Lokalne deskryptory cech obrazu
Temat 7

Piotr Kawa

2022

Struktury obrazu

Z czego zbudowane są obrazy:

- punkty,
- krawędzie,
- rogi,
- plamy.

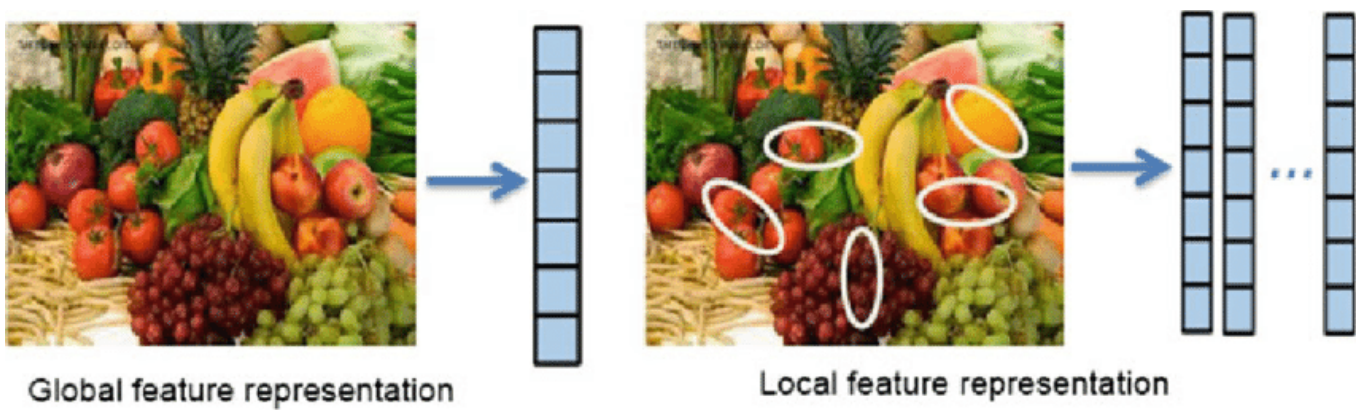


Rysunek: Struktury obrazu — detektor Harrisa, źródło: [1]

Lokalne deskryptory cech obrazu

Lokalne deskryptory cech obrazu (ang. local feature descriptors):

- powtarzalne: niezależne od rotacji, skali, zmienności oświetlenia (ta sama scena — te same cechy),
- charakterystyczne: na ich podstawie będziemy rozróżniać obrazy/przedmioty,
- kompaktowe: dużo informacji, mała reprezentacja,
- wydajność: muszą być szybkie.



Rysunek: Cechy globalne vs. cechy lokalne. [2]

LFD — zastosowania

- łączenie obrazów (ang. image stitching),
- detekcja obiektów,
- klasyfikacja obiektów,
- śledzenie obiektów,
- wyszukiwanie obrazów (ang. image retrieval),
- ...

LFD — NFLT

Choose a Descriptor Method

Descriptor	Binary	Function and Method	Invariance		Typical Use	
			Scale	Rotation	Finding Point Correspondences	Classification
HOG	No	`extractHOGFeatures(I, ...)`	No	No	No	Yes
LBP	No	`extractLBPFeatures(I, ...)`	No	Yes	No	Yes
SIFT	No	`extractFeatures(I,points,'Method','SIFT')`	Yes	Yes	Yes	Yes
SURF	No	`extractFeatures(I,points,'Method','SURF')`	Yes	Yes	Yes	Yes
KAZE	No	`extractFeatures(I,points,'Method','KAZE')`	Yes	Yes	Yes	Yes
FREAK	Yes	`extractFeatures(I,points,'Method','FREAK')`	Yes	Yes	Yes	No
BRISK	Yes	`extractFeatures(I,points,'Method','BRISK')`	Yes	Yes	Yes	No
ORB	Yes	`extractFeatures(I,points,'Method','ORB')`	No	Yes	Yes	No
• Block • Simple pixel neighborhood around a keypoint	No	`extractFeatures(I,points,'Method','Block')`	No	No	Yes	Yes

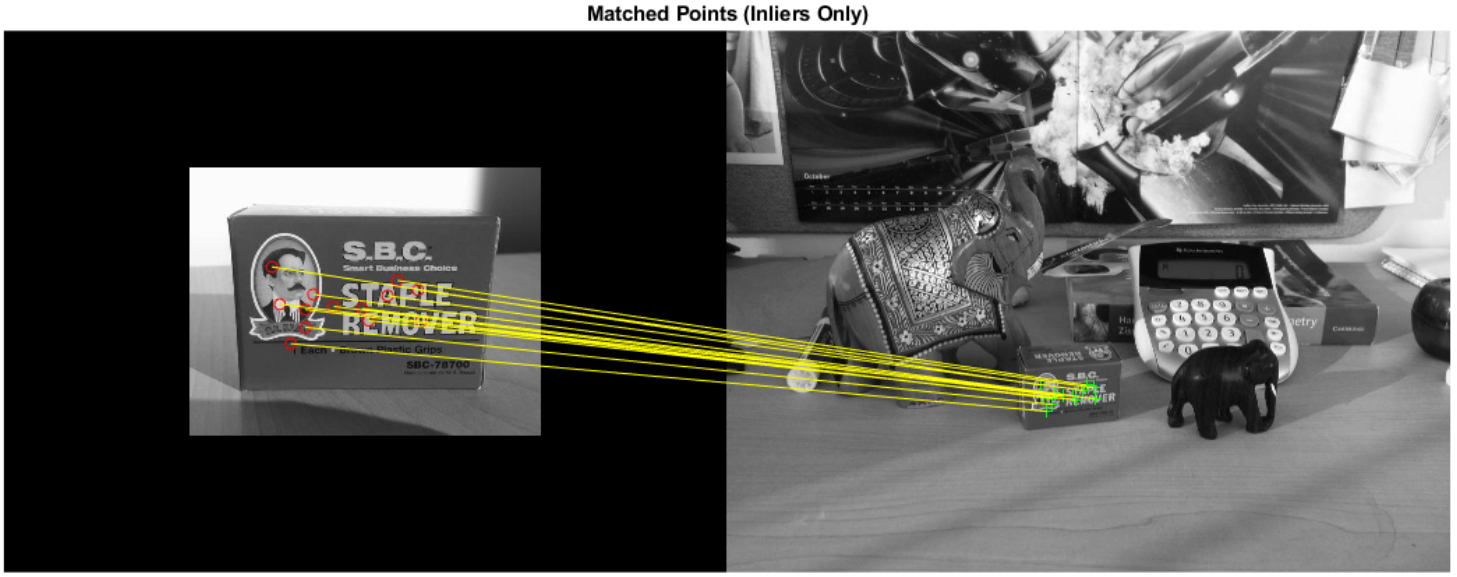
Rysunek: "No free lunch theorem"' — nie ma uniwersalnie najlepszego algorytmu.

LFD — Image stitching



Rysunek: Przykład łączenia obrazów [3]

LFD — Detekcja obiektów



Rysunek: Przykład detekcji obiektów [4]

LFD — przykłady algorytmów

- HOG (1986/2005),
- Harris (1988),
- Shi & Tomasi (1994),
- SIFT (2004),
- SURF (2006),
- ORB (2011),
- BRISK (2011).

SIFT

Scale Invariant Feature Transform (SIFT):

- odporny na skalowanie i rotację.

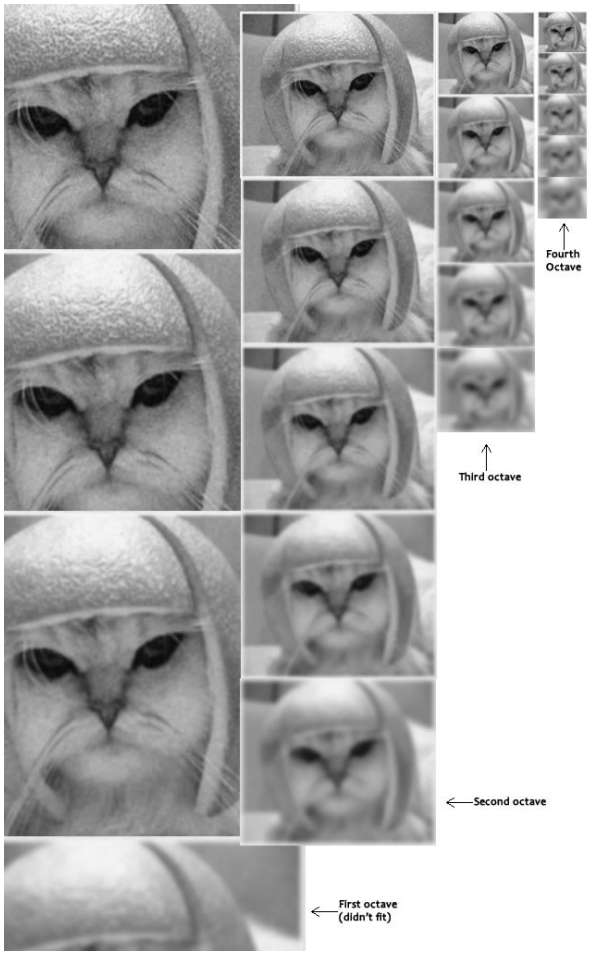
SIFT

Algorytm:

1. stworzenie N oktaw — każda 2x mniejsza od poprzedniej,
2. dla każdej oktawy:
   1. stworzenie L zaszumionych obrazów (Gaussian Blur),
   2. obliczenie różnicy gaussianów (ang. Difference of Gaussians),
   3. wyznaczenie punktów charakterystycznych

SIFT 1

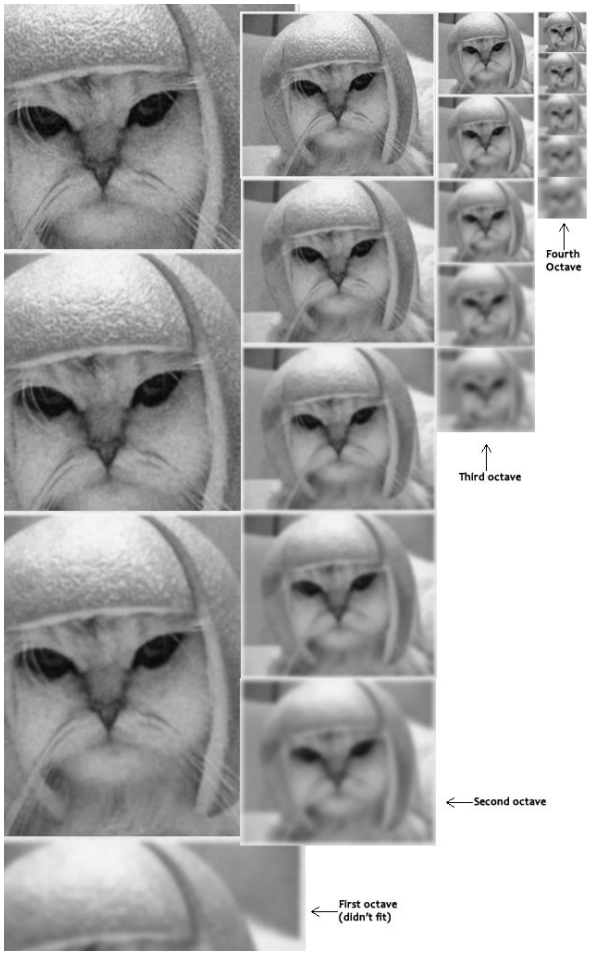
1. **stworzenie N (domyślnie: $\log_2(min(W, H))$) oktaw — każda 2x mniejsza od poprzedniej**,
2. dla każdej oktawy:
   1. stworzenie L zaszumionych obrazów (Gaussian Blur),
   2. obliczenie różnicy gaussianów (ang. Difference of Gaussians),
   3. wyznaczenie punktów charakterystycznych



Rysunek: N oktaw [5]

SIFT 2

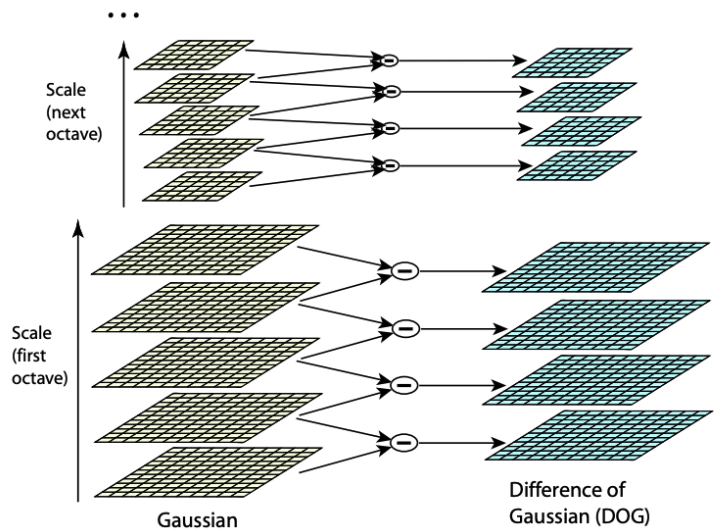
1. stworzenie N oktaw — każda 2x mniejsza od poprzedniej,
2. dla każdej oktawy:
    1. **stworzenie L (domyślnie 3) zaszumionych obrazów (Gaussian Blur)**,
    2. obliczenie różnicy gaussianów (ang. Difference of Gaussians),
    3. wyznaczenie punktów charakterystycznych



Rysunek: L zaszumionych obrazów [5]

SIFT — różnica Gaussianów

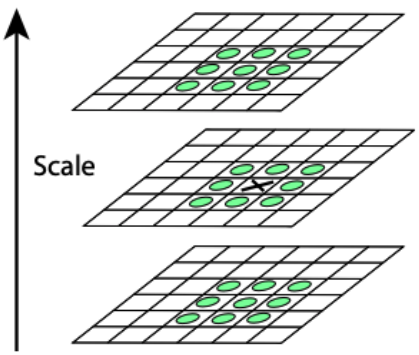
1. stworzenie N oktaw — każda 2x mniejsza od poprzedniej,
2. dla każdej oktawy:
   1. stworzenie L zaszumionych obrazów (Gaussian Blur),
   2. **obliczenie różnicy gaussianów (ang. Difference of Gaussians)**,
   3. wyznaczenie punktów charakterystycznych



Rysunek: Różnica Gaussianów [6]

SIFT 4

1. stworzenie N oktaw — każda 2x mniejsza od poprzedniej,
2. dla każdej oktawy:
    1. stworzenie L zaszumionych obrazów (Gaussian Blur),
    2. obliczenie różnicy gaussianów (ang. Difference of Gaussians),
    3. **wyznaczenie punktów charakterystycznych**.



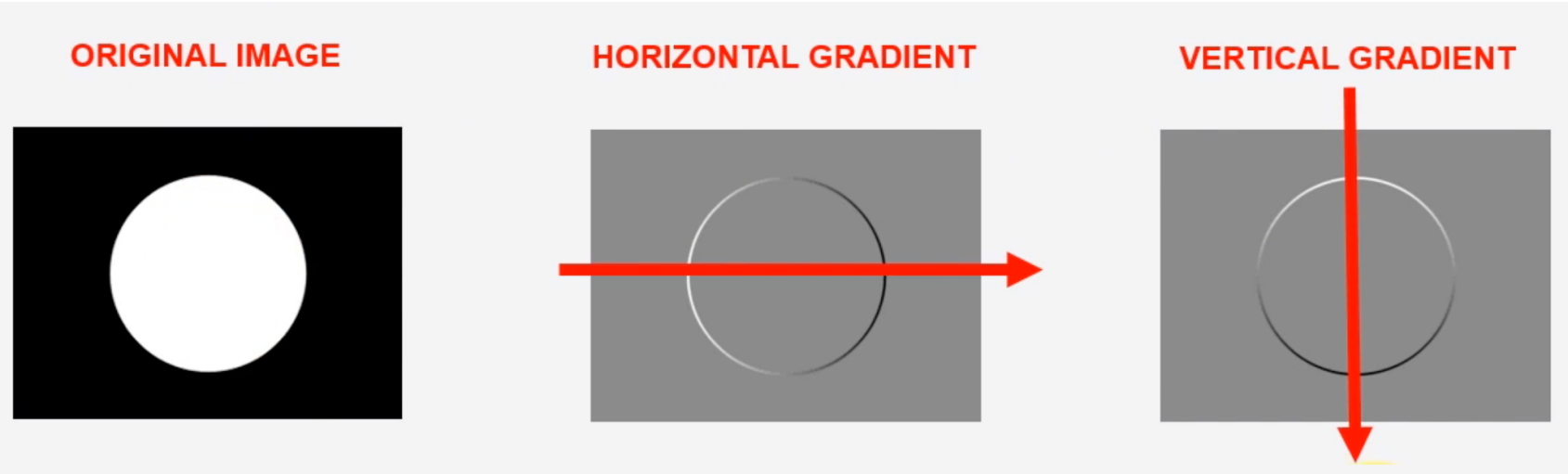
Rysunek: Porównanie piksela z 26 sąsiadami (8+9+9) i jeśli jest min/max — jest punktem charakterystycznym [5]

Histogram Zorientowanych Gradientów

Histogram of Oriented Gradients (HOG):

- używany głównie do detekcji obiektów — pierwotnie detekcji przechodniów.

Gradient w obrazie



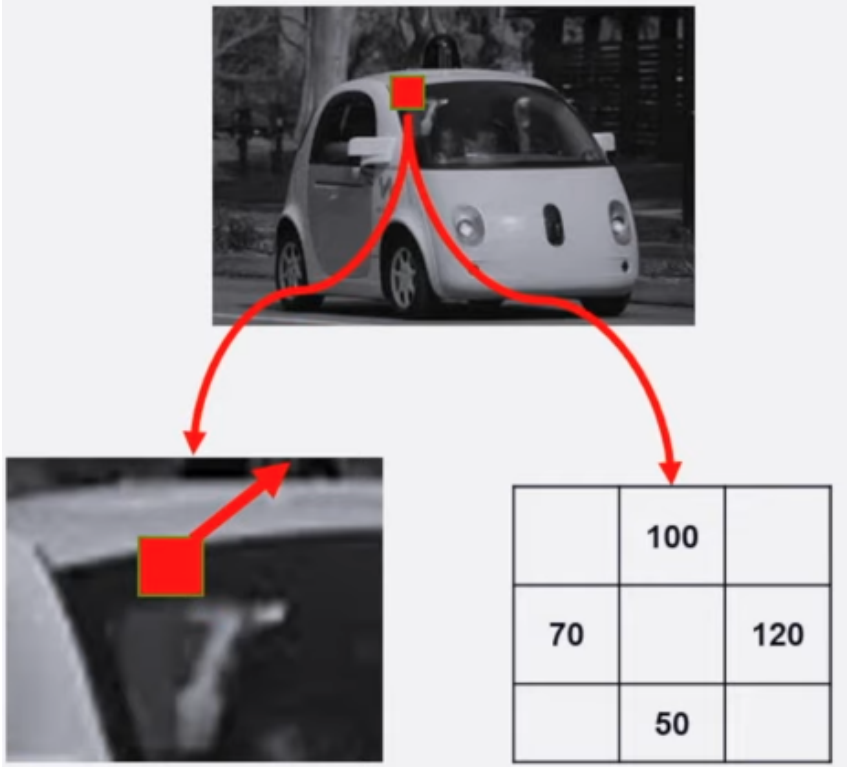
Rysunek: Gradient jako miara różnic między wartościami sąsiadujących pikseli, źródło: [5]

HOG — algorytm

1. Reskalowanie obrazu do rozmiaru 128x64
2. Dla każdego piksela:
   1. Oblicz gradient w pionie i poziomie
   2. Wylicz magnitude - $\sqrt{grad_x^2 + grad_y^2}$
   3. Wylicz kąt = $\mathrm{tg}^{-1} \frac{grad_x}{grad_y}$
3. Dla każdej łaty 8x8 stwórz histogram - podziel kąty na np. 9 wartości (180°/ 20°) i przyporządkuj magnitude do odpowiedniego kubełka.

HOG — algorytm 1

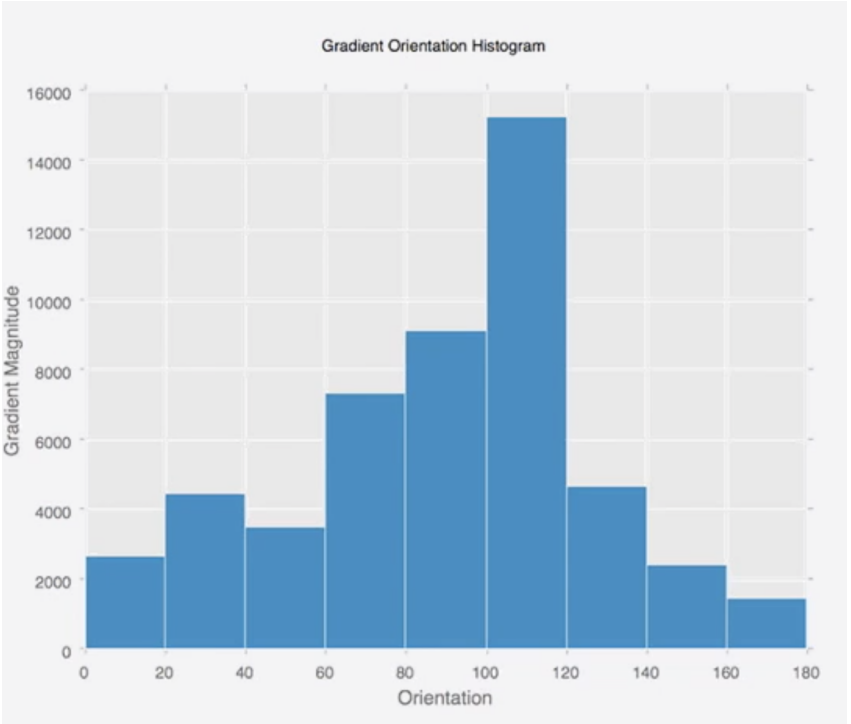
1. Reskalowanie obrazu do rozmiaru 128x64
2. **Dla każdego piksela:**
   1. Oblicz gradient w pionie i poziomie
   2. Wylicz magnitude - $\sqrt{grad_x^2 + grad_y^2}$
   3. Wylicz kąt $= \mathrm{tg}^{-1} \frac{grad_x}{grad_y}$
3. Dla każdej łaty 8x8 stwórz histogram - podziel kąty na np. 9 wartości (180°/ 20°) i przyporządkuj magnitude do odpowiedniego kubełka.



Rysunek: Liczenie gradientu [5]

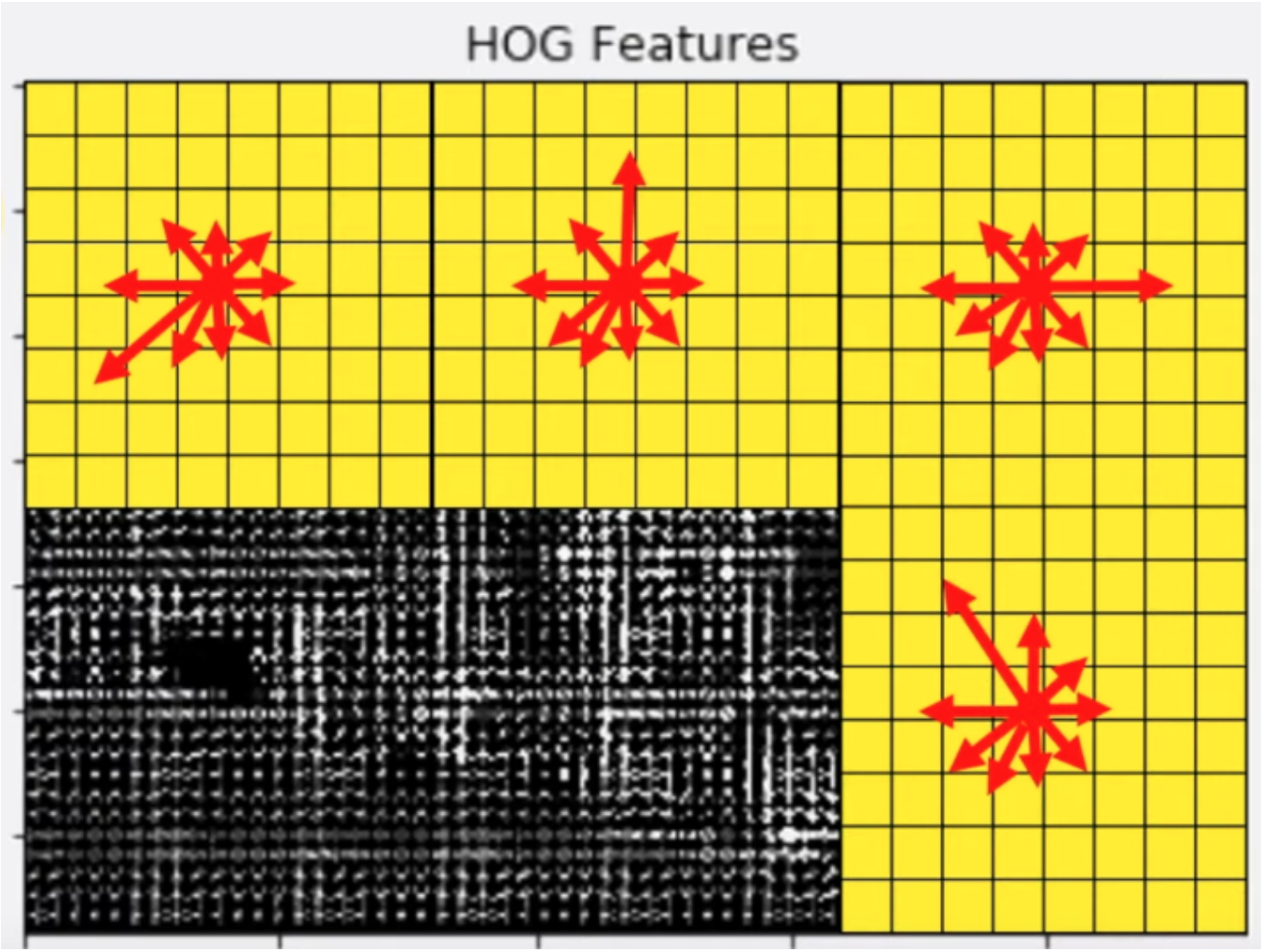
HOG — algorytm 2

1. Reskalowanie obrazu do rozmiaru 128x64
2. Dla każdego piksela:
   1. Oblicz gradient w pionie i poziomie
   2. Wylicz magnitude - $\sqrt{grad_x^2 + grad_y^2}$
   3. Wylicz kąt = $\mathrm{tg}^{-1} \frac{grad_x}{grad_y}$
3. **Dla każdej łaty 8x8 stwórz histogram** — podziel kąty na np. 9 wartości (180°/ 20°) i przyporządkuj magnitude do odpowiedniego kubełka.



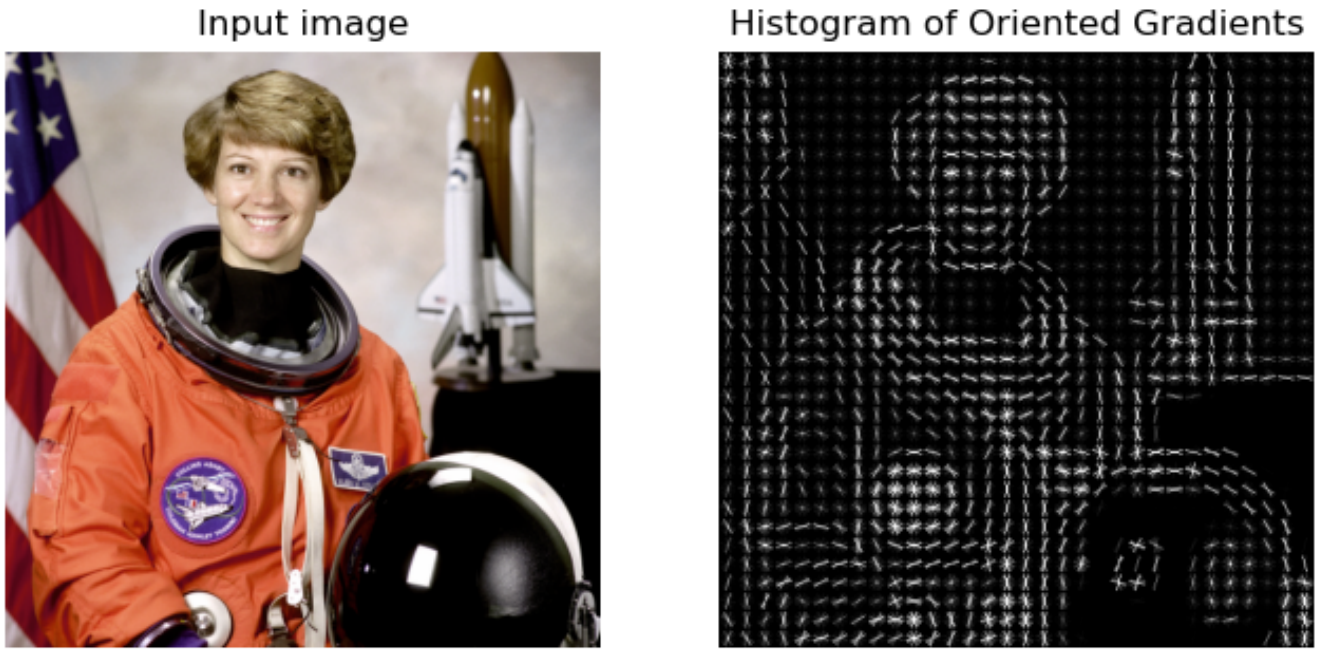
Rysunek: Stworzenie histogramu [5]

HOG — reprezentacja histogramu



Rysunek: Histogram zorientowanych gradientów — reprezentacja histogramu [5]

HOG — finalna reprezentacja



Rysunek: Histogram zorientowanych gradientów — finalna reprezentacja [7]

Przykłady użycia LFD

1. Łączenie obrazów — https://www.mathworks.com/help/vision/ug/feature-based-panoramic-image-stitching.html
2. Detekcja obiektów — https://www.mathworks.com/help/vision/ug/object-detection-in-a-cluttered-scene-using-point-feature-ma
html

Źródła I



Wikimedia writing desk with harris detector.

https://upload.wikimedia.org/wikipedia/commons/8/8d/Writing_Desk_with_Harris_Detector.png.
Accessed: 20-05-2022.



Mahmoud Hassaballah, Abdelmgeid Ali, and Hammam Alshazly.

Image Features Detection, Description and Matching, volume 630, pages 11–45.
02 2016.



Wikimedia image stitching example.

https://upload.wikimedia.org/wikipedia/commons/c/c9/Riola1.jpg.
Accessed: 20-05-2022.



MathWorks object detection example.

https://www.mathworks.com/help/examples/vision/win64/FeatureBasedObjectDetectionExample_06.png.
Accessed: 20-05-2022.



YT computer vision with opencv: Hog feature extraction.

https://www.youtube.com/watch?v=4ESLTAd3IOM.
Accessed: 20-05-2022.



David G Lowe.

Distinctive image features from scale-invariant keypoints.
International journal of computer vision, 60(2):91–110, 2004.



scikit-image hog - example.

https://scikit-image.org/docs/stable/auto_examples/features_detection/plot_hog.html.
Accessed: 20-05-2022.