{H}^{(a)}) = L_{xx}^{(a)} L_{yy}^{(a)} - (\omega L_{xy}^{(a)})^2 \quad (2.21) \quad \text{eqn:det_ap}$$

Gdzie $L_{xx}^{(a)}$, $L_{yy}^{(a)}$, $L_{xy}^{(a)}$ to sploty obrazu I z aproksymowanymi oknami rozmycia, a współczynnik ω to waga potrzebna zachowania energii jąder Gaussowskiego przy korzystaniu z ich

uproszczonej postaci. Obliczana jest ona według wyrażenia 2.22, gdzie $\|\cdot\|_F$ oznacza normę Frobeniusa.

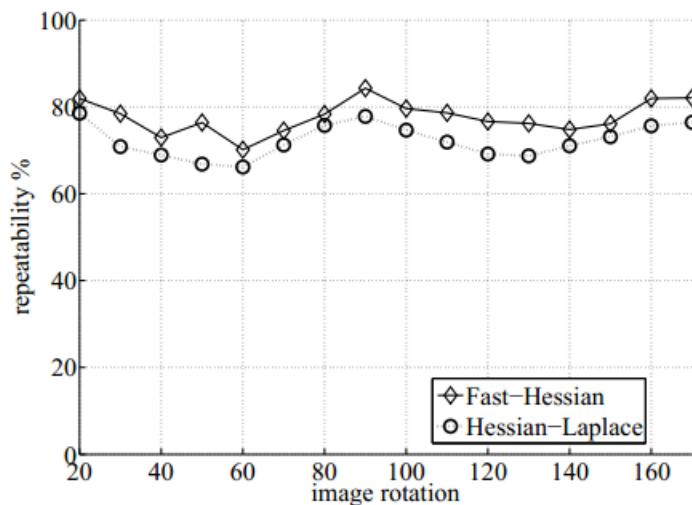
$$\omega = \frac{\|L_{xy}(\sigma)\|_F \|D_{yy}(rozmiar - okna)\|_F}{\|L_{yy}(\sigma)\|_F \|D_{xy}(rozmiar - okna)\|_F} \quad (2.22) \quad \{eqn:frobo\}$$

Dla okna rozmycia $\sigma = 1.2$ o wymiarach 9×9 współczynnik ω wynosi:

$$\omega = \frac{\|L_{xy}(1.2)\|_F \|D_{yy}(9)\|_F}{\|L_{1.2}(\sigma)\|_F \|D_{xy}(9)\|_F} = 0.9127 \simeq 0.9 \quad (2.23)$$

Teoretycznie każdemu rozmyciu i rozmiarowi okna odpowiada inna wartość wagi ω . Praktyka jednak pokazuje, że dla usprawnienia pracy algorytmu możemy ω traktować jaką stałą, bez dużej straty dokładności.

W sposób znaczny przyspiesza to obliczenia, a jak wykazały badania [3] nie tylko nie pogarsza to wyników, a wręcz je polepsza [rysunek 2.18].



Rysunek 2.18 Porównanie działania operatorów dokładnych i aproksymowanych w zależności od kąta rotacji obrazu

{fig:surf_hess}

Skalo-przestrzenna reprezentacja obrazu

Lokalizowanie punktów kluczowych odbywa się w różnych skalach obrazu. Obrazy skalo-przestrzenne są najczęściej reprezentowane w postaci piramidy obrazów. Kolejne obrazy są poddawane progresywnemu rozmyciu oraz podpróbkowaniu, tworząc oktawy.

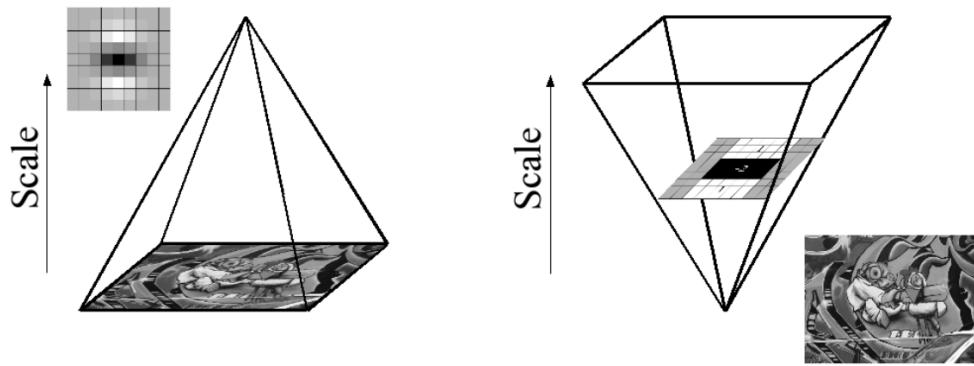
Dzięki zastosowaniu obrazów całkowych i błyskawicznie skalowalnych okien rozmycia, operacje skalo-przestrzenne dają się w łatwy sposób zrównoleglić.

Skalowanie jąder rozmycia

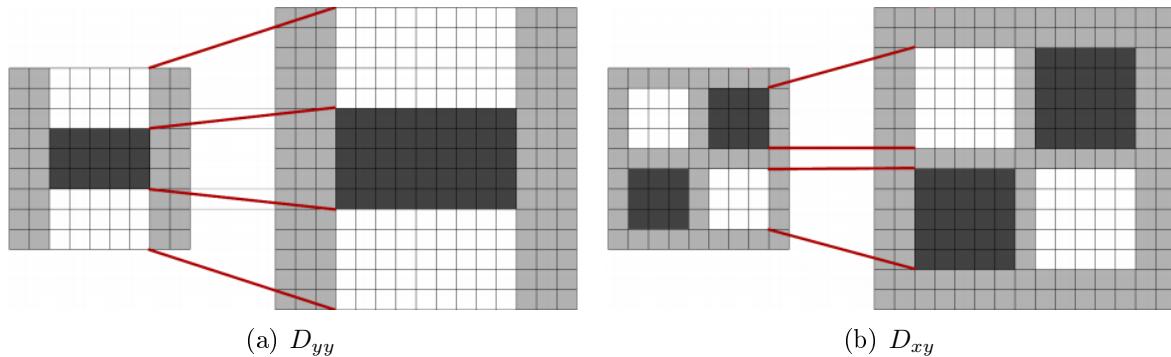
Każde okno rozmycia jest kwadratem o boku $3l$ które można opisać za pomocą następujących wyrażeń:

$$l = 2^o i + 1 \quad (2.24)$$

$$(2.25) \quad \{eqn:liczym\}$$



Rysunek 2.19 Piramida skalo-przestrzenna



Rysunek 2.20 Przeskalowane okna rozmycia

gdzie:

o - numer oktawy $o = 1, 2, 3, 4, \dots$

i - stopień rozmycia w oktawie $i = 1, 2, 3, 4$

W związku z powyższym wzór na odchylenie standardowe rozmycia Gaussa σ możemy zapisać:

$$\sigma = \frac{1.2}{3}l \quad (2.26)$$

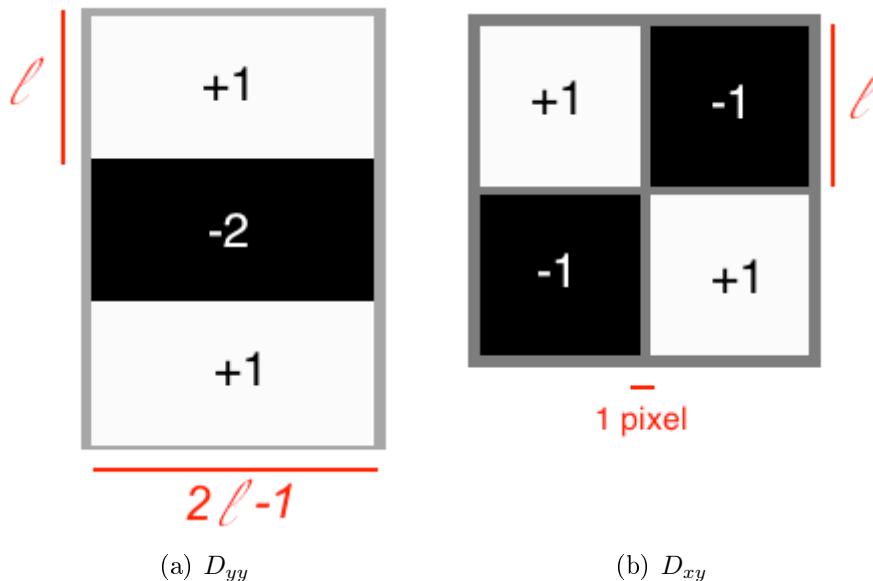
(2.27) {eqn:sigma_

$$D_{xx} = \begin{cases} -2 & (x, y) \in [-\frac{l}{2}, \frac{l}{2}] \times]-l, l[\\ +1 & (x, y) \in [-\frac{l}{2}, \frac{l}{2}] \times]-l, l[\times [-\frac{3l}{2}, \frac{3l}{2}] \setminus]-l, l[\\ 0 & reszta \end{cases} \quad (2.28)$$

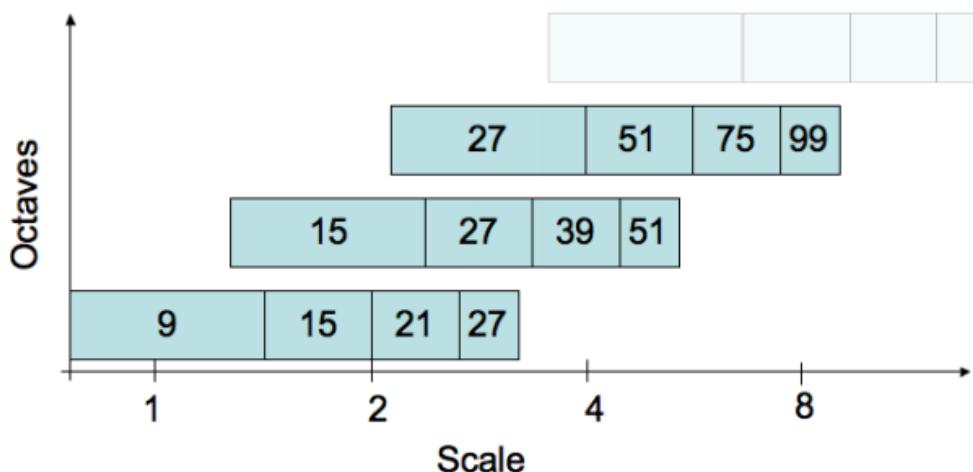
$$D_{yy} = \begin{cases} -2 & (x, y) \in]-l, l[\times [-\frac{l}{2}, \frac{l}{2}] \\ +1 & (x, y) \in]-l, l[\times [-\frac{3l}{2}, \frac{3l}{2}] \setminus]-l, l[\times [-\frac{l}{2}, \frac{l}{2}] \\ 0 & reszta \end{cases} \quad (2.29)$$

$$D_{xy} = \begin{cases} +1 & (x, y) \in]0, -l] \times]0, +l] \cap]0, +l] \times [-l, 0[\\ -1 & (x, y) \in [-l, 0[\times [-l, 0[\cap]0, +l] \times]0, +l] \\ 0 & reszta \end{cases} \quad (2.30)$$

$$\sigma^2 = \frac{L^2 - 1}{12} \quad (2.31) \quad \text{\{eqn:sigma_}$$



Rysunek 2.21 Okna rozmycia



Rysunek 2.22 Rozmiar okna filtru w zależności od oktawy i stopnia rozmycia

Tabela 2.1 przedstawia charakterystyczne parametry filtrów. Zostały w niej przedstawione właściwości dla 4 pierwszych oktaf. Algorytm zakłada możliwość analizowania większej ilości oktaf. Ich parametry są obliczane w analogiczny sposób, dopóki okno filtru nie przekracza rozmiaru obrazu. W praktyce jednak badania najczęściej ogranicza się do 4 pierwszych oktaf, gdyż ze wzrostem skali znaczająco spada liczba lokalizowanych punktów charakterystycznych [wykres 2.23].

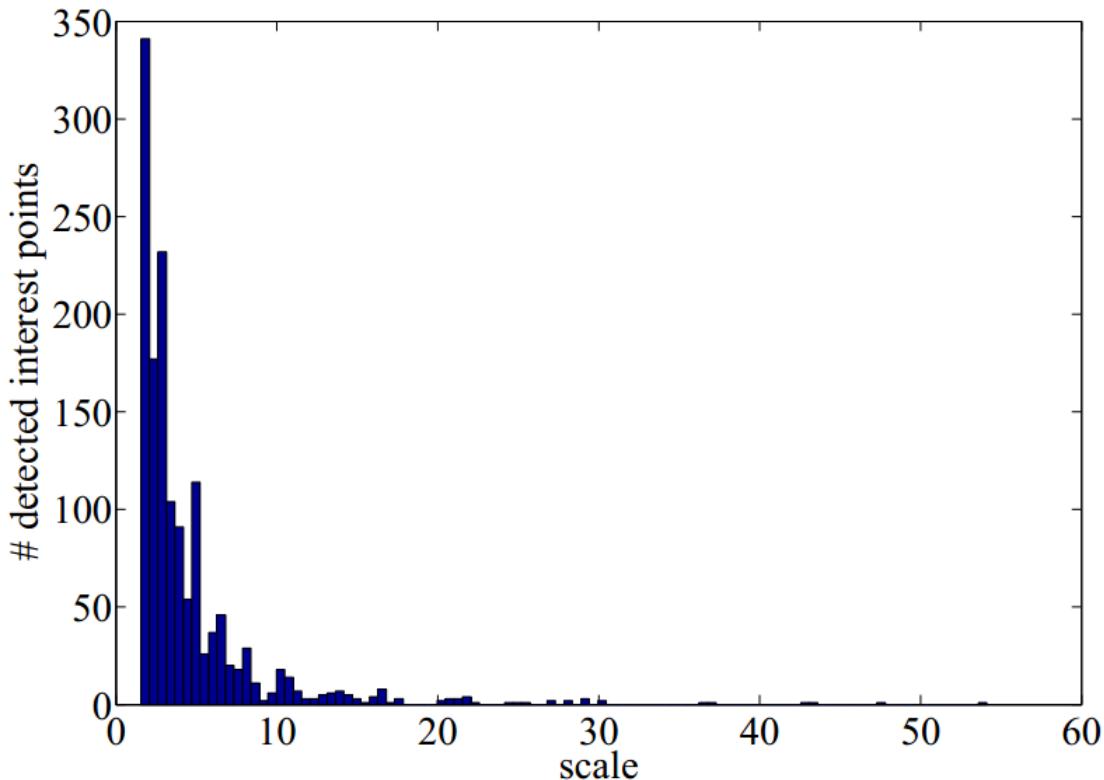
2.2.3 Określenie orientacji punktów charakterystycznych

Do określenia orientacji punktu charakterystycznego wykorzystywana jest falka Haara. Dla pikseli znajdujących się w otoczeniu punktu charakterystycznego liczone są odpowiedzi falki Haara w kierunku poziomym i pionowym. Poprzez otoczenie rozumiany jest kołowy obszar o promieniu 6σ , gdzie σ odpowiada skali w jakiej został zlokalizowany punkt. Podobnie rozmiar falki zależy od skali. Do jej liczenia wykorzystujemy okno o boku 4σ . Dzięki zastosowaniu przygotowanych wcześniej obrazów całkowych liczba operacji

σ	l	L	λ	ω
1	1	1,2	3	4,275
	2	2,0	5	7,000
	3	2,8	7	9,751
	4	3,6	9	12,511
2	1	2,0	5	8,000
	2	3,6	9	12,510
	3	5,2	13	18,041
	4	6,8	17	23,577
3	1	3,6	9	12,510
	2	6,8	17	23,577
	3	10,0	25	34,655
	4	13,2	33	45,737
4	1	6,8	17	23,577
	2	13,2	33	45,737
	3	19,6	49	67,903
	4	26,0	65	90,072
				0,9962

Tabela 2.1 Wartości charakterystyczne dla okien aproksymowanych filtrów rozmycia

{tab:surf_k}



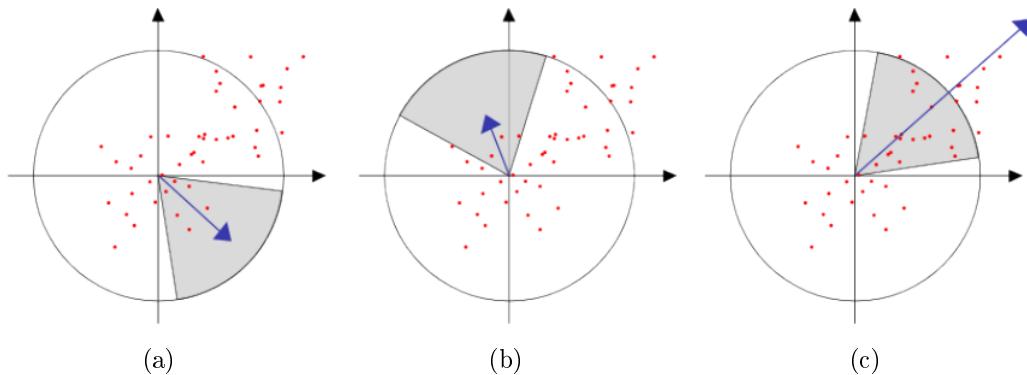
Rysunek 2.23 Ilość lokalizowanych punktów charakterystycznych w zależności od skali

{fig:surf_k}

potrzebnych do policzenia odpowiedzi w jednym kierunku została zredukowana do 6.

Odpowiedzi falek są reprezentowane na układzie współrzędnych. Oś OX reprezentuje odpowiedzi poziome, OY pionowe, a punkt przecięcia stanowi punkt charakterystyczny. Następnie wykorzystując obracające się wokół początku układu współrzędnych okno będące wycinkiem koła o kącie $\frac{\pi}{3}$. Odpowiedzi znajdujące się wewnątrz okna są sumowane. Suma odpowiedzi stanowi długość wektora związanego z kątem w jakim znajduje się

okno. Kąt orientacji punktu charakterystycznego jest kątem w którym długość wektora sum odpowiedzi jest największa [rys.2.24(c)].



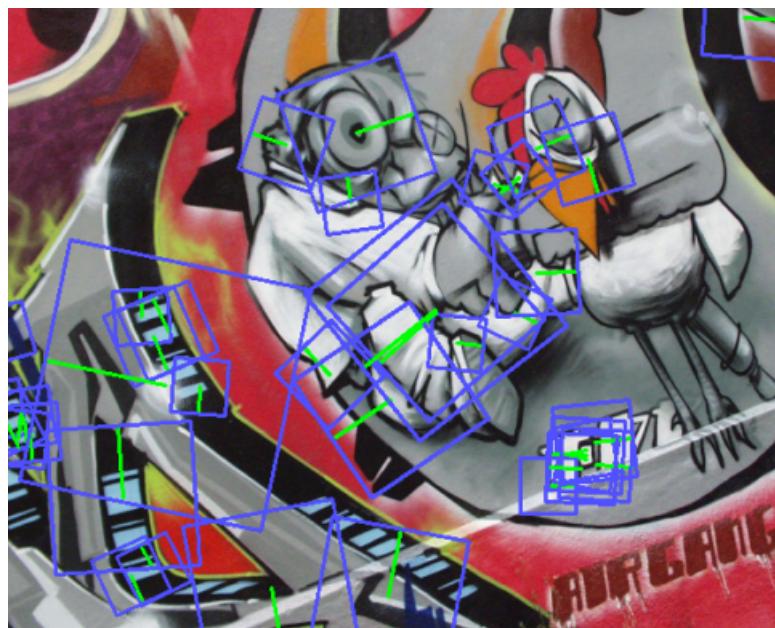
Rysunek 2.24 Badanie orientacji punktu charakterystycznego

{fig:surf_b}

2.2.4 Budowa deskryptora

Podobnie jak algorytm SIFT, SURF do budowy deskryptora wykorzystuje informacje o otoczeniu punktu charakterystycznego. Wokół badanego punktu analizowany jest kwadratowy obszar o boku 20σ .

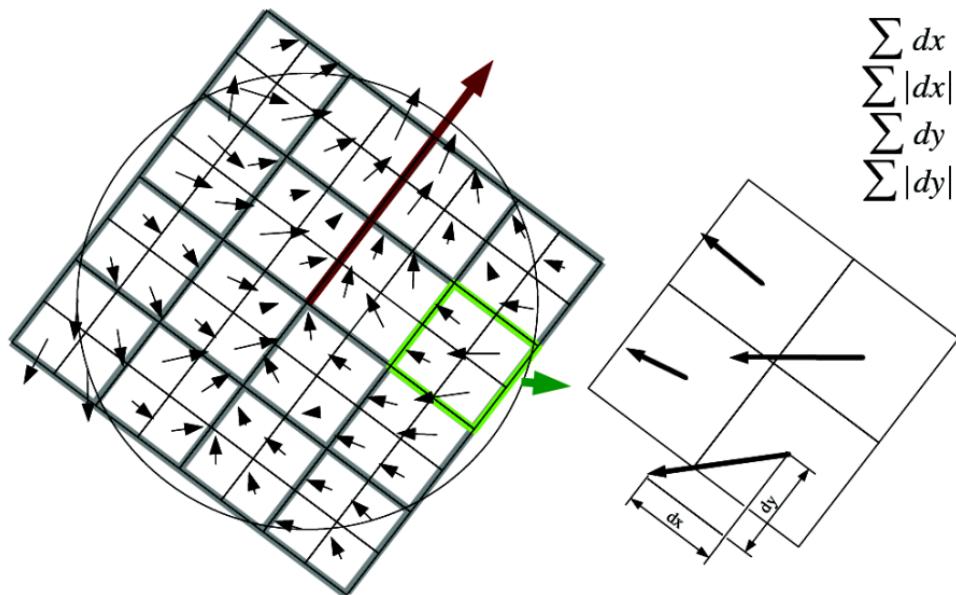
Okno to jest zorientowane zgodnie z kierunkiem wyznaczonym w poprzednim kroku i podzielone na 16 regularnych subregionów. Każdy subregion jest w równych odstępach próbkiowany w efekcie czego otrzymujemy zbiór 5×5 pikseli. Dla wycinka reprezentującego subregion liczona jest falka Haara w kierunku równoległym i normalnym do orientacji punktu charakterystycznego. Falka ta ma rozmiar 2σ . W praktyce aby zwiększyć szybkość działania algorytmu falki są liczone dla nie obróconego obrazu, a następnie ich odpowiedzi są interpolowane do kierunku rotacji. W celu minimalizacji błędów spowodowanych geo-



Rysunek 2.25 Badane obszary charakterystyczne

{fig:surf_k}

metrycznymi deformacjami odpowiedzi falek d_x i d_y są normalizowane rozkładem Gaussa z rozkładem równym 3.3σ (gdzie σ oznacza skale w jakiej badany jest obszar).

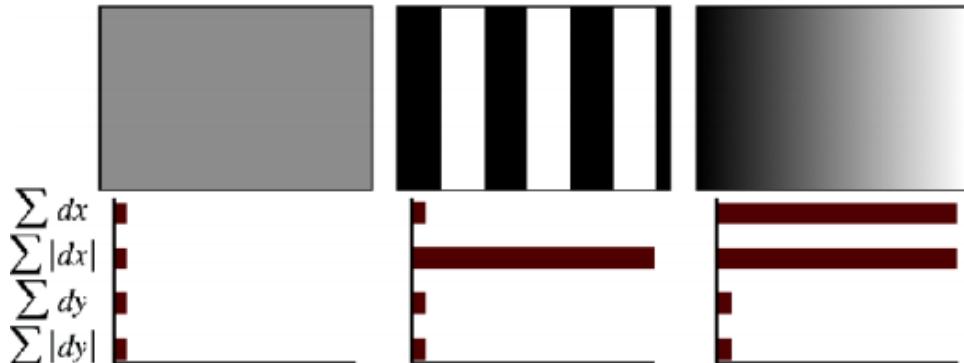


Rysunek 2.26 Badanie otoczenia punktu charakterystycznego

{fig:surf_b}

Falki w obrębie jednego subregionu są sumowane. Ponadto sumowaniu poddane są wartości bezwzględne falek. Zabieg ten ma na celu pozyskanie informacji o biegumowości intensywności zmian. W wyniku tych działań otrzymujemy 4 wymiarowy wektor opisujący subregion.

$$v_{subregion} = [\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|] \quad (2.32)$$

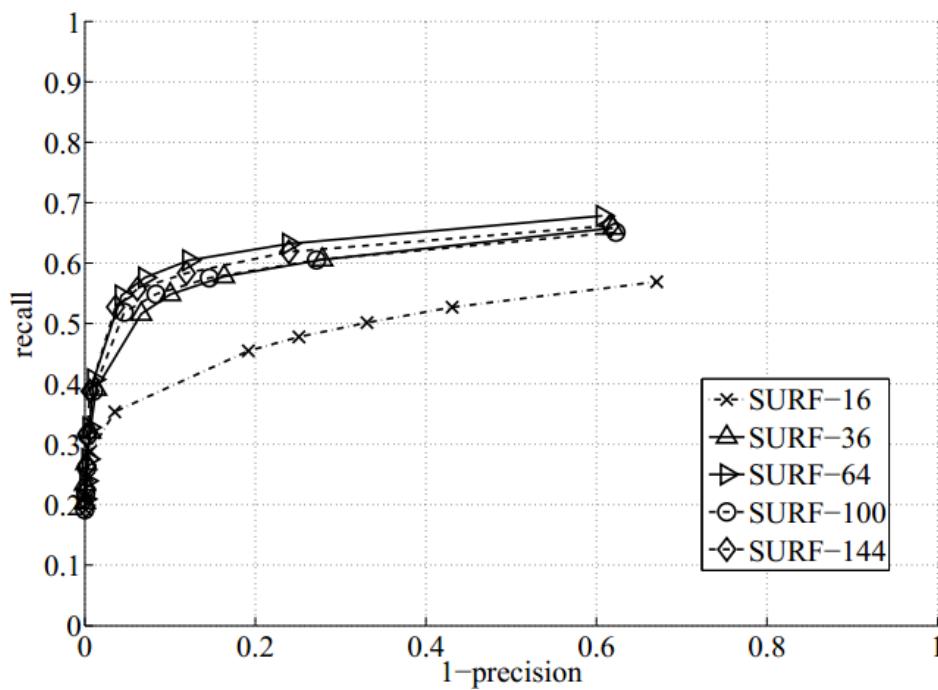


Rysunek 2.27 Przykłady wektorów opisujących subregiony w zależności od natury obszaru charakterystycznego

{fig:surf_b}

W skład deskryptora punktu charakterystycznego wchodzi 16 subregionów w efekcie czego otrzymujemy 64 wymiarowy wektor punktu charakterystycznego. Algorytm posiada wariancje różniące się rozdzielcością na jaką jest podzielony badany obszar charakterystyczny. Badania [4] wykazują jednak 64 wymiarowy deskryptor jest rozwiązaniem najlepszym i optymalnym [2.28]. W sytuacji, w której lekki spadek jakości jest akceptowalny na rzecz znaczającej poprawy czasu dopasowywania punktów charakterystycznych, wartym rozważenia jest deskryptor SURF-36 ($3 \times 3 \times 4$)

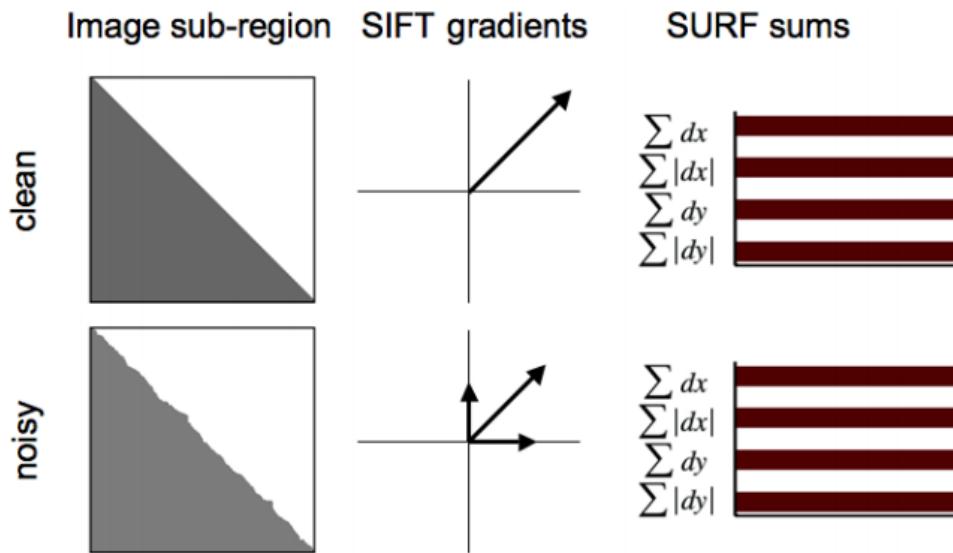
Otrzymany deskryptor dzięki zastosowaniu falek jest niezależny od rotacji, skali, zmiennego oświetlenia. Po normalizacji do wektora jednostkowego uzyskuje ponadto niezależność od kontrastu. Co więcej dzięki zastosowaniu całkowania informacji gradientach w



Rysunek 2.28 Porównanie deskryptorów SURF

{fig:surf_b}

subregionach, algorytm SURF jest również bardziej odporny na zaszumienie obrazu niż algorytm SIFT [rys. 2.29].



Rysunek 2.29 Wpływ szumu na wygląd deskryptorów subregionów algorytmów SIFT i SURF

{fig:surf_b}

2.3 Algorytm STAR

Algorytm STAR to zoptymalizowany algorytm CenSurE (skr. Center Surround Extremas) [2] rozwijany w laboratorium robotyki Willow Garage. Stanowi rozwinięcie podejścia zastosowanego w algorytmie SURF.

2.3.1 Obliczenie pochyłych obrazów całkowych

Algorytm STAR podobnie jak SURF korzysta z obrazów całkowych. Raz obliczona reprezentacja obrazu może być wielokrotnie wykorzystywana w trakcie analizy. W sposób znaczący przyśpiesza to prace algorytmu i zapewnia stały czas wykonywania operacji niezależnie od rozmiaru badanego obszaru. Ze względu na użycie w algorytmie STAR filtrów o skomplikowanych kształtach modyfikacji uległa funkcja obliczająca obrazy. Zamiast stosowanego w SURF wzoru:

$$I_{\sum}(x, y) = \sum_{x'=0}^x \sum_{y'=0}^y I(x', y') \quad (2.33) \quad \{eqn:star_i\}$$

obrazy całkowe w algorytmie STAR obliczane są według następującej formuły:

$$I_{\sum \alpha}(x, y) = \sum_{y'=0}^y \sum_{x'=0}^{x+\alpha(y-y')} I(x', y') \quad (2.34) \quad \{eqn:star_i\}$$

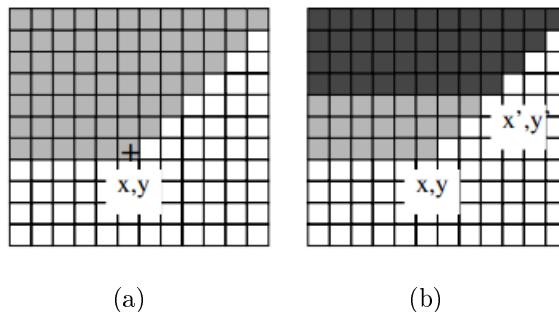
gdzie α jest stałą zależną od rodzaju stosowanego filtra. Gdy $\alpha = 0$ wzór 2.34 przyjmuje postać wzoru 2.33. Gdy $\alpha > 0$ obszar obrazu całkowego jest pochylony w prawo (jak na rysunku 2.30), a gdy $\alpha < 0$ w lewo.

2.3.2 Lokalizowanie punktów charakterystycznych

Do lokalizowania punktów charakterystycznych algorytm STAR wykorzystuje dwustopniowe filtry otoczeniowo-centryczne, stanowiące łatwą i szybką w obliczeniach aproksymacje Laplasjanu. W STAR zastosowano inspirowany falkami Haara dwustopniowy filtr otoczeniowo-centryczny [rys. 2.32].

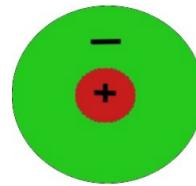
Dwustopniowe filtry otoczeniowo-centryczne

W ramach prac nad algorymem CenSurE przetestowano szereg dwustopniowych filtrów począwszy od kołowego operatora LoG po dwustopniowy filtr kwadratowy. Filtr kołowy



Rysunek 2.30 Przykład zastosowania pochyłych obrazów całkowych dla trapezoidalnego obszaru

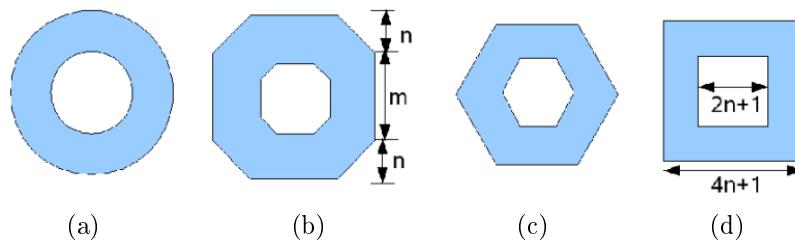
{fig:cs_trapez}



Rysunek 2.31 Schemat dwustopniowego filtru otoczeniowo-centrycznego

{fig:wg_haa}

jest najwierniejszą reprezentacją Laplasjanu, aczkolwiek jest najbardziej złożony obliczeniowo. Bardzo szybkie filtry kwadratowe są z kolei podatne na rotacje obrazu i ich efektywność spada przy obrotach o nieparzyste wielokrotności kąta 45° . Odpowiedni filtr powinien być kompromisem między liczbą stopni symetrii i złożonością obliczeniową. W algorytmie CenSurE zastosowano filtr ośmiokątny [rys. 2.32(b)] dający się łatwo obliczać dzięki zastosowaniu pochyłych obrazów całkowych. Optymalizując algorytm CenSurE wypracowano rozwiązanie złożone 2 obróconych względem siebie o kąt 45° kwadratów [rys. 2.33] tworzących gwiazdę - skąd nazwa algorytm STAR (w tłumaczeniu z angielskiego "gwiazda").



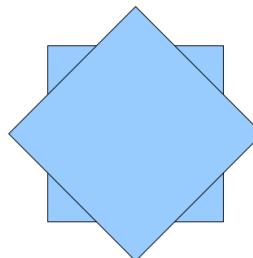
Rysunek 2.32 Dwustopniowe filtry otoczeniowo-centryczne

{fig:star_b}

Zastosowanie 2 okien o jednakowej powierzchni wyeliminowało problem równoważenia odpowiedzi dodatnich i ujemnych, oraz znacznie uprościło obliczenia. Podobnie jak w algorytmie SURF aby zwiększyć skale w jakiej lokalizujemy punkt charakterystyczny należy zwiększyć rozmiar filtra.

Ekstrema jako potencjalne punkty charakterystyczne

Dla pojedynczego piksela obrazu jest liczone 7 odpowiedzi falek otoczeniowo-centrycznych o różnych rozmiarach. Każda falka reprezentuje obraz w innej skali. Algorytm dopuszcza liczenie większej ilości filtrów, aczkolwiek do większości zastosowań 7 okien jest liczbą wystarczającą.



Rysunek 2.33 Dwustopniowy filtr otoczeniowo-centryczny wykorzystywany w algorytmie STAR

{fig:wg_cin}

Po obliczeniu odpowiedzi lokalizowane są lokalne ekstrema. Wartości falek są porównywane w pionowym i poziomym otoczeniu podobnie jak w omówionych wcześniej algorytmach. Jako potencjalne punkty charakterystyczne wybierane są piksele, które w swoim sąsiedztwie osiągają wartości ekstremalne.

2.3.3 Selekcja punktów charakterystycznych

Eliminacja słabych punktów

Wartość bezwzględna odpowiedzi filtru dostarcza informacji o jakości punktu charakterystycznego. Im wartość ta jest większa tym punkt jest bardziej powtarzalny. W celu poprawy stabilności lokalizowanych dokonuje się progowania i odrzuca punkty o małej wartości bezwzględnej.

Usuwanie punktów leżących na krawędziach

Jak zostało opisane przy omawianiu algorytmu SIFT aby wybrać najbardziej stabilne cechy obrazu ze zbioru punktów charakterystycznych należy usunąć punkty leżące na krawędziach. W odróżnieniu do algorytmów SIFT i SURF, w algorytmie STAR zamiast macierzy Hesjanu użyto miary Harrisza.

$$\mathbf{H} = \begin{bmatrix} \sum L_x^2 & \sum L_x L_y \\ \sum L_x L_y & \sum L_y^2 \end{bmatrix} \quad (2.35)$$

L_x i L_y są pochodnymi funkcji filtru w kierunkach x i y, a sumowanie odbywa się w obrębie okna w jakim punkt charakterystyczny został zlokalizowany.

Dalsza część postępowanie jest analogiczna jak we wspomnianym algorytmie SIFT. Badany jest wyznacznik i ślad macierzy na podstawie których określa się współczynnik głównych krzywizn. W oparciu o obliczony współczynnik piksele poddawane są progowaniu [wzór 2.16]. Domyślna wartość progu wynosi 10.

2.3.4 Budowa deskryptora

Algorytm STAR jest detektorem punktów charakterystycznych. Nie posiada on własnego mechanizmu opisu cech. Aby móc go stosować należy go połączyć, z algorytmem deskryptycznym takim jak np. deskryptor SIFT lub SURF. W raporcie dotyczącym badań nad algorytmem CenSurE autorzy posługują się zmodyfikowanym deskryptorem SURF.

2.4 Algorytm FAST

Algorytm FAST (skr. Features from Accelerated Segment Test) został opublikowany przez Edwarda Rostena i Toma Drummonda w 2006 roku [16] [17]. Algorytm ten służy do lokalizowania rogów w obrazie, które mogą być wykorzystane do zadań rozpoznawania i śledzenia.

W odróżnieniu do wcześniej omawianych algorytmów FAST nie dokonuje "globalnej" analizy obrazu w poszukiwaniu cech charakterystycznych. Wykorzystując algorytmy uczenia maszynowego bada małe wycinki obrazu.

FAST nie dokonuje wieloskalowego poszukiwania punktów charakterystycznych. Chcąc osiągnąć niezależność od skali z oryginalnego obrazu należy wygenerować piramidę skalową, czyli zbiór obrazów reprezentujących widoki w różnych skalach.

2.4.1 Zasady egzaminowania

Jak wspomniano we wstępnie, algorytm FAST należy do grupy algorytmów badających indywidualnie segmenty obrazu. Segment stanowi punkt p i utworzony wokół niego okrąg składający się z 16 pikseli [rys. 2.34].

Punkt p jest uznawany jako róg jeżeli istnieje zbiór n sąsiadujących pikseli okręgu, które są jaśniejsze od intensywności punktu p powiększonej o wartość progu t ($I_p + t$), lub ciemniejsze od intensywności punktu p pomniejszonej o próg t ($I_p - t$).

Jako wartość n najczęściej przyjmuje się liczbę 12. W praktyce badanie punktu rozpoczyna się od zbadania 4 pikseli umieszczonego w głównych osiach punktu p . Dla rysunku 2.34 są to piksele 1,5,9,13. Jeżeli punkt p jest rogiem, wówczas co najmniej 3 z 4 egzaminowanych pikseli muszą być jaśniejsze niż $I_p + t$ lub ciemniejsze niż $I_p - t$. Test ten pozwala w szybki sposób wykluczyć dużą liczbę punktów. Dla pozostałych punktów, które przeszły przez wstępny proces selekcji, badane są wszystkie piksele w otaczającym je segmencie.

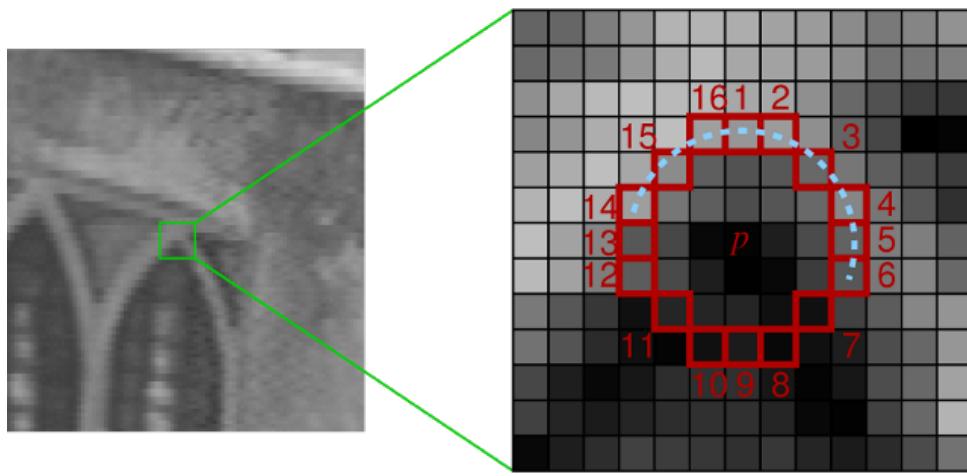
Przytoczona metoda cechuje się dużą wydajnością, aczkolwiek jej autorzy zwracają uwagę na następujące wady:

1. Aby móc skutecznie zastosować szybki test wstępny 4 pikseli n musi wynosić 12.
2. Wybór i kolejność testowania 4 pikseli w szybkim teście wpływa na rozkład lokalizowania punktów. Wada ta nabiera znaczenia przy zastosowaniach w czasie rzeczywistym.
3. Informacje uzyskane w szybkim teście 4 pikseli są niewykorzystywane przy dokładnym badaniu punktu.
4. Metoda może lokalizować wiele punków stykających się ze sobą.

2.4.2 Uczenie maszynowe

Algorytm FAST w swoim działaniu wykorzystuje metody uczenia maszynowego. Proces ten jest podzielony na dwa etapy:

1. budowa detektora rogów
2. budowa drzewa decyzyjnego



Rysunek 2.34 Badanie otoczenia punktu potencjalnego rogu

{fig:fast_t

W pierwszym etapie na podstawie zbioru uczącego jest budowany detektor rogów. Algorytm bada wszystkie piksele wchodzące w skład obrazu uczącego z zastosowaniem testów segmentowych. Używane testy są wolne, gdyż na tym etapie egzaminowane są wszystkie 16 piksele wchodzące w skład pojedynczego segmentu.

Wszystkie piksele w okręgu $x \in 1, 2, 3, \dots, 16$ są klasyfikowane względem jasności punktu p . Mogą się one znajdować w 3 stanach:

$$S_{p \rightarrow x} = \begin{cases} d, & I_{p \rightarrow x} \leq I_p - t \quad (\text{ciemniejszy}) \\ s, & I_p - t < I_{p \rightarrow x} < I_p + t \quad (\text{podobny}) \\ b, & I_p + t \leq I_{p \rightarrow x} \quad (\text{jaśniejszy}) \end{cases} \quad (2.36)$$

Niech P będzie zbiorem wszystkich piksele w obrazach uczących. Analizowane piksele są grupowane w 3 zbiorach.

$$P_d = \{p \in P : S_{p \rightarrow x} = d\} \quad (2.37)$$

$$P_s = \{p \in P : S_{p \rightarrow x} = s\} \quad (2.38)$$

$$P_b = \{p \in P : S_{p \rightarrow x} = b\} \quad (2.39)$$

Innymi słowy w zbiorze P_d znajdują się wszystkie piksele x , które są ciemniejsze od piksela centralnego po uwzględnieniu wartości progu. Analogicznie przedstawiają się zbiory P_b i P_s .

W drugim etapie pracy algorytmu, na podstawie danych zebranych w zbiorze uczących, budowane jest drzewo decyzyjne pozwalające określić czy badany piksel jest rogiem. Wykorzystywany w tym celu jest algorytm uczenia maszynowego ID3 opisany w [14].

Niech K_p będzie zmienną logiczną będącą prawdą jeżeli punkt centralny segmentu p jest rogiem i fałszem w przeciwnym wypadku. Za pomocą algorytmu ID3 wybierany jest piksel x z okręgu, który w sposób najbardziej skuteczny informuje czy piksel centralny jest rogiem. Skuteczność jest mierzona za pomocą entropii K_p .

Entropia K_p dla zbioru pikseli P opisana jest wzorem:

$$\begin{aligned} H(P) &= (c + \bar{c})\log_2(c + \bar{c}) - c\log_2c - \bar{c}\log_2\bar{c} \\ \text{gdzie: } c &= |\{p | K_p \text{ jest prawdą}\}| \quad (\text{liczba rogów}) \\ \bar{c} &= |\{p | K_p \text{ jest fałszem}\}| \quad (\text{liczba nie rogów}) \end{aligned} \quad (2.40)$$

Wybór x odbywa się poprzez porównanie wzrostów informacji skojarzonych z pikselami okręgu. Wzrost informacji wyraża się wzorem:

$$H_g = H(P) - H(P_d) - H(P_s) - H(P_b) \quad (2.41)$$

Po wyselekcjonowaniu piksela x wnoszącego najwięcej informacji, cały proces jest powtarzany rekursively na wszystkie trzy podzbiory. Przykładowo jeżeli w pierwszej iteracji wybrano piksel x_b (jaśniejszy od piksela centralnego) wówczas zbiór P_b jest rozwijany w podzbiory $P_{b,b}$, $P_{b,s}$, $P_{b,d}$. Zbiory są rozwijane w sposób rekursywny dopóki dopóty entropia podzbioru nie wyniesie 0. Oznacza to wówczas, że wszystkie piksele p w danym podzbiorze mają tą samą wartość zmiennej K_p . Gdy entropia wszystkich podzbiorów wyniesie 0 oznacza to, że drzewo decyzyjne zostało dopasowane do zbioru uczącego. W sytuacji gdy sąsiadujące poddrzewa mają taką samą strukturę różniący je test logiczny będący wspólnym korzeniem jest usuwany.

Po zbudowaniu drzewa decyzyjnego klasyfikującego wszystkie rogi w zbiorze uczącym, jest ono konwertowane na kod C. Odbywa się to poprzez utworzenie długiego łańcucha znakowego (ang. "string") z zakodowanymi regułami postępowania "if-else", które po skompilowaniu są używane jako klasyfikator rogów.

W celu polepszenia szybkości działania algorytmu otrzymany klasyfikator jest poddawany dwukrotnej optymalizacji.

2.4.3 Selekcja punktów charakterystycznych

W sytuacji gdy obok siebie lokalizowanych jest wiele punktów należy zastosować tłumienie mniej wyraźnych rogów. Zagęszczanie obiektów nie wnoszących dodatkowych informacji może wpływać niekorzystnie na lokalizowanie cechy charakterystycznej obrazu.

Niestety badanie testem segmentowym nie dostarcza żadnej wartości funkcji odpowiadzi rogu. Z tego powodu po zlokalizowaniu punktu charakterystycznego należy go poddać dodatkowemu badaniu funkcją oceniającą V . Po zlokalizowaniu rogów i obliczeniu ich wartości funkcji V , ze zbioru punktów charakterystycznych są usuwane elementy w którym sąsiedztwie mają mają narożnik o większej wartości V . Zalecanym przez autora sąsiedztwem jest okno o rozmiarze 3 na 3 piksele.

Funkcja oceniająca V może być zdefiniowana na kilka sposobów:

- Największa wartość ciągłych pikseli okręgu n , dla której punkt p jest dalej rogiem.
- Największa wartość progu t , dla której punkt p jest dalej rogiem.
- Suma wartości absolutnych różnic między pikselami tworzącymi łuk dookoła punktu p i wartością piksela centralnego.

W przypadku definicji 1 i 2, wartości funkcji V mieszczą się w wąskim przedziale wartości dyskretnych. W wielu przypadkach sąsiadujące piksele mają takie same wartości V co powoduje, że takie kryteria są niewystarczające. W praktyce stosuje się zmodyfikowane kryterium 3, które po optymalizacji wygląda następująco:

$$V = \max \left(\sum_{x \in S_{jasne}} |I_{p \rightarrow x} - I_p| - t, \sum_{x \in S_{ciemne}} |I_p - I_{p \rightarrow x}| - t \right) \quad (2.42)$$

gdzie:

$$S_{jasne} = \{x | I_{p \rightarrow x} \geq I_p + t\} \quad (2.43)$$

$$S_{ciemne} = \{x | I_{p \rightarrow x} \leq I_p - t\} \quad (2.44)$$

2.4.4 Budowa deskryptora

Algorytm FAST rozwiązuje jedynie problem lokalizowania punktów charakterystycznych jakimi są rogi. Nie dostarcza on, żadnego rozwiązania opisującego punkty charakterystyczne. Aby móc w praktyce stosować algorytm należy go połączyć z innym algorytmem dostarczającym deskryptor. Mogą to być deskryptory gradientowe takie jak zastosowano w algorytmach SIFT, SURF lub też opisany w kolejnym podrozdziale algorytm BRIEF.

2.5 Algorytm BRIEF

Algorytm BRIEF (Binary Robust Independent Elementary Features) [13] [6] należy do grupy algorytmów deskrypcyjnych i jest wykorzystywany do opisywania punktów charakterystycznych. Nie służy lokalizowaniu cech charakterystycznych obrazów, dlatego nie został ujęty w zestawieniu przedstawionym we wstępie niniejszej pracy.

Może być on wykorzystywany jako alternatywa dla deskryptorów algorytmów takich jak SIFT czy SURF lub uzupełnienie dla algorytmów nie posiadających deskryptorów jak FAST lub STAR. Twórcy algorytmu szczególny nacisk położyli na szybkość obliczania deskryptorów i ich dopasowywania.

Popularne stosowane metody optymalizacyjne skupiają się na redukcji redundantnych informacji zawartych w deskryptorach. Można wyróżnić dwa podejścia do tego zagadnienia.

W pierwszym przypadku dzięki zastosowaniu metod statystyki takich jak np. analiza głównych składowych (ang. PCA) lub liniowa analiza dyskryminacyjna (ang. LDA) ogranicza się wymiarowość deskryptora. Dokładny opis tego podejścia można znaleźć w pracach Deskryptory SIFT i SURF operują na liczbach zmiennoprzecinkowych. Są bardzo dokładne ale ich obliczanie jest czasochłonne. Jak pokazały prace [19][21] [5] przejście z reprezentacji zmiennoprzecinkowej do binarnej nie powoduje istotnego spadku jakości rozpoznawania obrazów a w sposób znaczący rozpoznawanie przyśpiesza.

Przedstawione podejścia często są stosowane razem, usprawniając rozpoznawanie i dopasowywanie punktów obrazów. Niestety aby móc zbudować "szybki" deskryptor w obu podejściach potrzebne jest "wolne" obliczenie dokładnego deskryptora, który następnie poddawany jest obróbce.

W przypadku algorytmu BRIEF końcowy deskryptor nie wymaga wcześniejszego obliczania innych reprezentacji, co czyni go szybkim zarówno podczas samego tworzenia jak i użytkowania.

2.5.1 Zasada działania

Działanie algorytmu oparte jest na porównywaniu parami intensywności pikseli znajdujących się w otoczeniu punktu charakterystycznego.

W otoczeniu punktu charakterystycznego zdefiniowanym jako kwadrat o boku S z poddanego rozmyciu obrazu \mathbf{p} wybiera się parę punktów \mathbf{x} i \mathbf{y} , które poddawane są następującemu testowi:

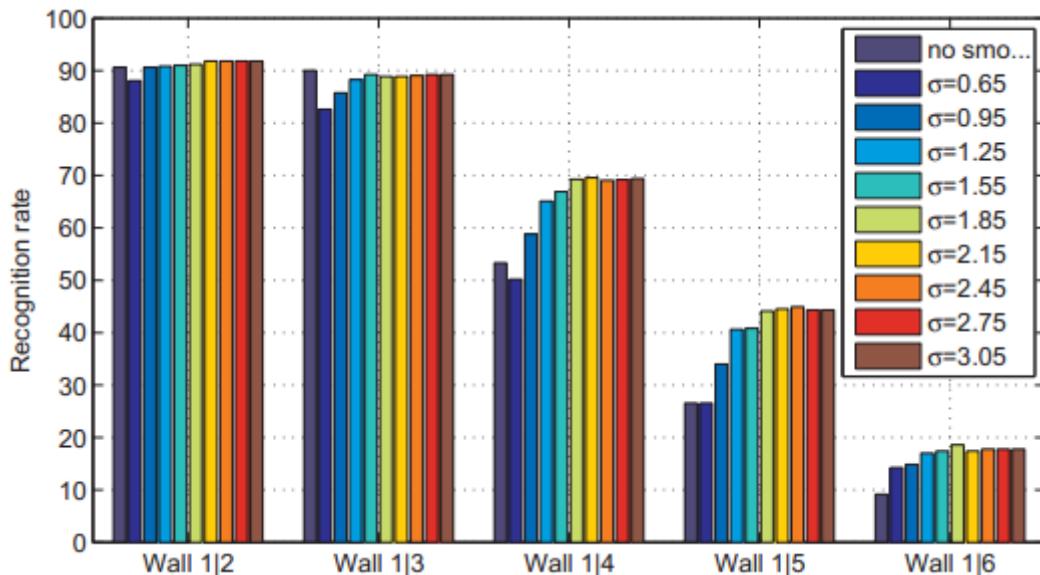
$$\tau(\mathbf{p}; \mathbf{x}, \mathbf{y}) := \begin{cases} 1 & \text{dla } \mathbf{p}(\mathbf{x}) < \mathbf{p}(\mathbf{y}) \\ 0 & \text{dla pozostałych} \end{cases} \quad (2.45) \quad \text{\{eqn:brie\}}$$

Powyższy test jest powtarzany dla kolejnych par. Ilość porównań jest oznaczona jest przez liczbę n_d . Określa ona jednocześnie długość deskryptora, który wyraża się wzorem:

$$f_{n_d}(\mathbf{p}) := \sum_{1 \leq i \leq n_d} 2^{i-1} \tau(\mathbf{p}; \mathbf{x}_i, \mathbf{y}_i) \quad (2.46)$$

Liczba porównywanych par n_d w zależności od wariantu wynosi od 128 do 512. Badanie przeprowadzone przez autorów dowodzi, że do zapisu efektywnego deskryptora wystarcza od 128 do 256 bitów. Jest to ogromny skok jakościowy w porównaniu do 128 liczb zmiennoprzecinkowych generowanych przez SIFT.

Zasada działania algorytmu jest bardzo prosta, aczkolwiek do pełnego jego opisu należy omówić dwa ważne aspekty jakimi są:



Rysunek 2.35 Skuteczność rozpoznawania deskryptorów w zależności od stopnia wstępniego rozmycia obrazu

{fig:brief}

- wstępne rozmycie obrazu
- wybór testowanych par punktów

2.5.2 Wstępne rozmycie obrazu

Dzięki zastosowaniu prostego testu jakim jest porównanie jasności udało się uniknąć nakładu obliczeń jaki towarzyszy deskryptorom gradientowym. Powoduje to jednakże zwiększoną wrażliwość algorytmu na zaszumienie obrazu. Z tego powodu aby poprawić stabilność deskryptora oryginalny obraz przed obliczeniem wskaźników do punktów charakterystycznych należy poddać rozmyciu.

Twórcy algorytmu zdecydowali się na zastosowanie rozmycia Gaussa. W trakcie prac nad algorytmem badano wpływ promienia rozmycia na jakość deskryptora i jego wydajność. Na zbiorze testowym Mikołajczyka¹ przetestowano promień o zakresie od 0 do 3. Jak pokazuje wykres 2.35 w przedziale promieni od 1 do 3 jakość jest bardzo zbliżona. Finalnie w algorytmie zastosowano dyskretne jądro rozmycia o wartości $\sigma = 2$ i rozmiarze 9 na 9 pikseli.

2.5.3 Wybór testowanych par punktów

Kluczowe znaczenie dla jakości generowanych deskryptorów, ma dobór par punktów poddawanych testowaniu. Od wyboru metody ich losowania zależy stopień powtarzalności i odporności deskryptora na rotacje.

W trakcie prac nad algorytmem autorzy przetestowali 5 sposobów selekcji par:

G I Lokalizacja punktów \mathbf{X} i \mathbf{Y} ma rozkład jednostajny na przedziale $(-\frac{S}{2}, \frac{S}{2})$. Początek układu współrzędnych umiejscowiony jest w punkcie centralnym badanego obszaru $S \times S$.

¹Zbiór testowy Mikołajczyka jest szerzej omówiony w rozdziale poświęconym badaniom

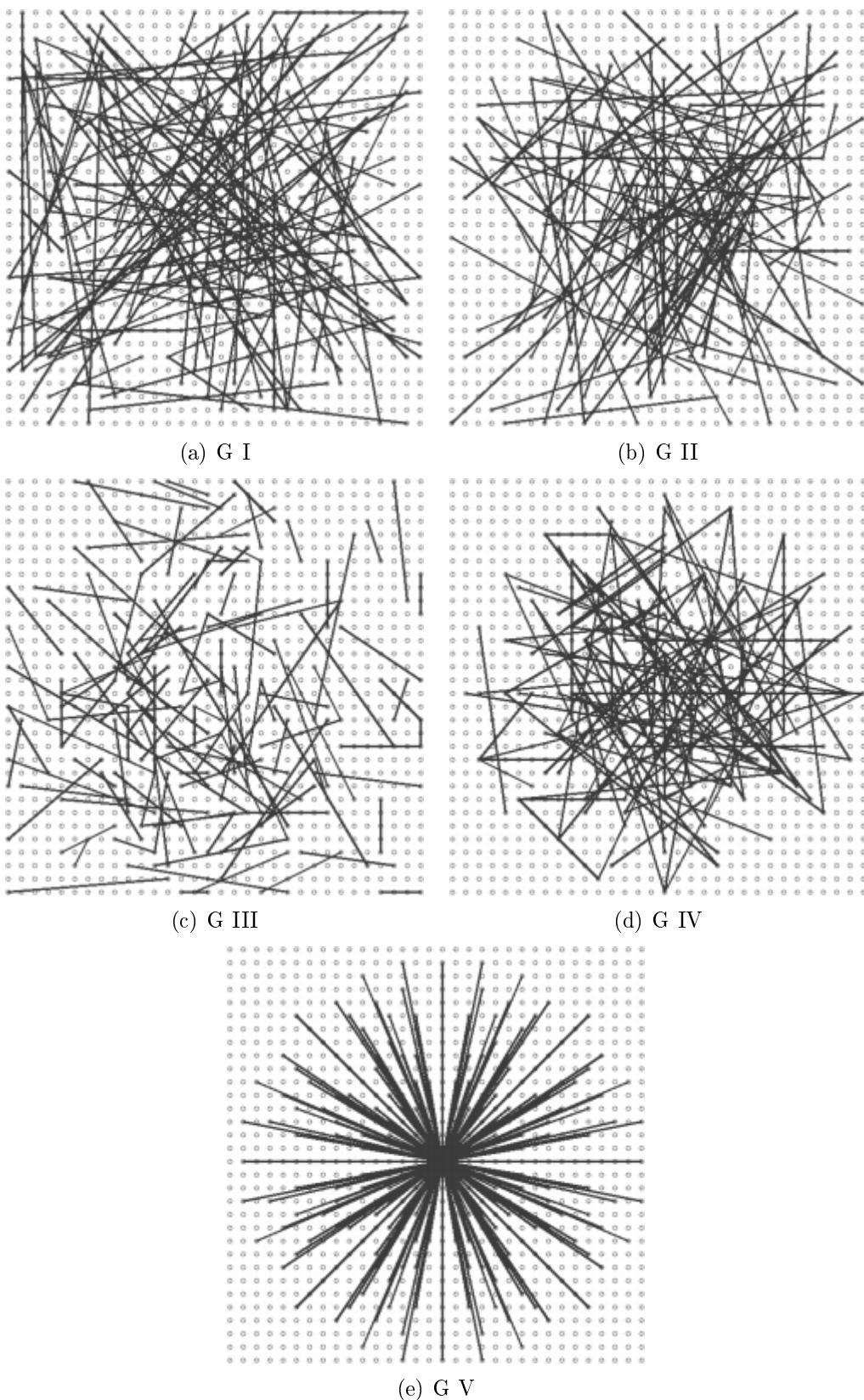
G II Zbiory punktów \mathbf{X} i \mathbf{Y} mają rozkład normalny o wartości oczekiwanej 0 i wariancji $\frac{S^2}{25}$. Wartość wariancji została dobrana eksperymentalnie i dla przedstawionej metody zapewnia najwyższy współczynnik rozpoznawalności.

G III W przypadku tej metody próbkowanie odbywa się 2 etapowo. W pierwszej kolejności z rozkładem normalnym o wartości oczekiwanej 0 i wariancji $\frac{S^2}{25}$ wybierane są punkty \mathbf{X} . Punkty \mathbf{Y} są lokalizowane w otoczeniu odpowiadających im punktów x_i . \mathbf{Y} ma rozkład normalny o wartości oczekiwanej x_i i wariancji $\frac{S^2}{100}$. W przypadku lokalizacji punktu y_i wypada poza badany obszar $S \times S$ przypisuje mu się położenie na krawędzi. Zastosowana metoda powoduje, że badane pary są badane w sposób bardziej lokalny. Podobnie jak w poprzedniej metodzie wartości wariancji zostały dobrane eksperymentalnie.

G IV Punkty x_i i y_i są wybierane w sposób losowy z dyskretnego zbioru zgrubnych współrzędnych biegunkowych.

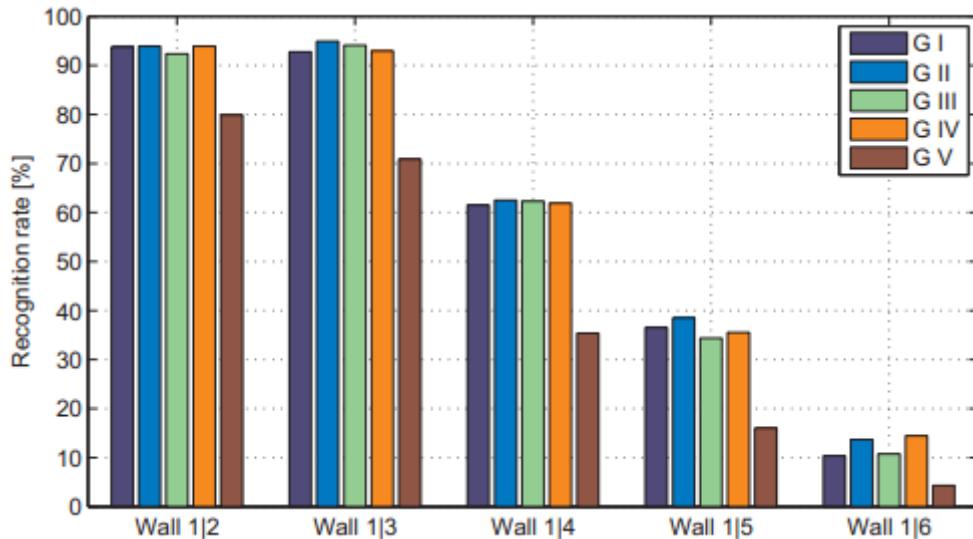
G V Wszystkie punkty x_i znajdują się w początku układu współrzędnych a punkty y_i lokalizowane są na liniach zgrubnych współrzędnych biegunkowych.

Zaproponowane metody próbkowania zostały przebadane na zbiorze testowym Mikołajczyka. Rezultaty tych badań przedstawione zostały na wykresie 2.37. Otrzymane wyniki dowodzą wysokiej skuteczności metod opartych o losowy dobór par. Na ich tle wyjątkowo słabo wypada regularna i symetryczna metoda G V. W ostatecznej wersji algorytmu zdecydowano się zastosować strategie próbkowania G II.



Rysunek 2.36 Przegląd metod losowania par punktów do testu binarnego

{fig:brief}



Rysunek 2.37 Skuteczność metod losowania par badana na zbiorze Mikołajczka WALL {fig:brief}

2.6 Algorytm ORB

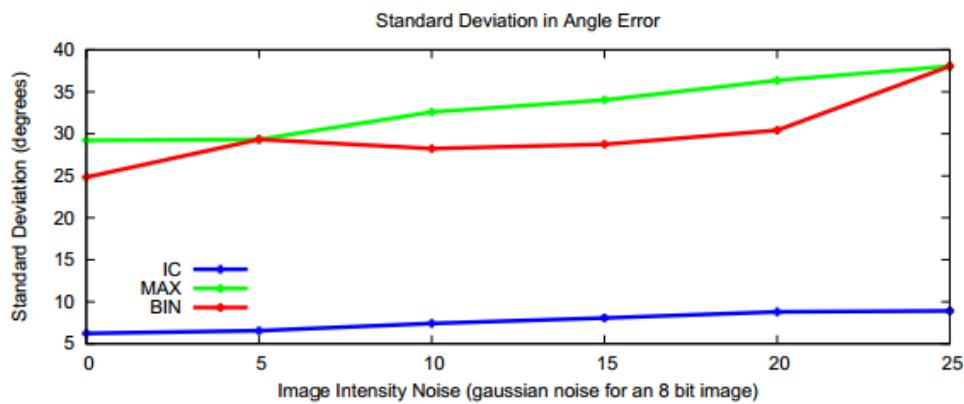
Algorytm ORB jest rozwinięciem wcześniejszej omówionych algorytmów: FAST i BRIEF. Został on opracowany podobnie jak algorytm STAR w laboratorium robotyki Willow Garage [18].

2.6.1 Lokalizowanie punktów charakterystycznych

Do lokalizowania punktów charakterystycznych algorytm ORB wykorzystuje zmodyfikowany detektor FAST. Jedna z modyfikacji polega na dodaniu do lokalizowanego punktu składowej określającej jego orientację, dlatego nowy detektor nazwano oFAST (od ang. oriented FAST).

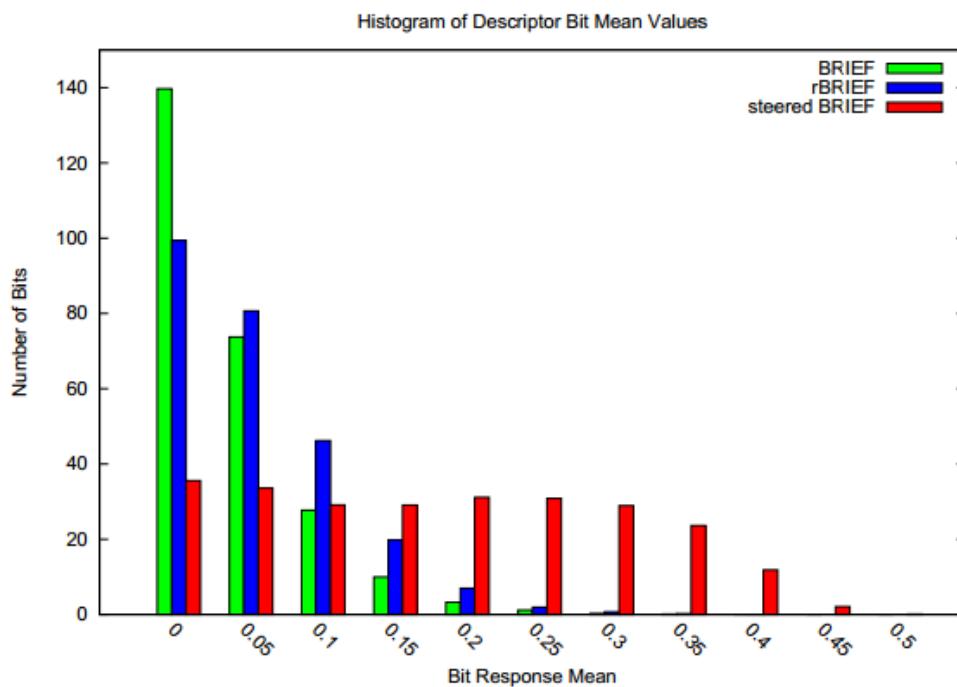
Pierwszym krokiem w działaniu algorytmu jest "zgrubne" zlokalizowanie punktów charakterystycznych. Realizuje się to za pomocą detektora FAST-9, gdzie 9 oznacza ilość ciągłych pikseli w okręgu lokalizującym róg.

Oryginalny algorytm FAST nie dostarcza żadnej miary jakości zlokalizowanego rogu, skutkiem czego jest dosyć wrażliwy na zakłócenia spowodowane odpowiedzią krawędziową. W efekcie bardzo często dostajemy bardzo dużą liczbę punktów charakterystycznych,



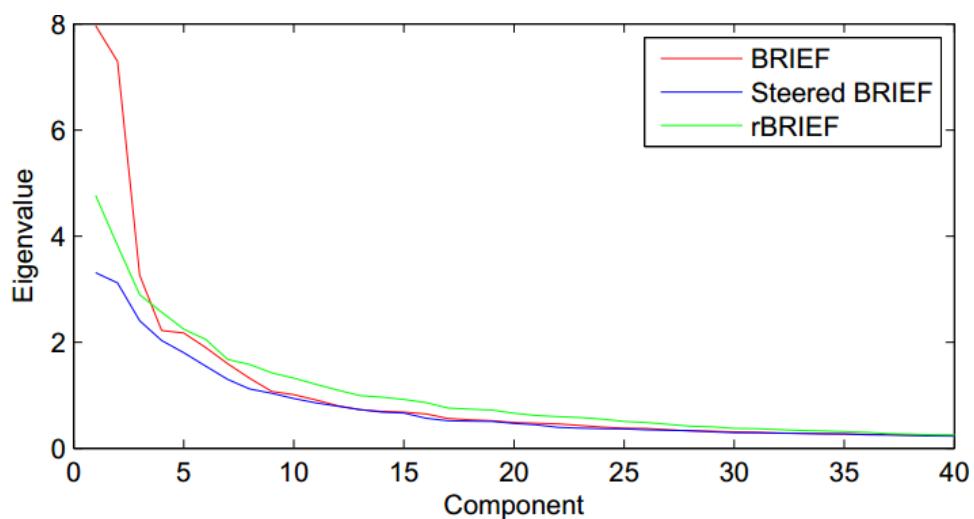
Rysunek 2.38 podpis

{fig:orb_wy}



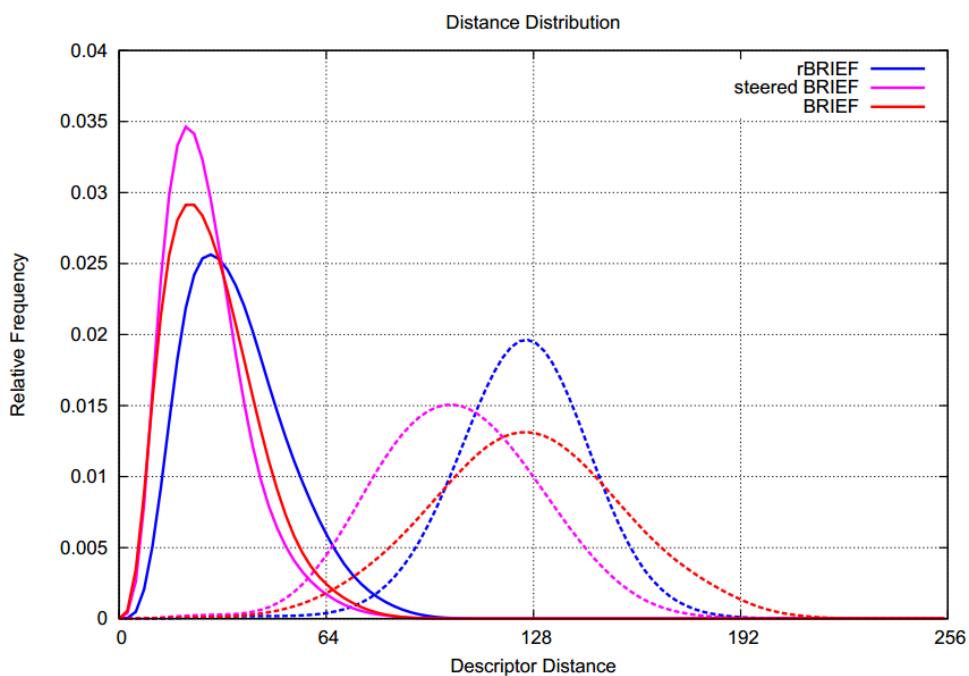
Rysunek 2.39 podpis

{fig:orb_wy1}



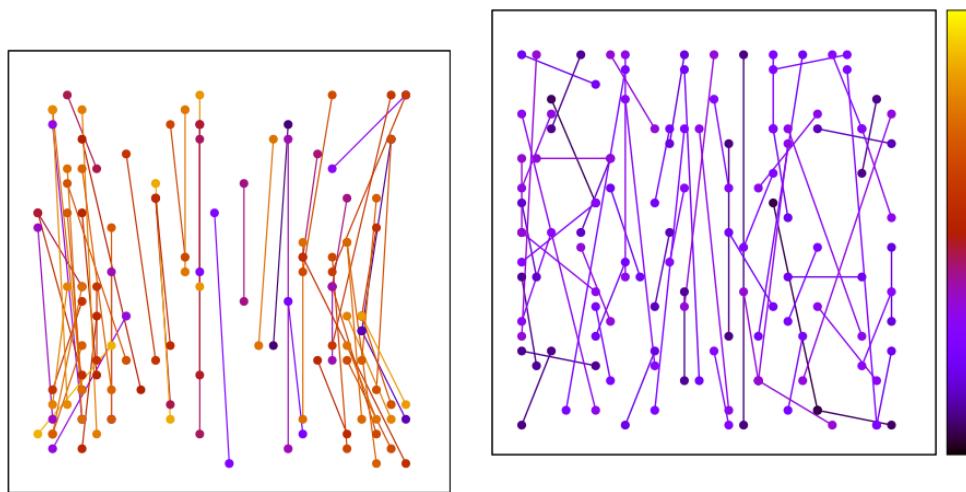
Rysunek 2.40 podpis

{fig:orb_wy2}



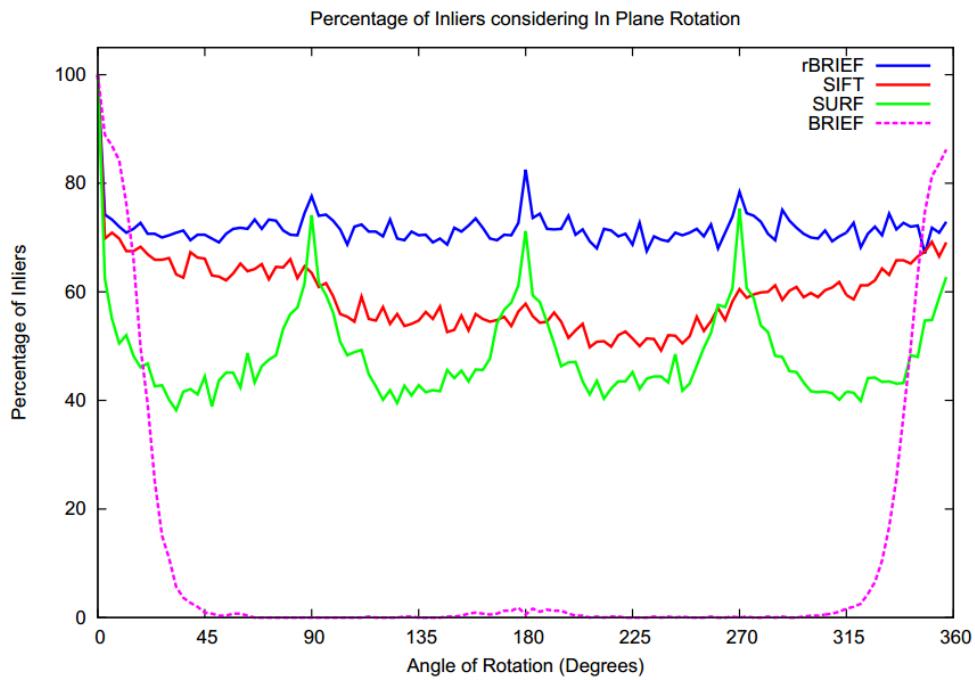
Rysunek 2.41 podpis

{fig:orb_wy



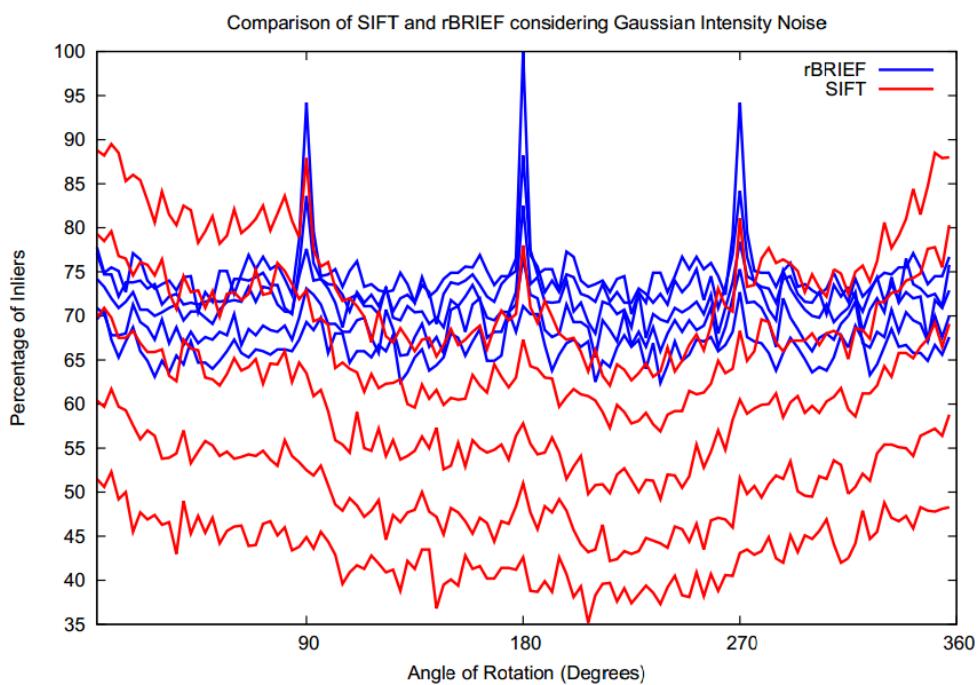
Rysunek 2.42 podpis

{fig:orb_wy}



Rysunek 2.43 podpis

{fig:orb_wy



Rysunek 2.44 podpis

{fig:orb_wy}

niestety o słabej jakości.

Rozwiąza niem powyższego problemu jest zastosowanie detektora Harrisza do oceniania punktów charakterystycznych. Algorytm ORB jako jeden z argumentów przyjmuje pożądaną liczbę punktów charakterystycznych oznaczoną przez N. Znając liczbę N algorytm tak ustala się wartość progu miary Harrisa aby uzyskać zbiór punktów, których ilość przekracza N. Następnie zbiór ten jest sortowany i wybiera się z niego N najlepszych punktów. W zastosowaniach praktycznych wartość liczby N ustala się najczęściej na 500. Zwiększenie tej liczby powoduje otrzymanie większej liczby punktów charakterystycznych, ale często gorszej jakości i nachodzące na punkty z grupy pięciuset.

Kolejnym wadę algorytmu FAST jest jego wrażliwość na skalowanie obrazu. W przypadku detektora oFAST zastosowano omawiany wcześniej mechanizm piramidy skalo-przestrzennej.

Do określenia orientacji rogu będącego punktem charakterystycznym zastosowano miarę intensywności środka ciężkości [15]. Metoda ta zakłada, że intensywność rogu jest przesunięta względem jego środka. Wektor przesunięcia jest używany do określenie orientacji punktu charakterystycznego.

Definiując dla badanego wycinka momenty jako:

$$m_{pq} = \sum_{x,y} x^p y^q I(x, y) \quad (2.47)$$

i środek ciężkości:

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \quad (2.48)$$

możemy skonstruować wektor \overrightarrow{OC} zaczepiony w środku rogu i skierowany w stronę środka ciężkości. Orientacje tego wektora możemy w prosty sposób obliczyć z zależności:

$$\theta = \text{atan2}(m_{01}, m_{10}) \quad (2.49)$$

Aby poprawić stabilność zaprezentowanej miary na rotacje, punkty x i y są wybierane z kołowego obszaru o promieniu r . W praktyce oznacza to, że r reprezentuje rozmiar badanego wycinka a współrzędne x i y zawierają się w przedziale $[-r, r]$. W sytuacji gdy wartość momentu badanego odcinka $|C|$ wynosi 0 algorytm zachowuje się w sposób niestabilny. W przypadku punktów lokalizowanych przez algorytm FAST jest to jednak bardzo rzadki przypadek.

Zaprezentowana metoda została porównana z metodami bazującymi na gradiencie obrazu, BIN i MAX. Metoda MAX wybiera największą składową gradientu w obrębie badanego punktu. Metoda BIN podobnie jak w algorytmie SIFT i jest oparta o histogram reprezentujący wszystkie składowe gradientu. Jak pokazuje wykres 2.38 metoda oparta o środek ciężkości jest o znacznie bardziej odporna na zakłócenia.

2.6.2 Budowa deskryptora

Do opisu zlokalizowanych cech wykorzystano ulepszoną zmodyfikowaną wersję deskryptora Steered BRIEF nazwaną rBRIEF.

Steered Brief jest rozwinięciem opisanego wcześniej deskryptora BRIEF. Oryginalny deskryptor BRIEF jest bardzo wrażliwy na rotacje [wykres 2.43]. Pewnym rozwiązaniem, aczkolwiek bardzo nieoptymalnym, jest zaproponowane przez autorów algorytmu BRIEF wyliczenie grupy obróconych względem siebie deskryptorów. Znacznie lepszym rozwiązaniem dostrojenie deskryptora BRIEF do orientacji punktu charakterystycznego. Deskryptor BRIEF jest generowany przez n testów binarny dla zbioru par punktów (x_i, y_i) . Zbiór

ten jest opisany przez macierz \mathbf{S} :

$$\mathbf{S} = \begin{pmatrix} x_1 & , & x_2 & , & \dots & , & x_n \\ y_1 & , & y_2 & , & \dots & , & y_n \end{pmatrix} \quad (2.50)$$

Na podstawie obliczonej wcześniej orientacji punktu charakterystycznego θ i odpowiadającej jej macierzy rotacji \mathbf{R}_θ , konstruowany jest "ukierunkowany" zbiór testowy \mathbf{S}_θ :

$$\mathbf{S}_\theta = \mathbf{R}_\theta \mathbf{S} \quad (2.51)$$

Otrzymany zbiór jest poddawany temu samemu testowi jak w przypadku oryginalnego algorytmu BRIEF (wzór 2.45).

W celu usprawnienia pracy algorytmu orientacje punktów charakterystycznych poddawane są kwantyzacji. Dla zakresu 360° przyjęto się 30 przedziałów 12° do których może zostać zakwalifikowana orientacja punktu. Każdy z przedziałów posiada własny wzorzec punktów poddawanych testom.

Jedną z niewątpliwych zalet algorytmu BRIEF jest to że każdy bit jego deskryptora ma dużą wariancję i wartość średnią na poziomie 0,5. Wykres 2.39 prezentuje rozrzut wartości średniej dla typowego gaussowskiego wzorca 256 bitowego deskryptora BRIEF dla zbioru 100 tysięcy punktów charakterystycznych. Wartość średnia 0,5 daje największą wariancję próbkowania na poziomie 0,25 dla pojedynczego bitu. Zastosowanie "ukierunkowanego" algorytmu BRIEF powoduje przesunięcie wartości średniej w stronę rozkładu typowego dla wzorca odpowiadającego danemu kątowi orientacji. Powoduje to zbytnie upodobnienie się deskryptorów punktów zakwalifikowanych do tego samego przedziału orientacji.

Wysoka wariancja czyni deskryptory bardziej dyskryminacyjnymi co jest pożądany zjawiskiem, gdyż generuje bardziej zróżnicowane odpowiedzi dla punktów o podobnych parametrach.

Kolejnym ważnym aspektem jest również to aby kolejne bity deskryptora były ze sobą nieskorelowane. Autorzy algorytmu dla badanego zbioru 100 tysięcy punktów dokonali analizy głównych składowych (ang. PCA) otrzymywanych deskryptorów. Wyniki tego badania przedstawia wykres 2.40. Na podstawie analizy wykreślono 40 najwyższych wartości własnych, po których zarówno deskryptory BRIEF jak steered BRIEF się pokrywają. Obie metody wykazują wysokie wejściowe wartości własne, wskazujące na korelacje testów binarnych. Większość informacji jest zawarta w pierwszych 15 komponentach. Algorytm steered BRIEF ma znacznie mniejszą wariancję, aczkolwiek przy mniejszych wartościach własnych przestaje być dyskryminacyjnym. Na podstawie tego można stwierdzić, że BRIEF zawdzięcza swoja dobre wyniki losowej orientacji punktów charakterystycznych.

Innym niekorzystnym efektem towarzyszącym deskryptorowi steered BRIEF jest zmniejszenie dystansu między punktami pokrywającymi się i odstającymi co przedstawiono na wykresie 2.41. Może to powodować nie pożądane nachodzenie na siebie punktów poprawnych i złych, a co za tym idzie powodować fałszywe dopasowania.

Aby zapobiec opisanym zjawiskom spadku wariancji i wzrostu korelacji pomiędzy testami, autorzy algorytmu zaproponowali metodę uczenia służącą wyborowi dobrego podzbioru binarnych testów.

W zaproponowanej metodzie na podstawie zbioru obrazów PASCAL 2006 [7] przygotowano zbiór uczący zawierający około 300 tysięcy punktów charakterystycznych. Następnie dla okna o wymiarach 31×31 pikseli przygotowano kombinacje wszystkich możliwych binarnych testów. Każdy test jest wykonywany na parze mniejszych okien 5×5 . Zapisując szerokość dużego okna jako $w_p = 31$ i szerokość małego okna jako $w_t = 5$ otrzymujemy liczbę $N = (w_p - w_t)^2$ oznaczającą liczbę możliwych małych okien. Wybierając 2 okna do

pojedynczego badania otrzymujemy zbiór $\binom{N}{2}$ testów. Eliminując testy nakładające się na siebie otrzymujemy zbiór 205590 porównań.

Przygotowany zbiór uczący poddano badaniu następującemu algorytmowi:

1. Wszystkie obrazy zbioru uczącego przetestowano wszystkimi kombinacjami testów.
2. Testy zostały uporządkowane na podstawie odległości wartości średniej od wartości 0,5 i zawarte w wektorze \mathbf{T} .
3. Następnie wektor \mathbf{T} jest poddawany przeszukiwaniu zachłannemu:
 - (a) Pierwszy test wektora \mathbf{T} jest kopowany do 256 wymiarowego wektora \mathbf{R} i usuwany ze zbioru \mathbf{T} .
 - (b) Kolejny test ze zbioru \mathbf{T} jest porównywany ze wszystkimi testami znajdującymi się w wektorze \mathbf{R} . W sytuacji gdy wartość absolutna korelacji między testami jest mniejsza niż zadany próg, test zostaje przeniesiony do zbioru wektora \mathbf{R} .
 - (c) Punkt (b) jest powtarzany aż do momentu, w którym w wektor \mathbf{R} będzie posiadał 256 elementów. W sytuacji niespełnienia warunku stopu i wyczerpania się elementów wektora \mathbf{T} wartość progu jest podnoszona i algorytm powtarza swoje działanie.

Wynik działania tego algorytmu został nazwany rBRIEF. Algorytm ten ma znaczaco lepsze parametry wariancji i korelacji niż algorytm steered BRIEF. Jak przedstawia wykres 2.40 wygenerowany deskryptorem ma wartości własne głównych składowych są wyższe i zmniejszają się znacznie wolniej niż w przypadku algorytmów steered BRIEF i BRIEF. Badania przeprowadzone przez autorów algorytmów wykazują ponadto wysoką odporność na rotacje [wykres 2.43].

Część II

Badania

Rozdział 3

Opis badań

Omówione w części teoretycznej algorytmy zostały przebadane pod kątem efektywności działania. W tym celu został napisany program komputerowy, którego szerszy opis można znaleźć w dodatku A.

3.1 Badane algorytmy

Problem poprawnego rozpoznawania obrazów opartego o cechy charakterystyczne można zdekomponować na dwa mniejsze jakimi są:

- lokalizowanie punktów charakterystycznych,
- opisywanie punktów charakterystycznych.

Oba aspekty są kluczowe dla realizacji postawionego zadania, a efekt finalny jest wypadkową pracy detektorów i deskryptorów. Z tego względu przyjęto założenie, że algorytmy kompletnie tj. zawierające zarówno detektor jak i deskryptor zostaną zestawione z algorytmami mieszanymi. W przypadku algorytmów nie posiadających własnych mechanizmów opisu cech, pulę dostępnych deskryptorów wzbogacono o metody algorytmów kompletnych. Zastosowane podejście pozwoliło porównać ze sobą niezależnie stosowane detektory i deskryptory. Postanowiono przebadać następujące algorytmy:

- ORB
- SIFT
- SURF
- detektor STAR + deskryptor BRIEF
- detektor STAR + deskryptor ORB (*wł. rBRIEF*)
- detektor STAR + deskryptor SIFT
- detektor STAR + deskryptor SURF
- detektor FAST + deskryptor BRIEF ¹
- detektor FAST + deskryptor ORB (*wł. rBRIEF*) ¹
- detektor FAST + deskryptor SIFT ¹

- detektor FAST + deskryptor SURF ¹

3.2 Kryteria badań

W porównywaniu algorytmów wzięto pod uwagę następujące kryteria:

- ilość lokalizowanych cech
- czas potrzebny na zlokalizowanie pojedynczego punktu charakterystycznego
- czas potrzebny na wygenerowanie deskryptora pojedynczego punktu charakterystycznego
- powtarzalność zlokalizowanych cech
- procent poprawnych dopasowań

Ze względu na to, że algorytmy różnią się czasem działania i ilością lokalizowanych cech w różnych obrazach, wszystkie kryteria czasowe zdecydowano się odnieść do pojedynczego punktu.

Powtarzalność zlokalizowanych cech jest rozumiana jako:

$$p_{AB} = \frac{d_{AB}}{i_{AB}} \quad (3.1)$$

gdzie:

p_{AB} - powtarzalność lokalizowanych cech między obrazami A i B

d_{AB} - ilość dopasowanych cech między obrazami A i B metodą k najbliższych sąsiadów

i_{AB} - średnia arytmetyczna ilości punktów zlokalizowanych w obrazach A i B

Do oceny jakości parowania cech wykorzystano algorytm RANSAC [1]. Pozwala on wyeliminować błędne dopasowania nie pasujące do estymowanego wzorca. Za procent poprawnych dopasowań rozumiemy stosunek ilość par punktów po zastosowaniu algorytmu RANSAC do ilości wstępnie dopasowanych punktów.

3.3 Zbiory testowe

Do oceny pracy algorytmów wykorzystano dwa zbiory testowe:

- zbiór testowy Mikołajczyka
- autorski zbiór testowy przedstawiający obrazy skalne

W skład obu zbiorów wchodzą zestawy obrazów pogrupowane wg, przekształceń takich jak rotacja, zmiana skali, zmiana oświetlenie lub położenia obserwatora. Każdy pojedynczy zestaw składa się z 6 zdjęć. Pierwsze zdjęcie jest obiektem referencyjnym względem którego dokonywane są dopasowania. Kolejne zdjęcia zestawu są poddawane coraz mocniejszym przekształceniom.

¹Pierwsze testy wykazały jednakże, że ilość punktów charakterystycznych lokalizowanych przez algorytm FAST jest zbyt duża, aby można było dokonać dopasowania w rozsądny czasie pracy maszyny. W związku z powyższym zrezygnowano z opisywania i dopasowywania cech lokalizowanych przez ten algorytm

Rozdział 4

Zbiór testowy Mikołajczyka

Zbiór testowy Mikołajczyka to opracowany przez doktora Krystiana Mikołajczyka zespół zestawów obrazów testowych służących ocenie afinicznych, kowariantnych cech. Zawiera on przekształcenia takie jak:

- zmiana położenia obserwatora
- zmiana skali
- rotacja
- zmiana oświetlenia
- rozmycie
- kompresja JPEG

Oryginalny zbiór zawiera 8 zestawów zawierające powyższe transformacje. Zbiory dotyczące położenia obserwatora, skali i rozmycia są zdublowane. Uczyniono to aby uniezależnić ocenę jakości pracy z przekształceniem od charakterystyki obrazu. W przeprowadzonych badaniach zrezygnowano z testowania odporności algorytmów na kompresje JPEG. Ze względu na fakt, że zbiór Mikołajczyka jest często wykorzystywany w badaniach, zdecydowano się zachować angielskie nazewnictwo zestawów. Zbiór jest dostępny w internecie na stronie <http://www.robots.ox.ac.uk/~vgg/research/affine/>.

4.1 Rozmycie obrazu

4.1.1 BIKES



(a) BIKES 1



(b) BIKES 2



(c) BIKES 3



(d) BIKES 4



(e) BIKES 5



(f) BIKES 6

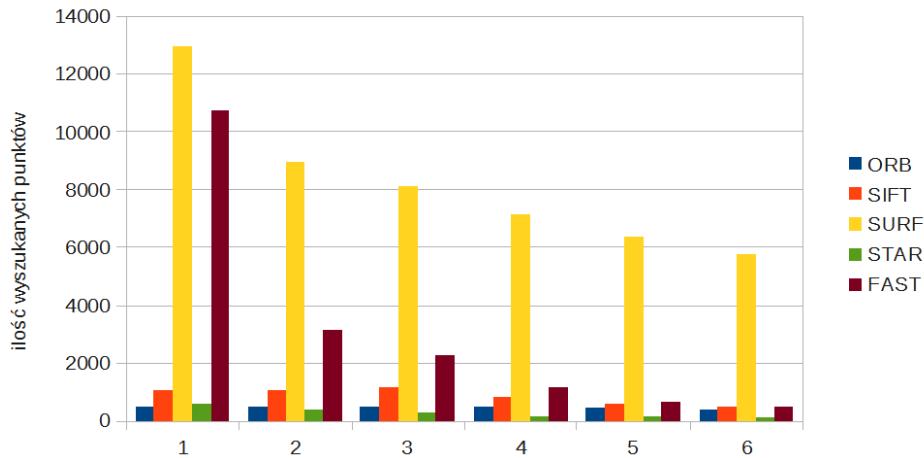
Rysunek 4.1 BIKES - zbiór obrazów testowych

{fig:bikes_}

Tabela. 4.1 BIKES - ilość wyszukanych cech

obraz	ORB	SIFT	SURF	STAR	FAST
1	500	1014	12911	586	10702
2	500	1062	8974	362	3128
3	500	1159	8100	290	2247
4	485	839	7151	181	1129
5	420	596	6344	139	675
6	366	473	5751	118	500
średnia	462	857	8205	279	3064

{tab:bikes_1}



Rysunek 4.2 BIKES - ilość wyszukanych cech

{fig:bikes_1}

Tabela. 4.2 BIKES - czas lokalizowania pojedynczego punktu charakterystycznego

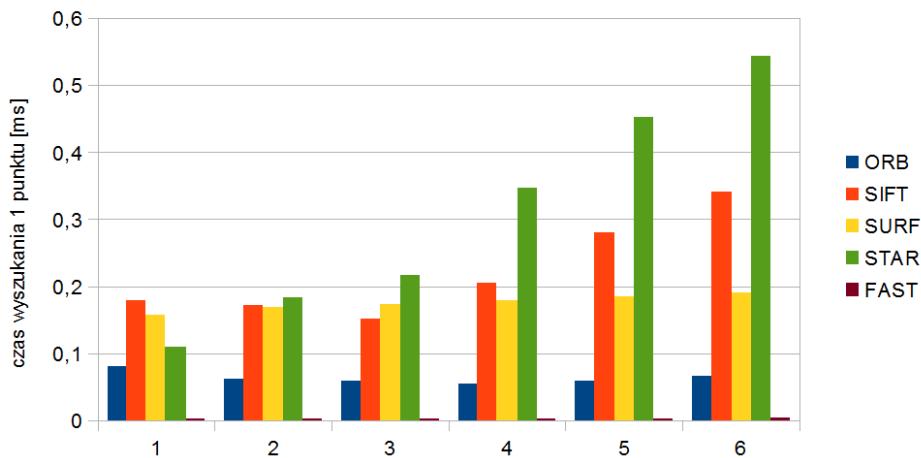
obraz	ORB	SIFT	SURF	STAR	FAST
1	0,080	0,179	0,156	0,109	0,001
2	0,062	0,171	0,169	0,182	0,002
3	0,060	0,153	0,173	0,217	0,002
4	0,056	0,204	0,179	0,348	0,003
5	0,060	0,282	0,185	0,453	0,003
6	0,066	0,340	0,191	0,542	0,004
średnia	0,064	0,222	0,175	0,309	0,002

{tab:bikes_2}

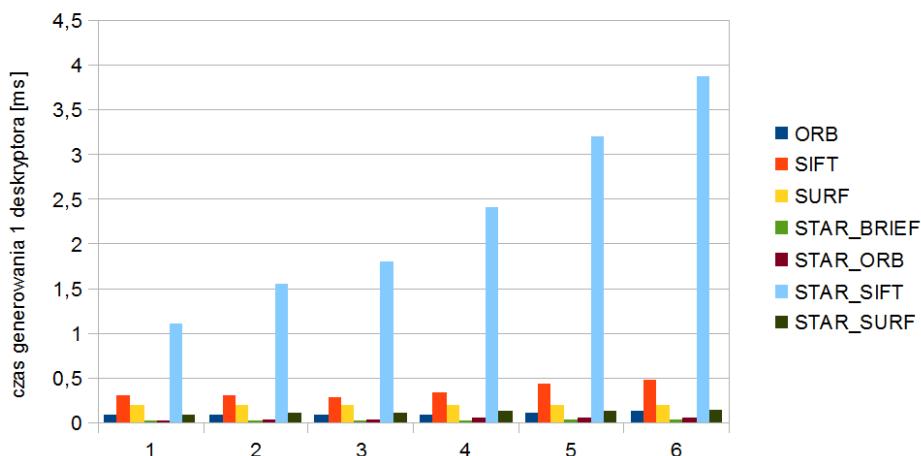
Tabela. 4.3 BIKES - czas generowania pojedynczego deskryptora punktu charakterystycznego

obraz	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[ms]	[ms]	[ms]	[ms]	[ms]	[ms]	[ms]
1	0,092	0,300	0,193	0,015	0,019	1,109	0,094
2	0,090	0,293	0,201	0,019	0,025	1,561	0,099
3	0,090	0,283	0,199	0,017	0,031	1,803	0,103
4	0,093	0,344	0,194	0,022	0,044	2,420	0,116
5	0,107	0,424	0,195	0,029	0,050	3,201	0,129
6	0,120	0,474	0,198	0,025	0,059	3,873	0,136
średnia	0,099	0,353	0,197	0,021	0,038	2,328	0,113

{tab:bikes_3}



Rysunek 4.3 BIKES - czas lokalizowania pojedynczego punktu charakterystycznego



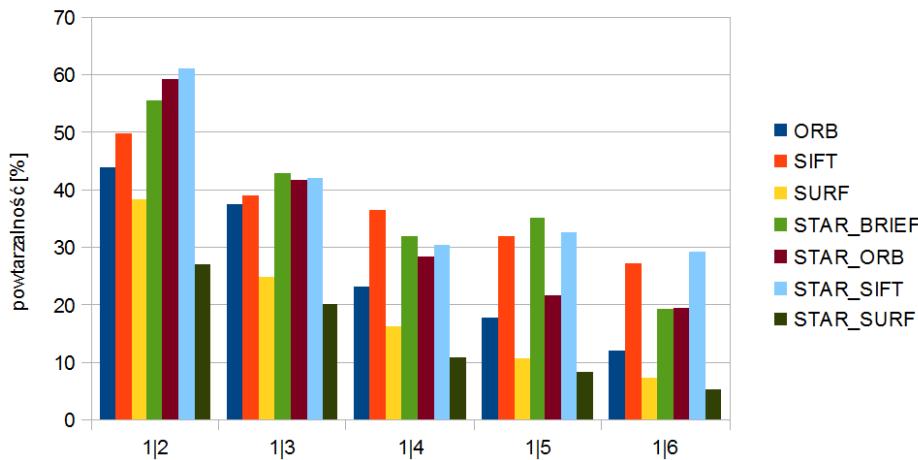
Rysunek 4.4 BIKES - czas generowania pojedynczego deskryptora punktu charakterystycznego

{fig:bikes}

Tabela 4.4 BIKES - powtarzalność wykrywanych cech

obrazy	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[%]	[%]	[%]	[%]	[%]	[%]	[%]
1 2	44	50	38	55	59	61	27
1 3	38	39	25	43	42	42	20
1 4	23	36	16	32	28	30	11
1 5	18	32	11	35	22	33	8
1 6	12	27	7	19	19	29	5
średnia	27	37	19	37	34	39	14

{tab:bikes}



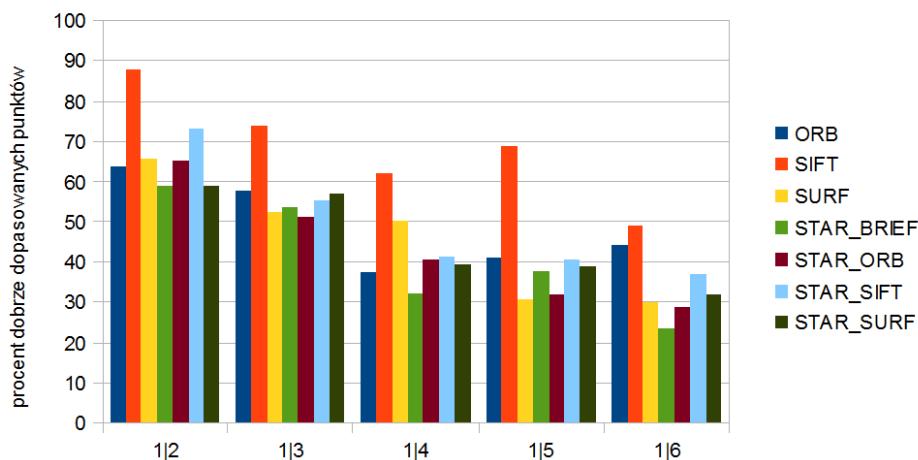
Rysunek 4.5 BIKES - powtarzalność wykrywanych cech

{fig:bikes_1}

Tabela 4.5 BIKES - procent poprawnych dopasowań

obrazy	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[%]	[%]	[%]	[%]	[%]	[%]	[%]
1 2	64	88	66	59	65	73	59
1 3	58	74	52	54	51	55	57
1 4	37	62	50	32	40	41	39
1 5	41	69	31	38	32	40	39
1 6	44	49	30	23	28	37	32
średnia	49	68	46	41	43	49	45

{tab:bikes_1}



Rysunek 4.6 BIKES - procent poprawnych dopasowań

{fig:bikes_2}

4.1.2 TREES



(a) TREES 1



(b) TREES 2



(c) TREES 3



(d) TREES 4



(e) TREES 5



(f) TREES 6

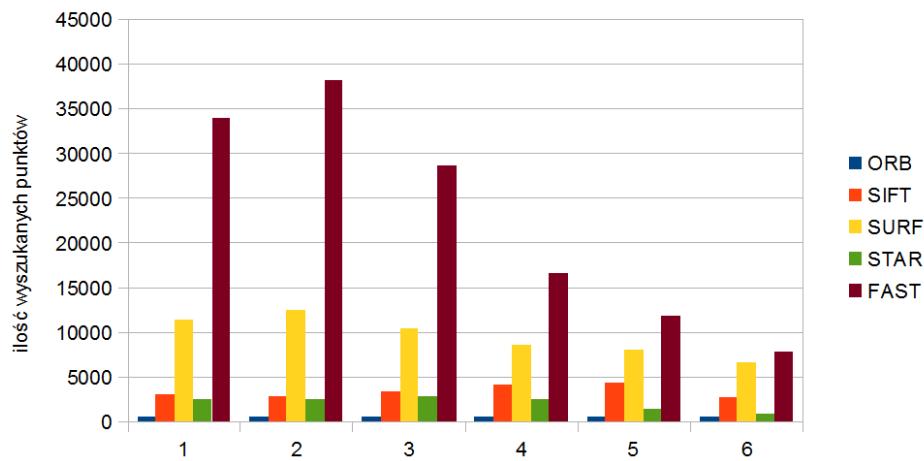
Rysunek 4.7 TREES - zbiór obrazów testowych

{fig:trees}

Tabela. 4.6 TREES - ilość wyszukanych cech

obraz	ORB	SIFT	SURF	STAR	FAST
1	500	2941	11343	2539	34024
2	500	2848	12563	2536	38235
3	500	3363	10377	2846	28516
4	500	4150	8613	2398	16582
5	500	4244	8022	1383	11716
6	500	2709	6605	824	7883
średnia	500	3376	9587	2088	22826

{tab:trees}



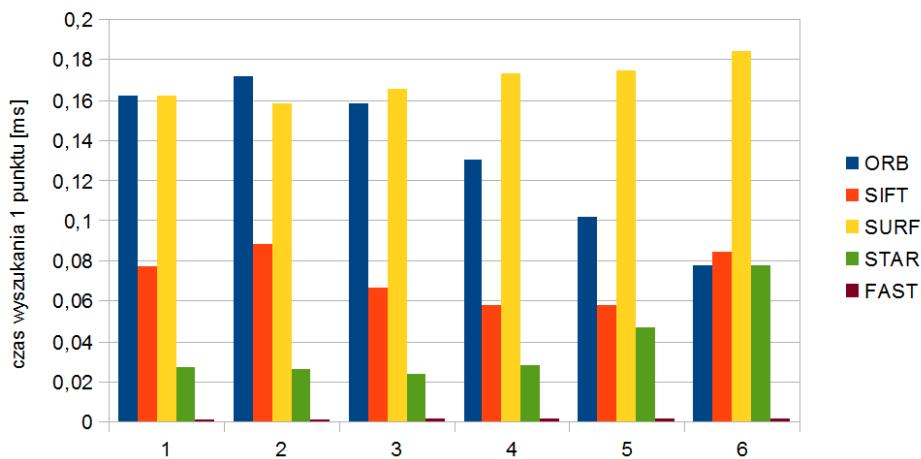
Rysunek 4.8 TREES - ilość wyszukanych cech

{fig:trees}

Tabela. 4.7 TREES - czas lokalizowania pojedynczego punktu charakterystycznego w obrazach z zestawu TREES

obraz	ORB	SIFT	SURF	STAR	FAST
-	[ms]	[ms]	[ms]	[ms]	[ms]
1	0,162	0,077	0,162	0,027	0,001
2	0,172	0,088	0,158	0,026	0,001
3	0,158	0,067	0,165	0,024	0,001
4	0,130	0,058	0,173	0,028	0,002
5	0,102	0,058	0,175	0,047	0,002
6	0,078	0,085	0,185	0,078	0,001
średnia	0,134	0,072	0,170	0,038	0,001

{tab:trees}



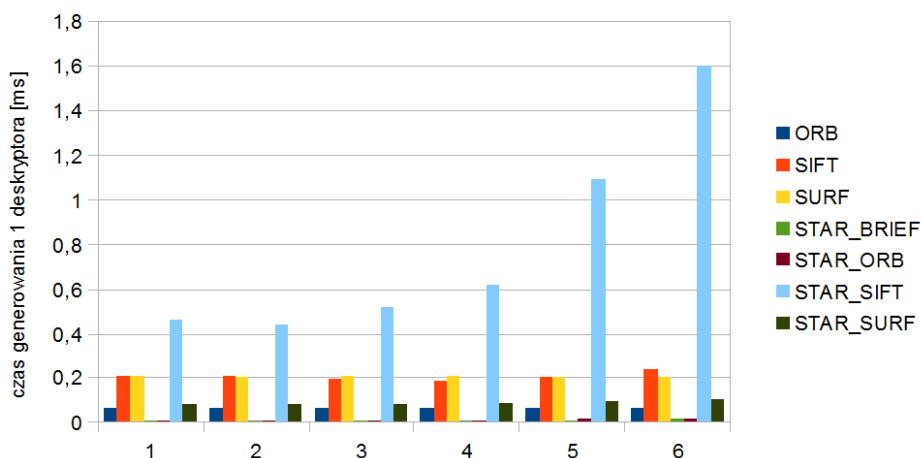
Rysunek 4.9 TREES - czas lokalizowania pojedynczego punktu charakterystycznego

{fig:trees_1}

Tabela. 4.8 TREES - czas generowania pojedynczego deskryptora punktu charakterystycznego

obraz	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[ms]						
1	0,064	0,207	0,209	0,012	0,012	0,460	0,083
2	0,064	0,210	0,203	0,012	0,012	0,442	0,084
3	0,064	0,197	0,209	0,012	0,011	0,516	0,084
4	0,064	0,188	0,213	0,012	0,012	0,620	0,085
5	0,064	0,200	0,204	0,012	0,014	1,091	0,093
6	0,064	0,240	0,202	0,013	0,017	1,602	0,103
średnia	0,064	0,207	0,207	0,012	0,013	0,789	0,089

{tab:trees_1}



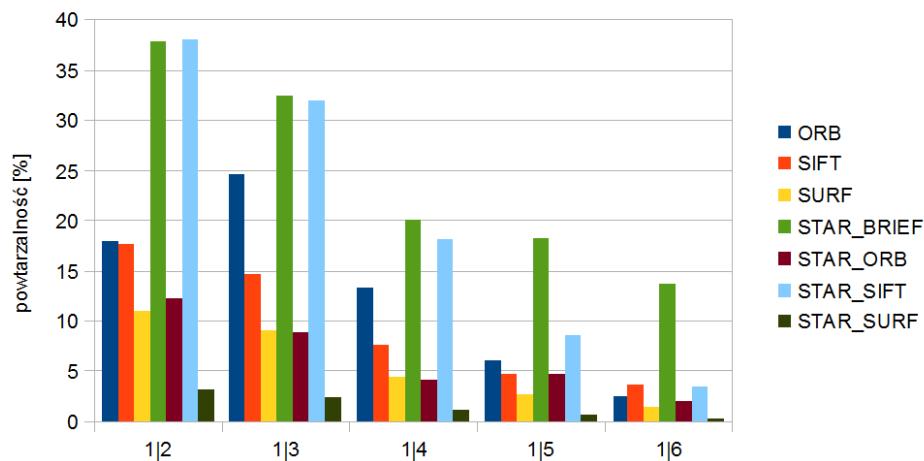
Rysunek 4.10 TREES - czas generowania pojedynczego deskryptora punktu charakterystycznego

{fig:trees_2}

Tabela. 4.9 TREES - powtarzalność wykrywanych cech

obrazy	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[%]	[%]	[%]	[%]	[%]	[%]	[%]
1 2	18	18	11	38	12	38	3
1 3	25	15	9	32	9	32	2
1 4	13	8	4	20	4	18	1
1 5	6	5	3	18	5	9	1
1 6	3	4	1	14	2	3	0
średnia	13	10	6	24	6	20	1

{tab:trees}



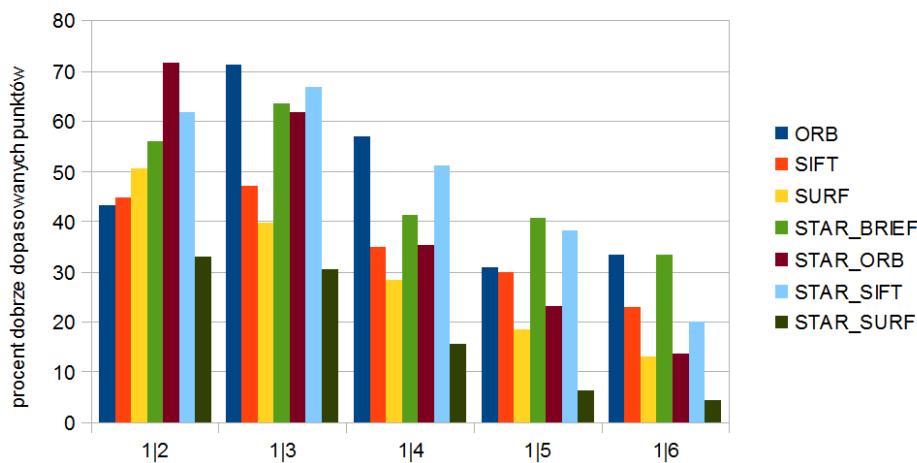
Rysunek 4.11 TREES - powtarzalność wykrywanych cech

{fig:trees}

Tabela. 4.10 TREES - procent poprawnych dopasowań

obrazy	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[%]	[%]	[%]	[%]	[%]	[%]	[%]
1 2	43	45	50	56	71	62	33
1 3	71	47	40	63	62	67	30
1 4	57	35	28	41	35	51	16
1 5	31	30	18	41	23	38	6
1 6	33	23	13	33	14	20	4
średnia	47	36	30	47	41	48	18

{tab:trees}



Rysunek 4.12 TREES - procent poprawnych dopasowań

{fig:trees}

4.1.3 Dyskusja wyników

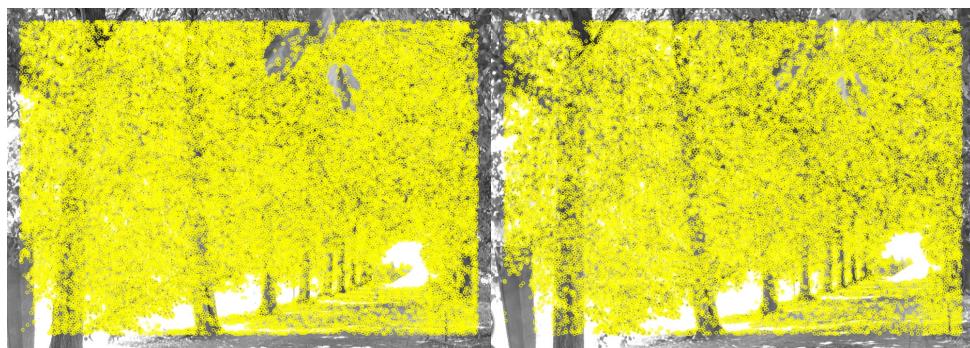
Jak możemy zaobserwować na wykresach 4.2 i 4.8 wraz ze wzrostem stopnia rozmycia obrazu maleje liczba lokalizowanych punktów charakterystycznych. Wśród ilości lokalizowanych punktów prym wiodą w zależności od scenerii algorytmy SURF i FAST, i to wśród ich wyników spadek ilości punktów jest najbardziej zauważalny. Algorytmy te generują o rząd wielkości więcej punktów niż pozostałe metody. Niestety jakość tych punktów pozostawia często wiele do życzenia, a ich ilość w sposób znaczny utrudnia proces dopasowywania. Rysunek 4.13 przedstawia punkty wyszukane przez algorytm FAST w obrazach z zestawu TREES. Jak widzimy ich ilość w sposób znaczący zaciemnia obraz.

Większość algorytmów jest wrażliwa na stopień rozmycia, co przekłada się na czas lokalizowania punktów charakterystycznych. Na tym tle korzystnie wypada algorytm ORB, poprawiający swoje charakterystyki czasowe przy zachowanej stałej liczbie lokalizowanych punktów.

W przypadku generowania deskryptora punktu charakterystycznego większość algorytmów zachowuje względnie stałe wartości czasowe. Wyniki te mogą się nieznacznie różnić w zależności od analizowanej sceny. Wyjątkiem tu są dokładne deskryptory gradientowe SIFT. Wraz ze wzrostem rozmycia obserwujemy wzrost czasu potrzebnego na obliczenie deskryptora. Szczególnie dramatyczny wzrost ma miejsce w przypadku kombinacji deskryptora SIFT z detektorem STAR. Częściową tą wadę rekompensuje wysoka na tle innych algorytmów powtarzalność dopasowań.

Powtarzalność dopasowań dla wszystkich algorytmów spada wraz ze wzrostem rozmycia. Parametr ten w zależności od scenerii osiąga maksymalną wartość od 30 do 60 %. Jak pokazują algorytmy są wrażliwe w przypadku rozmycia obrazów o dużej ilości drobnych, podobnych elementów jakimi są liście.

W przypadku obrazów o zróżnicowanych elementach najlepszy współczynnik trafnych dopasowań osiąga algorytm SIFT. W sytuacji gdy obraz wskutek rozmycia w znacznym stopniu staje się nieczytelny jak w przypadku obrazów TREES lepszą odpornością wykazują się algorytm ORB i kombinacja algorytmów STAR i BRIEF.



Rysunek 4.13 Zlokalizowane przez algorytm FAST punkty charakterystyczne w obrazach TREES 1 i TREES 3

{tt13}

4.2 Zmiana położenia obserwatora

4.2.1 GRAFFITI



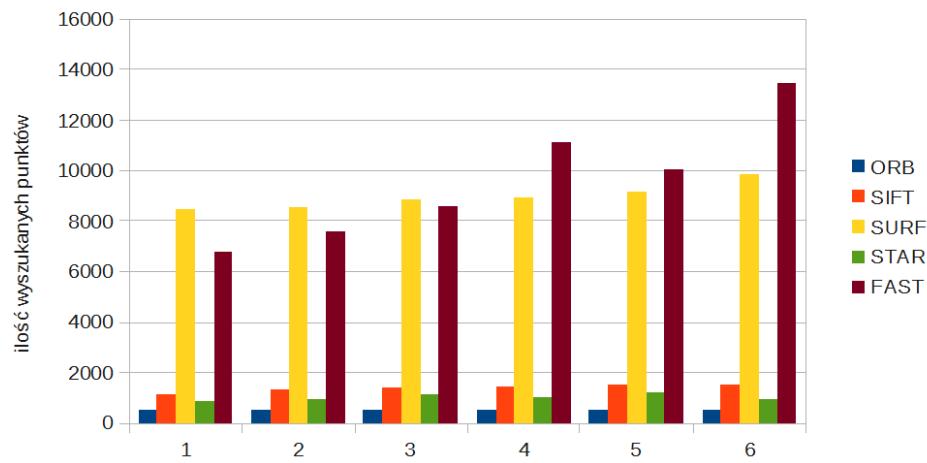
Rysunek 4.14 GRAFFITI - zbiór obrazów testowych

{fig:graffiti}

Tabela. 4.11 GRAFFITI - ilość wyszukanych cech

obraz	ORB	SIFT	SURF	STAR	FAST
1	500	1128	8444	878	6727
2	500	1296	8538	930	7567
3	500	1358	8869	1147	8579
4	500	1445	8888	973	11146
5	500	1500	9165	1164	10053
6	500	1478	9875	910	13459
średnia	500	1368	8963	1000	9589

{tab:graffiti}

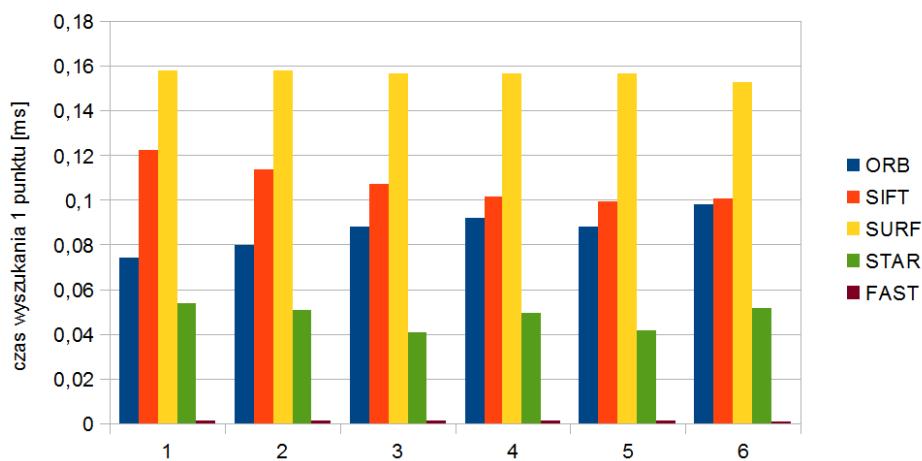


Rysunek 4.15 GRAFFITI - ilość wyszukanych cech

Tabela. 4.12 GRAFFITI - czas lokalizowania pojedynczego punktu charakterystycznego

obraz	ORB	SIFT	SURF	STAR	FAST
-	[ms]	[ms]	[ms]	[ms]	[ms]
1	0,074	0,122	0,158	0,054	0,001
2	0,080	0,113	0,158	0,051	0,001
3	0,088	0,108	0,156	0,041	0,001
4	0,092	0,102	0,156	0,049	0,001
5	0,088	0,099	0,156	0,041	0,001
6	0,098	0,101	0,153	0,052	0,001
średnia	0,087	0,108	0,156	0,048	0,001

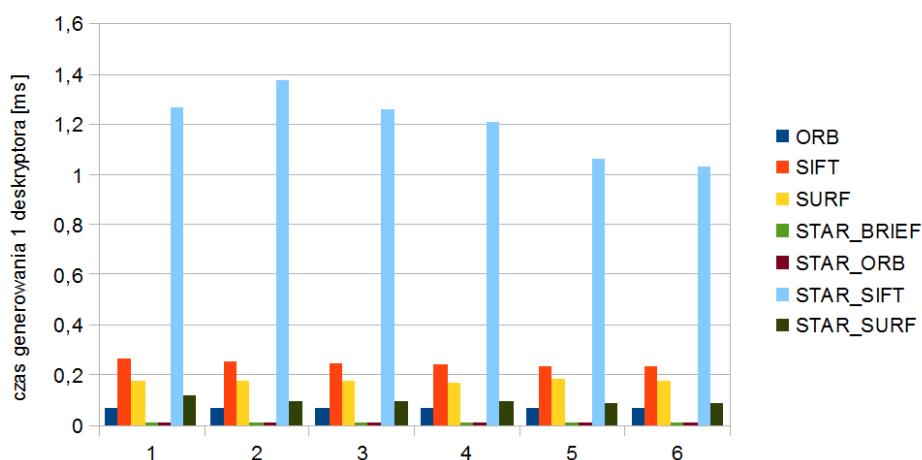
{tab:graffiti}



Rysunek 4.16 GRAFFITI - czas lokalizowania pojedynczego punktu charakterystycznego

Tabela 4.13 GRAFFITI - czas generowania pojedynczego deskryptora punktu charakterystycznego

obraz	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[ms]						
1	0,070	0,269	0,180	0,013	0,014	1,264	0,118
2	0,068	0,255	0,177	0,013	0,014	1,372	0,097
3	0,068	0,244	0,179	0,012	0,013	1,261	0,095
4	0,070	0,239	0,172	0,012	0,013	1,209	0,095
5	0,068	0,235	0,185	0,012	0,013	1,062	0,090
6	0,070	0,235	0,180	0,012	0,014	1,031	0,091
średnia	0,069	0,246	0,179	0,012	0,014	1,200	0,098

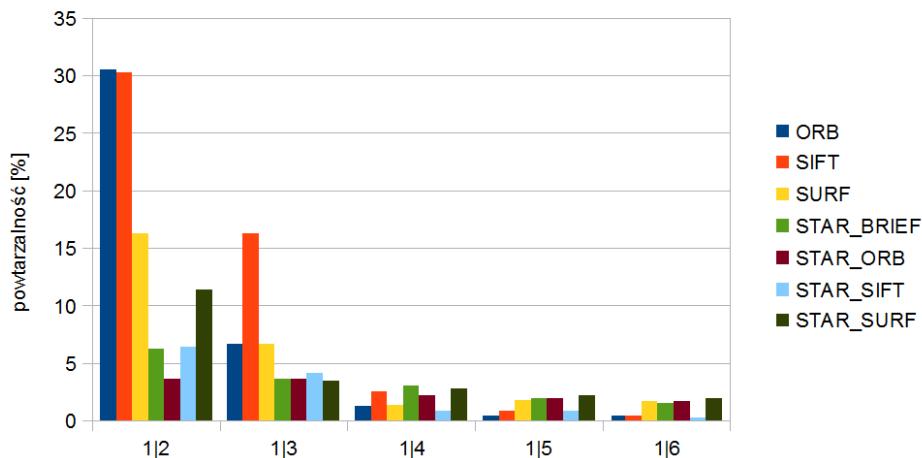


Rysunek 4.17 GRAFFITI - czas generowania pojedynczego deskryptora punktu charakterystycznego

Tabela. 4.14 GRAFFITI - powtarzalność wykrywanych cech

obrazy	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[%]	[%]	[%]	[%]	[%]	[%]	[%]
-	[%]	[%]	[%]	[%]	[%]	[%]	[%]
1 2	31	30	16	6	4	6	11
1 3	7	16	7	4	4	4	3
1 4	1	2	1	3	2	1	3
1 5	0	1	2	2	2	1	2
1 6	0	0	2	2	2	0	2
średnia	8	10	6	3	3	2	4

{tab:graffiti}

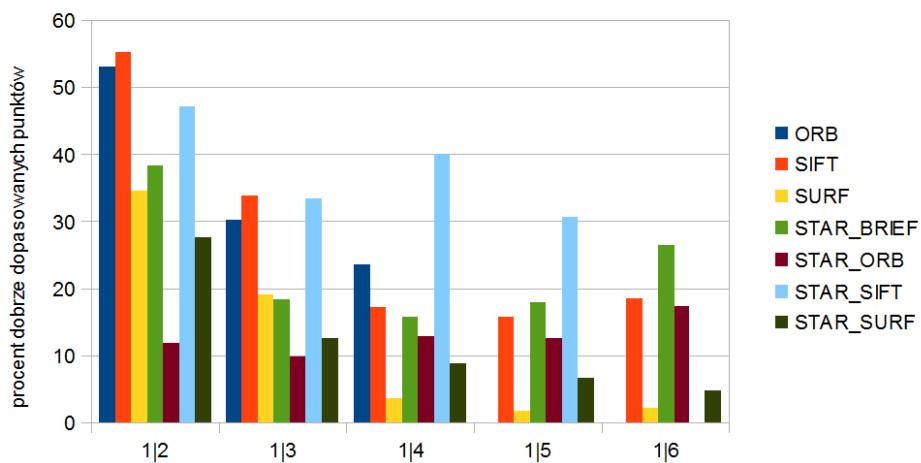


Rysunek 4.18 GRAFFITI - powtarzalność wykrywanych cech

Tabela. 4.15 GRAFFITI - procent poprawnych dopasowań

obrazy	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[%]	[%]	[%]	[%]	[%]	[%]	[%]
1 2	53	55	34	38	12	47	28
1 3	30	34	19	18	10	33	13
1 4	24	17	4	16	13	40	9
1 5	0	16	2	18	13	31	7
1 6	0	19	2	26	17	0	5
średnia	21	28	12	23	13	30	12

{tab:graffiti}



Rysunek 4.19 GRAFFITI - procent poprawnych dopasowań

4.2.2 WALL

Tabela 4.16 WALL - ilość wyszukanych cech

obraz	ORB	SIFT	SURF	STAR	FAST
1	500	1821	14494	1461	36935
2	500	1870	11050	1434	27704
3	500	1840	10763	1330	26384
4	500	1859	10471	1309	27751
5	500	1922	10355	1256	27321
6	500	1919	10118	1302	27181
średnia	500	1872	11209	1349	28879

{tab:wall_f}

Tabela. 4.17 WALL - czas lokalizowania pojedynczego punktu charakterystycznego

obraz	ORB	SIFT	SURF	STAR	FAST
-	[ms]	[ms]	[ms]	[ms]	[ms]
1	0,124	0,108	0,155	0,044	0,001
2	0,102	0,094	0,158	0,038	0,001
3	0,098	0,093	0,159	0,041	0,001
4	0,104	0,092	0,159	0,042	0,001
5	0,102	0,089	0,160	0,044	0,001
6	0,104	0,090	0,160	0,042	0,001
średnia	0,106	0,094	0,158	0,042	0,001

{tab:wall_l}

Tabela. 4.18 WALL - czas generowania pojedynczego deskryptora punktu charakterystycznego

obraz	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[ms]	[ms]	[ms]	[ms]	[ms]	[ms]	[ms]
1	0,064	0,506	0,230	0,012	0,014	0,272	0,081
2	0,054	0,491	0,236	0,012	0,013	0,262	0,080
3	0,054	0,495	0,233	0,012	0,013	0,271	0,080
4	0,054	0,496	0,224	0,012	0,013	0,286	0,082
5	0,056	0,495	0,227	0,012	0,014	0,312	0,082
6	0,056	0,489	0,216	0,012	0,013	0,323	0,082
średnia	0,056	0,495	0,227	0,012	0,013	0,288	0,081

{tab:wall_g}

Tabela. 4.19 WALL - powtarzalność wykrywanych cech

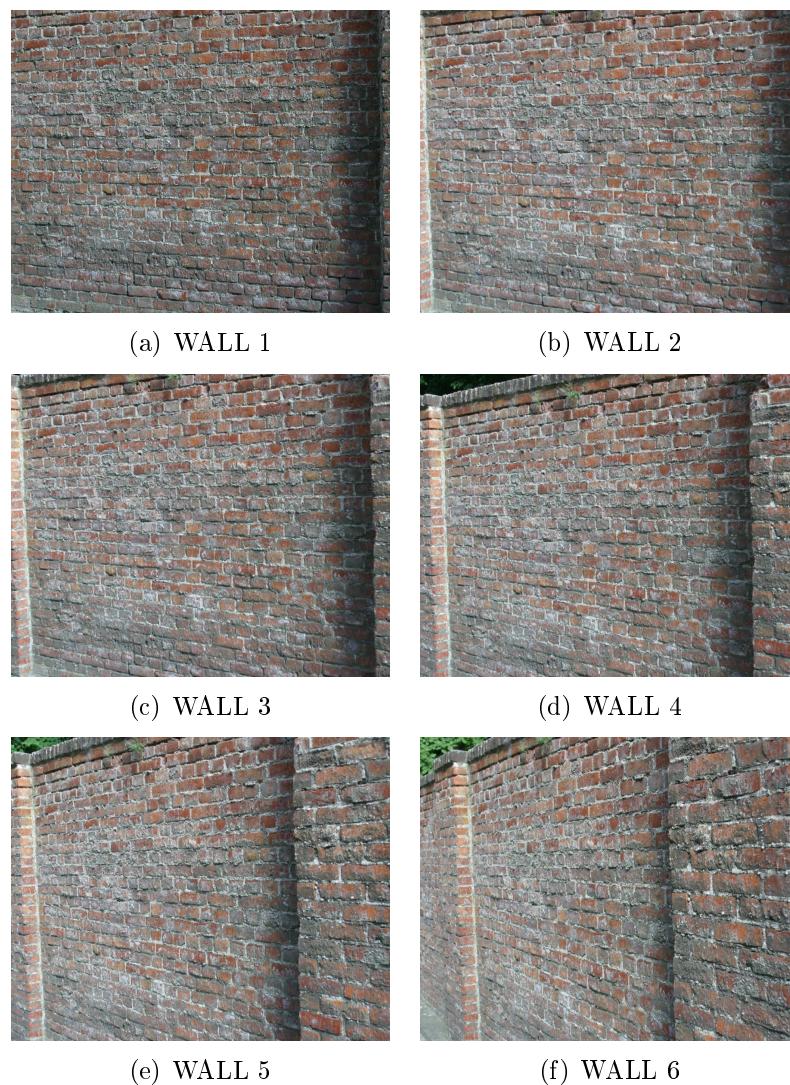
obrazy	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[%]	[%]	[%]	[%]	[%]	[%]	[%]
1 2	32	47	27	59	32	61	6
1 3	22	39	19	43	16	47	2
1 4	7	20	8	23	8	28	1
1 5	1	6	2	9	3	13	1
1 6	1	0	0	1	2	1	1
średnia	13	22	11	27	12	30	2

{tab:wall_r}

Tabela. 4.20 WALL - procent poprawnych dopasowań

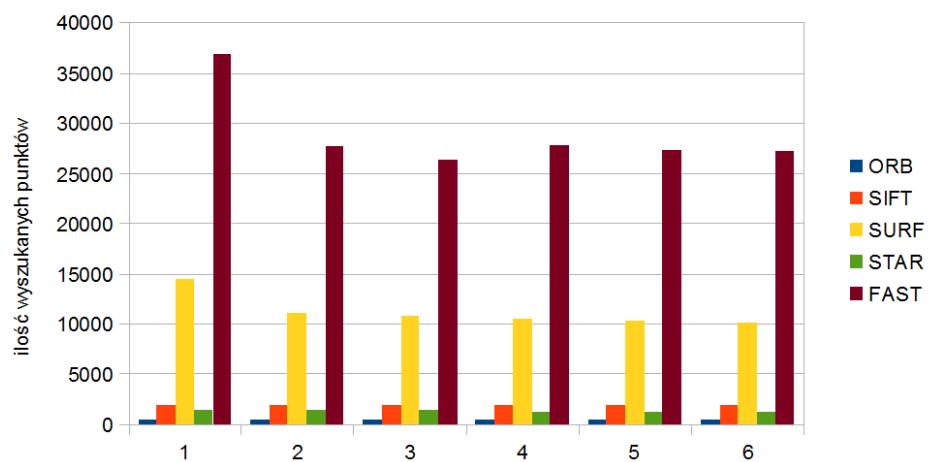
obrazy	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
1 2	65	89	66	71	89	83	61
1 3	65	76	65	62	69	80	28
1 4	51	68	56	61	45	63	16
1 5	50	43	28	50	18	55	10
1 6	67	25	5	25	10	52	8
średnia	59	60	44	53	46	67	25

{tab:wall_m}



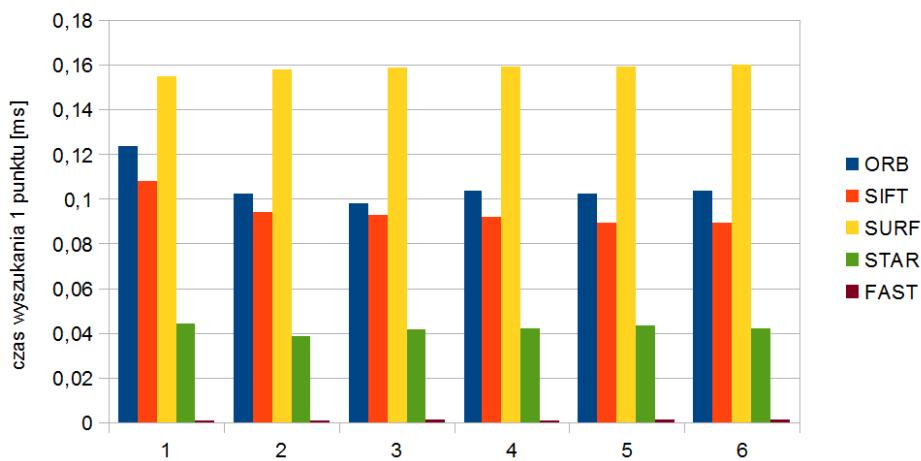
Rysunek 4.20 WALL - zbiór obrazów testowych

{fig:wall_s}



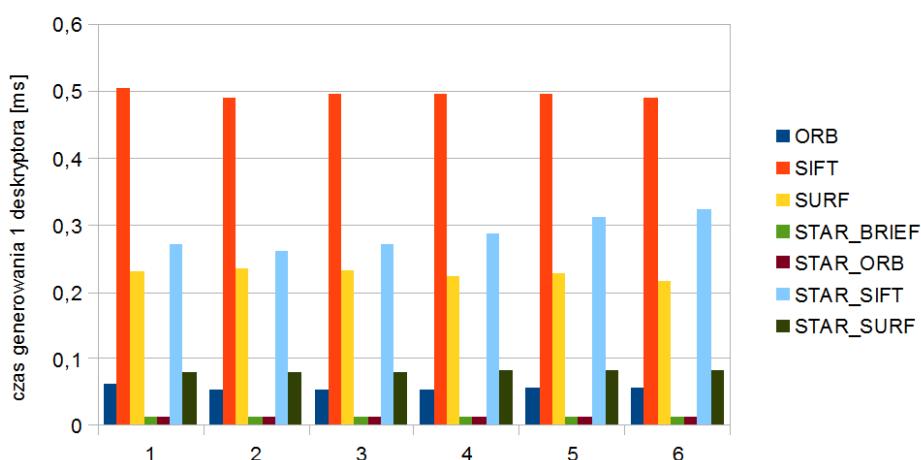
Rysunek 4.21 WALL - ilość wyszukanych cech

{fig:wall_f}



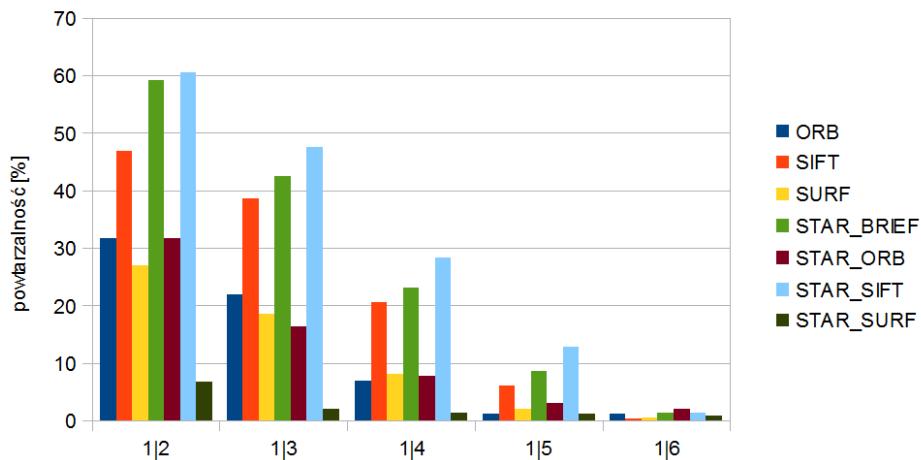
Rysunek 4.22 WALL - czas lokalizowania pojedynczego punktu charakterystycznego

{fig:wall_f}



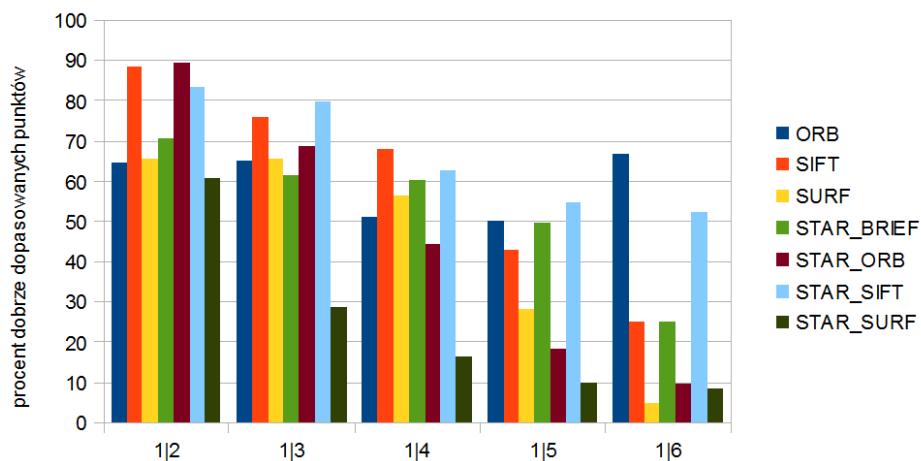
Rysunek 4.23 WALL - czas generowania pojedynczego deskryptora punktu charakterystycznego

{fig:wall_f}



Rysunek 4.24 WALL - powtarzalność wykrywanych cech

{fig:wall_repeatability}



Rysunek 4.25 WALL - procent poprawnych dopasowań

{fig:wall_matches}

4.2.3 Dyskusja wyników

W przypadku zmiany położenia obserwatora algorytmy SURF i FAST górują wyraźnie nad pozostałymi w ilości lokalizowanych punktów charakterystycznych. Dla reszty algorytmów ilość lokalizowanych cech pozostaje na względnie stałym poziomie. Ze względu na różną charakterystykę zbiorów WALL i GRAFFITI ciężko mówić o ujednoliconym trendzie zależnym od zmiany położenia.

Stał czas lokalizowania niezależny od scenerii ma algorytm SURF. Należy jednak zwrócić uwagę, że metoda ta wskaźnik ten ma najwyższy spośród badanych metod. W przypadku pozostałych algorytmów wahania czasu rzadko przekraczają 0,02 milisekundy, co pozwala nie rozpatrywać tych zmian jako bardzo istotnych. Z grupy algorytmów wykorzystywanych w badaniu dopasowań najlepsze wyniki osiąga algorytm STAR, algorytm SIFT i ORB osiągają zbliżone wyniki.

Najkorzystniejszy współczynnik czasu generowania deskryptora osiąga algorytm BRIEF i jego ewolucja w postaci algorytmu ORB. Wyjątkową niestabilnością charakteryzuje się kombinacja algorytmów STAR - SIFT. Metoda ta jest bardzo zależna od badanej sceny. Podobnie jednak jak w przypadku rozmycia obrazów zestawienie tych algorytmów wykazuje się najodporniejszym współczynnikiem powtarzalności w obrazach o "chropowatej" teksturze. Trzeba jednakże zaznaczyć, że wszystkie algorytmy są bardzo wrażliwe na zmianę położenia obserwatora co skutkuje ogromnym spadem powtarzalności.

W przypadku zmiany położenia obserwatora w znaczący sposób spada również procent trafnych dopasowań. Z analizy wyników dla scen WALL i GRAFFITI, widać, że algorytmy lepiej radzą sobie dla zbiorów o "chropowatej" teksturze. Spowodowane to jest faktem, że większość metod opiera swoje działanie o detekcje rogów. Na tym tle najlepiej wypadają algorytmy ORB, SIFT i STAR-SIFT.

4.3 Rotacja i zmiana skali

4.3.1 BARK

Tabela. 4.21 BARK - ilość wyszukanych cech

obraz	ORB	SIFT	SURF	STAR	FAST
1	500	1942	4958	892	15151
2	500	1986	4809	784	13496
3	500	1729	5482	1015	17422
4	500	1672	5773	1014	18678
5	500	1593	5887	854	18524
6	500	1643	6226	1063	20723
średnia	500	1761	5523	937	17332

{tab:bark_1}

Tabela. 4.22 BARK - czas lokalizowania pojedynczego punktu charakterystycznego

obraz	ORB	SIFT	SURF	STAR	FAST
-	[ms]	[ms]	[ms]	[ms]	[ms]
1	0,096	0,065	0,169	0,037	0,001
2	0,082	0,064	0,171	0,040	0,001
3	0,098	0,071	0,166	0,032	0,001
4	0,102	0,074	0,163	0,032	0,001
5	0,098	0,076	0,163	0,036	0,001
6	0,110	0,074	0,162	0,030	0,001
średnia	0,098	0,071	0,166	0,034	0,001

{tab:bark_f}

Tabela. 4.23 BARK - czas generowania pojedynczego deskryptora punktu charakterystycznego

obraz	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[ms]	[ms]	[ms]	[ms]	[ms]	[ms]	[ms]
1	0,056	0,216	0,199	0,012	0,012	0,855	0,092
2	0,056	0,207	0,205	0,011	0,013	0,810	0,091
3	0,056	0,211	0,205	0,012	0,013	0,710	0,089
4	0,058	0,217	0,199	0,012	0,013	0,624	0,086
5	0,056	0,220	0,202	0,012	0,014	0,562	0,087
6	0,056	0,216	0,201	0,012	0,012	0,568	0,087
średnia	0,056	0,215	0,202	0,012	0,013	0,688	0,089

{tab:bark_f}

Tabela. 4.24 BARK - powtarzalność wykrywanych cech

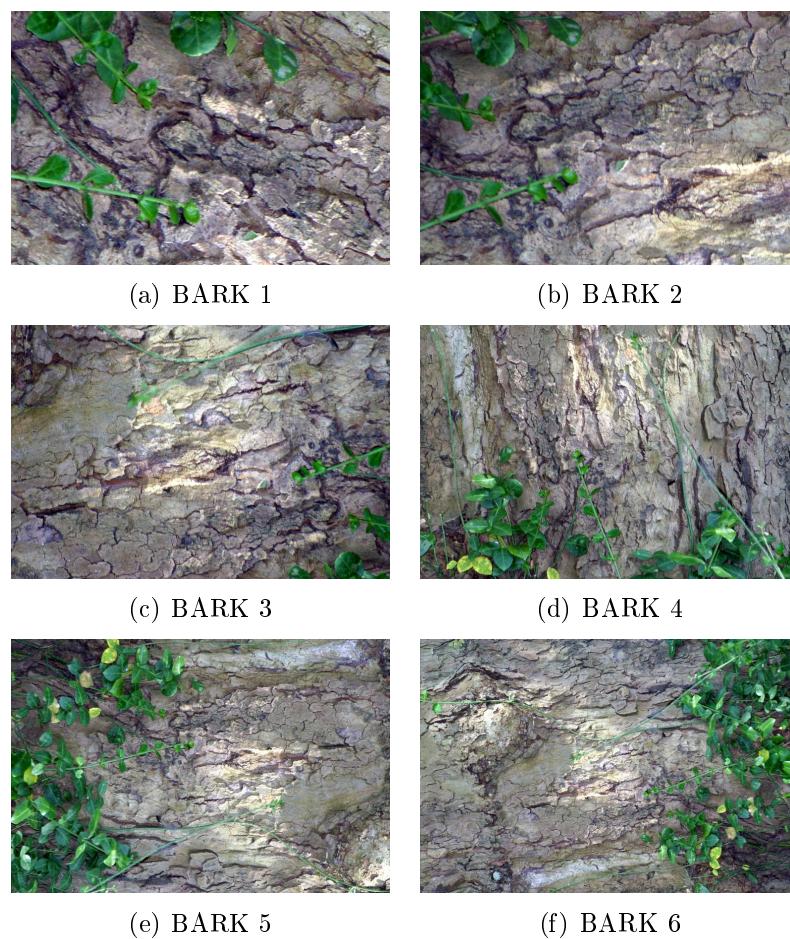
obrazy	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[%]	[%]	[%]	[%]	[%]	[%]	[%]
1 2	15	19	10	2	3	0	2
1 3	8	10	6	1	2	0	2
1 4	6	11	5	2	1	0	2
1 5	3	8	2	1	2	0	2
1 6	1	4	2	1	2	0	2
średnia	6	10	5	1	2	0	2

{tab:bark_f}

Tabela. 4.25 BARK - procent poprawnych dopasowań

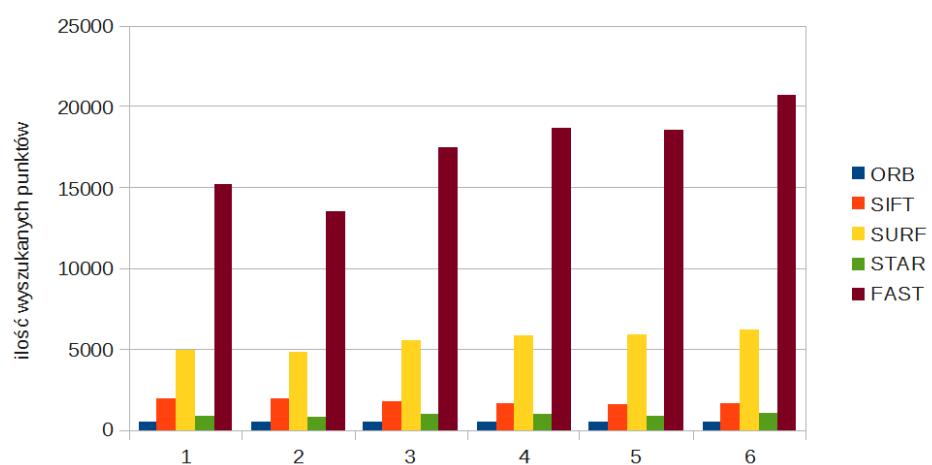
obrazy	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[%]	[%]	[%]	[%]	[%]	[%]	[%]
1 2	49	50	46	24	12	0	18
1 3	33	43	38	21	18	0	22
1 4	44	80	28	19	19	0	20
1 5	36	78	14	21	14	0	23
1 6	0	71	12	19	18	0	8
średnia	33	64	28	21	16	0	18

{tab:bark_f}



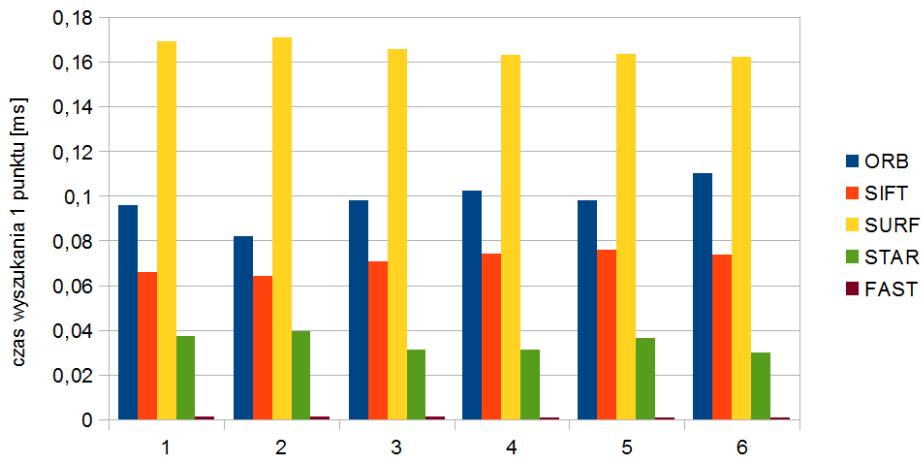
Rysunek 4.26 BARK - zbiór obrazów testowych

{fig:bark_s}



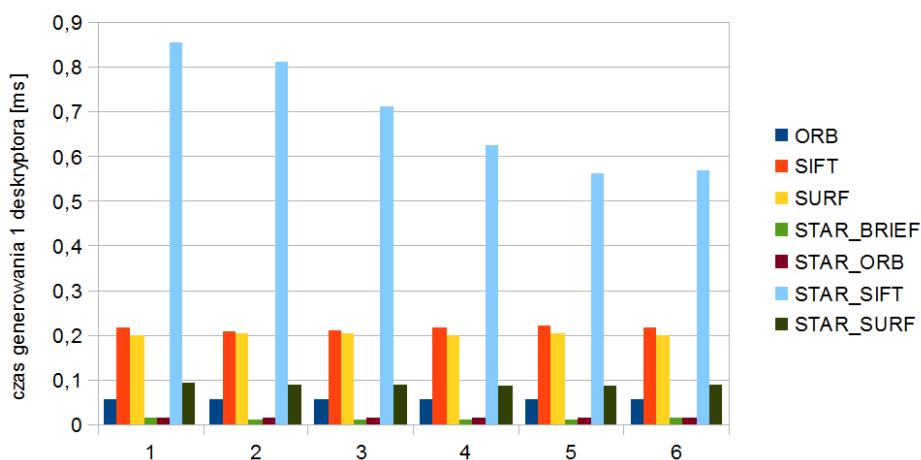
Rysunek 4.27 BARK - ilość wyszukanych cech

{fig:bark_f}



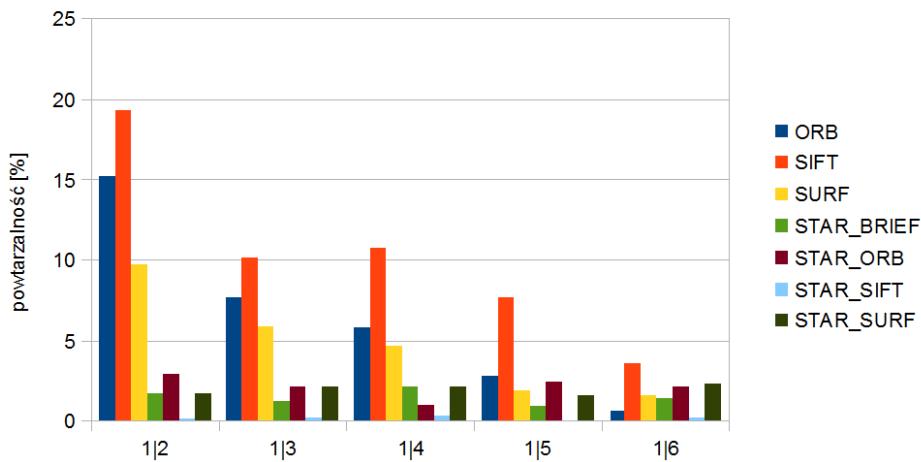
Rysunek 4.28 BARK - czas lokalizowania pojedynczego punktu charakterystycznego

{fig:bark_1}



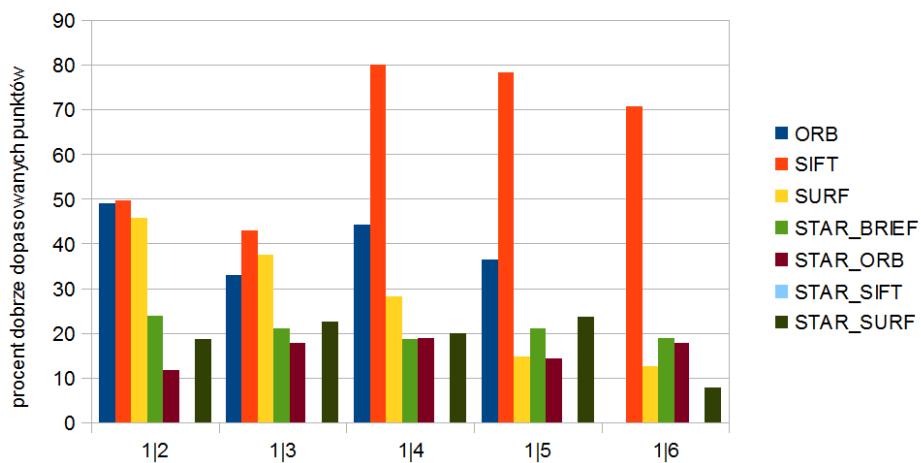
Rysunek 4.29 BARK - czas generowania pojedynczego deskryptora punktu charakterystycznego

{fig:bark_1}



Rysunek 4.30 BARK - powtarzalność wykrywanych cech

{fig:bark_repeatability}



Rysunek 4.31 BARK - procent poprawnych dopasowań

{fig:bark_matches}

4.3.2 BOAT



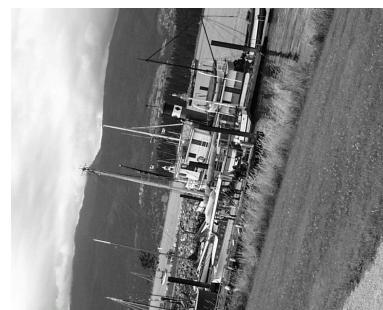
(a) BOAT 1



(b) BOAT 2



(c) BOAT 3



(d) BOAT 4



(e) BOAT 5



(f) BOAT 6

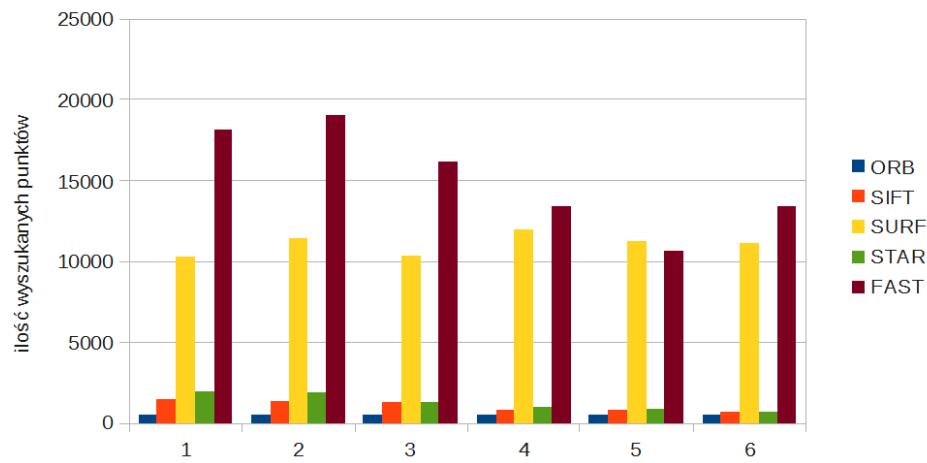
Rysunek 4.32 BOAT - zbiór obrazów testowych

{fig:boat_s}

Tabela. 4.26 BOAT - ilość wyszukanych cech

obraz	ORB	SIFT	SURF	STAR	FAST
1	500	1451	10302	1923	18122
2	500	1352	11467	1853	19076
3	500	1238	10375	1306	16222
4	500	727	11912	924	13377
5	500	726	11274	896	10668
6	500	687	11099	658	13394
średnia	500	1030	11072	1260	15143

{tab:boat_f}



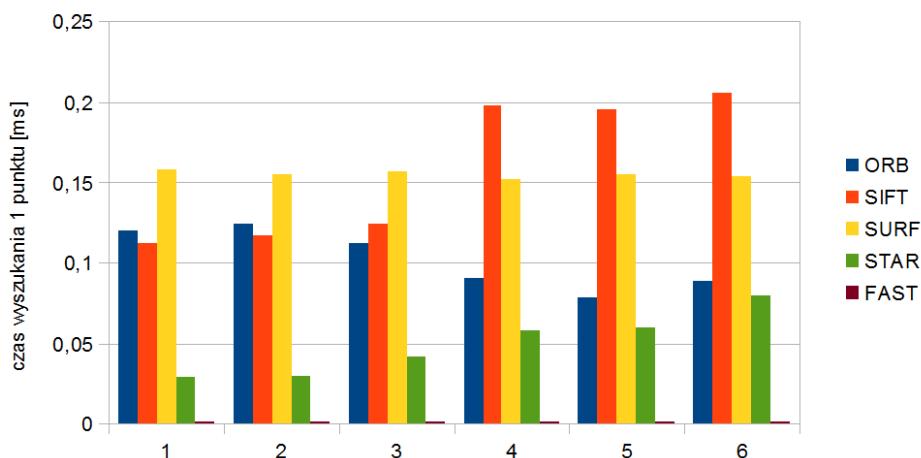
Rysunek 4.33 BOAT - ilość wyszukanych cech

{fig:boat_f}

Tabela. 4.27 BOAT - czas lokalizowania pojedynczego punktu charakterystycznego

obraz	ORB	SIFT	SURF	STAR	FAST
-	[ms]	[ms]	[ms]	[ms]	[ms]
1	0,120	0,112	0,158	0,029	0,001
2	0,124	0,118	0,155	0,030	0,001
3	0,112	0,124	0,156	0,041	0,001
4	0,090	0,198	0,152	0,057	0,001
5	0,078	0,196	0,155	0,059	0,001
6	0,088	0,205	0,154	0,079	0,001
średnia	0,102	0,159	0,155	0,049	0,001

{tab:boat_f}



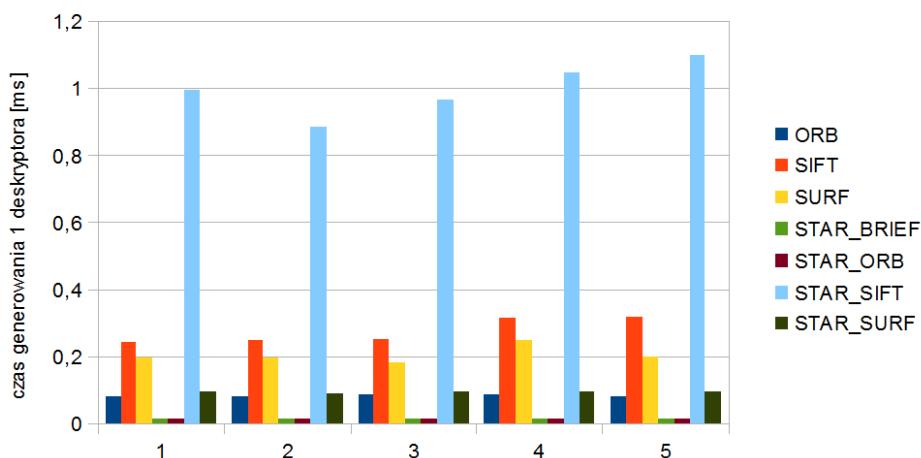
Rysunek 4.34 BOAT - czas lokalizowania pojedynczego punktu charakterystycznego

{fig:boat_f1}

Tabela. 4.28 BOAT - czas generowania pojedynczego deskryptora punktu charakterystycznego

obraz	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[ms]						
1	0,082	0,241	0,196	0,011	0,013	0,996	0,093
2	0,082	0,249	0,195	0,012	0,013	0,887	0,091
3	0,084	0,254	0,183	0,012	0,014	0,969	0,093
4	0,084	0,315	0,249	0,013	0,015	1,049	0,093
5	0,082	0,320	0,202	0,012	0,016	1,103	0,094
6	0,084	0,332	0,185	0,014	0,017	1,195	0,096
średnia	0,083	0,285	0,202	0,012	0,015	1,033	0,093

{tab:boat_f1}



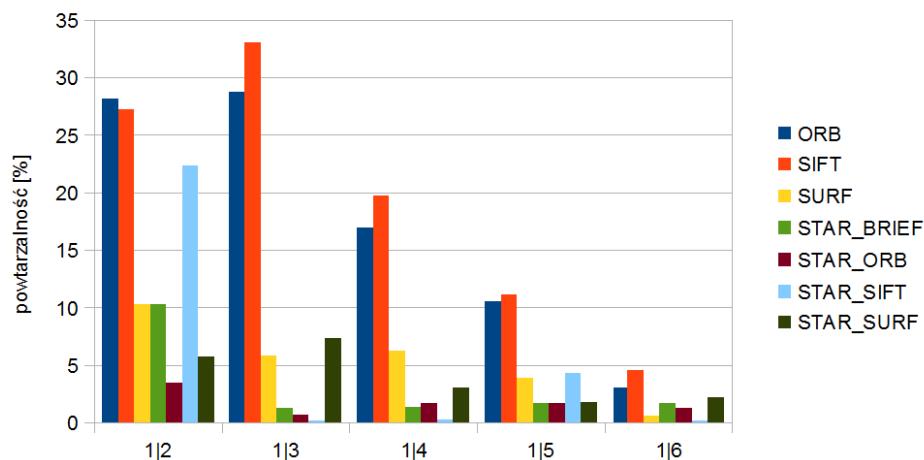
Rysunek 4.35 BOAT - czas generowania pojedynczego deskryptora punktu charakterystycznego

{fig:boat_f1}

Tabela. 4.29 BOAT - powtarzalność wykrywanych cech

obrazy	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[%]	[%]	[%]	[%]	[%]	[%]	[%]
1 2	28	27	10	10	3	22	6
1 3	29	33	6	1	1	0	7
1 4	17	20	6	1	2	0	3
1 5	11	11	4	2	2	4	2
1 6	3	5	1	2	1	0	2
średnia	18	19	5	3	2	5	4

{tab:boat_r}



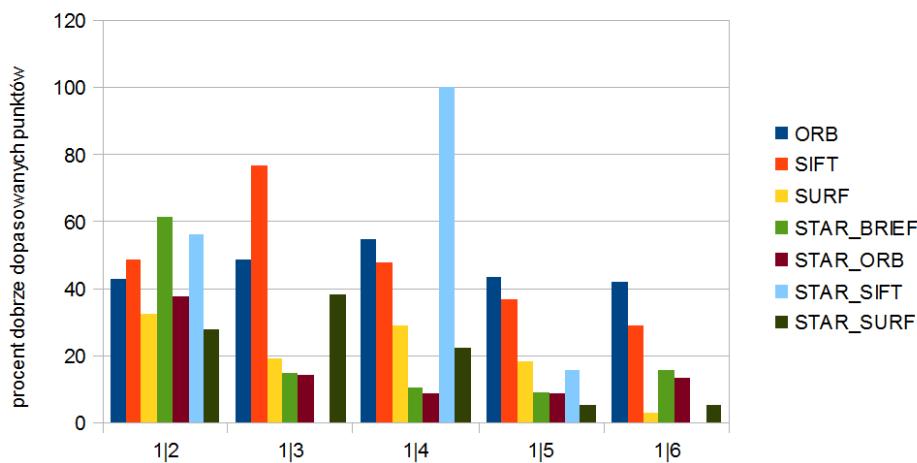
Rysunek 4.36 BOAT - powtarzalność wykrywanych cech

{fig:boat_r}

Tabela. 4.30 BOAT - procent poprawnych dopasowań

obrazy	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[%]	[%]	[%]	[%]	[%]	[%]	[%]
1 2	43	49	32	61	38	56	28
1 3	49	77	19	15	14	0	38
1 4	55	48	29	10	9	100	22
1 5	43	37	18	9	9	15	5
1 6	42	29	3	16	13	0	5
średnia	46	48	20	22	16	34	20

{tab:boat_r}



Rysunek 4.37 BOAT - procent poprawnych dopasowań

{fig:boat_m}

4.3.3 Dyskusja wyników

Dla zestawów poddanych rotacji i zmianie skali możemy zaobserwować większość trendów widocznych w poprzednich próbach. Największe wahnięcia w ilości lokalizowanych punktów obserwujemy dla algorytmu FAST. Pozostałe metody lokalizują zblzoną liczbę punktów w obrębie zestawów.

Średni czas lokalizowania cechy dla większości algorytmów jest stały niezależnie od zbioru. Najstabilniejszy pod tym względem jest algorytm ORB. Wyniki czasowe algorytmu SIFT są najbardziej zróżnicowane. Algorytm ten osiąga jednakże najlepsze wyniki pod względem powtarzalności lokalizowanych cech.

Pod względem czasu generowanie deskryptora zachowana zostaje zachowana klasyfikacja algorytmów jak w przypadku wcześniejszych zbiorów. Tradycyjnie najgorsze wyniki w tym polu osiąga kombinacja STAR-SIFT. O ile w przypadku wcześniejszych zbiorów kombinacja ta osiągała dobre wyniki w obszarze dopasowań, o tyle dla zbiorów BARK i BOAT, algorytm nie działa.

Rotacja połączona ze zmianą skali jest najbardziej wymagającym zbiorem i wyniki osiągane przez wszystkie algorytmy są słabe. Po analizie wyników zdecydowano się w przypadku badań obszarów skalnych zbiór ten podzielić na dwa, reprezentujące jeden typ przekształcenia.

4.4 Zmiana oświetlenia

4.4.1 CARS

Tabela. 4.31 CARS - ilość wyszukanych cech

obraz	ORB	SIFT	SURF	STAR	FAST
1	500	688	12062	477	9431
2	500	595	12278	299	7216
3	500	515	12234	253	6301
4	500	481	12141	219	5806
5	500	463	12215	204	5336
6	500	445	12008	190	4974
średnia	500	531	12156	274	6511

{tab:cars_f}

Tabela. 4.32 CARS - czas lokalizowania pojedynczego punktu charakterystycznego

obraz	ORB	SIFT	SURF	STAR	FAST
-	[ms]	[ms]	[ms]	[ms]	[ms]
1	0,066	0,199	0,150	0,105	0,001
2	0,058	0,220	0,150	0,167	0,001
3	0,056	0,258	0,149	0,198	0,001
4	0,054	0,277	0,150	0,247	0,001
5	0,054	0,283	0,150	0,230	0,001
6	0,052	0,299	0,150	0,247	0,001
średnia	0,057	0,256	0,150	0,199	0,001

{tab:cars_f}

Tabela. 4.33 CARS - czas generowania pojedynczego deskryptora punktu charakterystycznego

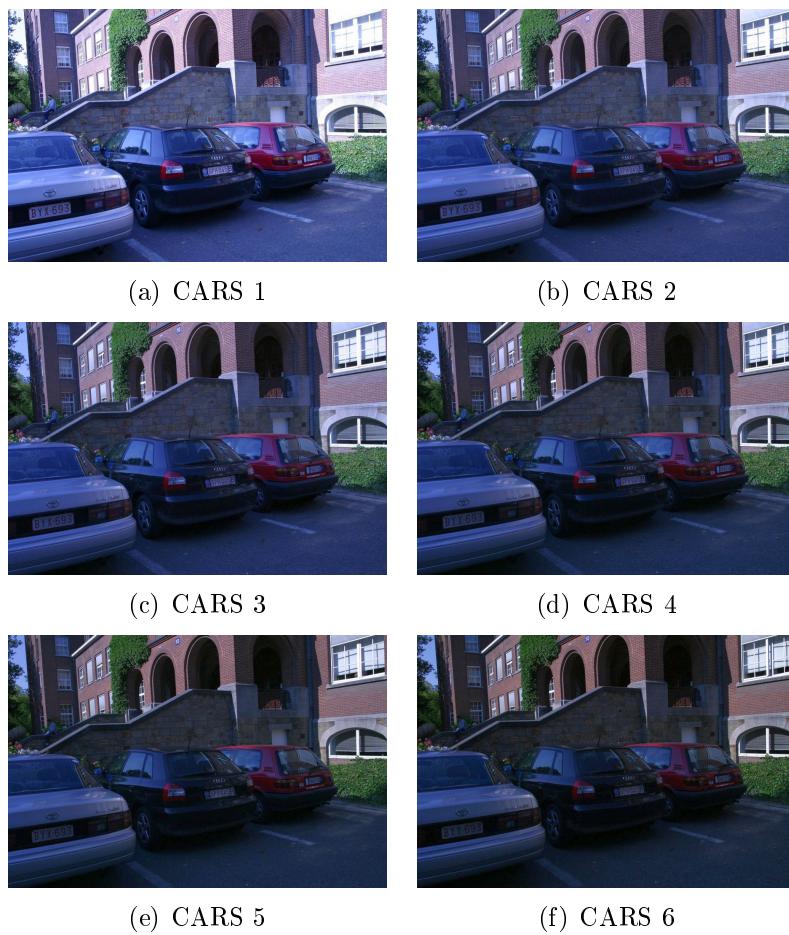
obraz	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[ms]	[ms]	[ms]	[ms]	[ms]	[ms]	[ms]
1	0,072	0,320	0,178	0,015	0,019	1,537	0,101
2	0,072	0,343	0,180	0,017	0,023	1,722	0,104
3	0,070	0,367	0,181	0,020	0,028	1,846	0,107
4	0,072	0,391	0,182	0,018	0,032	1,968	0,105
5	0,072	0,389	0,183	0,025	0,029	1,917	0,108
6	0,072	0,409	0,185	0,021	0,032	1,900	0,105
średnia	0,072	0,370	0,181	0,019	0,027	1,815	0,105

{tab:cars_f}

Tabela. 4.34 CARS - powtarzalność wykrywanych cech

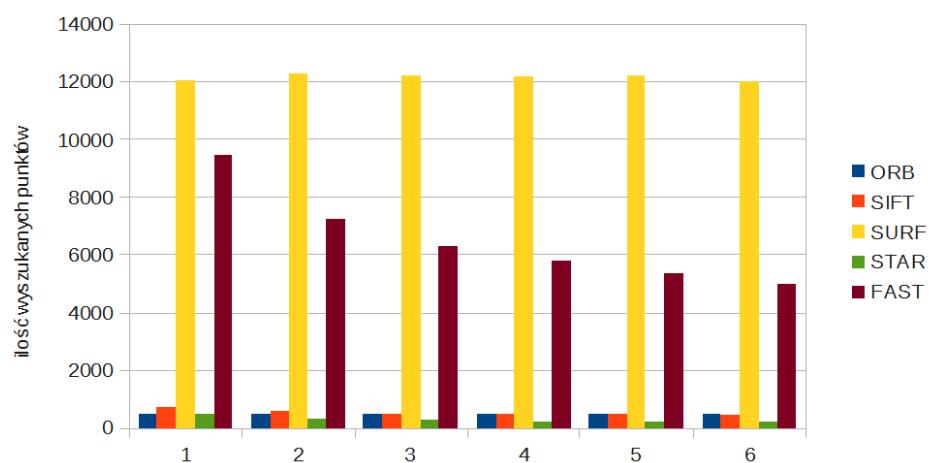
obrazy	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[%]	[%]	[%]	[%]	[%]	[%]	[%]
1 2	33	54	39	51	51	53	25
1 3	20	48	34	46	44	48	21
1 4	17	41	34	39	39	40	20
1 5	11	38	31	40	36	37	18
1 6	7	36	22	31	30	31	11
średnia	18	43	32	41	40	42	19

{tab:cars_f}



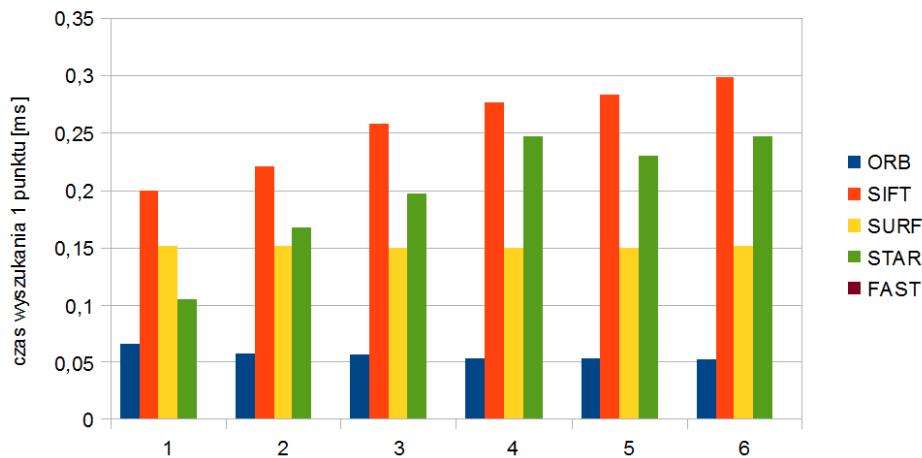
Rysunek 4.38 CARS -Zbiór obrazów testowych

{fig:cars_1}



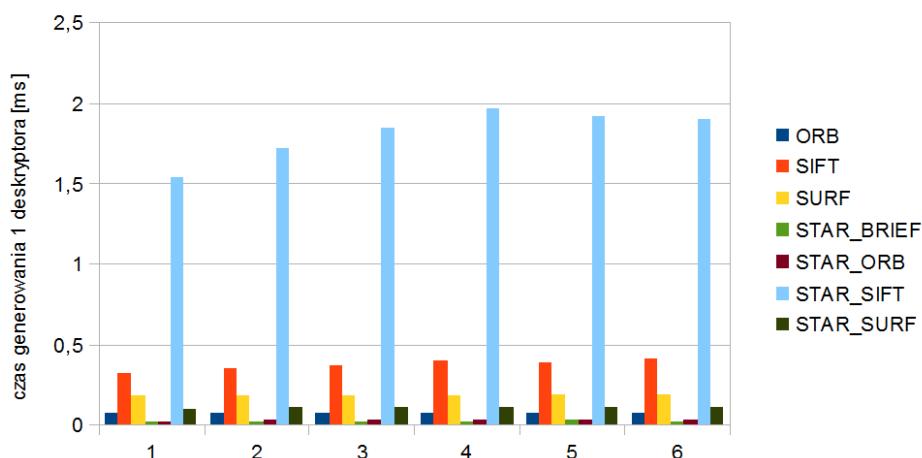
Rysunek 4.39 CARS -ilość wyszukanych cech

{fig:cars f



Rysunek 4.40 CARS - czas lokalizowania pojedynczego punktu charakterystycznego

{fig:cars_f}



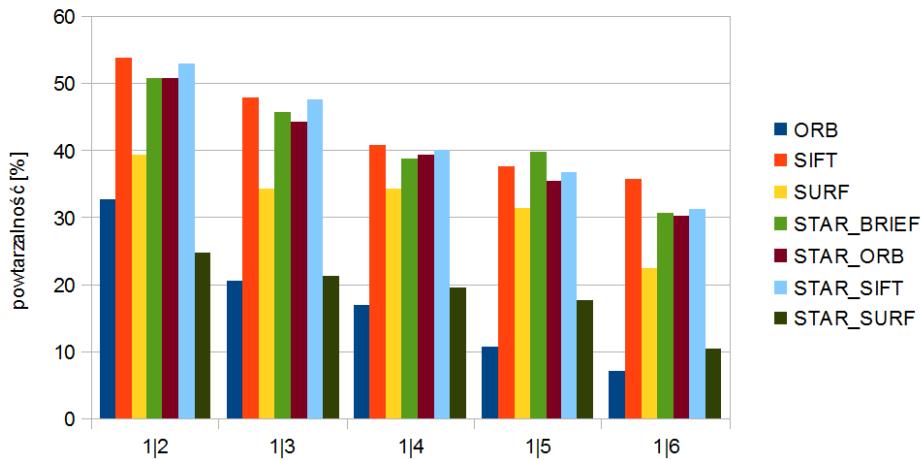
Rysunek 4.41 CARS - czas generowania pojedynczego deskryptora punktu charakterystycznego

{fig:cars_i}

Tabela 4.35 CARS - procent poprawnych dopasowań

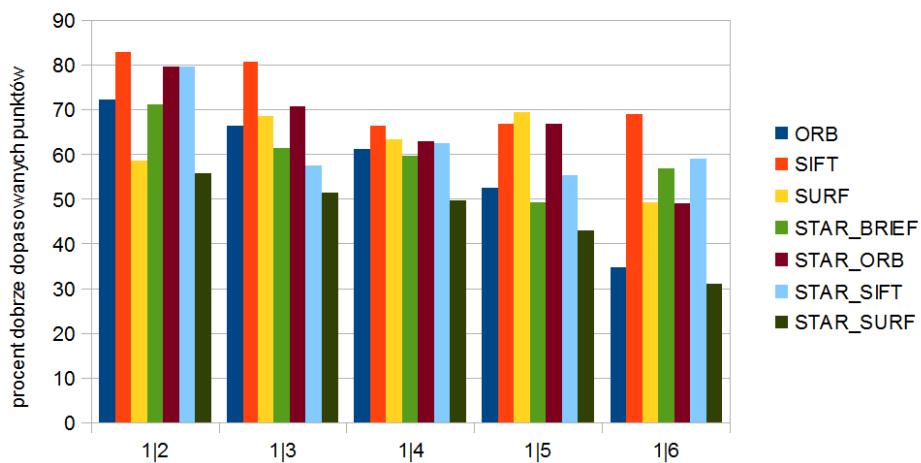
obrazy	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[%]	[%]	[%]	[%]	[%]	[%]	[%]
1 2	72	83	59	71	80	80	56
1 3	66	81	68	61	71	58	51
1 4	61	66	63	60	63	62	50
1 5	53	67	69	49	67	55	43
1 6	35	69	49	57	49	59	31
średnia	57	73	62	60	66	63	46

{tab:cars_r}



Rysunek 4.42 CARS - powtarzalność wykrywanych cech

{fig:cars_repe}



Rysunek 4.43 CARS - procent poprawnych dopasowań

{fig:cars_mis}

4.4.2 Dyskusja wyników

Większość algorytmów badanych w tym zbiorze wykazuje wrażliwość na stopień oświetlenia. Przejawia się to spadkiem ilości lokalizowanych cech. Spadek ten jest szczególnie widoczny w przypadku algorytmu FAST. Wyjątkiem w tym zestawieniu jest algorytm SURF, którego ilość lokalizowanych cech wydaje się być niezależna od oświetlenia.

Podobnie sytuacja kształtuje się w obszarze charakterystyk czasowych. O ile większość algorytmów wraz z przyciemnieniem zdjęcia potrzebuje więcej czasu na zlokalizowanie punktu, SURF ma stały czas wykonania zadania.

Czas generowanie deskryptorów w obrębie zbioru wydaje się być stabilny dla wszystkich algorytmów. Najlepsze wyniki w tym obszarze osiągają kombinacje algorytmów STAR z deskryptorami BRIEF i ORB oraz kompletny algorytm ORB. Ponownie możemy zaobserwować długi czas generowania deskryptora STAR-SIFT.

Najwyższy procent powtarzalności posiada algorytm SIFT. Wolna kombinacja STAR-SIFT również osiąga wysokie parametry powtarzalności. Dobre wskaźniki posiadają również algorytmy SURF, STAR-BRIEF i STAR-ORB.

Wspomniane algorytmy również osiągają zadowalające poprawności dopasowań. Naj słabiej pod tym względem wypadają algorytmy ORB i STAR-SURF.

Rozdział 5

Zbiór testowy obrazów skalnych

Głównym celem niniejszej pracy był dobór metod selekcji punktów charakterystycznych pod kątem rozpoznawania obrazów skalnych. W tym celu opracowano autorski zbiór zestawów testowych służący ocenie pracy algorytmów. Zdjęcie powstały w lecie 2012 roku, w rejonie Gór Sokolich na Dolnym Śląsku. Zbiór składa się z 6 zestawów odpowiadających transformacjom takim jak:

- rotacja
- zmiana położenia obserwatora
- zmiana skali

5.1 Rotacja

Rotacja jest reprezentowana przez 3 zbiory. Zbiory te różnią się między sobą skala wykadrowania zdjęcia i stopniem oświetlenia. Zostały wykonane w rejonie formacji Sukiennic, Krzywej Turni i Małego Sokolika.

5.1.1 R1

Transformacja: Rotacja

Rejon: Sukiennice

Powiększenie: Duże

Oświetlenie: Duże

Rozdzielcość: 768×768

Tabela. 5.1 R1 - ilość wyszukanych cech

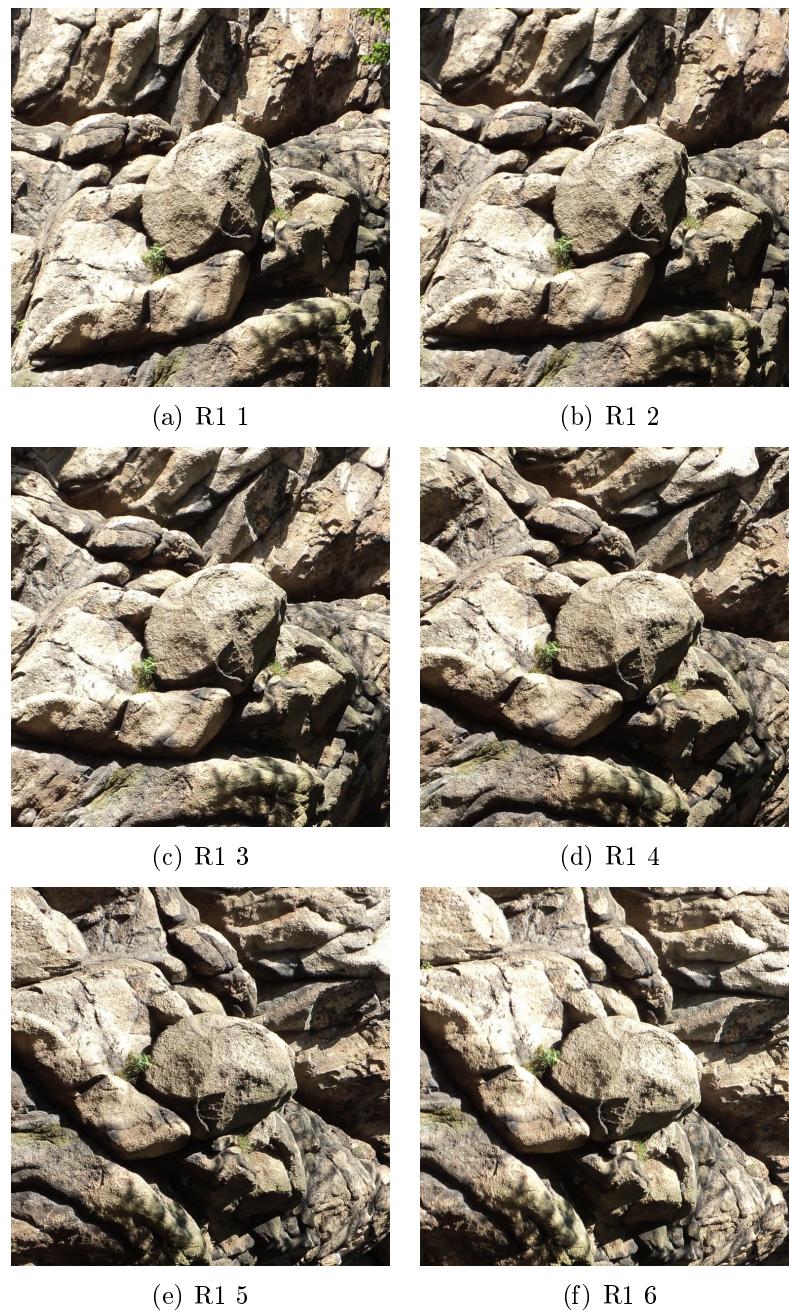
obraz	ORB	SIFT	SURF	STAR	FAST
1	500	1795	8212	1545	24151
2	500	1762	8059	1555	23688
3	500	1729	8560	1566	25404
4	500	1749	8951	1601	25762
5	500	1678	8952	1591	26614
6	500	1827	8240	1596	23200
średnia	500	1757	8496	1576	24803

{tab:r1_f1}

Tabela. 5.2 R1 - czas lokalizowania pojedynczego punktu charakterystycznego

obraz	ORB	SIFT	SURF	STAR	FAST
-	[ms]	[ms]	[ms]	[ms]	[ms]
1	0,152	0,096	0,166	0,039	0,001
2	0,150	0,099	0,166	0,039	0,001
3	0,154	0,099	0,165	0,038	0,001
4	0,158	0,101	0,164	0,038	0,001
5	0,158	0,103	0,163	0,038	0,001
6	0,152	0,095	0,166	0,038	0,001
średnia	0,154	0,099	0,165	0,038	0,001

{tab:r1_f2}



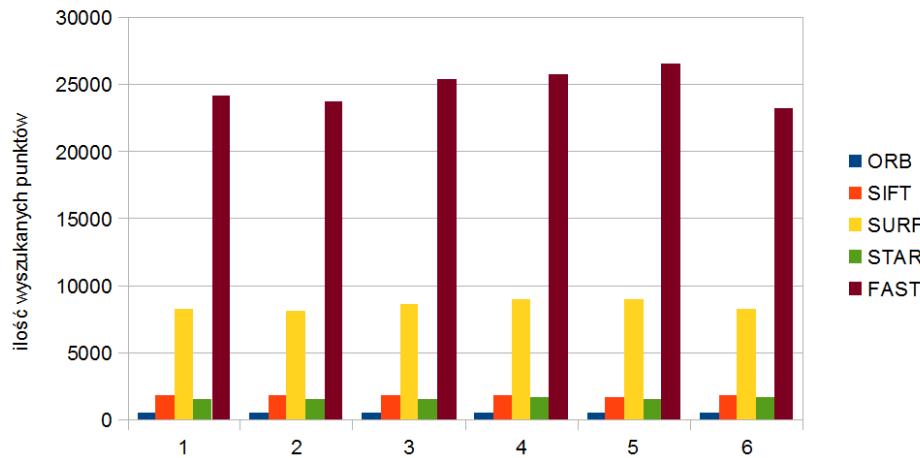
Rysunek 5.1 R1 - zbiór obrazów testowych

{fig:r1_set}

Tabela. 5.3 R1 - czas generowania pojedynczego deskryptora punktu charakterystycznego

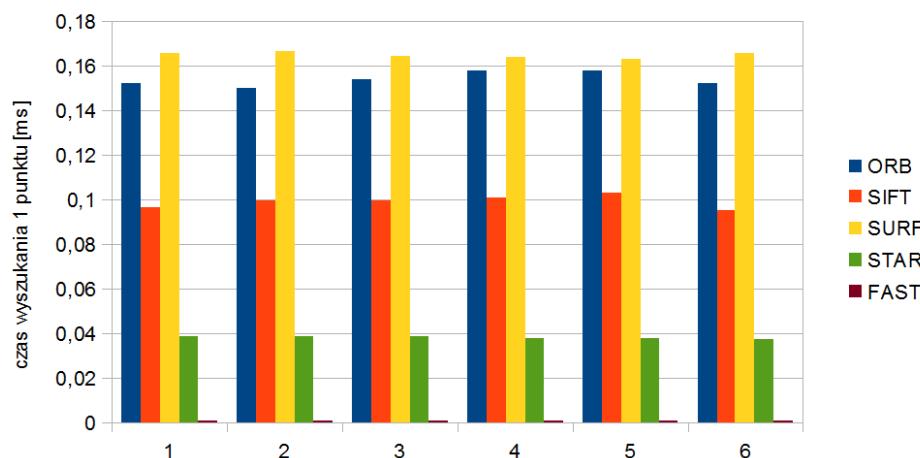
obraz	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[ms]	[ms]	[ms]	[ms]	[ms]	[ms]	[ms]
1	0,078	0,233	0,187	0,012	0,012	1,197	0,098
2	0,080	0,236	0,186	0,012	0,013	1,194	0,098
3	0,082	0,239	0,187	0,012	0,012	1,238	0,099
4	0,078	0,236	0,185	0,012	0,012	1,213	0,097
5	0,078	0,240	0,185	0,013	0,013	1,200	0,099
6	0,080	0,230	0,190	0,013	0,013	1,274	0,098
średnia	0,079	0,236	0,187	0,012	0,012	1,219	0,098

{tab:r1_f3}



Rysunek 5.2 R1 - ilość wyszukanych cech

{fig:r1_f1}



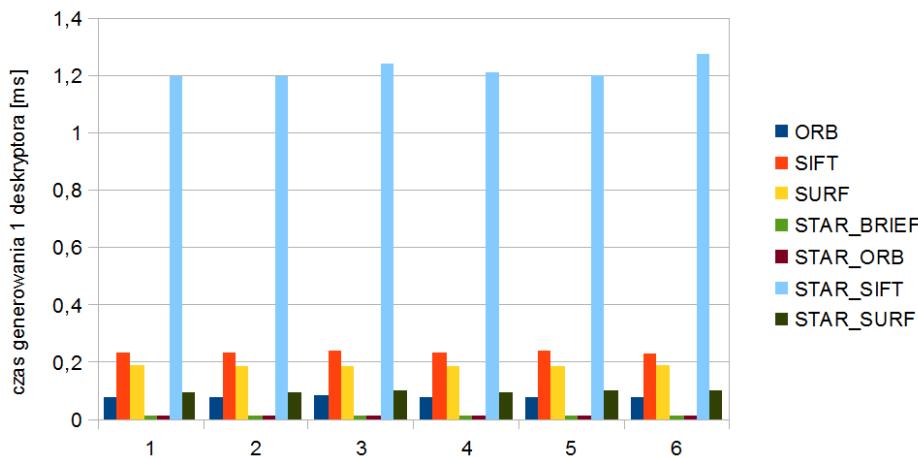
Rysunek 5.3 R1 - czas lokalizowania pojedynczego punktu charakterystycznego

{fig:r1_f2}

Tabela. 5.4 R1 - powtarzalność wykrywanych cech

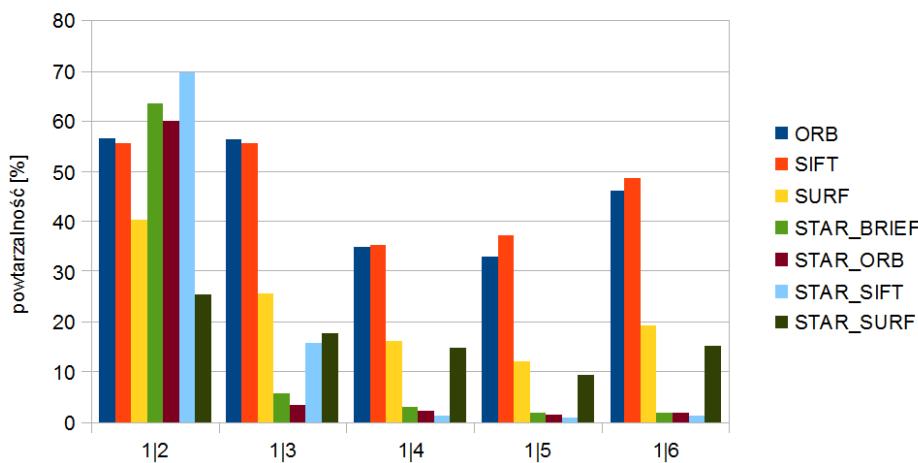
obrazy	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[%]	[%]	[%]	[%]	[%]	[%]	[%]
1 2	57	56	40	63	60	70	25
1 3	56	56	26	6	3	16	18
1 4	35	35	16	3	2	1	15
1 5	33	37	12	2	2	1	9
1 6	46	49	19	2	2	1	15
średnia	45	46	23	15	14	18	17

{tab:r1_m1}



Rysunek 5.4 R1 - czas generowania pojedynczego deskryptora punktu charakterystycznego

{fig:r1_f3}



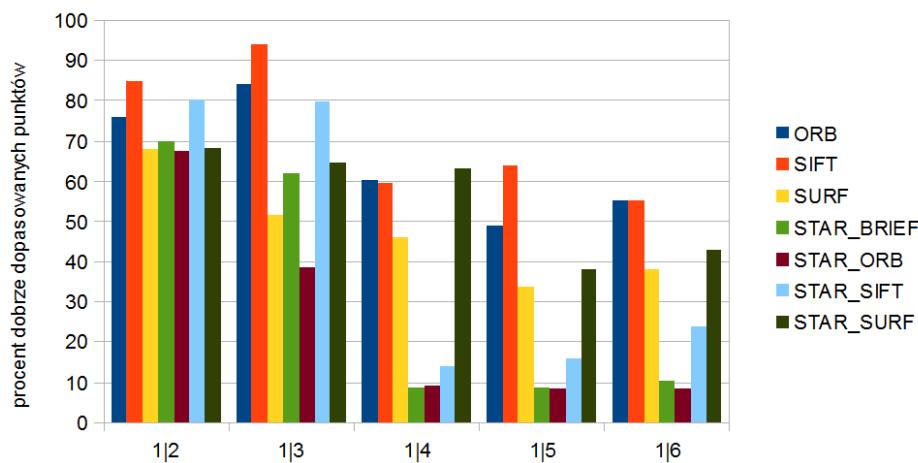
Rysunek 5.5 R1 - powtarzalność wykrywanych cech

{fig:r1_m1}

Tabela. 5.5 R1 - procent poprawnych dopasowań

obrazy	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[%]	[%]	[%]	[%]	[%]	[%]	[%]
1 2	76	85	68	70	68	80	68
1 3	84	94	52	62	39	80	65
1 4	60	60	46	8	9	14	63
1 5	49	64	34	9	8	16	38
1 6	55	55	38	10	8	24	43
średnia	65	71	47	32	26	43	55

{tab:r1_m2}



Rysunek 5.6 R1 - procent poprawnych dopasowań

{fig:r1_m2}

5.1.2 R2

Transformacja: Rotacja

Rejon: Sokolik Mały

Powiększenie: Małe

Oświetlenie: Duże

Rozdzielczość: 768×1024

Tabela. 5.6 ilość wyszukanych cech

obraz	ORB	SIFT	SURF	STAR	FAST
1	500	2879	12776	2834	39644
2	500	2844	14042	2852	47948
3	500	2874	13905	2912	48224
4	500	2741	13416	2543	44424
5	500	2686	12601	2343	40263
6	500	2600	12938	2274	40797
średnia	500	2771	13280	2626	43550

{tab:r2_f1}

Tabela. 5.7 R2 - czas lokalizowania pojedynczego punktu charakterystycznego w obrazach z zestawu R2

obraz	ORB	SIFT	SURF	STAR	FAST
-	[ms]	[ms]	[ms]	[ms]	[ms]
1	0,214	0,083	0,162	0,030	0,001
2	0,244	0,085	0,159	0,029	0,001
3	0,244	0,082	0,555	0,029	0,001
4	0,228	0,088	0,160	0,033	0,001
5	0,212	0,086	0,162	0,035	0,001
6	0,206	0,087	0,162	0,036	0,001
średnia	0,225	0,085	0,227	0,032	0,001

{tab:r2_f2}

Tabela. 5.8 R2 - czas generowania pojedynczego deskryptora punktu charakterystycznego

obraz	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[ms]	[ms]	[ms]	[ms]	[ms]	[ms]	[ms]
1	0,100	0,215	0,202	0,012	0,012	0,709	0,089
2	0,102	0,217	0,196	0,012	0,012	0,678	0,089
3	0,100	0,213	0,192	0,012	0,012	0,702	0,089
4	0,100	0,221	0,192	0,012	0,012	0,773	0,090
5	0,102	0,219	0,192	0,012	0,012	0,733	0,090
6	0,100	0,221	0,195	0,012	0,012	0,690	0,089
średnia	0,101	0,218	0,195	0,012	0,012	0,714	0,089

{tab:r2_f3}

Tabela. 5.9 R2 - powtarzalność wykrywanych cech

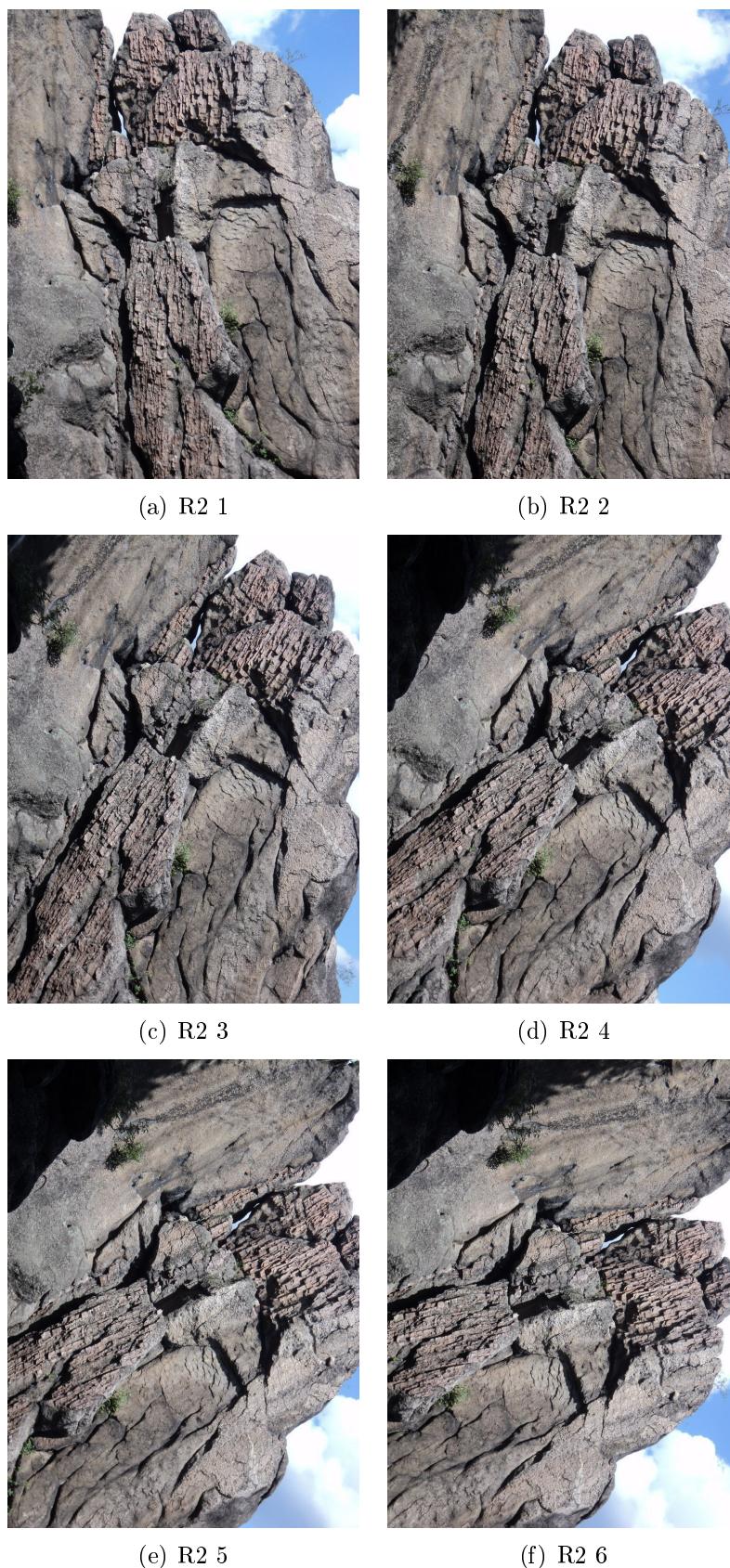
obrazy	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[%]	[%]	[%]	[%]	[%]	[%]	[%]
1 2	54	51	16	24	8	43	10
1 3	48	55	12	1	1	0	9
1 4	57	45	14	1	1	0	12
1 5	55	55	16	1	1	0	9
1 6	44	47	12	2	1	0	9
średnia	52	51	14	6	2	9	10

{tab:r2_m1}

Tabela. 5.10 R2 - procent poprawnych dopasowań

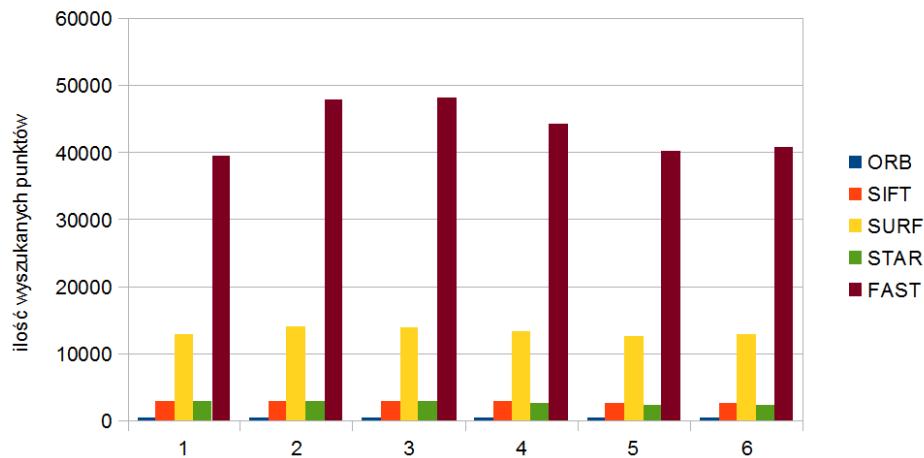
obrazy	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[%]	[%]	[%]	[%]	[%]	[%]	[%]
1 2	75	77	39	72	67	82	61
1 3	71	90	49	10	13	33	57
1 4	84	69	49	8	9	24	71
1 5	72	93	60	9	7	33	61
1 6	51	71	32	10	11	20	55
średnia	71	80	46	22	22	39	61

{tab:r2_m2}



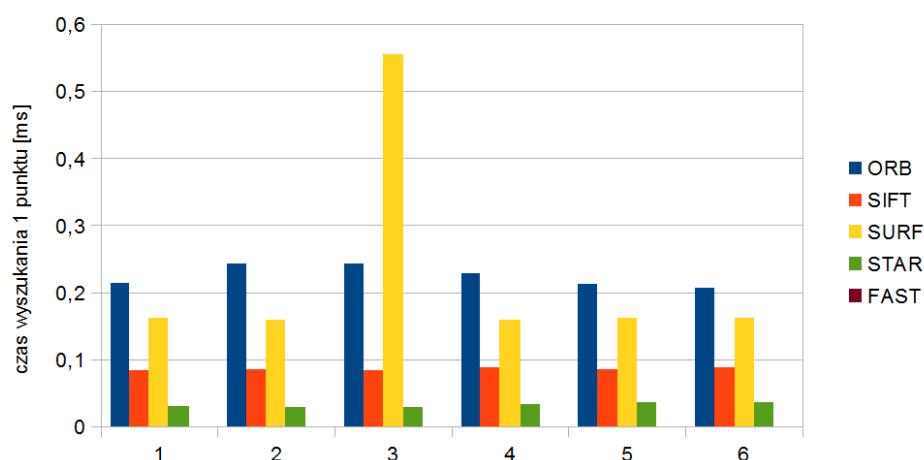
Rysunek 5.7 R2 - zbiór obrazów testowych

{fig:r2_set}



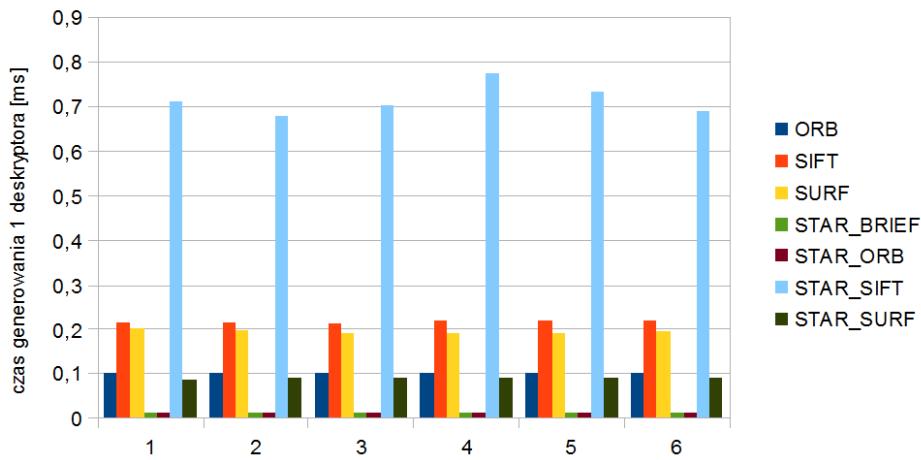
Rysunek 5.8 R2 - ilość wyszukanych cech

{fig:r2_f1}



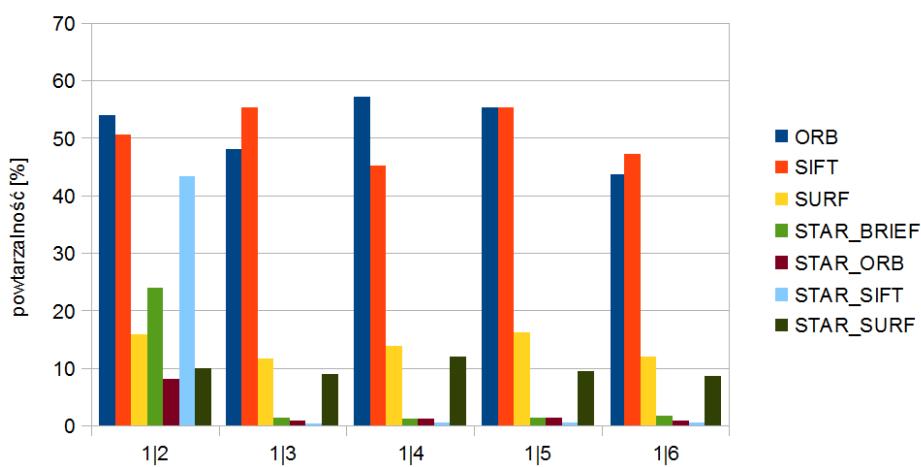
Rysunek 5.9 R2 - czas lokalizowania pojedynczego punktu charakterystycznego

{fig:r2_f2}



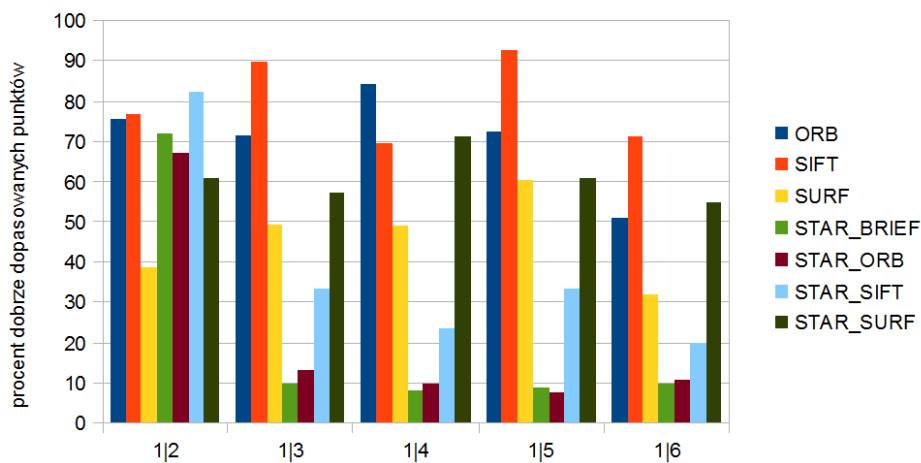
Rysunek 5.10 R2 - czas generowania pojedynczego deskryptora punktu charakterystycznego

{fig:r2_f3}



Rysunek 5.11 R2 - powtarzalność wykrywanych cech

{fig:r2_m1}



Rysunek 5.12 R2 - procent poprawnych dopasowań

{fig:r2_m2}

5.1.3 R3

Transformacja: Rotacja

Rejon: Krzywa Turnia

Powiększenie: Średnie

Oświetlenie: Mieszane

Rozdzielcość: 1024×768

Tabela. 5.11 R3 - ilość wyszukanych cech

obraz	ORB	SIFT	SURF	STAR	FAST
1	500	1847	10957	690	18567
2	500	2021	10701	686	20103
3	500	2054	10452	579	21226
4	500	2413	10839	785	24483
5	500	1787	10631	653	18969
6	500	1869	10903	722	20906
średnia	500	1999	10747	686	20709

{tab:r3_f1}

Tabela. 5.12 R3 - czas lokalizowania pojedynczego punktu charakterystycznego

obraz	ORB	SIFT	SURF	STAR	FAST
-	[ms]	[ms]	[ms]	[ms]	[ms]
1	0,110	0,122	0,172	0,126	0,001
2	0,122	0,113	0,172	0,125	0,001
3	0,116	0,111	0,173	0,149	0,001
4	0,128	0,101	0,171	0,111	0,001
5	0,112	0,128	0,173	0,132	0,001
6	0,112	0,121	0,171	0,120	0,001
średnia	0,117	0,116	0,172	0,127	0,001

{tab:r3_f2}

Tabela. 5.13 R3 - czas generowania pojedynczego deskryptora punktu charakterystycznego

obraz	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[ms]	[ms]	[ms]	[ms]	[ms]	[ms]	[ms]
1	0,102	0,254	0,198	0,020	0,019	1,462	0,106
2	0,104	0,243	0,193	0,020	0,019	1,443	0,106
3	0,100	0,243	0,196	0,022	0,021	1,577	0,109
4	0,100	0,234	0,193	0,020	0,018	1,367	0,104
5	0,100	0,259	0,191	0,021	0,020	1,424	0,106
6	0,100	0,252	0,198	0,021	0,019	1,385	0,105
średnia	0,101	0,247	0,195	0,021	0,019	1,443	0,106

{tab:r3_f3}

Tabela. 5.14 R3 - powtarzalność wykrywanych cech dla obrazów z zestawu R3

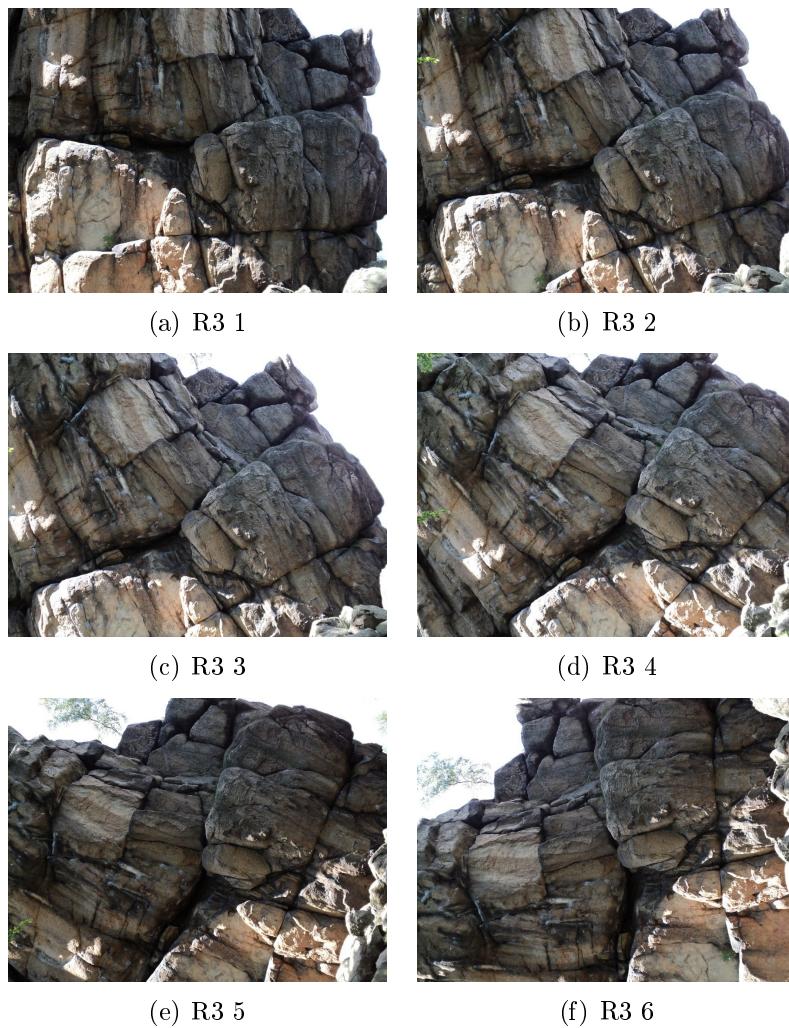
obrazy	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[%]	[%]	[%]	[%]	[%]	[%]	[%]
1 2	53	54	25	19	7	28	20
1 3	36	44	20	2	2	1	11
1 4	36	44	17	4	2	1	14
1 5	37	46	19	1	3	1	14
1 6	38	50	35	2	2	1	23
średnia	40	47	23	6	3	6	16

{tab:r3_m1}

Tabela. 5.15 R3 - procent poprawnych dopasowań

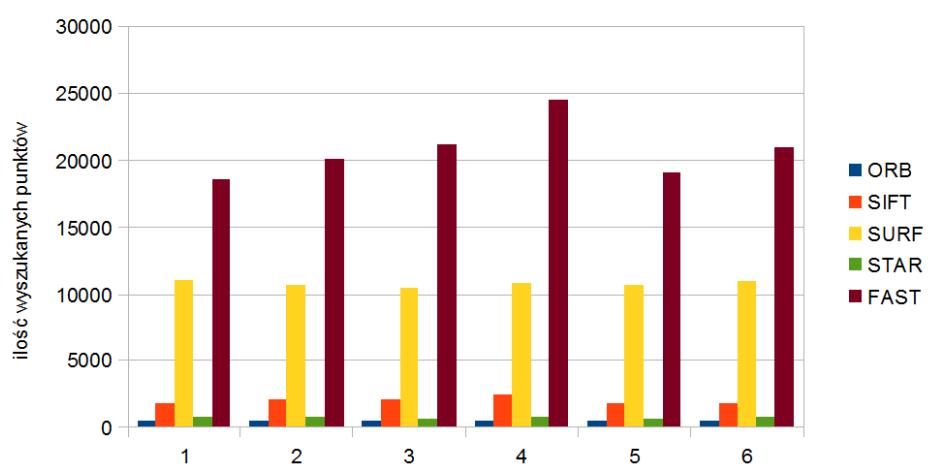
obrazy	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[%]	[%]	[%]	[%]	[%]	[%]	[%]
1 2	79	90	51	70	59	73	50
1 3	65	66	41	18	17	36	55
1 4	72	67	46	16	16	57	54
1 5	54	80	48	14	13	36	47
1 6	69	82	48	18	14	57	76
średnia	68	77	47	27	24	52	56

{tab:r3_m2}



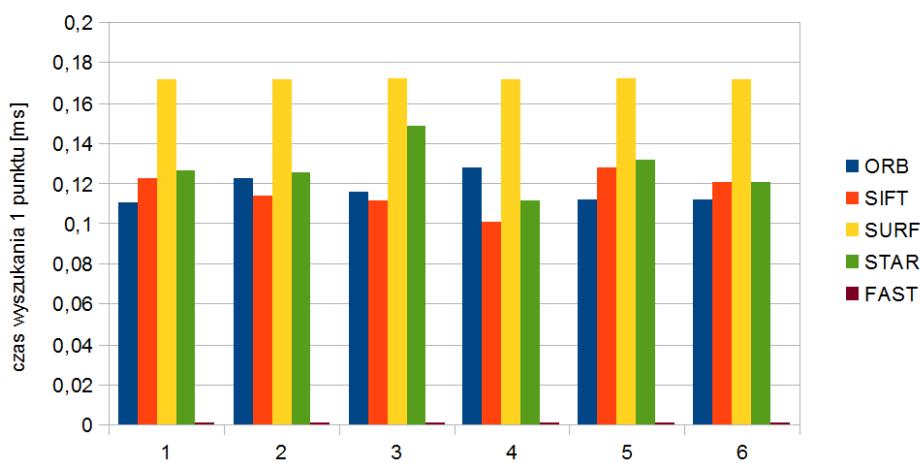
Rysunek 5.13 R3 - zbiór obrazów testowych

{fig:r3_set}



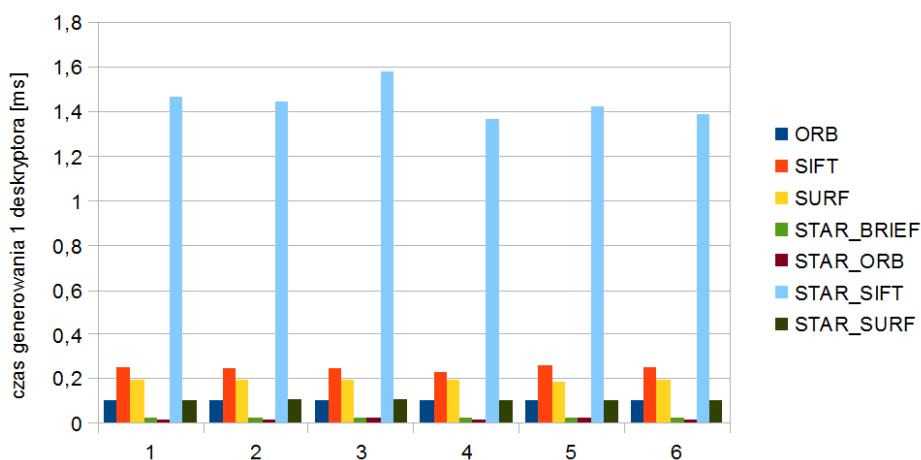
Rysunek 5.14 R3 - ilość wyszukanych cech

{fig:r3_f1}



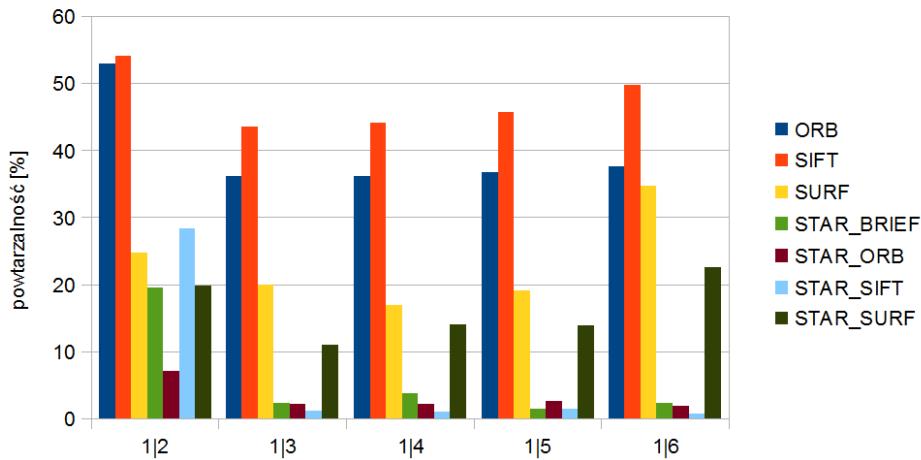
Rysunek 5.15 R3 - czas lokalizowania pojedynczego punktu charakterystycznego

{fig:r3_f2}



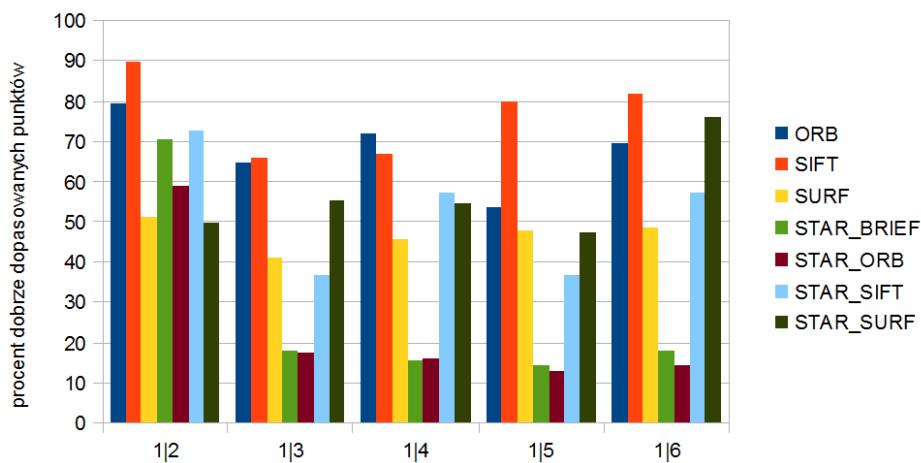
Rysunek 5.16 R3 - czas generowania pojedynczego deskryptora punktu charakterystycznego

{fig:r3_f3}



Rysunek 5.17 R3 - powtarzalność wykrywanych cech

{fig:r3_m1}



Rysunek 5.18 R3 - procent poprawnych dopasowań

{fig:r3_m2}

5.1.4 Dyskusja wyników

Dla przygotowanych zbiorów wszystkie algorytmy wykazują stabilność wskaźników takich jak ilość generowanych lokalizowanych punktów i charakterystyki czasowe. Tradycyjnie o algorytm FAST lokalizuje o rząd wielkości więcej punktów, kolejne w kolejności są algorytmy SURF, SIFT, STAR i ograniczony parametrycznie ORB.

Czasy lokalizowania punktu przez FAST są również znikomo małe. W zależności od badanej sceny czas lokalizowania przez algorytm ORB może zmieniać się o blisko 100%. Algorytm ten spowalnia dla obrazów o dużym stopniu przybliżenia. Prawdopodobnie spowodowane dużą szczegółowością obrazów, a co za tym idzie dużą liczbą zgrubnych punktów charakterystycznych, które trzeba ocenić miarą Harrisza. Na tle tych charakterystyk czasowych najlepsze wyniki osiąga algorytm STAR.

Czas generowania deskryptora można uznać za stały niezależnie od typu obrazów. Podobnie jak w badaniach dla zbioru Mikołajczyka kombinacja STAR-SIFT jest w sposób radykalny najwolniejszą w tej kategorii.

Badając powtarzalność algorytmów możemy zaobserwować większe zróżnicowanie w wynikach. W tej dziedzinie jako wiodące możemy uznać algorytmy ORB i SIFT. Dla obrazów o dużym stopniu przybliżenia ich wyniki są niemal równe. Większość algorytmów dobrze radzi sobie z niewielkim stopniem rotacji obrazu przy dużym powiększeniu. Jakość ich jednak spada w sposób znaczący przy większych kątach obrotu. Najlepsze wyniki na tym polu osiągają algorytmy ORB i SIFT, których minimum wyników przypada w okolicach kąta 45° .

Najkorzystniejszy współczynnik poprawnych dopasowań posiada algorytm SIFT. Dobre wyniki osiągają również wyniki dla algorytmów ORB i STAR-SURF. Przewaga algorytmu SIFT ujawnia się w kątach powyżej 60° . Pozostałe metody dla większych kątów osiągają wskaźnik poprawności poniżej 50 %.

5.2 Zmiana położenia obserwatora

5.2.1 V1

Transformacja: Zmiana położenia obserwatora

Rejon: Mały Sokolik

Powiększenie: Duże

Oświetlenie: Duże

Rozdzielcość: 585×1024



(a) V1 1



(b) V1 2



(c) V1 3



(d) V1 4



(e) V1 5



(f) V1 6

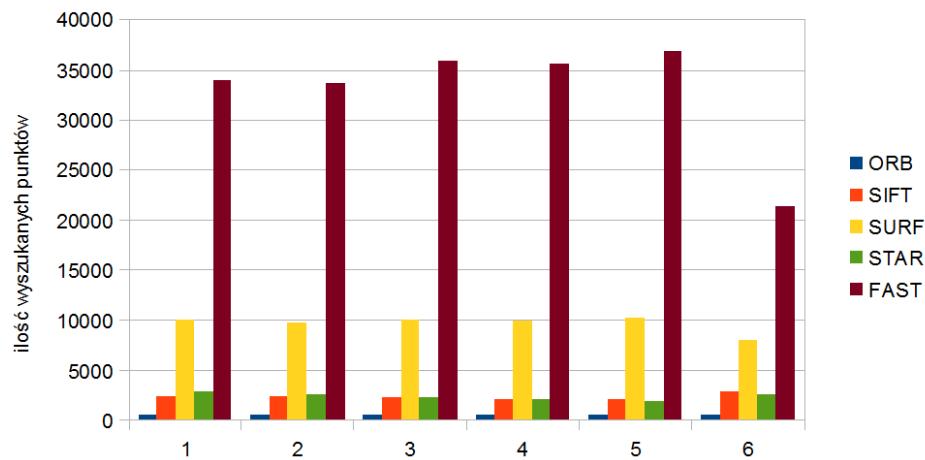
Rysunek 5.19 V1 - zbiór obrazów testowych

{fig:v1_set}

Tabela. 5.16 V1 - ilość wyszukanych cech

obraz	ORB	SIFT	SURF	STAR	FAST
1	500	2463	10032	2951	34032
2	500	2472	9698	2601	33659
3	500	2317	10096	2344	35807
4	500	2125	9966	2133	35498
5	500	2106	10194	2032	36830
6	500	2844	8021	2562	21418
średnia	500	2388	9668	2437	32874

{tab:v1_f1}



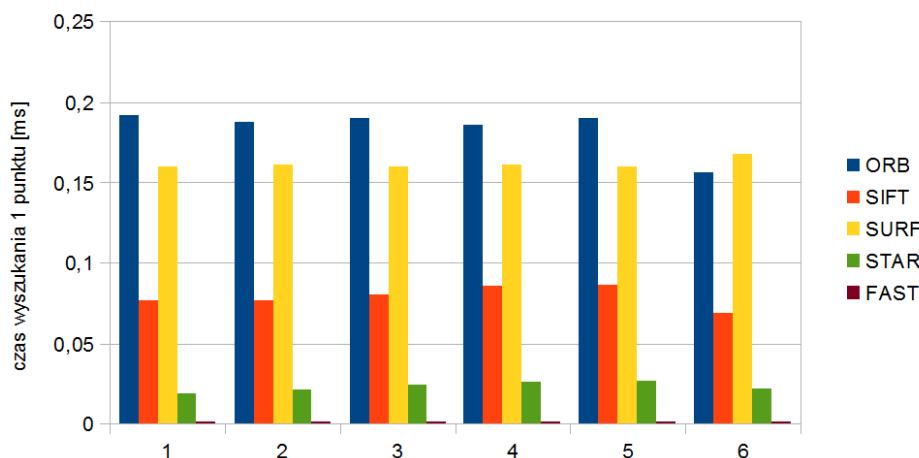
Rysunek 5.20 V1 - ilość wyszukanych cech

{fig:v1_f1}

Tabela. 5.17 V1 - czas lokalizowania pojedynczego punktu charakterystycznego

obraz	ORB	SIFT	SURF	STAR	FAST
-	[ms]	[ms]	[ms]	[ms]	[ms]
1	0,192	0,076	0,160	0,019	0,001
2	0,188	0,076	0,161	0,021	0,001
3	0,190	0,080	0,160	0,023	0,001
4	0,186	0,085	0,161	0,025	0,001
5	0,190	0,086	0,160	0,027	0,001
6	0,156	0,069	0,168	0,021	0,001
średnia	0,184	0,079	0,162	0,023	0,001

{tab:v1_f2}



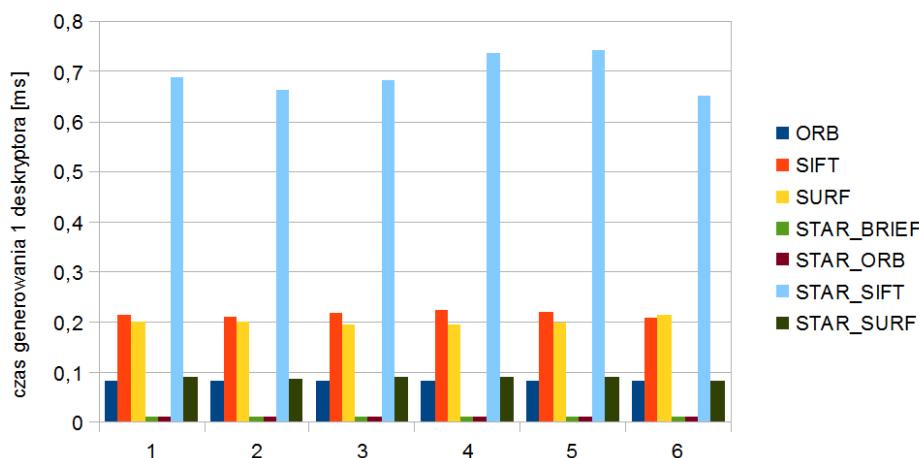
Rysunek 5.21 V1 - czas lokalizowania pojedynczego punktu charakterystycznego

{fig:v1_f2}

Tabela 5.18 V1 - czas generowania pojedynczego deskryptora punktu charakterystycznego

obraz	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[ms]						
1	0,082	0,213	0,201	0,011	0,011	0,688	0,088
2	0,082	0,211	0,201	0,012	0,011	0,665	0,088
3	0,082	0,216	0,196	0,012	0,012	0,684	0,089
4	0,082	0,223	0,196	0,012	0,012	0,737	0,090
5	0,082	0,221	0,197	0,012	0,012	0,742	0,090
6	0,082	0,206	0,214	0,011	0,011	0,651	0,085
średnia	0,082	0,215	0,201	0,012	0,011	0,694	0,088

{tab:v1_f3}



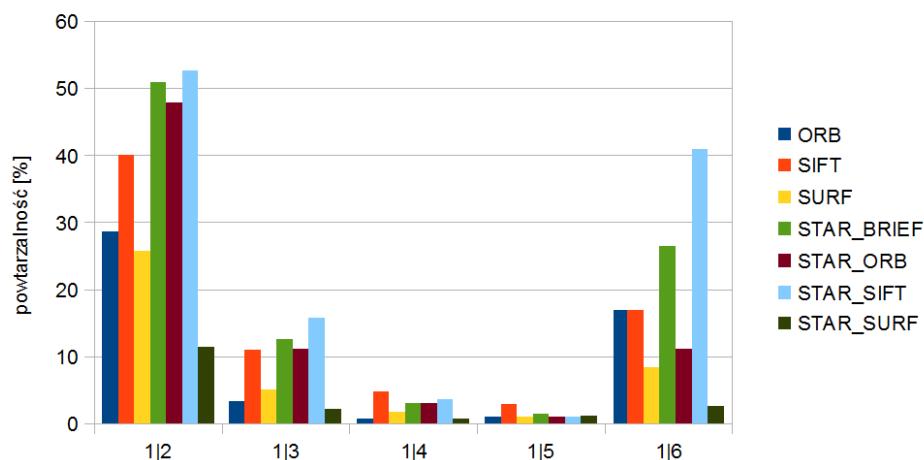
Rysunek 5.22 V1 - czas generowania pojedynczego deskryptora punktu charakterystycznego

{fig:v1_f3}

Tabela. 5.19 V1 - powtarzalność wykrywanych cech

obrazy	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[%]	[%]	[%]	[%]	[%]	[%]	[%]
1 2	29	40	26	51	48	53	11
1 3	3	11	5	13	11	16	2
1 4	1	5	2	3	3	4	1
1 5	1	3	1	1	1	1	1
1 6	17	17	8	26	11	41	3
średnia	10	15	8	19	15	23	4

{tab:v1_m1}



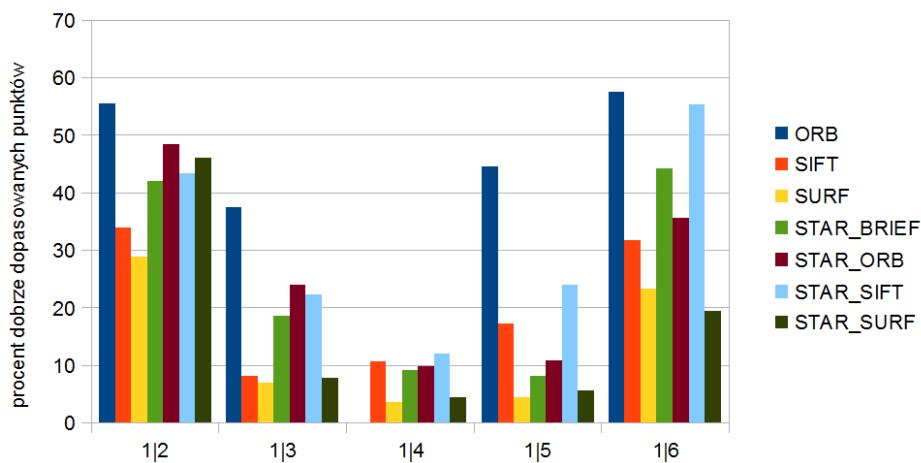
Rysunek 5.23 V1 - powtarzalność wykrywanych cech

{fig:v1_m1}

Tabela. 5.20 V1 - procent poprawnych dopasowań

obrazy	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[%]	[%]	[%]	[%]	[%]	[%]	[%]
1 2	56	34	29	42	48	43	46
1 3	38	8	7	18	24	22	8
1 4	0	11	4	9	10	12	4
1 5	44	17	4	8	11	24	6
1 6	57	32	23	44	36	55	19
średnia	39	20	13	24	26	31	17

{tab:v1_m2}



Rysunek 5.24 V1 - procent poprawnych dopasowań

{fig:v1_m2}

5.2.2 V2

Transformacja: Zmiana położenia obserwatora

Rejon: Signum

Powiększenie: Średnie

Oświetlenie: Umiarkowane

Rozdzielcość: 1024×768

Tabela 5.21 V2 - ilość wyszukanych cech

obraz	ORB	SIFT	SURF	STAR	FAST
1	500	2389	11418	886	29038
2	500	2207	12586	1004	35894
3	500	2426	11923	873	31295
4	500	2568	10984	990	28111
5	500	2380	11698	1245	28736
6	500	2067	11915	551	27275
średnia	500	2340	11754	925	30058

{tab:v2_f1}

5.2.3 Dyskusja wyników

Problem zmiany położenia obserwatora wydaje się być najtrudniejszym dla badanych algorytmów. O ile charakterystyki czasowe i ilościowe dla obrazów w ramach pojedynczego zbioru pozostają stałe, o tyle w przypadku wskaźników dopasowań obserwujemy duże zróżnicowanie wyników i lawinowy spadek skuteczności wraz ze zmianą położenia. Klasyfikacja w dziedzinach ilości punktu, czasów detekcji i opisywanie nie różni się w zasadniczy sposób od tych zaobserwowanych w poprzednich badaniach.

W trakcie konstruowania zbioru testowego obserwator przemieszczał się średnio o 2 metry pomiędzy kolejnymi ujęciami. Badania wykazują dużą wrażliwość algorytmów na przemieszczenie obserwatora powyżej 4 metrów. Powtarzalność dla takiej transformacji w

Tabela. 5.22 V2 - czas lokalizowania pojedynczego punktu charakterystycznego

obraz	ORB	SIFT	SURF	STAR	FAST
-	[ms]	[ms]	[ms]	[ms]	[ms]
1	0,134	0,098	0,170	0,097	0,001
2	0,154	0,104	0,166	0,087	0,001
3	0,144	0,096	0,168	0,100	0,001
4	0,142	0,094	0,172	0,088	0,001
5	0,146	0,100	0,169	0,071	0,001
6	0,126	0,111	0,169	0,158	0,001
średnia	0,141	0,100	0,169	0,100	0,001

{tab:v2_f2}

Tabela. 5.23 V2 - czas generowania pojedynczego deskryptora punktu charakterystycznego

obraz	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[ms]	[ms]	[ms]	[ms]	[ms]	[ms]	[ms]
1	0,102	0,234	0,210	0,019	0,017	1,073	0,098
2	0,100	0,242	0,206	0,018	0,016	1,036	0,098
3	0,100	0,235	0,209	0,019	0,017	1,094	0,100
4	0,102	0,229	0,212	0,018	0,016	1,088	0,099
5	0,102	0,232	0,216	0,017	0,014	0,882	0,095
6	0,102	0,243	0,205	0,022	0,022	0,902	0,096
średnia	0,101	0,236	0,210	0,019	0,017	1,012	0,098

{tab:v2_f3}

Tabela. 5.24 V2 - powtarzalność wykrywanych cech

obrazy	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[%]	[%]	[%]	[%]	[%]	[%]	[%]
1 2	22	38	44	67	64	69	27
1 3	4	20	12	30	25	33	4
1 4	1	1	0	3	2	3	2
1 5	0	0	0	2	2	1	1
1 6	2	2	1	3	3	6	1
średnia	6	12	11	21	19	22	7

{tab:v2_m1}

Tabela. 5.25 V2 - procent poprawnych dopasowań

obrazy	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[%]	[%]	[%]	[%]	[%]	[%]	[%]
1 2	45	52	86	69	65	71	78
1 3	54	33	30	28	32	27	21
1 4	0	15	4	10	17	29	9
1 5	0	19	5	16	19	57	12
1 6	40	16	7	13	14	29	11
średnia	28	27	26	27	29	43	26

{tab:v2_m2}

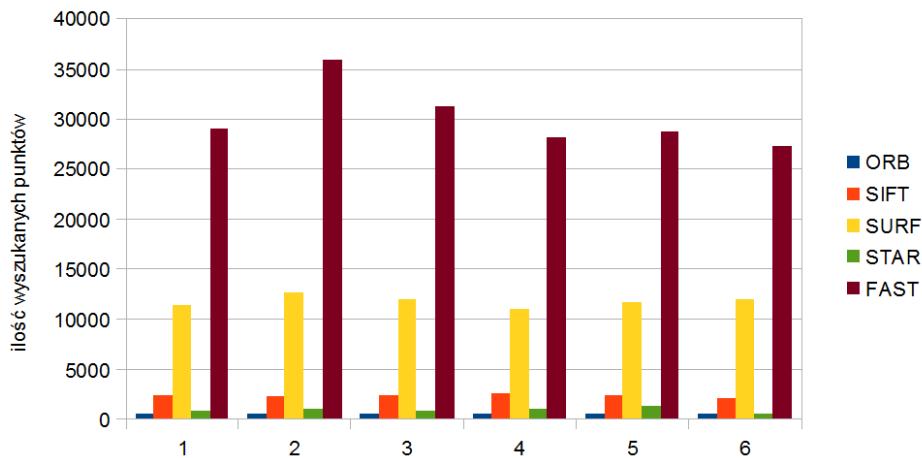


Rysunek 5.25 V2 - zbiór obrazów testowych

{fig:v2_set}

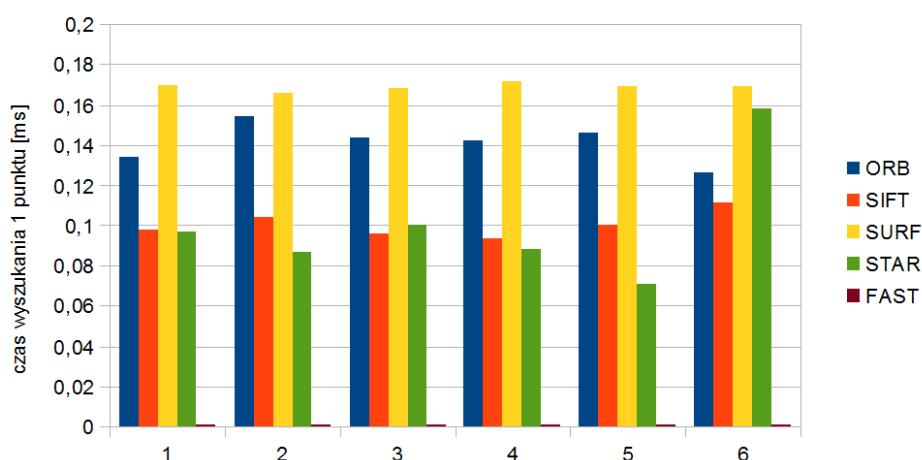
najlepszych warunkach wynosi około 30%, by finalnie spaść niemal do 0. Zjawisko to jest obserwowane dla obu zestawów.

Procent dobrze dopasowanych punktów jest wskaźnikiem silnie powiązany z powtarzalnością. W sytuacji gdy powtarzalność spada do 0, bezzasadnym staje się dyskusja poprawności. Biorąc pod uwagę te pary obrazów z niezerową powtarzalnością, najlepsze wyniki w zależności od sceny osiągają algorytmy ORB i STAR-SIFT.



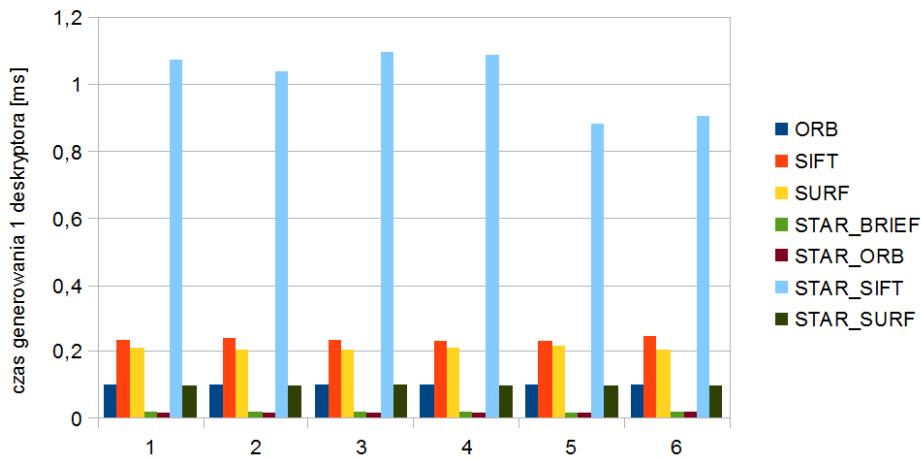
Rysunek 5.26 V2 - ilość wyszukanych cech

{fig:v2_f1}



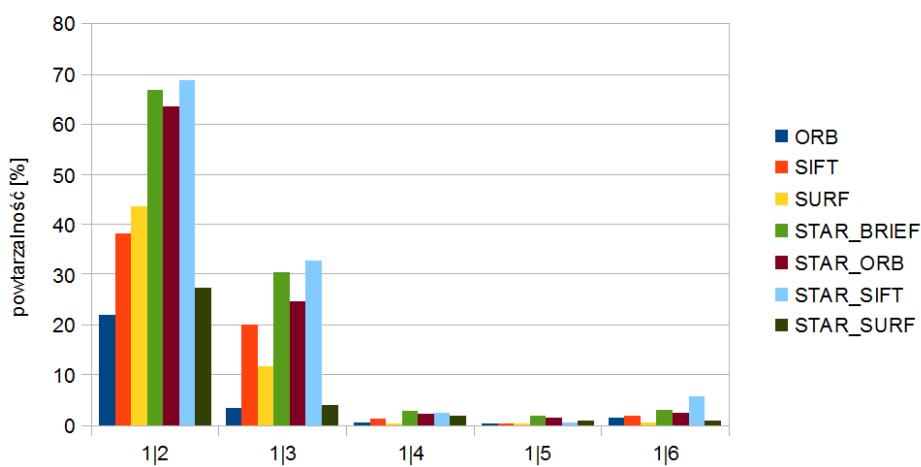
Rysunek 5.27 V2 - czas lokalizowania pojedynczego punktu charakterystycznego

{fig:v2_f2}



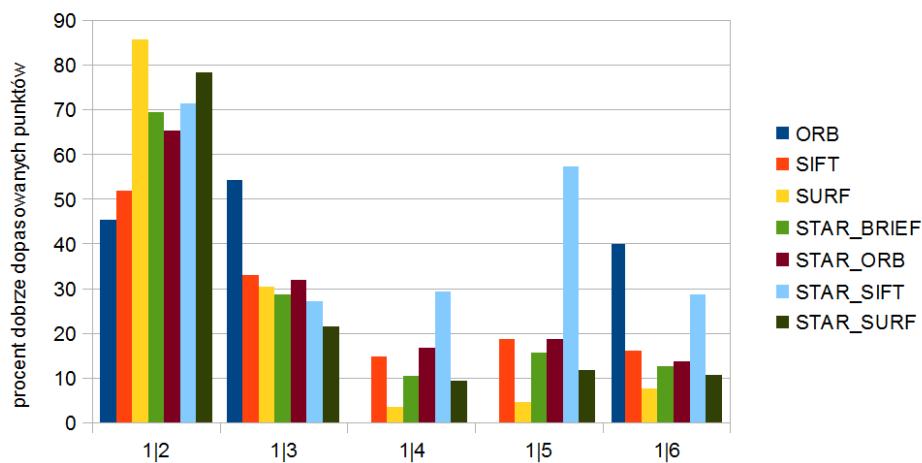
Rysunek 5.28 V2 - czas generowania pojedynczego deskryptora punktu charakterystycznego

{fig:v2_f3}



Rysunek 5.29 V2 - powtarzalność wykrywanych cech

{fig:v2_m1}



Rysunek 5.30 V2 - procent poprawnych dopasowań

{fig:v2_m2}

5.3 Zmiana skali

5.3.1 Z1

Transformacja: Zmiana skali

Rejon: Krzywa Turnia

Powiększenie: Duże

Oświetlenie: Umiarkowane/Mieszane

Rozdzielcość: 1024×768

Tabela 5.26 Z1 - ilość wyszukanych cech

obraz	ORB	SIFT	SURF	STAR	FAST
1	500	2318	10357	874	24877
2	500	2360	10257	833	24590
3	500	2303	9109	763	17416
4	500	2281	10120	1033	24754
5	500	2503	9943	1220	23616
6	500	3012	9473	1485	21403
średnia	500	2463	9877	1035	22776

{tab:z1_f1}

Tabela. 5.27 Z1 - czas lokalizowania pojedynczego punktu charakterystycznego

obraz	ORB	SIFT	SURF	STAR	FAST
-	[ms]	[ms]	[ms]	[ms]	[ms]
1	0,132	0,099	0,176	0,100	0,001
2	0,132	0,098	0,174	0,104	0,001
3	0,120	0,103	0,180	0,114	0,001
4	0,144	0,102	0,175	0,086	0,001
5	0,142	0,096	0,177	0,072	0,001
6	0,146	0,083	0,178	0,060	0,001
średnia	0,136	0,097	0,177	0,089	0,001

{tab:z1_f2}

Tabela. 5.28 Z1 - czas generowania pojedynczego deskryptora punktu charakterystycznego

obraz	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[ms]	[ms]	[ms]	[ms]	[ms]	[ms]	[ms]
1	0,100	0,234	0,194	0,019	0,017	1,292	0,104
2	0,100	0,231	0,195	0,019	0,018	1,414	0,106
3	0,104	0,237	0,197	0,020	0,018	1,507	0,107
4	0,102	0,243	0,195	0,018	0,015	1,370	0,105
5	0,104	0,235	0,198	0,017	0,015	1,245	0,102
6	0,100	0,222	0,201	0,016	0,014	1,213	0,100
średnia	0,102	0,234	0,197	0,018	0,016	1,340	0,104

{tab:z1_f3}

Tabela. 5.29 Z1 - powtarzalność wykrywanych cech

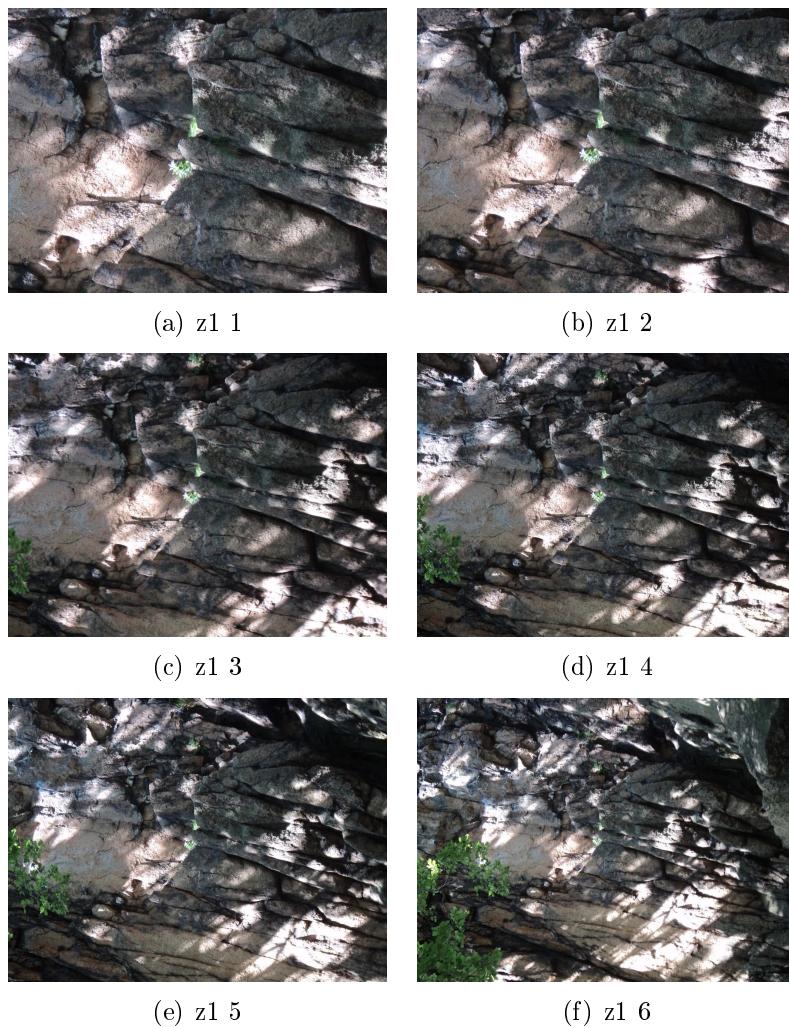
obrazy	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[%]	[%]	[%]	[%]	[%]	[%]	[%]
1 2	37	38	33	46	37	63	12
1 3	11	16	12	7	4	27	5
1 4	5	9	8	4	3	18	3
1 5	3	7	5	3	2	7	2
1 6	1	4	2	1	1	1	1
średnia	12	15	12	12	10	23	4

{tab:z1_m1}

Tabela. 5.30 Z1 - procent poprawnych dopasowań

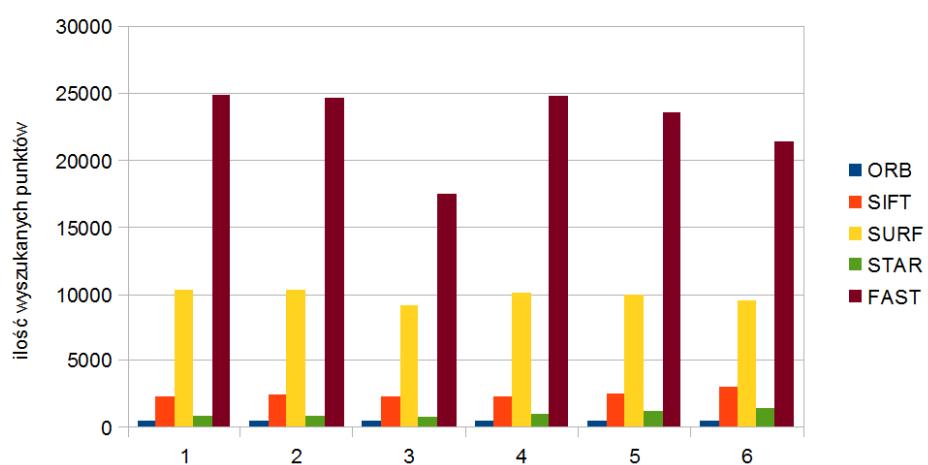
obrazy	ORB	SIFT	SURF	ST-BRIEF	ST-ORB	ST-SIFT	ST-SURF
-	[%]	[%]	[%]	[%]	[%]	[%]	[%]
1 2	69	69	71	54	67	66	42
1 3	47	51	42	26	9	59	22
1 4	32	47	41	12	10	47	21
1 5	24	49	36	15	13	45	26
1 6	19	38	19	12	13	52	16
średnia	38	51	42	24	22	54	25

{tab:z1_m2}



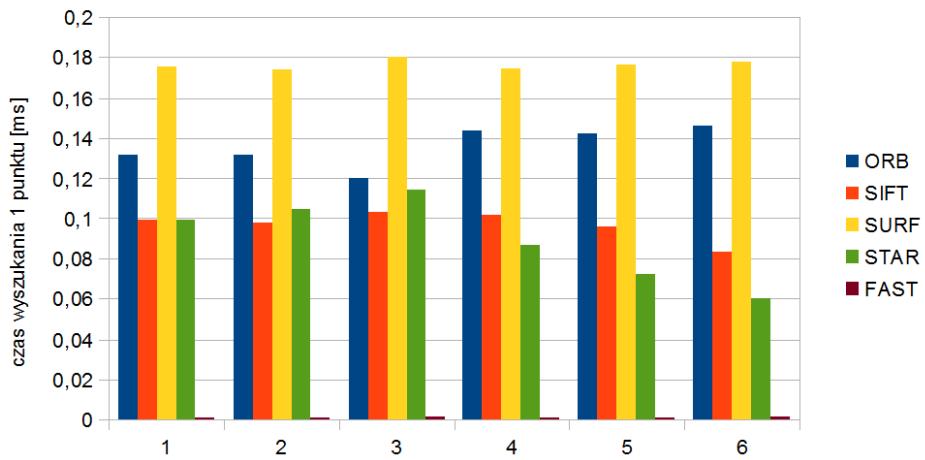
Rysunek 5.31 Z1 - zbiór obrazów testowych

{fig:z1_set}



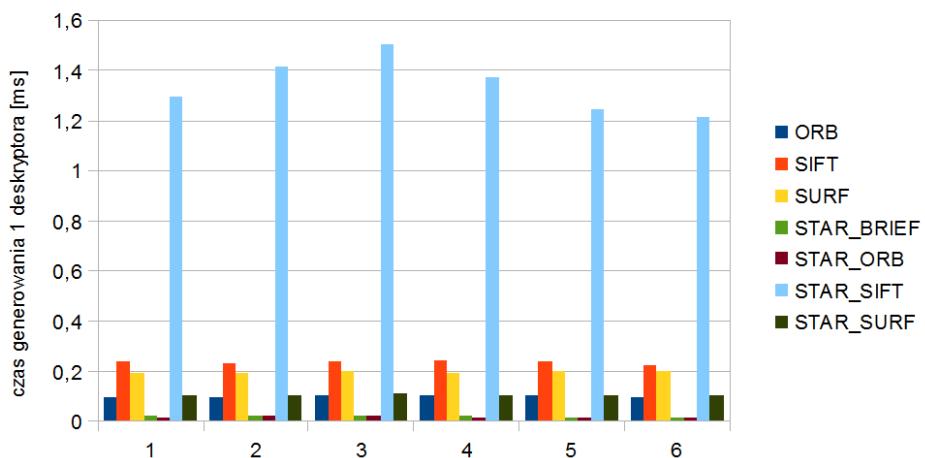
Rysunek 5.32 Z1 - ilość wyszukanych cech

{fig:z1_f1}



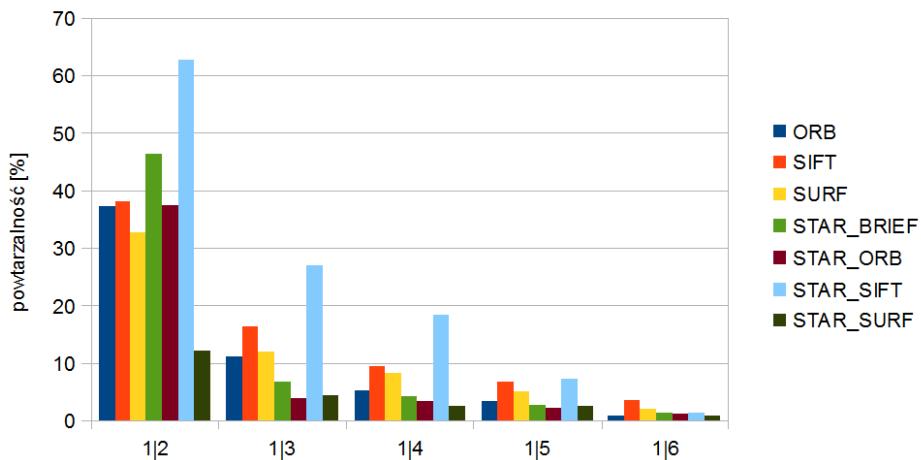
Rysunek 5.33 Z1 - czas lokalizowania pojedynczego punktu charakterystycznego

{fig:z1_f2}



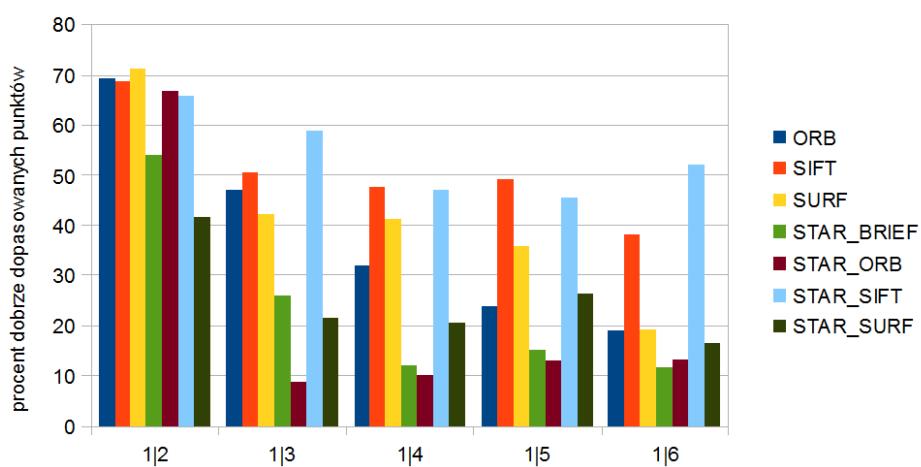
Rysunek 5.34 Z1 - czas generowania pojedynczego deskryptora punktu charakterystycznego

{fig:z1_f3}



Rysunek 5.35 Z1 - powtarzalność wykrywanych cech

{fig:z1_m1}



Rysunek 5.36 Z1 - procent poprawnych dopasowań

{fig:z1_m2}

5.3.2 Dyskusja wyników

Większość algorytmów niezależnie od stopnia przybliżenia generuje zbliżoną liczbę punktów charakterystycznych. Podobnie jak w poprzednich testach najwięcej cech lokalizuje algorytm FAST, potem SURF, następnie SIFT, STAR i stało wartościowy ORB.

Charakterystyki czasowe również pozostają stałe dla obrazów w zbiorze. Najwolniej punkty lokalizują algorytmy SURF i ORB, z kolei najwolniejszym deskryptorem jest kooperujący z algorytmem STAR, deskryptor SIFT.

Metoda ta jednak osiąga najlepszy współczynnik powtarzalności. Dla większości algorytmów wskaźnik ten znaczaco spada przy zmianie skali większej niż 0,5.

Metoda STAR-SIFT osiąga również najlepsze wyniki pod względem trafności dopasowań. Nieznacznie gorzej przy niej wypada czysty algorytm SIFT. Pozostałe metody w tym obszarze rzadko osiągają poprawność przekraczającą 30% .

Rozdział 6

Podsumowanie

W niniejszej pracy dokonano zestawienia grupy najpopularniejszych algorytmów służących lokalizowaniu i opisywaniu lokalnych cech charakterystycznych obrazów. Metody te zostały w dokładny sposób opisane i przebadane.

Do oceny jakości działań algorytmów wykorzystano często stosowany w ewaluacji zbiór Mikołajczyka oraz opracowany na potrzeby niniejszej pracy autorski zbiór obrazów skalnych. Stabilność kryteriów czasowych i ilościowych pozwala wysnuć wniosek o poprawnym doborze obrazów wchodzących w skład zbioru autorskiego.

W ramach badań sprawdzono odporność algorytmów na transformacje obrazu. Większość algorytmów w sposób umiarkowanie dobry radzi sobie z przekształceniami takimi jak rozmycie obrazu czy zmiana oświetlenia. Największe wyzwanie dla algorytmów stanowią obrazy, w których zmienia się sposób patrzenia na scenę. Zarówno w przypadku obrazów skalnych i Mikołajczyka algorytmy są wrażliwe na przekształcenia rotacji, zmianę skali lub położenia obserwatora. Na tym tle najlepiej wypadają algorytmy ORB i SIFT oraz kombinacja STAR-SIFT.

Algorytm SIFT w większości przypadków okazuje się bardziej dokładnym od algorytmu ORB. Różnica ta jednak często jest znikoma, dlatego w większości aplikacji, w których istotnymi są parametry czasowe lepszym rozwiązaniem wydaje się algorytm ORB. Należy jednak pamiętać, że w zależności od badanej sceny szybszym może się okazywać algorytm SIFT. Ciekawym rozwiązaniem wydaje się również kombinacja STAR-SIFT. W rankingach czasowych osiąga ona wyniki o rząd wielkości gorsze wyniki niż pozostałe algorytmy. Kombinacja ta jednak działa w sytuacjach, gdy pozostałe metody zawodzą.

Reasumując wyniki badań, nie ma jednego idealnego algorytmu pozwalającego lokalizować cechy obrazu do rozpoznawania. Osiągi algorytmów różnią się w zależności od rodzaju sceny i transformacji. Rozwiązaniem przyszłościowym może być opracowanie metody operującej na wyższym stopniu abstrakcji, działającej na kompleksach cech zamiast na pojedynczych punktach.

Dodatek A

Opis aplikacji testowej

Na potrzeby niniejszej pracy powstał program testowy napisany w języku C#. Wykorzystuje on bibliotekę przetwarzania obrazów OpenCV w wersji 2.4. OpenCV jest wieloplatformową biblioteką o otwartym kodzie dostarczającą szereg rozwiązań z zakresu obróbki obrazów i widzenia maszynowego. Ze względu, na to, że bibliotek jest napisana w językach C i C++ w programie testowym zastosowano nakładkę EmguCV. EmguCV pozwala na wykorzystywanie OpenCV w programach działających na platformie .NET.

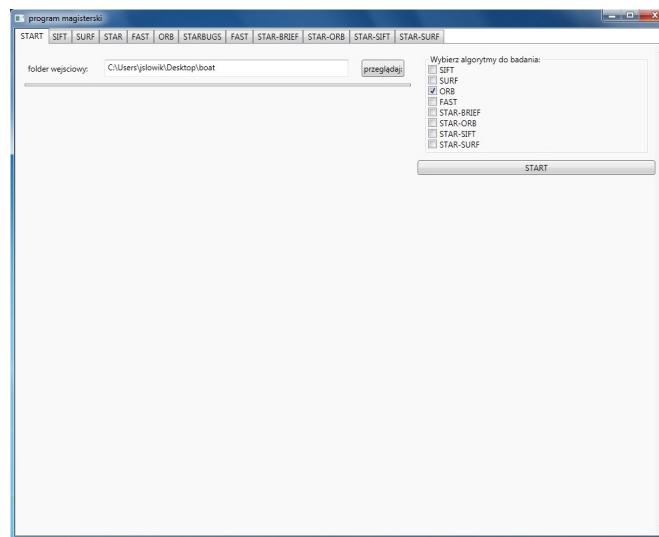
Źródła bibliotek OpenCV i EmguCV można znaleźć pod adresami:

- www.opencv.org
- www.emgu.com

Aplikacja posiada interfejs graficzny napisany z wykorzystaniem technologii Windows Presentation Foundation.

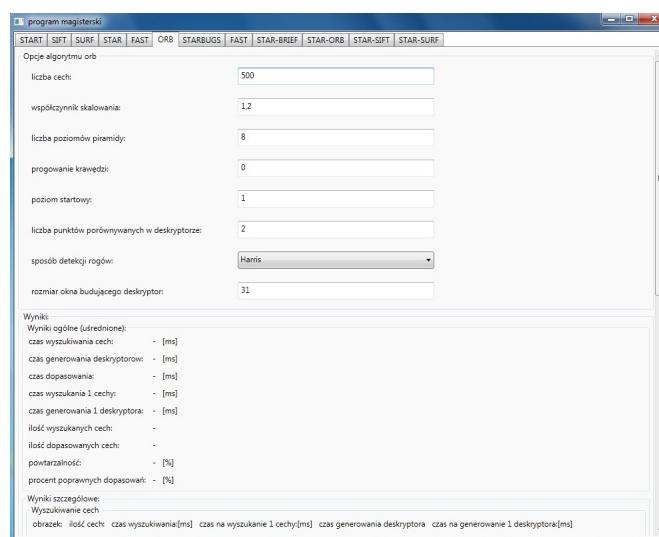
Instrukcja obsługi:

- Po uruchomieniu programu użytkownikowi ukazuje się okno aplikacji ustawione na zakładce START. Służy ona wybraniu folderu z zestawem obrazów oraz algorytmów poddawanych badaniu. [rysunek A.1]
- Użytkownik ma możliwość zdefiniowania parametrów wejściowych testowanego algorytmu. W tym celu należy przejść do zakładki odpowiadającej badanej metodzie. Ze względu na olbrzymią liczbę dostępnych kombinacji wszystkie algorytmy były testowane na ustawieniach domyślnych. [rysunek A.2]
- Badania uruchamiane są po kliknięciu przycisk START widoczny w pierwszej zakładce. Wyniki prezentowane są w zakładce odpowiadającej badanemu algorytmowi oraz zapisywane do pliku w folderze ze zbiorem wejściowym. [rysunek A.3]



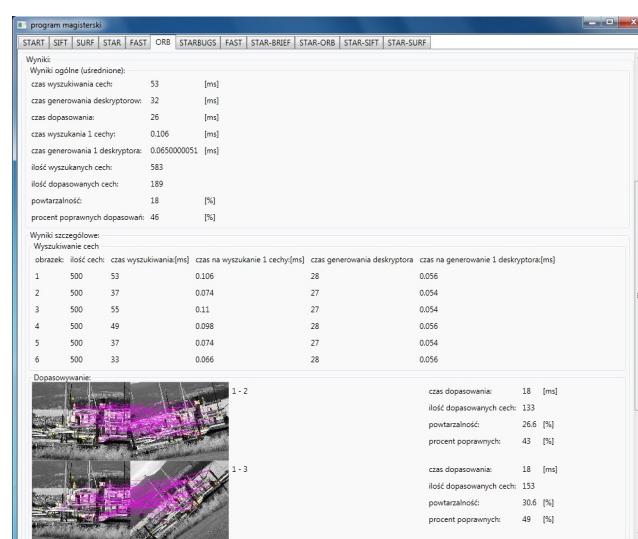
Rysunek A.1 Okno startowe aplikacji testowej

{pict:a1}



Rysunek A.2 Przykładowa zakładka algorytmu - konfiguracja

{pict:a2}



Rysunek A.3 Przykładowa zakładka algorytmu - prezentacja wyników

{pict:a3}

Dodatek B

Zawartość płyty DVD

Do pracy została dołączona płyta DVD zawierająca:

- wersje elektroniczną niniejszej pracy zapisaną w formacie PDF
- zbiory testowe Mikołajczyka i obrazów skalnych
- projekt MS Visual Studio aplikacji testowej

Bibliografia

- [1] Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. strona 381–395, 1981.
- [2] M. Agrawal, K. Konolige, M. R. Blas. Censure: Center surround extrema for realtime feature detection and matching. *European Conference on Computer Vision*, wolumen 4, strony 102–115, 2008.
- [3] H. Bay. *From wide-baseline point and line correspondences to 3D*. Praca doktorska, Eidgenössische Technische Hochschule (ETH) Zürich, 2006.
- [4] H. Bay, A. Ess, T. Tuytelaars, L. V. Gool. Surf: Speeded up robust features. *Computer Vision and Image Understanding*, 2008.
- [5] M. Calonder, V. Lepetit, K. Konolige, J. Bowman, P. Mihelich, P. Fua. Compact signatures for high-speed interest point description and matching. *International Conference on Computer Vision*, 2009.
- [6] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, P. Fua. Brief: Computing a local binary descriptor very fast. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.
- [7] M. Everingham. The pascal visual object classes challenge 2006. <http://pascallin.ecs.soton.ac.uk/challenges/VOC/databases.html>, 2006.
- [8] C. Harris, M. Stephens. A combined corner and edge detector. *Proceedings of Fourth Alvey Vision Conference*, 1988.
- [9] I. Khvedchenia. Comparison of the OpenCV's feature detection algorithms 2. <http://computer-vision-talks.com/2011/07/comparison-of-the-opencv-feature-detection-algorithms-ii/>, dostęp z dnia 11.06.2012.
- [10] I. Khvedchenia. Comparison of the OpenCV's feature detection algorithms. <http://computer-vision-talks.com/2011/01/comparison-of-the-opencv-feature-detection-algorithms-2/>, dostęp z dnia 11.06.2012.
- [11] D. G. Lowe. Object recognition from local scale-invariant features. *Proceedings of the International Conference on Computer Vision*, 1999.
- [12] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [13] C. S. M. Calonder, V. Lepetit, P. Fua. Brief: Binary robust independent elementary features. *European Conference on Computer Vision*, Heraklion, Greece, 2010.

- [14] J. Quinlan. *Induction of decision trees*. *Machine Learning* 1. 1986.
- [15] P. L. Rosin. Measuring corner properties. *Computer Vision and Image Understanding*, 73(2):291 – 307, 1999.
- [16] E. Rosten, T. Drummond. Machine learning for high-speed corner detection. *European Conference on Computer Vision*, wolumen 1, strony 430–443, Maj 2006.
- [17] E. Rosten, R. Porter, T. Drummond. Faster and better: A machine learning approach to corner detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 32:105–119, 2010.
- [18] E. Rublee, V. Rabaud, K. Konolige, G. Bradski. Orb: an efficient alternative to sift or surf. Raport instytutowy, Willow Garage, Menlo Park, California, 2011.
- [19] T. Tuytelaars, C. Schmid. Vector quantizing feature space with a regular lattice. *International Conference on Computer Vision*, 2007.
- [20] T. Tuytellars, K. Mikolajczyk. Local invariant feature detectors: A survey. *Foundations and Trends® in Computer Graphics and Vision*, wolumen 3, strony 177–280, 2007.
- [21] S. Winder, G. Hua, M. Brown. Picking the best daisy. *Conference on Computer Vision and Pattern Recognition*, 2009.

Spis rysunków

2.1	Zestaw obrazów testowych do porównania algorytmów	5
2.2	Liczba wykrytych cech	5
2.3	czas potrzebny na wykrycie cechy	6
2.4	Skuteczność śledzenia wyszukanych cech	6
2.5	Zestawienie kolejnych oktafów badanego obrazu.	8
2.6	Schemat generowania obrazów różnicy Gaussianów	9
2.7	Porównywanie otoczenia piksela w celu znalezienia lokalnego ekstremum . .	10
2.8	Badanie otoczenia punktu charakterystycznego - rozmycie	12
2.9	Badanie otoczenia punktu charakterystycznego - badanie gradientów	12
2.10	Histogram reprezentujący orientacje gradientów wokół punktu charakterystycznego	13
2.11	Tworzenie deskryptora punktu charakterystycznego - lokalna siatka histogramów	13
2.12	Tworzenie deskryptora punktu charakterystycznego - lokalna siatka histogramów	14
2.13	Siatka histogramów uwzględniająca odległość od punktu charakterystycznego	14
2.14	Obliczanie intensywności wybranego obszaru w oparciu o obrazy całkowe .	15
2.15	Pochodne cząstkowe drugiego stopnia funkcji rozmycia Gaussa	16
2.16	Dyskretnie pochodne cząstkowe drugiego stopnia Gaussianów	16
2.17	Aproksymacje pochodnych cząstkowych drugiego stopnia Gaussianów	16
2.18	Porównanie działania operatorów dokładnych i aproksymowanych w zależności od kąta rotacji obrazu	17
2.19	Piramida skalo-przestrzenna	18
2.20	Przeskalowane okna rozmycia	18
2.21	Okna rozmycia	19
2.22	Rozmiar okna filtra w zależności od oktawy i stopnia rozmycia	19
2.23	Ilość lokalizowanych punktów charakterystycznych w zależności od skali . .	20
2.24	Badanie orientacji punktu charakterystycznego	21
2.25	Badane obszary charakterystyczne	21
2.26	Badanie otoczenia punktu charakterystycznego	22
2.27	Przykłady wektorów opisujących subregiony w zależności od natury obszaru charakterystycznego	22
2.28	Porównanie deskryptorów SURF	23
2.29	Wpływ szumu na wygląd deskryptorów subregionów algorytmów SIFT i SURF	23
2.30	Przykład zastosowania pochyłych obrazów całkowych dla trapezoidalnego obszaru	24
2.31	Schemat dwustopniowego filtra otoczeniowo-centrycznego	25
2.32	Dwustopniowe filtry otoczeniowo-centryczne	25

2.33 Dwustopniowy filtr otoczeniowo-centryczny wykorzystywany w algorytmie STAR	25
2.34 Badanie otoczenia punktu potencjalnego rogu	28
2.35 Skuteczność rozpoznawania deskryptorów w zależności od stopnia wstęp- nego rozmycia obrazu	32
2.36 Przegląd metod losowania par punktów do testu binarnego	34
2.37 Skuteczność metod losowania par badana na zbiorze Mikołajczka WALL . .	35
2.38 podpis	35
2.39 podpis	36
2.40 podpis	36
2.41 podpis	37
2.42 podpis	37
2.43 podpis	38
2.44 podpis	38
4.1 BIKES - zbiór obrazów testowych	46
4.2 BIKES - ilość wyszukanych cech	47
4.3 BIKES - czas lokalizowania pojedynczego punktu charakterystycznego . .	48
4.4 BIKES - czas generowania pojedynczego deskryptora punktu charaktery- stycznego	48
4.5 BIKES - powtarzalność wykrywanych cech	49
4.6 BIKES - procent poprawnych dopasowań	49
4.7 TREES - zbiór obrazów testowych	50
4.8 TREES - ilość wyszukanych cech	51
4.9 TREES - czas lokalizowania pojedynczego punktu charakterystycznego . .	52
4.10 TREES - czas generowania pojedynczego deskryptora punktu charaktery- stycznego	52
4.11 TREES - powtarzalność wykrywanych cech	53
4.12 TREES - procent poprawnych dopasowań	54
4.13 Zlokalizowane przez algorytm FAST punkty charakterystyczne w obrazach TREES 1 i TREES 3	55
4.14 GRAFFITI - zbiór obrazów testowych	55
4.15 GRAFFITI - ilość wyszukanych cech	56
4.16 GRAFFITI - czas lokalizowania pojedynczego punktu charakterystycznego	57
4.17 GRAFFITI - czas generowania pojedynczego deskryptora punktu charak- terystycznego	57
4.18 GRAFFITI - powtarzalność wykrywanych cech	58
4.19 GRAFFITI - procent poprawnych dopasowań	59
4.20 WALL - zbiór obrazów testowych	61
4.21 WALL - ilość wyszukanych cech	61
4.22 WALL - czas lokalizowania pojedynczego punktu charakterystycznego . .	62
4.23 WALL - czas generowania pojedynczego deskryptora punktu charaktery- stycznego	62
4.24 WALL - powtarzalność wykrywanych cech	63
4.25 WALL - procent poprawnych dopasowań	63
4.26 BARK - zbiór obrazów testowych	66
4.27 BARK - ilość wyszukanych cech	66
4.28 BARK - czas lokalizowania pojedynczego punktu charakterystycznego . .	67

4.29 BARK - czas generowania pojedynczego deskryptora punktu charakterystycznego	67
4.30 BARK - powtarzalność wykrywanych cech	68
4.31 BARK - procent poprawnych dopasowań	68
4.32 BOAT - zbiór obrazów testowych	69
4.33 BOAT - ilość wyszukanych cech	70
4.34 BOAT - czas lokalizowania pojedynczego punktu charakterystycznego	71
4.35 BOAT - czas generowania pojedynczego deskryptora punktu charakterystycznego	71
4.36 BOAT - powtarzalność wykrywanych cech	72
4.37 BOAT - procent poprawnych dopasowań	73
4.38 CARS -Zbiór obrazów testowych	75
4.39 CARS -ilosc wyszukanych cech	75
4.40 CARS - czas lokalizowania pojedynczego punktu charakterystycznego	76
4.41 CARS - czas generowania pojedynczego deskryptora punktu charakterystycznego	76
4.42 CARS - powtarzalność wykrywanych cech	77
4.43 CARS - procent poprawnych dopasowań	77
5.1 R1 - zbiór obrazów testowych	81
5.2 R1 - ilość wyszukanych cech	82
5.3 R1 - czas lokalizowania pojedynczego punktu charakterystycznego	82
5.4 R1 - czas generowania pojedynczego deskryptora punktu charakterystycznego	83
5.5 R1 - powtarzalność wykrywanych cech	83
5.6 R1 - procent poprawnych dopasowań	84
5.7 R2 - zbiór obrazów testowych	86
5.8 R2 - ilość wyszukanych cech	87
5.9 R2 - czas lokalizowania pojedynczego punktu charakterystycznego	87
5.10 R2 - czas generowania pojedynczego deskryptora punktu charakterystycznego	88
5.11 R2 - powtarzalność wykrywanych cech	88
5.12 R2 - procent poprawnych dopasowań	89
5.13 R3 - zbiór obrazów testowych	91
5.14 R3 - ilość wyszukanych cech	91
5.15 R3 - czas lokalizowania pojedynczego punktu charakterystycznego	92
5.16 R3 - czas generowania pojedynczego deskryptora punktu charakterystycznego	92
5.17 R3 - powtarzalność wykrywanych cech	93
5.18 R3 - procent poprawnych dopasowań	93
5.19 V1 - zbiór obrazów testowych	95
5.20 V1 - ilość wyszukanych cech	96
5.21 V1 - czas lokalizowania pojedynczego punktu charakterystycznego	97
5.22 V1 - czas generowania pojedynczego deskryptora punktu charakterystycznego	97
5.23 V1 - powtarzalność wykrywanych cech	98
5.24 V1 - procent poprawnych dopasowań	99
5.25 V2 - zbiór obrazów testowych	101
5.26 V2 - ilość wyszukanych cech	102
5.27 V2 - czas lokalizowania pojedynczego punktu charakterystycznego	102
5.28 V2 - czas generowania pojedynczego deskryptora punktu charakterystycznego	103
5.29 V2 - powtarzalność wykrywanych cech	103
5.30 V2 - procent poprawnych dopasowań	104

5.31 Z1 - zbiór obrazów testowych	106
5.32 Z1 - ilość wyszukanych cech	106
5.33 Z1 - czas lokalizowania pojedynczego punktu charakterystycznego	107
5.34 Z1 - czas generowania pojedynczego deskryptora punktu charakterystycznego	107
5.35 Z1 - powtarzalność wykrywanych cech	108
5.36 Z1 - procent poprawnych dopasowań	108
A.1 Okno startowe aplikacji testowej	112
A.2 Przykładowa zakładka algorytmu - konfiguracja	112
A.3 Przykładowa zakładka algorytmu - prezentacja wyników	112

Spis tabel

2.1 Wartości charakterystyczne dla okien aproksymowanych filtrów rozmycia	20
4.1 BIKES - ilość wyszukanych cech	47
4.2 BIKES - czas lokalizowania pojedynczego punktu charakterystycznego	47
4.3 BIKES - czas generowania pojedynczego deskryptora punktu charakterystycznego	47
4.4 BIKES - powtarzalność wykrywanych cech	48
4.5 BIKES - procent poprawnych dopasowań	49
4.6 TREES - ilość wyszukanych cech	51
4.7 TREES - czas lokalizowania pojedynczego punktu charakterystycznego w obrazach z zestawu TREES	51
4.8 TREES - czas generowania pojedynczego deskryptora punktu charakterystycznego	52
4.9 TREES - powtarzalność wykrywanych cech	53
4.10 TREES - procent poprawnych dopasowań	53
4.11 GRAFFITI - ilość wyszukanych cech	56
4.12 GRAFFITI - czas lokalizowania pojedynczego punktu charakterystycznego	56
4.13 GRAFFITI - czas generowania pojedynczego deskryptora punktu charakterystycznego	57
4.14 GRAFFITI - powtarzalność wykrywanych cech	58
4.15 GRAFFITI - procent poprawnych dopasowań	58
4.16 WALL - ilość wyszukanych cech	59
4.17 WALL - czas lokalizowania pojedynczego punktu charakterystycznego	60
4.18 WALL - czas generowania pojedynczego deskryptora punktu charakterystycznego	60
4.19 WALL - powtarzalność wykrywanych cech	60
4.20 WALL - procent poprawnych dopasowań	60
4.21 BARK - ilość wyszukanych cech	64
4.22 BARK - czas lokalizowania pojedynczego punktu charakterystycznego	65
4.23 BARK - czas generowania pojedynczego deskryptora punktu charakterystycznego	65
4.24 BARK - powtarzalność wykrywanych cech	65
4.25 BARK - procent poprawnych dopasowań	65
4.26 BOAT - ilość wyszukanych cech	70
4.27 BOAT - czas lokalizowania pojedynczego punktu charakterystycznego	70
4.28 BOAT - czas generowania pojedynczego deskryptora punktu charakterystycznego	71
4.29 BOAT - powtarzalność wykrywanych cech	72
4.30 BOAT - procent poprawnych dopasowań	72

4.31 CARS - ilość wyszukanych cech	74
4.32 CARS - czas lokalizowania pojedynczego punktu charakterystycznego	74
4.33 CARS - czas generowania pojedynczego deskryptora punktu charakterystycznego	74
4.34 CARS - powtarzalność wykrywanych cech	74
4.35 CARS - procent poprawnych dopasowań	76
5.1 R1 - ilość wyszukanych cech	80
5.2 R1 - czas lokalizowania pojedynczego punktu charakterystycznego	80
5.3 R1 - czas generowania pojedynczego deskryptora punktu charakterystycznego	81
5.4 R1 - powtarzalność wykrywanych cech	82
5.5 R1 - procent poprawnych dopasowań	83
5.6 ilość wyszukanych cech	84
5.7 R2 - czas lokalizowania pojedynczego punktu charakterystycznego w obrazach z zestawu R2	85
5.8 R2 - czas generowania pojedynczego deskryptora punktu charakterystycznego	85
5.9 R2 - powtarzalność wykrywanych cech	85
5.10 R2 - procent poprawnych dopasowań	85
5.11 R3 - ilość wyszukanych cech	89
5.12 R3 - czas lokalizowania pojedynczego punktu charakterystycznego	90
5.13 R3 - czas generowania pojedynczego deskryptora punktu charakterystycznego	90
5.14 R3 - powtarzalność wykrywanych cech dla obrazów z zestawu R3	90
5.15 R3 - procent poprawnych dopasowań	90
5.16 V1 - ilość wyszukanych cech	96
5.17 V1 - czas lokalizowania pojedynczego punktu charakterystycznego	96
5.18 V1 - czas generowania pojedynczego deskryptora punktu charakterystycznego	97
5.19 V1 - powtarzalność wykrywanych cech	98
5.20 V1 - procent poprawnych dopasowań	98
5.21 V2 - ilość wyszukanych cech	99
5.22 V2 - czas lokalizowania pojedynczego punktu charakterystycznego	100
5.23 V2 - czas generowania pojedynczego deskryptora punktu charakterystycznego	100
5.24 V2 - powtarzalność wykrywanych cech	100
5.25 V2 - procent poprawnych dopasowań	100
5.26 Z1 - ilość wyszukanych cech	104
5.27 Z1 - czas lokalizowania pojedynczego punktu charakterystycznego	105
5.28 Z1 - czas generowania pojedynczego deskryptora punktu charakterystycznego	105
5.29 Z1 - powtarzalność wykrywanych cech	105
5.30 Z1 - procent poprawnych dopasowań	105