

Wizja Maszynowa

**Algorytmy detekcja i porównywani cech obrazu -
wykład 6**

Adam Szmigielski

aszmigie@pjwstk.edu.pl

materiały: *ftp(public) : //aszmigie/WM*

Właściwości dobrych cech

- **lokalne:** cechy są lokalne, odporne na okluzję i bałagan (bez wcześniejszej segmentacji),
- **Dokładna:** dokładna lokalizacja.
- **Niezmienna** albo **kowariantna**
- **Solidna:** szum, rozmycie, kompresja itp. nie mają dużego wpływu na cechę.
- **Charakterystyczne:** poszczególne cechy można dopasować do dużej bazy danych obiektów.
- **Wydajne:** wydajność zbliżona do czasu rzeczywistego.

Punkty i łąty

- Funkcje punktowe można wykorzystać do znalezienia rzadkiego zestawu odpowiednich lokalizacji na różnych obrazach, często jako odniesienie do pozycji kamery,
- Istnieją dwa główne podejścia do znajdowania punktów charakterystycznych i ich odpowiedników.
 - aby znaleźć cechy na jednym obrazie, które można dokładnie prześledzić za pomocą lokalnej techniki wyszukiwania, takiej jak korelacja lub metoda najmniejszych kwadratów,
 - niezależnie wykrywać cechy na wszystkich rozważanych obrazach, a następnie dopasowywać cechy na podstawie ich lokalnego wyglądu

Podstawowy algorytm wykrywania cech

- Oblicz pochodne poziome i pionowe obrazu I_x i I_y , konwertując oryginalny obraz z pochodnymi Gaussa,
- Oblicz trzy obrazy odpowiadające iloczynom zewnętrznym tych gradientów (macierz A jest symetryczna, więc potrzebne są tylko trzy wpisy),
- Przekształć każdy z tych obrazów w większy Gaussian,
- Oblicz miarę skalarną za pomocą jednej z formuł,
- Znajdź lokalne maksima powyżej określonego progu i zapisz je jako wykryte lokalizacje punktów charakterystycznych.

Algorytmy wykrywania cech

- **Scale Invariant Feature Transform (SIFT)** David Lowe's (2004),
- **SURF (Bay, Ess i in. 2008)**, który wykorzystuje integralne obrazy do szybszych konwekcji,
- **FAST and FASTER (Rosten, Porter i Drummond 2010)**, jeden z pierwszych nauczonych detektorów,
- **BRISK (Leutenegger, Chli i Siegwart 2011)**, który wykorzystuje detektor FAST w przestrzeni skali wraz z deskryptorem ciągu bitowego,
- **ORB (Rublee, Rabaud et al. 2011)**, który dodaje orientacji do FAST,
- **KAZE i Accelerated-KAZE (Alcantarilla, Nuevo i Bartoli 2013)**, które wykorzystują dyfuzję nieliniową do wyboru skali wykrywania cech.

Konwolucyjne sieci neuronowe do wykrywania deskryptorów

1. Uczenie się kowariantnych detektorów cech (Lenc i Vedaldi 2016);
2. Nauka przypisywania orientacji punktom charakterystycznym (Yi, Verdie i in. 2016);
3. LIFT, wyuczone niezmiennie transformacje cech (Yi, Trulls et al.2016),
4. SuperPoint, samonadzorowane wykrywanie i opis punktów zainteresowania (DeTone, Malisiewicz i Rabinovich 2018),
5. LF-Net, uczenie się lokalnych cech z obrazów (Ono, Trulls et al.2018),
6. Key.Net (Barroso-Laguna, Riba i in. 2019), który wykorzystuje połączenie ręcznie opracowanych i wyuczonych funkcji CNN, aby stworzyć metodę, która daje najnowocześniejsze wyniki
7. D2D, Describe-to-Detect, (Tian, Balntas et al.2020), który wyodrębnia gęste lokalne deskryptory cech, które mają wysoką istotność.

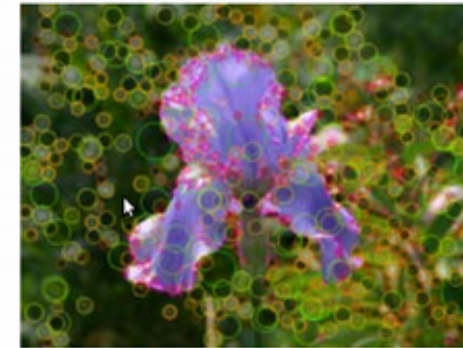
Wykrywanie punktów zainteresowania - przykład



(a)



(b)



(c)

(a) Sample image, (b) Harris response, (c) DoG response
Rozmiary i kolory okręgów wskazują skalę, w której wykryto punkt.

- *Adaptive non-maximal suppression (ANMS)* - większość detektorów cech po prostu szuka lokalnych maksimów w funkcji zainteresowania (punkty będą gęstsze w obszarach o wyższym kontraście),
- Aby uniknąć tego problemu, wykrywaj tylko te cechy, które są jednocześnie lokalnymi maksimami i których wartość odpowiedzi jest istotna.

Znajdowanie cech

Biorąc pod uwagę dużą liczbę detektorów cech, jak możemy zdecydować, których użyć?

- Mierzac *powtarzalność cech* - określ częstotliwość, z jaką powtarzają się punkty kluczowe wykryte na jednym obrazie,
- Mierzac *informacyjną zawartość* - zdefiniuj entropię zbioru niezmiennych rotacyjnie lokalnych deskryptorów w skali szarości.

SIFT

Zaimplementowany w OpenCV (ograniczenia patentowe)

SIFT (Scale Invariant Feature Transform)

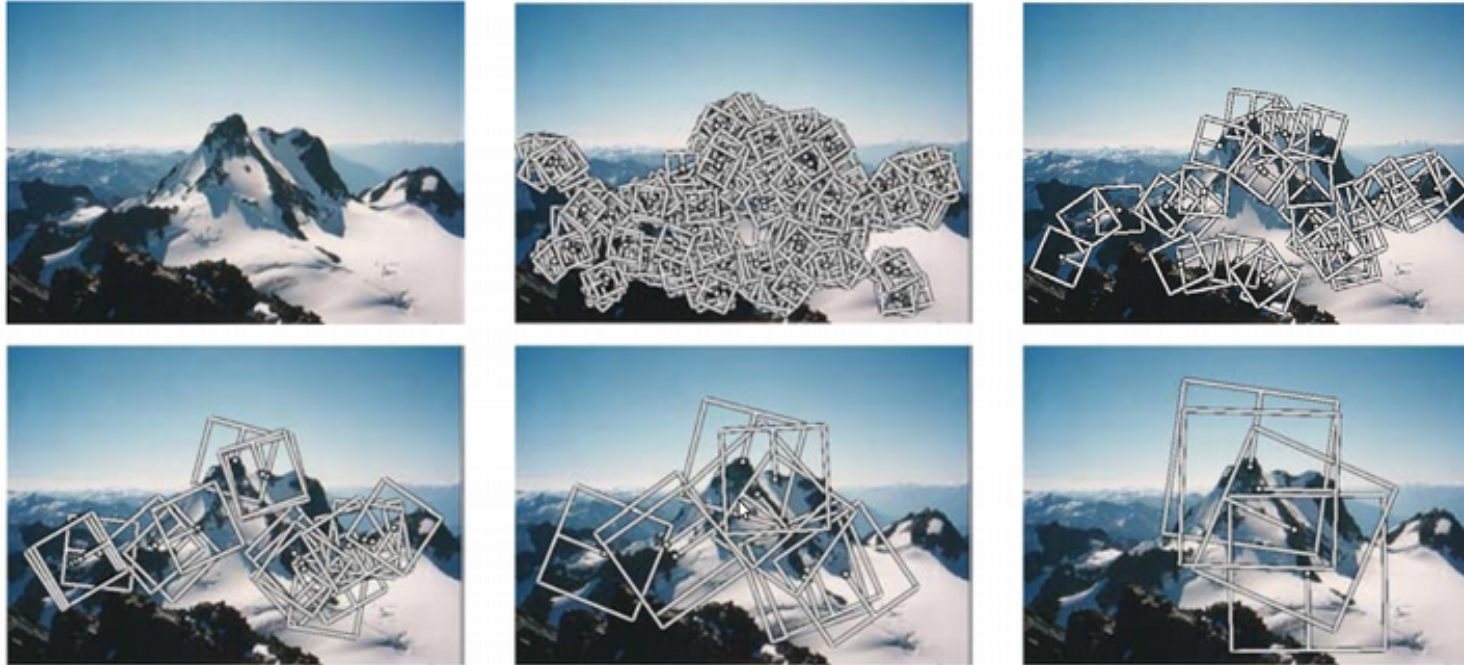
SIFT opisuje zarówno bf detektor, jak i bf deskryptor

Proces wykrywania zwraca: $\{x, y, \sigma, \theta\}$ - lokalizacja, skala, orientacja



- Wykrywanie ekstrem w wielu skalach
- Lokalizacja punktów kluczowych
- Przypisanie orientacji
- Deskryptor punktu kluczowego

Niezmiennność skali



- Wykrywanie obiektów w możliwie najdokładniejszej stabilnej skali może nie być odpowiednie (np. Chmury) - cechy w precyzyjnej skali mogą nie istnieć.
- Rozwiązaniem problemu jest wyodrębnienie funkcji w różnych skalach (poprzez wykonanie tych samych operacji) w wielu rozdzielczościach w piramidzie, a następnie dopasowanie funkcji na tym samym poziomie.

Wykrywanie cech za pomocą różnicy Gaussa

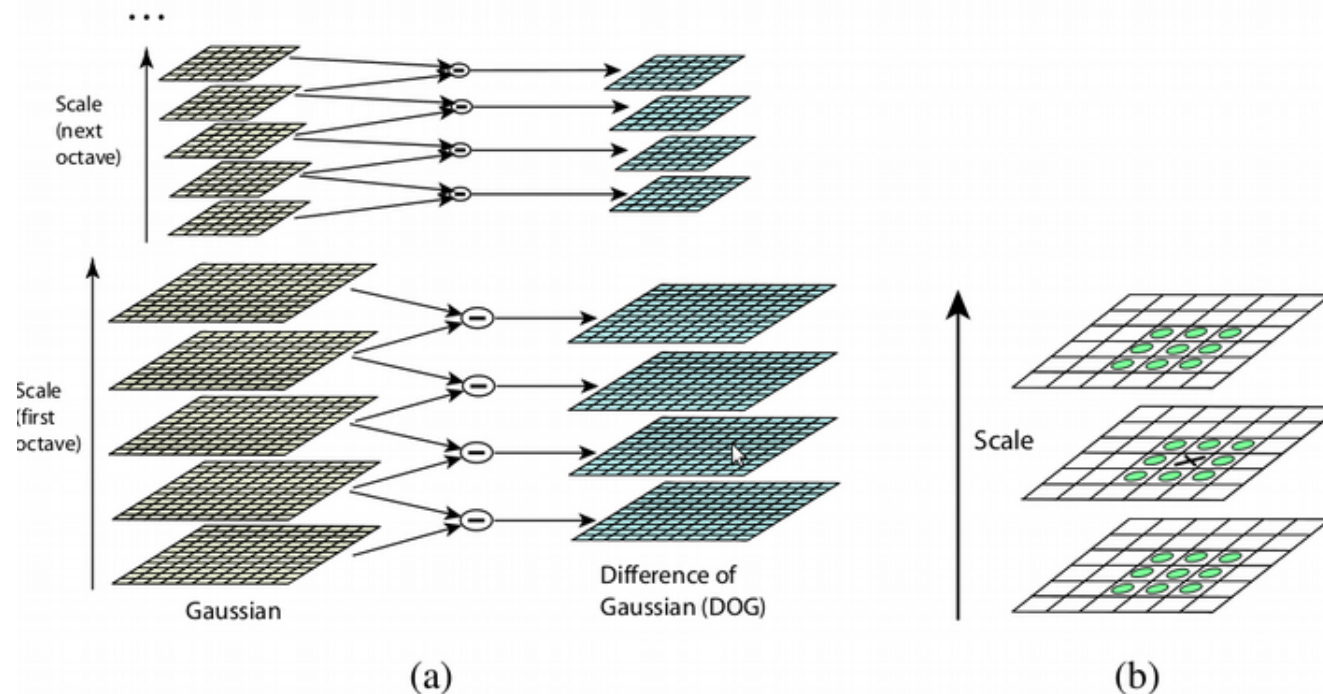
- Podobnie jak w przypadku operatora Harrisa, piksele, w których występuje silna asymetria w lokalnej krzywizny funkcji wskaźnika (w tym przypadku DoG), są odrzucane.
- Jest to realizowane przez najpierw obliczenie lokalnego hesjanu obrazu różnicy D ,

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix},$$

- a następnie odrzucanie punktów kluczowych dla których

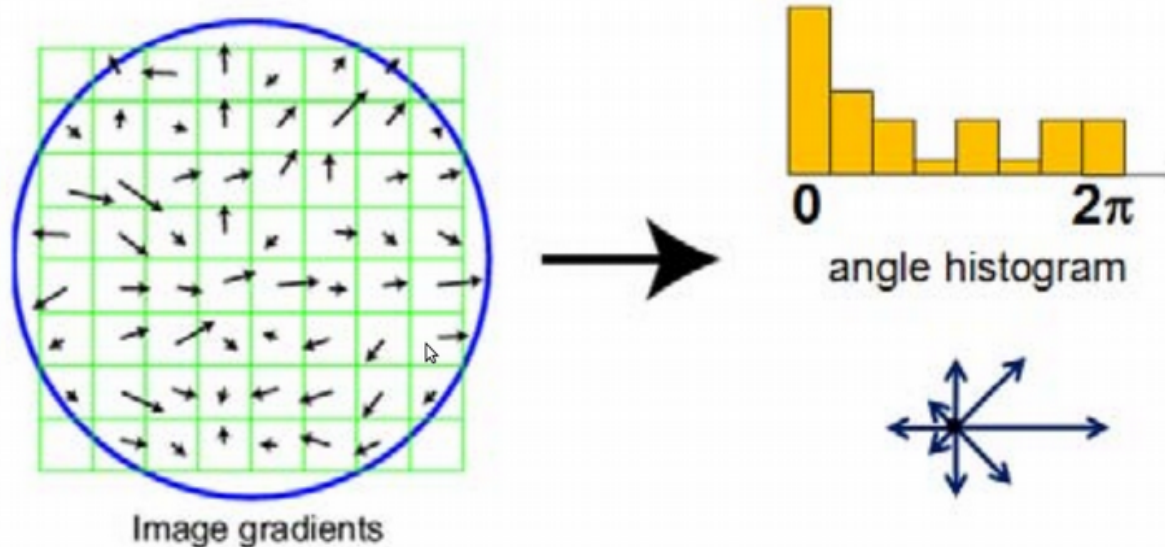
$$\frac{Tr(H)^2}{Det(H)} > 10.$$

Różnica suboktawowa filtrów Gaussa



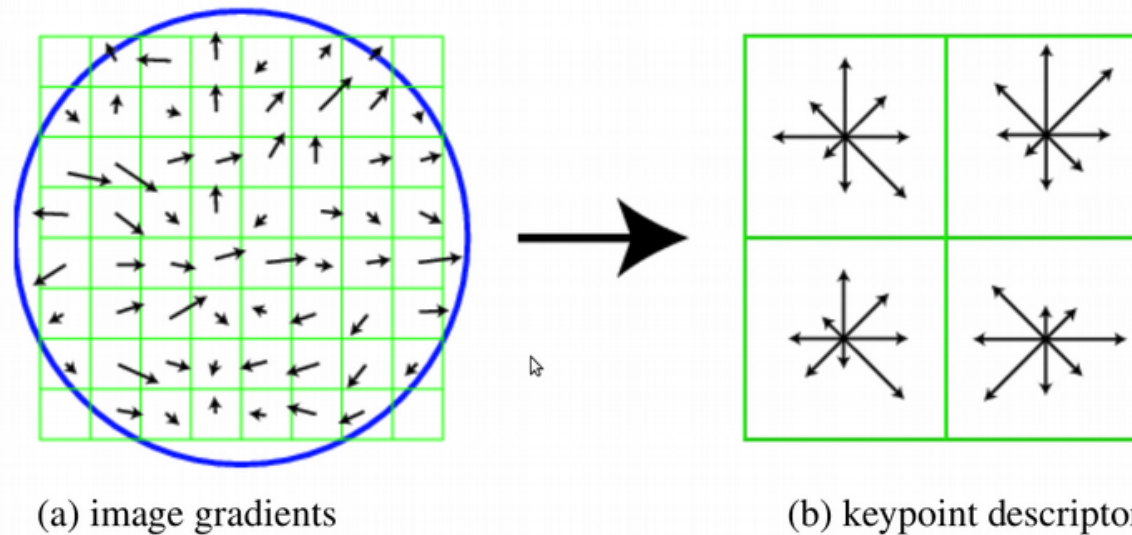
- (a) Sąsiadujące poziomy suboktawowej piramidy Gaussa są odejmowane, aby uzyskać Różnicę obrazów Gaussa,
- (b) Extrema (maksima i minima) w wynikowej objętości 3D wykryte przez porównanie piksela z jego 26 sąsiadami,
- Zachowujemy tylko te punkty, dla których Laplasjan jest ekstremalny.

Estymacja niezmienności rotacji i orientacji



- Oprócz radzenia sobie ze zmianami skali, większość algorytmów dopasowywania obrazu i rozpoznawania obiektów musi zajmować się obracaniem obrazu,
- Jednym ze sposobów rozwiązania tego problemu jest zaprojektowanie deskryptora, który jest niezmienny rotacyjnie,
- Oszacowanie dominującej orientacji można obliczyć, tworząc histogram wszystkich orientacji gradientu.

Deskryptory cech - Scale invariant feature transform (SIFT)

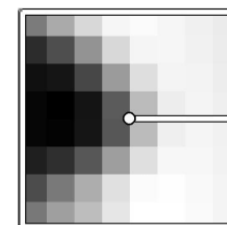
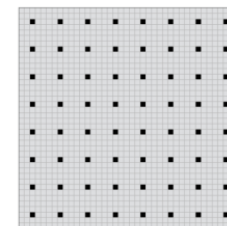


- **Transformacja cech niezmiennych w skali (SIFT)** - są tworzone przez obliczenie gradientu w każdym pikselu w oknie 16×16 wokół wykrytego punktu kluczowego, przy użyciu odpowiedniego poziomu piramidy Gaussa, na której został wykryty punkt kluczowy.

MOPS

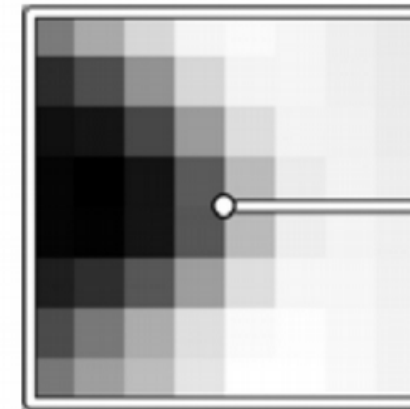
Nie zaimplementowano w OpenCV

Multi-Scale Oriented Patches (MOPS)



- Dane cechy: $\{x, y, s, \theta\}$
- Pobierz łate 40×40 , podpróbka co 5 piksel (filtrowanie niskich częstotliwości, pochłanianie błędy lokalizacji)
- Odejmij średnią, podziel przez odchylenie standardowe (usuwa bias i wzmacnienie)
- Transformacja falkowa Haara (projekcja niskiej częstotliwości)

Deskryptory cech - MOPS



Po wykryciu cech (punktów kluczowych) musimy je dopasować, aby określić, które cechy pochodzą z odpowiednich lokalizacji na różnych obrazach.

- **Bias and gain normalization (MOPS)** są tworzone przy użyciu próbkowania 8×8 odchylenia i uzyskują znormalizowane wartości intensywności, z odstępem między próbkami wynoszącym pięć pikseli w stosunku do skali wykrywania

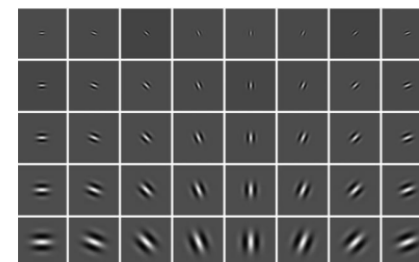
GIST

Nie zaimplementowano w OpenCV

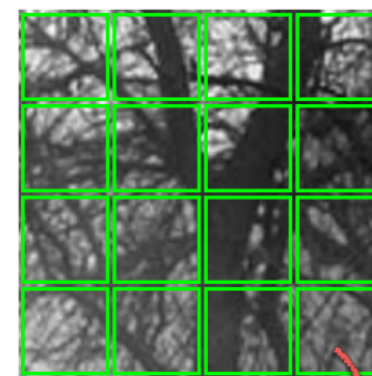
GIST

- Oblicz odpowiedzi filtrów (bank filtrów filtrów Gabora)
- Podziel plik obrazu na 4×4 komórki
- Oblicz średnie odpowiedzi filtru dla każdej komórki
- Rozmiar deskryptora wynosi $4 \times 4 \times N$, gdzie N jest rozmiarem banku filtrów
- GIST opisuje zgrubny przestrzenny rozkład gradientów obrazu

Filter bank



4 x 4 cell

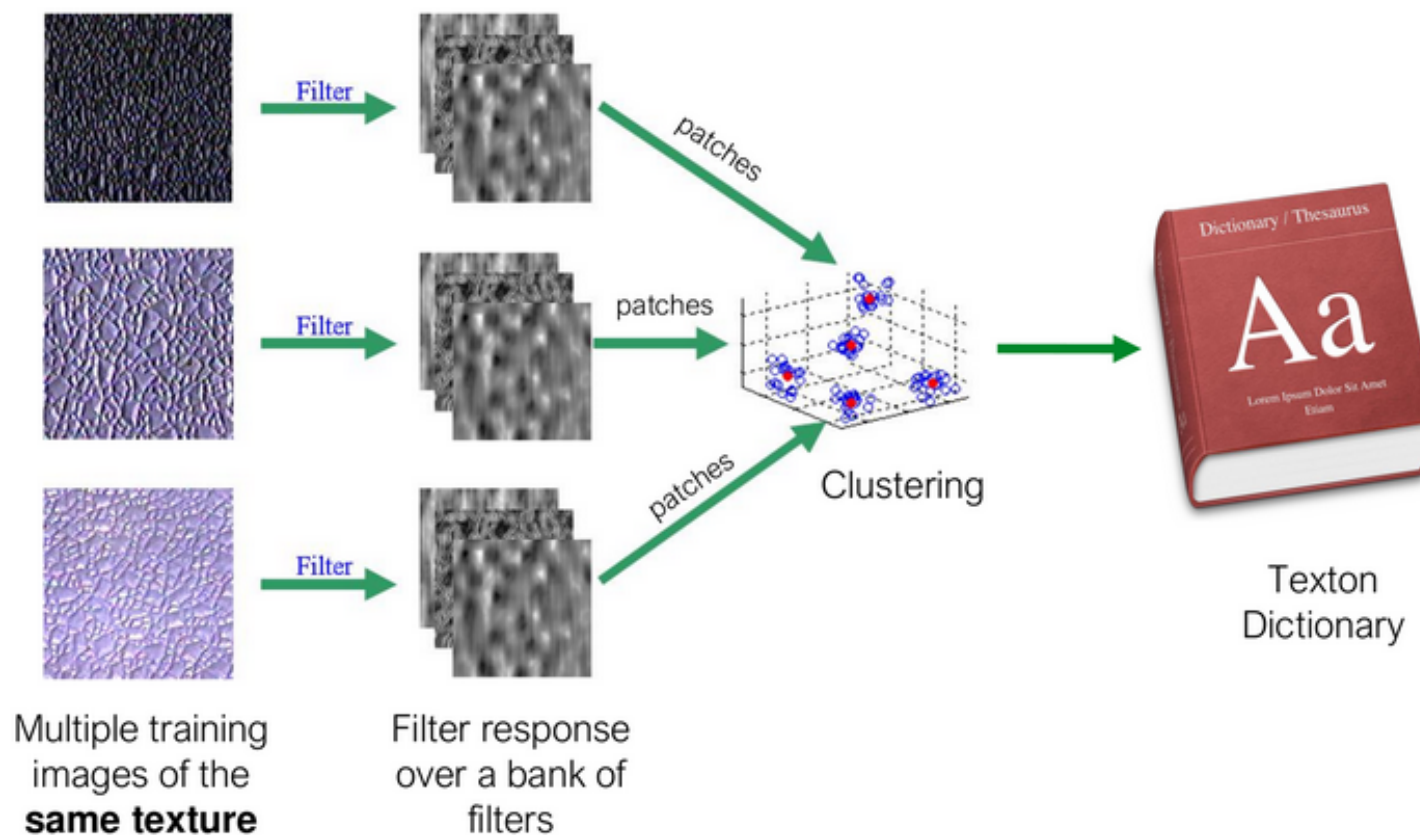


Deskryptor Histogramu Textons

Nie zaimplementowano w OpenCV

Textons

Teksturę charakteryzuje powtarzalność podstawowych elementów lub tekstur - uczenie się tekstur.

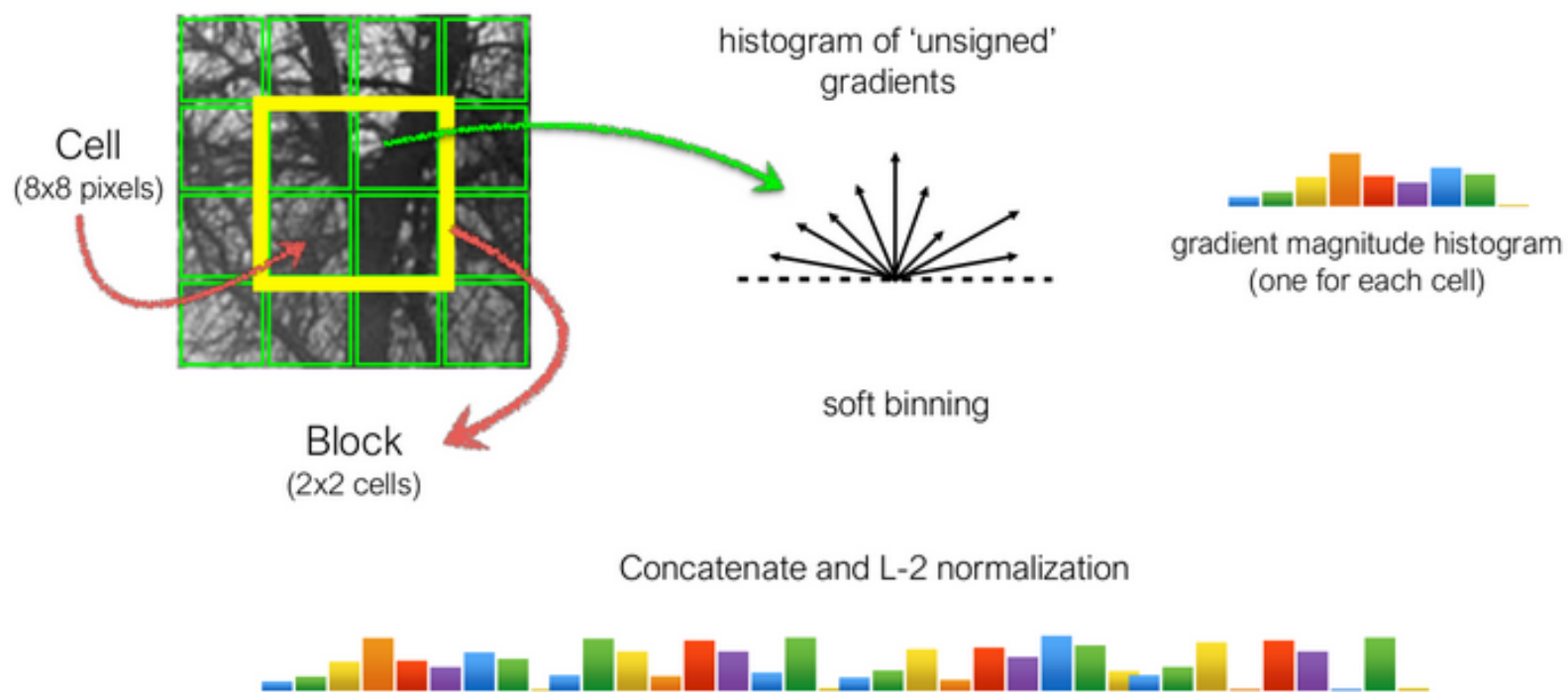


Deskryptor HOG

Nie zaimplementowano w OpenCV

HOG Histograms of Oriented Gradients

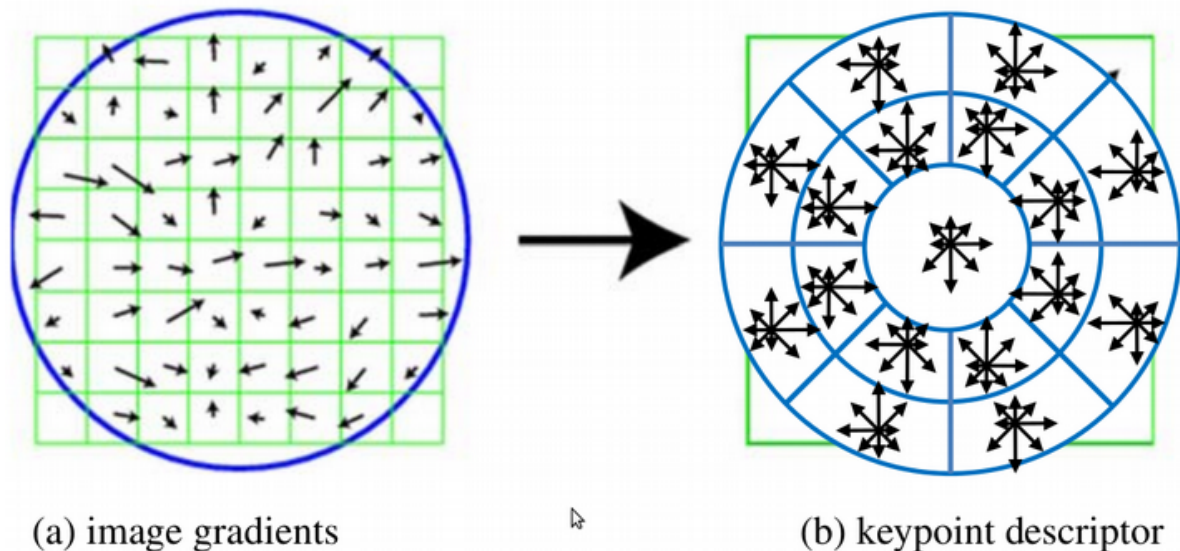
Pojedyncza skala, brak dominującej orientacji.



Deskryptor GLOH

Nie zaimplementowano w OpenCV

The Gradient Location-Orientation Histogram (GLOH)

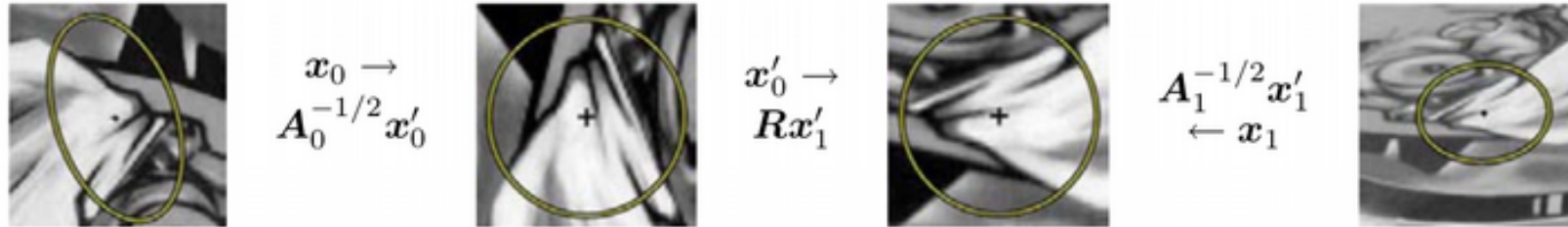


- **Deskryptor gradientu lokalizacji-orientacji (GLOH)** używa log-biegunowych pojemników zamiast kwadratowych do obliczania histogramów orientacji

Deskryptor niezmiennika afinicznego

Nie zaimplementowano w OpenCV

Niezmiennik afiniczny



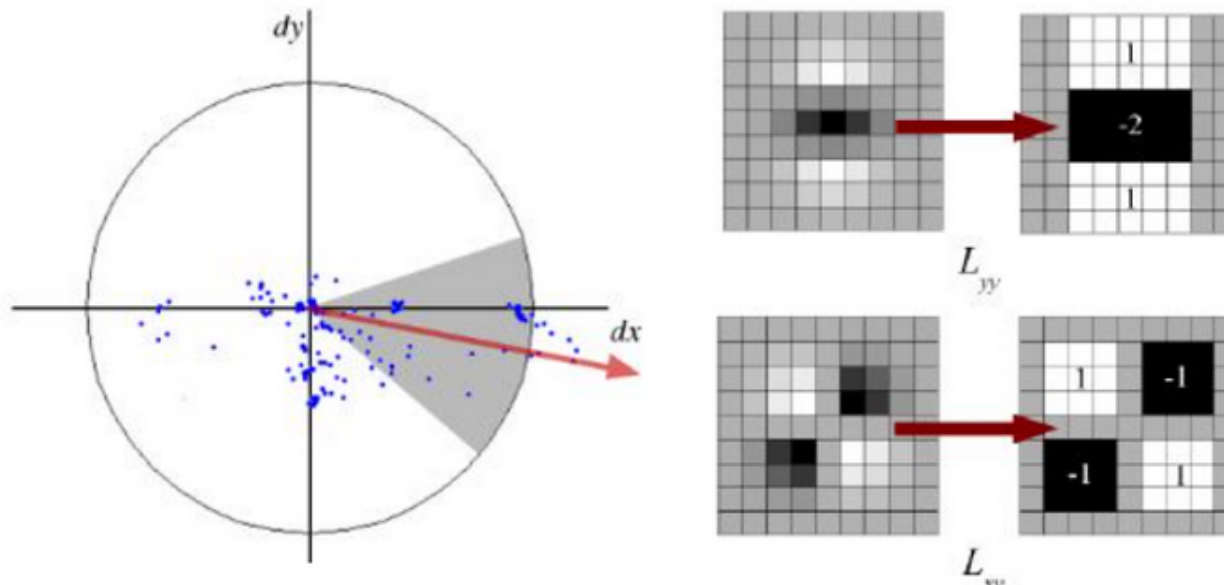
Normalizacja afiniczna z wykorzystaniem macierzy drugiego momentu. Po przekształceniu współrzędnych obrazu za pomocą macierzy $A_0^{-\frac{1}{2}}$ i $A_1^{\frac{1}{2}}$ i są powiązane z rotacją R ,

- W przypadku wielu zastosowań (takich jak szerokie dopasowanie stereo) preferowana jest pełna niezmiennność afiniczna.
- Detektory niezmiennicze afinicznie nie tylko reagują w stałych lokalizacjach po zmianach skali i orientacji, ale także konsekwentnie reagują na deformacje afiniczne.

Deskryptor SURF

Zaimplementowany w OpenCV (ograniczenia patentowe)

SURF (Speeded-Up Robust Features)



- SURF przybliża LoG za pomocą **Box Filter**.
- Konwolucję z filtrem pudełkowym można łatwo obliczyć za pomocą integralnych obrazów - można to zrobić równoległe dla różnych skal.
- SURF opierają się na wyznaczniku hesjanu zarówno dla skali, jak i położenia.
- Dla orientacji SURF używa odpowiedzi falkowych w kierunku poziomym i pionowym dla sąsiedztwa rozmiaru 6s (okno o kącie 60 stopni)

SURF - przykład



- Istotną poprawą jest użycie znaku laplasjński (śląd macierzy Hessa) dla bazowego punktu zainteresowania,
- Znak laplasjński odróżnia jasne plamki na ciemnym tle od sytuacji odwrotnej,
- Możesz zobaczyć, że SURF bardziej przypomina detektor kropelek,
- Jeśli można pominąć orientację, rozmiar deskryptora jest zmniejszany.

Deskryptor BRIEF

Zaimplementowany w OpenCV

BRIEF (Binary Robust Independent Elementary Features)

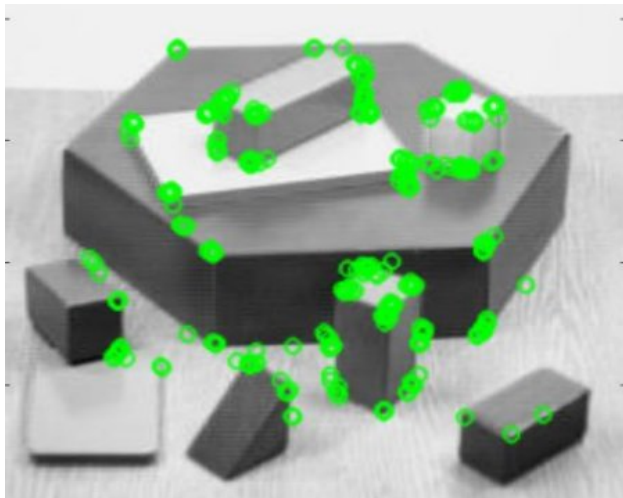
- Możemy skompresować rozmiar deskryptora za pomocą kilku metod, takich jak PCA, LDA lub LSH (Locality Sensitive Hashing), aby przekonwertować deskryptory SIFT w liczby zmiennoprzecinkowych na ciągi binarne.
- Ciągi binarne są używane do dopasowywania cech przy użyciu odległości Hamminga (wystarczy zastosować XOR) i licznik bitów,
- Najpierw musimy znaleźć deskryptory, a dopiero potem możemy zastosować haszowanie (nie rozwiązuje to naszego początkowego problemu z pamięcią).
- Jedną ważną kwestią jest to, że BRIEF jest deskryptorem funkcji, nie zapewnia żadnej metody znajdowania funkcji. Będziesz więc musiał użyć innych detektorów funkcji, takich jak SIFT, SURF itp.

- BRIEF to szybsza metoda obliczania i dopasowywania deskryptorów cech. Zapewnia również wysoki współczynnik rozpoznawania, chyba że występuje duża rotacja w płaszczyźnie.

Deskryptor ORB

Zaimplementowany w OpenCV

ORB (Oriented FAST and Rotated BRIEF)



- ORB jest w zasadzie połączeniem detektora punktu kluczowego FAST i deskryptora BRIEF z wieloma modyfikacjami w celu zwiększenia wydajności,
- FAST służy do znajdowania punktów kluczowych, a następnie zastosuj miarę narożną Harrisa, aby znaleźć wśród nich N górnych punktów,
- Używa również piramidy do tworzenia funkcji wieloskalowych,
- Oblicza ważoną intensywność z narożnikiem położonym w środku - kierunek wektora od tego punktu narożnego do środka ciężkości podaje orientację.

Cechy z uczenia maszynowego

Ten slajd powinien być punktem wyjścia dla konwolucyjnych sieci neuronowych i uczenia głębokiego.

Feature matching

Dopasowanie cech

Po wyodrębnieniu cech i ich deskryptorów z dwóch lub więcej obrazów następnym krokiem jest ustalenie wstępnych dopasowań cech między tymi obrazami.

- Pierwszym jest wybranie **strategii dopasowania**, która określa, które korespondencje są przekazywane do następnego etapu w celu dalszego przetwarzania.
- Drugim jest wymyślenie wydajnych **struktur danych i algorytmów**, aby przeprowadzić to dopasowanie tak szybko, jak to możliwe.

Kryterium dopasowania

- *(Ważona) zsumowana różnica kwadratowa,*

$$E_{WSSD}(u) = \sum_i w(x_i)[I_1(x_i + u) - I_0(x_i)]^2,$$

gdzie I_0 i I_1 to dwa porównywane obrazy, $u = (u, v)$ to wektor przemieszczenia, $w(x)$ to przestrzennie zmieniająca się funkcja ważenia (lub okna), a sumowanie i znajduje się nad wszystkimi pikselami w łańcu.

- *Autokorelacji funkcji lub powierzchni*

$$E_{AC}(\Delta u) = \sum_i w(x_i)[I_0(x_i + \Delta u) - I_0(x_i)]^2$$

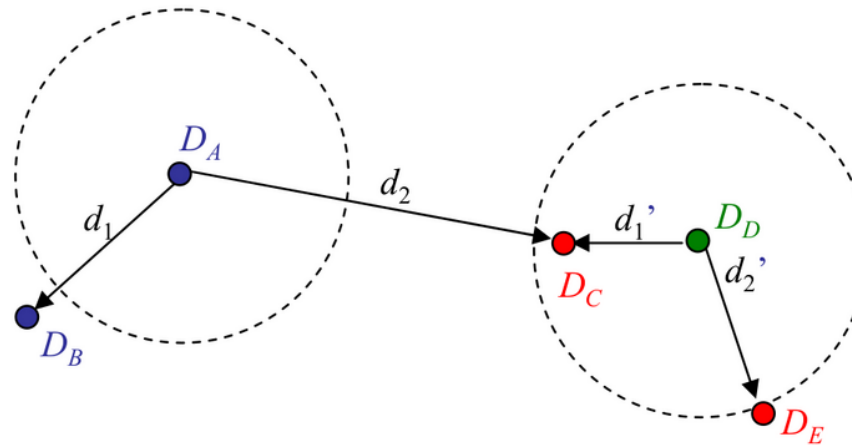
Dlatego możemy obliczyć, jak stabilna jest ta metryka w odniesieniu do małych odchyłeń pozycji Δu , porównując poprawkę obrazu ze sobą,

Strategia dopasowania - progowanie

Najprostszą strategią dopasowania jest ustawienie progu (maksymalnej odległości) i zwrócenie wszystkich dopasowań z innych obrazów w tym progu.

- Ustawienie zbyt wysokiego progu skutkuje zbyt dużą liczbą fałszywych alarmów, tj. Zwracaniem nieprawidłowych dopasowań,
- Ustawienie zbyt niskiego progu skutkuje zbyt dużą liczbą fałszywych wyników negatywnych, tj. zbyt dużą liczbą pominięte prawidłowych porównań.

Dopasowywanie odległości do najbliższego sąsiada i najbliższego sąsiada

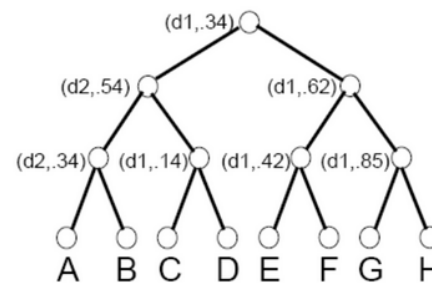
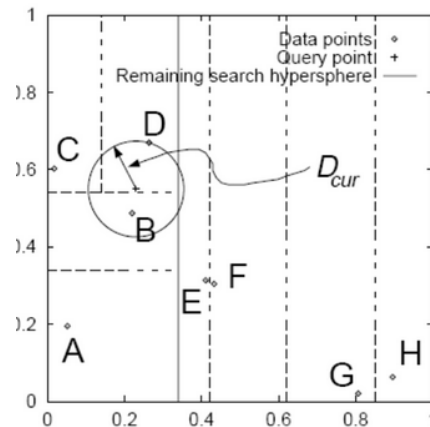


- Problem ze stosowaniem ustalonego progu polega na tym, że trudno go ustawić,
- Przy ustalonym progu odległości (przerywane kółka) deskryptor D_A nie pasuje do D_B , a D_D niepoprawnie pasuje do D_C i D_E .
- Możemy zdefiniować *Nearest Neighbor Distance Ratio* jako:

$$NNDR = \frac{d_1}{d_2} = \frac{|D_A - D_B|}{|D_A - D_C|}$$

gdzie d_1 i d_2 to najbliższa i druga najbliższa odległość sąsiada, D_A to deskryptor celu, a D_B i D_C to jego najbliżsi sąsiedzi.

Skuteczne dopasowanie



- Lepszym podejściem jest opracowanie struktury indeksowania, takiej jak wielowymiarowe drzewo wyszukiwania lub tabela haszowania,
- Takie struktury indeksujące można zbudować niezależnie dla każdego obrazu,
- Inną klasą struktur indeksujących są wielowymiarowe drzewa wyszukiwania, takie jak drzewa k-d (które dzielą wielowymiarową przestrzeń cech wzdłuż naprzemiennych hiperpłaszczyzn wyrównanych do osi,

Dopasowanie strategii i wskaźników błędów



Określenie, które dopasowania cech są uzasadnione do dalszego przetwarzania, zależy od kontekstu, w którym przeprowadzane jest dopasowywanie.

- Są one dopasowywane do zagraconej sceny pośrodku za pomocą funkcji SIFT, pokazanych jako małe kwadraty na prawym obrazie.
- Wypaczenie afiniczne każdego rozpoznanego obrazu bazy danych na scenie jest pokazane jako większy równoległobok na prawym obrazie.

Miary klasyfikacyjne

- TP: true positives, i.e., liczba poprawnych dopasowań,
- FN: false negatives, porównania, które nie zostały poprawnie wykryte,
- FP: false positives, proponowane dopasowania, które są nieprawidłowe,
- TN: true negatives, niedopasowania, które zostały poprawnie odrzucone
- true positive rate (TPR): $TPR = \frac{TP}{TP+FN} = \frac{TP}{P}$
- false positive rate (FPR): $FPR = \frac{FP}{FP+TN} = \frac{FP}{N}$
- positive predictive value (PPV): $PPV = \frac{TP}{TP+FP} = \frac{TP}{P'}$
- accuracy (ACC): $ACC = \frac{TP+TN}{P+N}$.

Funkcje dopasowywania w OpenCV

- **Basics of Brute-Force Matcher** - Pobiera deskryptor jednej cechy z pierwszego zestawu i jest dopasowywany do wszystkich innych cech w drugim zestawie przy użyciu obliczenia odległości.
- **FLANN based Matcher** - Zawiera zbiór algorytmów zoptymalizowanych pod kątem szybkiego wyszukiwania najbliższego sąsiada w dużych zbiorach danych i dla cech o dużych wymiarach.

Śledzenie cech - formułowanie problemu

Aby znaleźć zestaw prawdopodobnych lokalizacji elementów na pierwszym obrazie, a następnie wyszukać odpowiadające im lokalizacje na kolejnych obrazach.

- Proces wybierania dobrych cech do śledzenia jest ściśle powiązany z wybieraniem dobrych cech do bardziej ogólnych zastosowań rozpoznawania.
- Regiony zawierające wysokie gradienty w obu kierunkach (które mają wysokie wartości własne w macierzy autokorelacji) zapewniają stabilne lokalizacje do znalezienia odpowiedników.

Zadania na laboratoria

- **Deskryptor punktu zainteresowania** - znajdź punkty kluczowe na obrazach za pomocą deskryptorów opencv (np. ORB lub SHIFT)
- **Feature matcher:** Po wyodrębnieniu cech dopasuj je według ich deskryptorów - możesz użyć funkcji opencv.
- **Feature tracker:** Zamiast znajdować punkty cech niezależnie na wielu obrazach i dopasowywać je, znajdź cechy na pierwszym obrazie filmu lub sekwencji obrazów, a następnie ponownie zlokalizuj odpowiadające im punkty w następnych klatkach.

Zastosuj swój tracker cech do zadania śledzenia twarzy lub do filmu z piłą łańcuchową.