Deskryptory punktów charakterystycznych

Bartosz Wieloch

Przetwarzanie i Rozpoznawanie Obrazów

May 18, 2016

Wstęp

Często spotykany (typowy) schemat przetwarzanie obrazu/sekwencji wideo:

1. Detekcja punktów charakterystycznych
2. Opis wyznaczonych punktów
3. Konkretne zastosowanie:
   - estymacja (np. wykrywanie linii, określanie pozycji/orientacji 3d)
   - dopasowywanie (np. szukanie konkretnego obiektu w obrazie)
   - indeksowanie (np. szukanie podobnych do siebie obrazów)
   - detekcja (np. wykrywanie w obrazie obiektów z bazy danych)

Przykładowe zastosowania

- Wyznaczanie macierzy przekształceń pomiędzy parą obrazów
  (jaka transformacja przekształci jeden obraz w drugi)
- Rekonstrukcja 3D
- Estymacja ruchu
- Wykrywanie różnych zdjęć tego samego obiektu
- Tworzenie panoramy z mniejszych obrazów
- Detekcja obiektu w obrazie
  (np. wykrywanie konkretnych produktów na zdjęciu ze sklepu)
- Detekcja akcji w sekwencji wideo
  (np. wykrywanie osoby wzywającej pomocy, wykrywanie paniki w tłumie)
- Nawigacja w robotyce

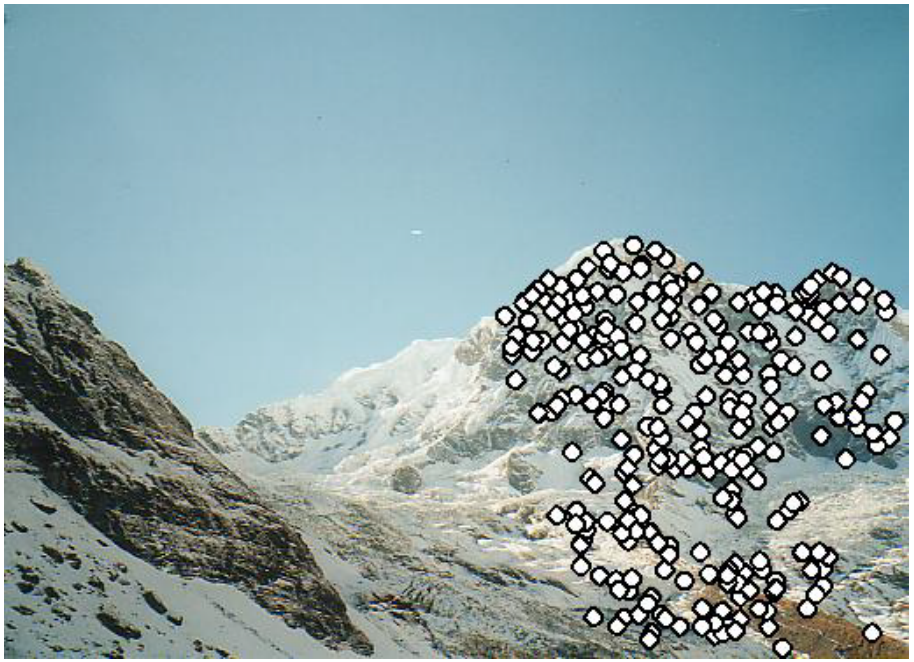
Dlaczego dopasowywanie obrazów jest trudne?

Jak dopasować do siebie te obrazy?

Jak dopasować do siebie te obrazy?

- Znajdź charakterystyczne punkty w obu obrazach.
- Dopasuj odpowiadające sobie pary punktów.
- Użyj ich do wyrównania/nałożenia na siebie obrazów.

Jak dopasować do siebie te obrazy?



- Znajdź charakterystyczne punkty w obu obrazach.
- Dopasuj odpowiadające sobie pary punktów.
- Użyj ich do wyrównania/nałożenia na siebie obrazów.

Jak dopasować do siebie te obrazy?

- Znajdź charakterystyczne punkty w obu obrazach.
- Dopasuj odpowiadające sobie pary punktów.
- Użyj ich do wyrównania/nałożenia na siebie obrazów.

Dopasowywanie punktów

Problemy:

- wykrycie tych samych widocznych punktów niezależnie w obu obrazach,
- znalezienie dla każdego punktu odpowiadającego mu punktu w drugim obrazie,

A więc potrzebujemy:

- detektora który jest *powtarzalny* (wskazuje te same punkty),
- deskryptora który jest *wiarygodny* (opisuje tak samo te same punkty) i *odróżniający* (nie opisuje wszystkiego tak samo)

Model zniekształceń

Geometryczne:

- obrót
- skalowanie
- zmiana punktu widzenia (perspektywa)

Fotometryczne:

- zmiana jasności

Detekcja punktów

Podział

- Narożnik — przecięcie się krawędzi
  - dokładna lokalizacja
  - mniej wyróżniający się (w porównaniu do blobów)
- Blob (plama) — wzorzec który wyraźnie różni się od sąsiednich rejonów obrazu w jasności/teksturze (np. podobny kolor, kółko, itp.)
  - mniejsza dokładność lokalizacji
  - bardziej wyróżniający się od narożników

Detekcja narożników

- Patrzymy na punkt przez niewielkie okno.
- Przesuwamy okno w różnych kierunkach i obserwujemy zmiany jasności obserwowanych punktów (np. licząc sumę kwadratów różnic poszczególnych punktów — SSD):
  - dla regionów "płaskich" różnica jest mała (≈ 0) w każdym kierunku,
  - dla krawędzi różnica jest ≈ 0 wzdłuż niej, i $\gg 0$ w innych kierunkach,
  - dla narożników różnica jest $\gg 0$ w każdym kierunku.

- Rozważmy 2 fragmenty obrazu (ang. patches) o środkach w punktach (x, y) oraz $(x + \Delta x,\ y + \Delta y)$
- Suma kwadratów różnic pomiędzy nimi jest równa

$$SSD(\Delta x, \Delta y) = \sum_{x,y \in P} (I(x, y) - I(x + \Delta x,\ y + \Delta y))^2$$

- Pochodne cząstkowe z obrazu I to $I_x = \frac{\partial I(x,y)}{\partial x}$ oraz $I_y = \frac{\partial I(x,y)}{\partial y}$. Przybliżając szeregiem Taylora 1go stopnia mamy:

$$I(x + \Delta x,\ y + \Delta y) \approx I(x, y) + I_x(x, y)\Delta x + I_y(x, y)\Delta y$$

- Stąd mamy przybliżenie

$$SSD(\Delta x, \Delta y) \approx \sum_{x,y \in P} (I_x(x, y)\Delta x + I_y(x, y)\Delta y)^2$$

$$SSD(\Delta x, \Delta y) \approx \sum_{x,y \in P} (I_x(x,y)\Delta x + I_y(x,y)\Delta y)^2$$

- Możemy zapisać w postaci macierzowej:

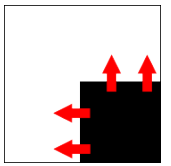
$$SSD(\Delta x, \Delta y) \approx \sum \begin{bmatrix} \Delta x & \Delta y \end{bmatrix} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

$$SSD(\Delta x, \Delta y) \approx \begin{bmatrix} \Delta x & \Delta y \end{bmatrix} M \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

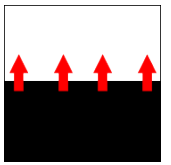
- gdzie

$$M = \sum_{x,y \in P} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}$$

- Zakładamy narożnik wyrównany do osi układu współrzędnych czyli dominujący gradient wzdłuż osi x albo y



- $M = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$
- Jeśli λ_1 albo λ_2 jest bliska 0 to punkt nie jest narożnikiem, np.:

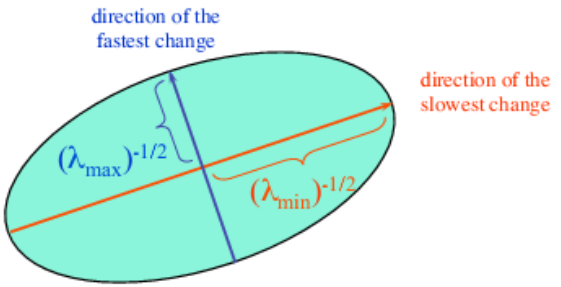


$$M = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

- Ponieważ macierz M jest symetryczna możemy ją zdekomponować:

$$M = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$

- λ_1 i λ_2 to wartości własne macierzy M
- Wektory własne pokazują kierunek największej i najmniejszej zmiany SSD



Kryterium na wartościach własnych

- Punkt klasyfikujemy na podstawie wartości własnych macierzy M – jeśli λ_1 i λ_2 są odpowiednio duże to punkt jest narożnikiem:

$$R = min(\lambda_1, \lambda_2) > threshold$$

- Detektor używający takiego kryterium nazywa się detektorem “Shi-Tomasi”
  J. Shi and C. Tomasi (June 1994). "Good Features to Track,". 9th IEEE Conference on Computer Vision and Pattern Recognition
- Obliczenie λ_1 i λ_2 jest kosztowne (trzeba policzyć dla każdego punktu obrazu) dlatego Harris&Stephens zapronowali inne kryterium:

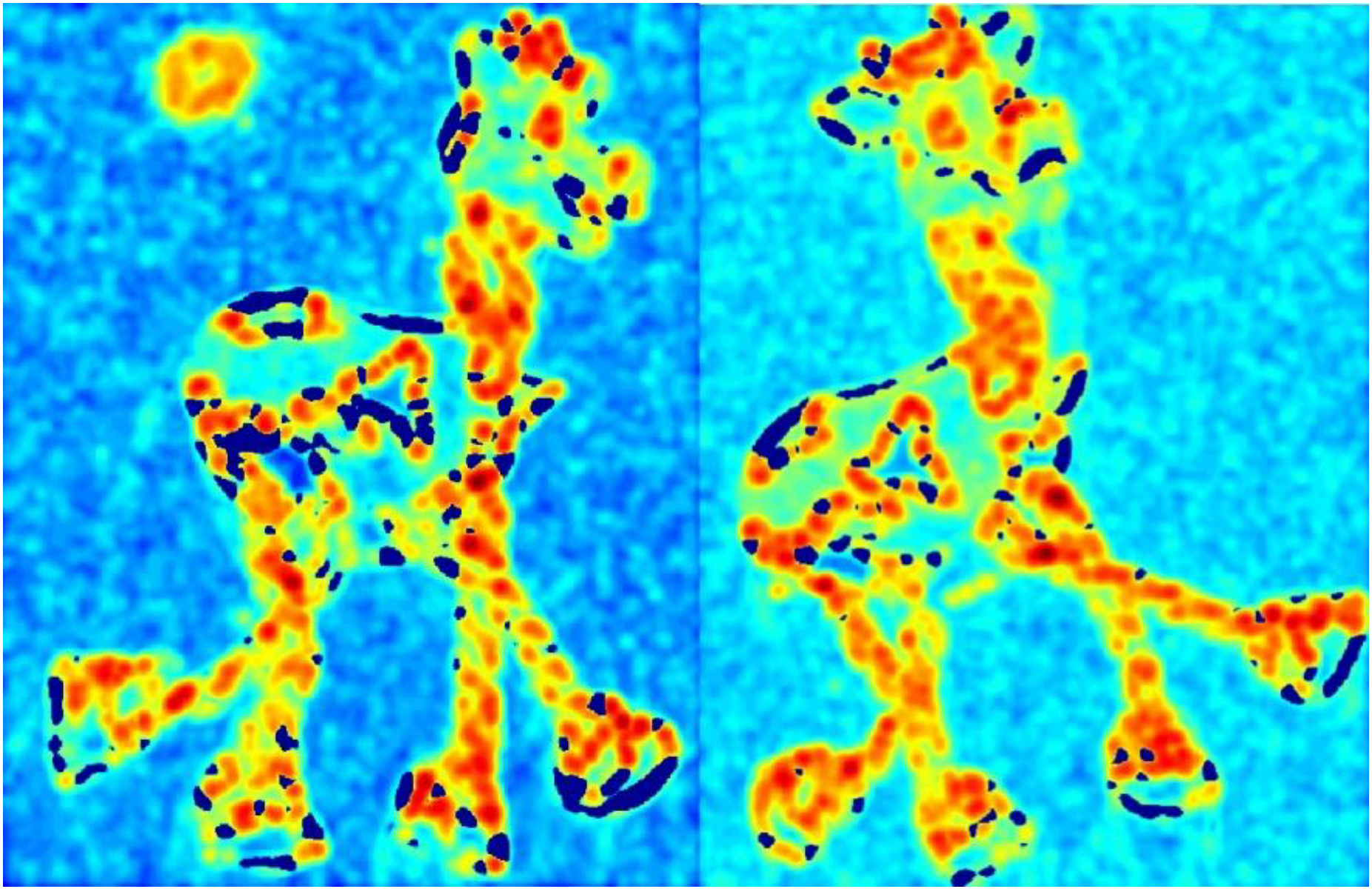
$$R = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2 = det(M) - k \cdot trace^2(M)$$

gdzie k jest “magiczną liczbą” z zakresu $(0.04; 0.15)$

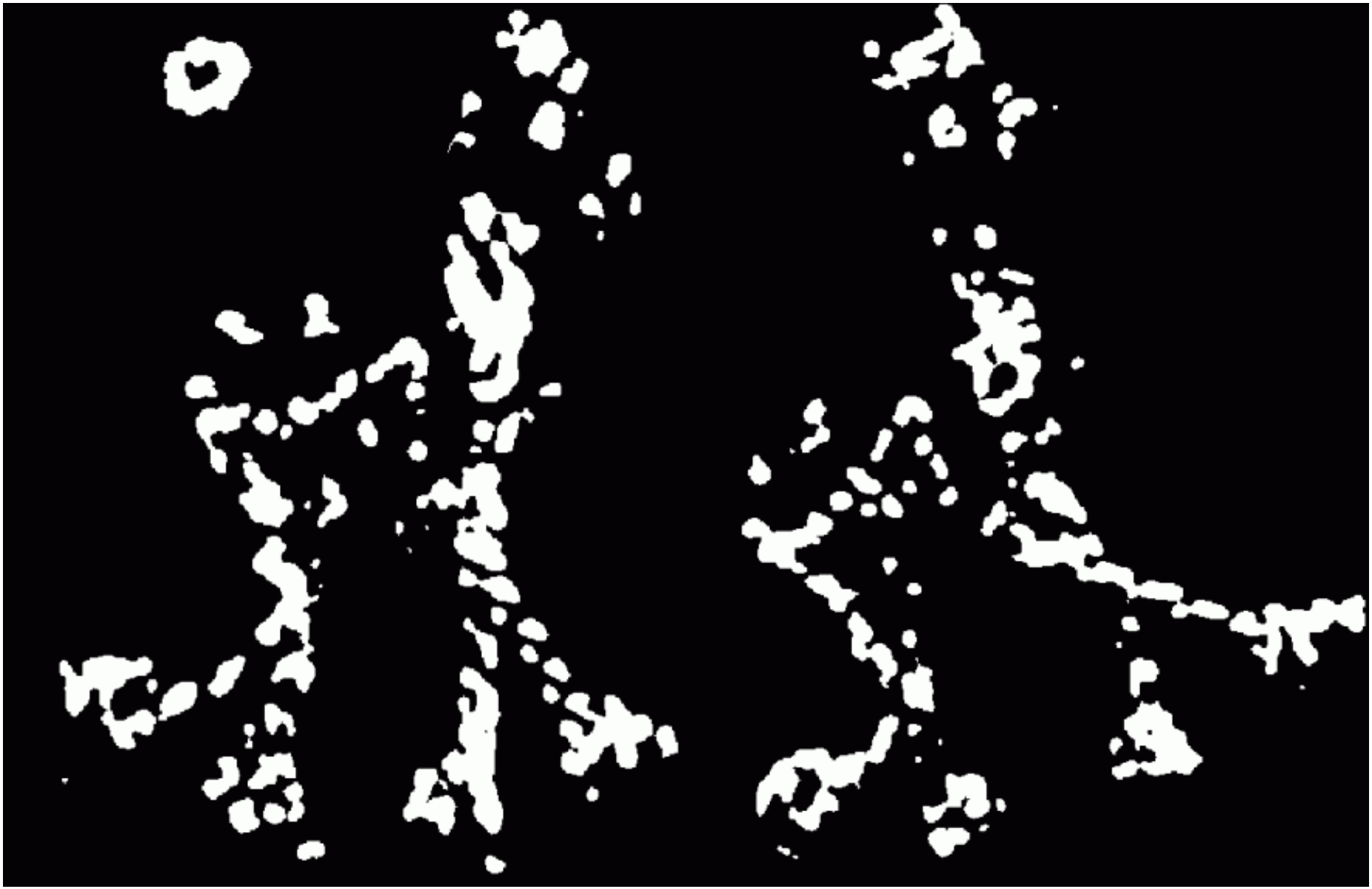
1. Policz pochodne cząstkowe dla obrazu (np. filtrem Sobela).
2. Policz I_x^2, I_y^2 i $I_x I_y$.
3. Oblicz splot I_x^2, I_y^2 i $I_x I_y$ z filtrem prostokątnym aby obliczyć: $\sum I_x^2$, $\sum I_y^2$ i $\sum I_x I_y$ (elementy macierzy M)
   Alternatywnie można zastosować filtr Gaussowski.
4. Oblicz R wg. wzoru Harrisa lub Shi-Tomasi.
5. Znajdź punkty $R > \mathit{threshold}$.
6. Weź lokalne maxima.

Obrazki

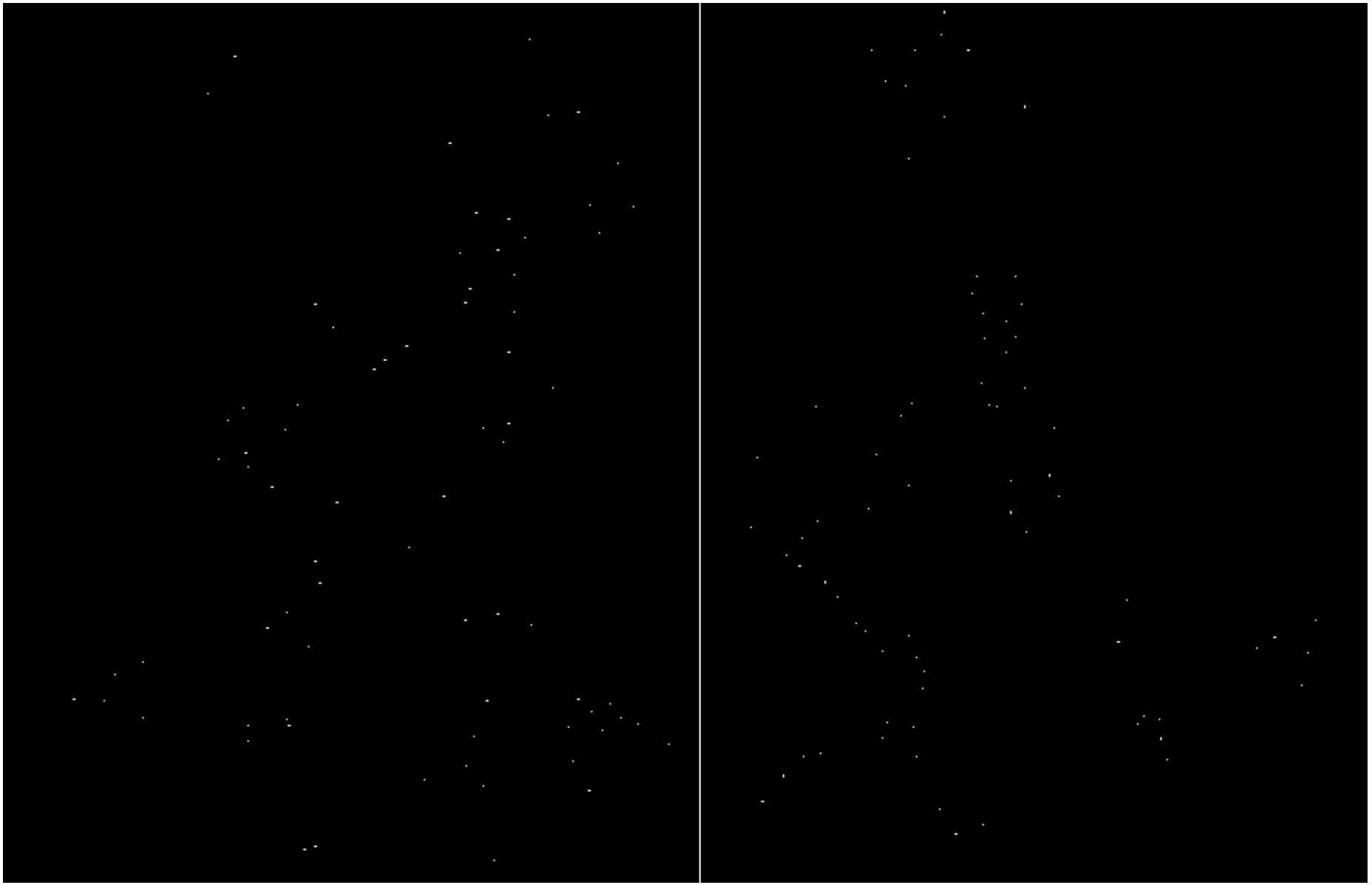
Odpowiedź R



Punkty dla których R jest większe od progu



Lokalne maxima R po progowaniu



Wyznaczone punkty na oryginalnym obrazie

Cechy detektora Harrisa

- Jest niezmienniczy ze względu na obrót
  - obrót powoduje obrócenie osi elipsoidy, ale
  - jej kształt (a więc wartości własne λ_1 i λ_2) pozostaje taki sam
- Ale zależy od skali obrazka!
  - przy dużym powiększeniu punkty mogą być zaklasyfikowane jako krawędzie
  - przy mniejszym obrazku fragment może wyglądać na narożnik

Dopasowywanie fragmentów

- Jak możemy sobie poradzić w sytuacji gdy oba obrazy mają różną skalę?
- Jeden ze sposobów: przeskaluj fragment przed porównaniem.
- Wada: jest wolne — dla każdej porównywanej pary trzeba skalować rozmiar jednego z fragmentów indywidualnie.
- Możliwe rozwiązanie: przypisz każdemu punktowi jego własną optymalną "skalę" (rozmiar)
  - Jaki jest optymalny rozmiar fragmentu?

Automatyczny wybór skali

- Zaprojektuj funkcję:
  - argument: fragment obrazu
  - ma taką samą wartość dla odpowiadających sobie regionów obrazu nawet gdy są one w różnej skali (wtedy wartość funkcji taka sama gdy argumenty są fragmentami o różnych rozmiarach)
- Oblicz wartość tej funkcji dla różnych wielkości fragmentów.
- Wybierz rozmiar fragmentu (szukana skala) dla której zaprojektowana funkcja osiąga lokalne maximum.
- Znormalizuj wielkość fragmentu.
- Zaleta: szybkie — "skala" jest ustalana dla każdego punktu zupełnie niezależnie!

Automatyczny wybór skali

- Dobra funkcja powinna mieć jedno, wyraźne maximum.
- Ostre, lokalne zmiany intensywności wartości idealnie nadają się do identyfikowania skali.
- Laplacian of Gaussian (LoG)

$$LoG = \nabla^2 G(x, y) = \frac{\partial^2 G(x, y)}{\partial x^2} + \frac{\partial^2 G(x, y)}{\partial y^2}$$

- "Skala" jest ustalana na lokalnym maximum po kolejnych rozmyciach obrazu

Deskryptory

Pożądana cecha — niezmienniczość ze względu na transformacje:

- translację,
- obrót 2D,
- skalowanie,
- niewielkie zmiany perspektywy (obrót 3D),
- liniową zmianę jasności punktów.

Jak uzyskać niezmienniczość?

Normalizacja skali i obrotu fragmentu (patcha):

- Znajdź poprawną skalę (np. używając operatora LoG).
- Przeskaluj fragment do ustalonego rozmiaru (np. 8x8 punktów)
- Znajdź lokalną orientację
  (np. dominujący kierunek gradientu w fragmencie — wektory własne)
- Obróć fragment obrazu.

Jak uzyskać niezmienniczość?

Normalizacja afiniczna:

- Umożliwia osiągnięcie pewnej niezmienniczości w stosunku do zmiany punktu obserwacji (perspektywy).
- Macierz M może być wykorzystana do wyznaczenia kierunku najszybszej oraz najwolniejszej zmiany jasności pikseli wokół wykrytego punktu.
- Wyznacz eliptyczny fragment obrazu (w skali wyznaczonej np. operatorem LoG) o osiach zgodnych z wyznaczonymi kierunkami zmian intensywności punktów.
- Znormalizuj eliptyczny region do koła.

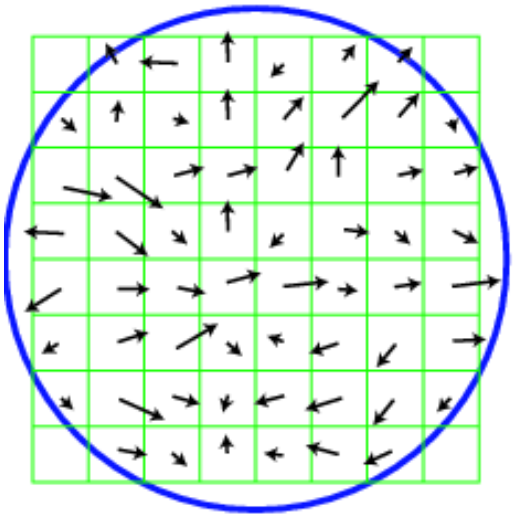
Fragment jako deskryptor

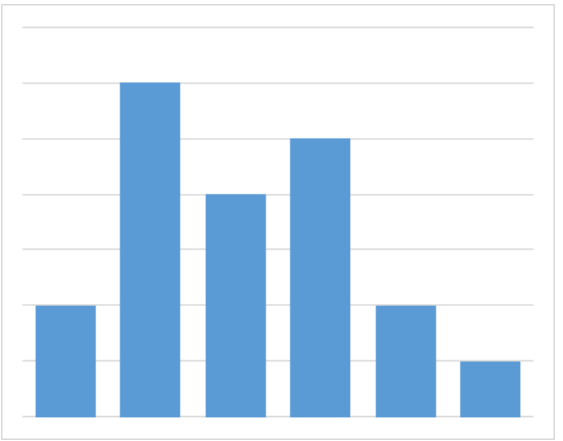
Wady:

- niewielkie różnice w obrocie, skalowaniu, kąta patrzenia, jasności puntów itp. mogą istotnie wpłynąć na wartość funkcji dopasowania fragmentów,
- kosztowne obliczeniowo — wymaga fizycznego wykonania normalizacji każdego fragmentu

Dlatego lepszym rozwiązaniem jest budowanie deskryptora na podstawie np. histogramu orientacji gradientów (ang. Histogram of Oriented Gradients - HOG)

- Wyznacz histogram orientacji gradientów (z jasności punktów).
- Największa wartość w histogramie to dominująca orientacja (orientacja punktu)
  - jeśli maksymalna wartość w histogramie niejednoznaczna to stwórz osobne punkty charakterystyczne dla każdej takiej orientacji
- Obróć zgodnie z wyznaczoną orientacją.

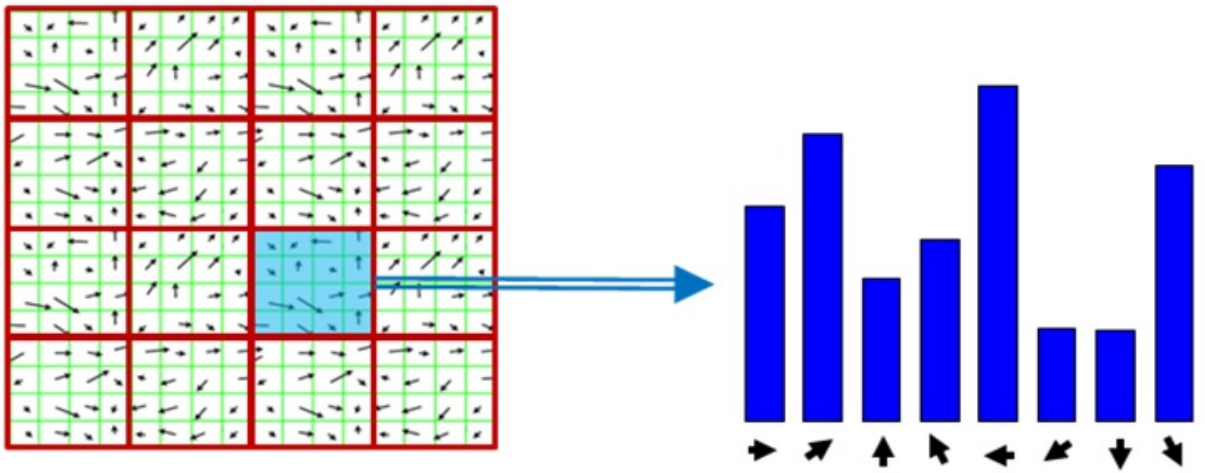




- Scale Invariant Feature Transform
- Wynaleziony przez Davida Lowe w 2004

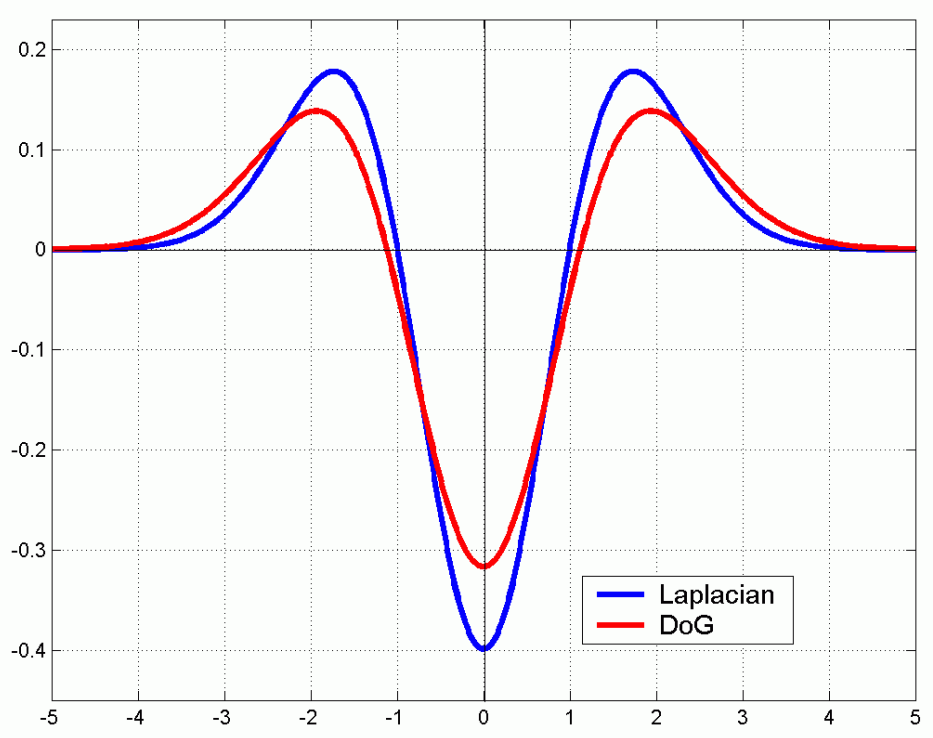
Obliczenia:

- Podziel fragment na 4x4 podfragmenty (16 komórek)
- Oblicz histogram orientacji gradientów (8 kubełków) z każdego punktu wewnątrz danego podfragmentu
- Wynikowy deskryptor: 4x4x8=128 wartości

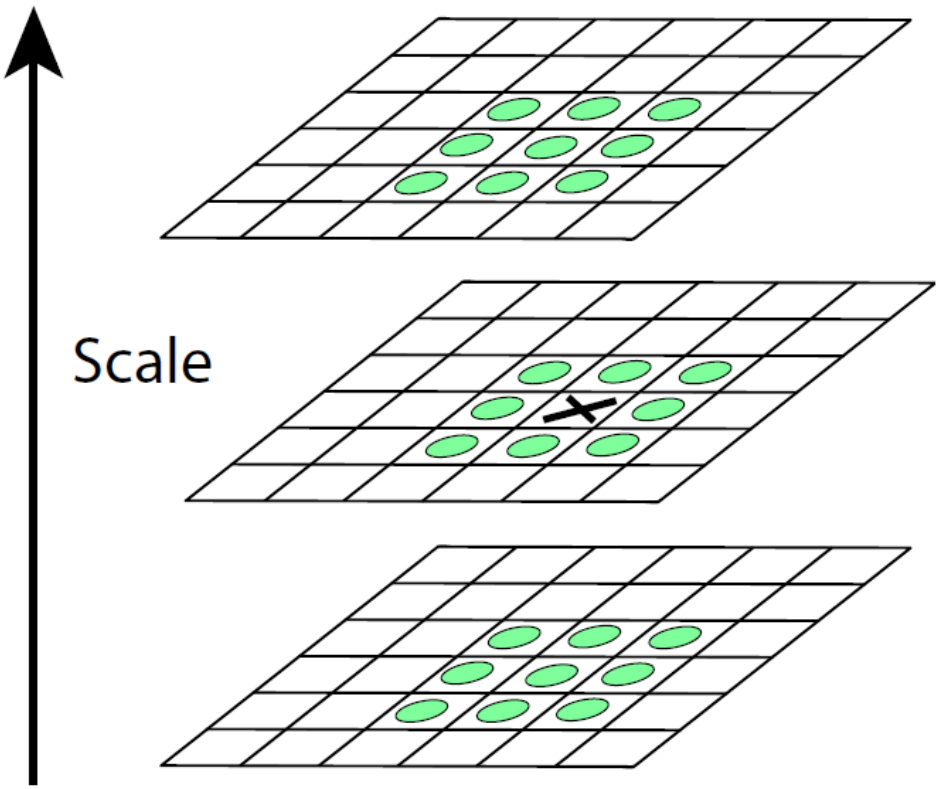


- Dedykowany detektor punktów charakterystycznych
- Przybliża Laplacian of Gaussian (LoG)
  za pomocą Difference of Gaussian (DoG)

$$LoG \approx DoG = G_{k\sigma}(x, y) - G_{\sigma}(x, y)$$



- Lokalne ekstrema w piramidzie DoG
- Jednoczesna detekcja lokalizacji oraz skali



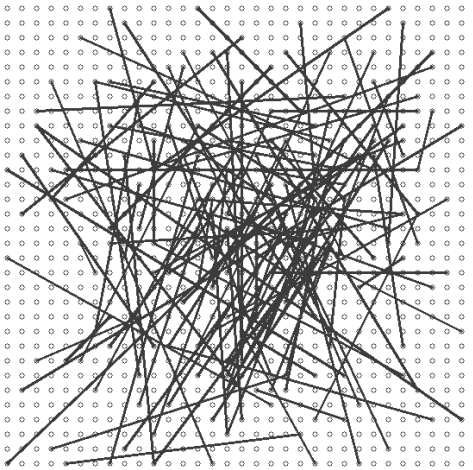
Cechy deskryptora SIFT

- Odporny na obroty, skalowanie, i niewielkie zmiany perspektywy
- Odporny na zmiany oświetlenia
  - czasami nawet pomiędzy dniem a nocą
- Niezbyt szybki
- Opatentowany

Binary Robust Independent Elementary Features

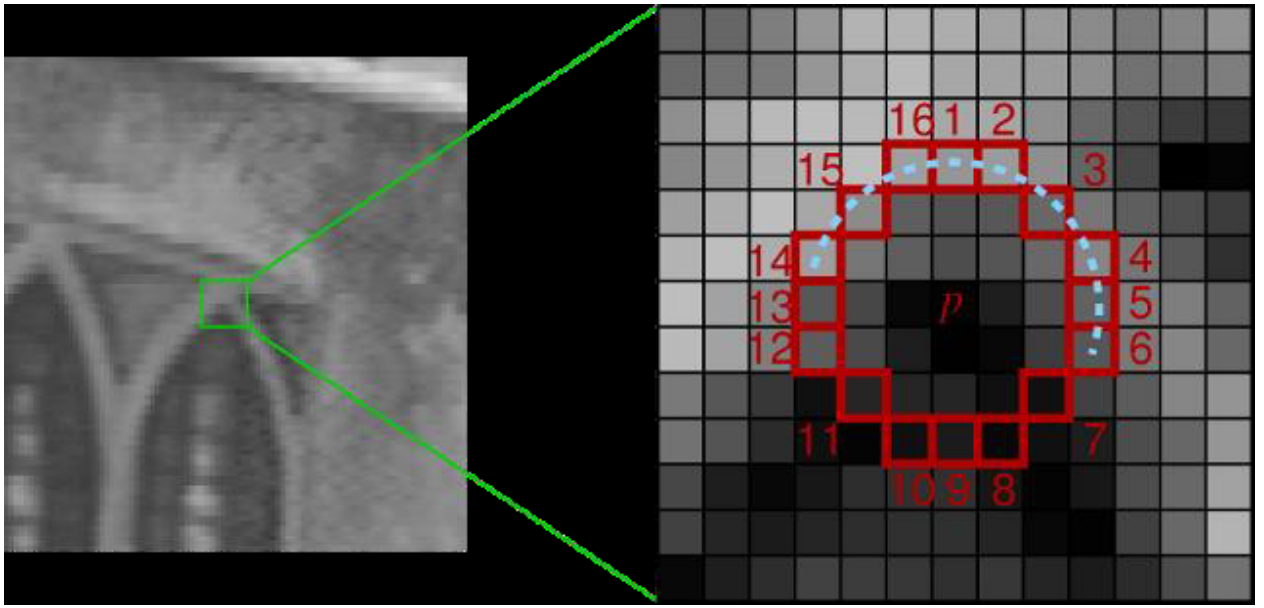
- Rozmyj obraz
- Dla każdego punktu charakterystycznego
- sprawdź w jego otoczeniu 256 par punktów (p_1, p_2) (jednorazowo wylosowanych)
- Jeśli $p_1 < p_2$to ustaw odpowiedni bit deskryptora na 1,
  w przeciwnym wypadku na 0

- Wzorzec punktów jest losowany tylko raz — potem używa się tego samego wzorca.
- Deskryptor nie jest odporny na zmianę skali i obroty.
- Jest bardzo szybki. Porównanie deskryptorów za pomocą odległości Hamminga.



Features from Accelerated Segment Test

- Analizuje jasności punktów na okręgu dookoła rozważanego punktu
- Punkt jest narożnikiem jeśli N kolejnych punktów na okręgu jest:
  - jaśniejszych od niego ($+$ próg), albo
  - ciemniejszy od niego ($+$ próg)
- Zazwyczaj detektor FAST: $N = 9$, okrąg ma 16 punktów



Deskryptor BRISK

Binary Robust Invariant Scalable Keypoints

- Punkty charakterystyczne przy użyciu FAST
- Odporny na obroty i zmianę skali
- Szybki

- Deskryptor binarny — porównywanie par punktów podobnie jak w BRIEF
- Wzorzec jest zdefiniowany
- Porównuje ze sobą punkty bliskie i dalekie