



# **Monitorowanie i Diagnostyka w Systemach Sterowania**

**Wydział Elektrotechniki i Automatyki  
Katedra Elektrotechniki, Systemów Sterowania i Informatyki  
Dr inż. Michał Grochowski**

# Monitorowanie i Diagnostyka w Systemach Sterowania

na studiach II stopnia specjalności: Systemy Sterowania i Podejmowania Decyzji

---

---

## Klasteryzacja danych

*na podstawie:*

Leszek Rutkowski. Metody i techniki sztucznej inteligencji. Wydawnictwo Naukowe PWN,  
Warszawa 2005;

Matlab Statistics Toolbox User's Guide. The MathWorks, Inc.

Opracował: dr inż. Michał Grochowski

kiss.pg.mg@gmail.com

michal.grochowski@pg.gda.pl

## Motywacja

---

- rozwój technologii „cyfrowej” (informacje z czujników pomiarowych, zdjęcia, bazy danych, digitalizacja zbiorów „bibliotecznych”, badania genetyczne, przeszukiwanie internetu, obróbka dźwięku i wideo etc...) kreuje coraz większe zbiory danych;
- szacuje się że w 2011 roku przetwarzanych było ok.  $3 \cdot 10^{20}$  B danych;
- Youtube - ponad 800 mln miesięcznych użytkowników, którzy „uploadują” ponad godzinne materiały każdej sekundy (2013r.), co minutę przybywa ok 300g filmów (2015r);
- 40 000 zapytań w Google na sekundę (2015r)
- Dzienna ilość użytkowników Twittera 100 mln (2015);
- „like” na Facebooku 3 mld dziennie (2013r);

# Motywacja

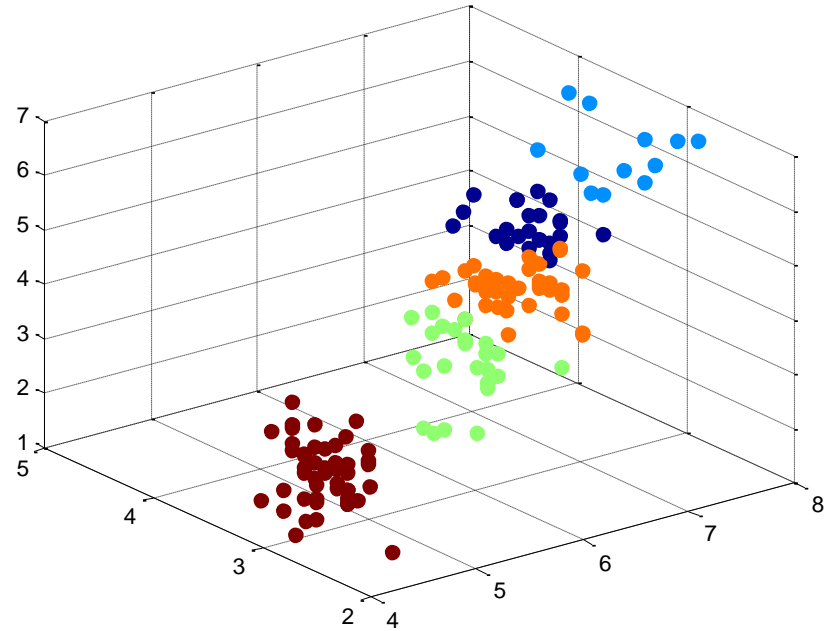
---

- 10 miliardów urządzeń jest podłączonych do internetu. Szacuje się że do 2020 będzie ich 50 mld i 100 trylionów do roku 2040 (“Internet of Things - IoT”);
- National Oceanic and Atmospheric Administration (NOAA) aktualnie zbiera ponad 19 TB danych każdego roku (2013r).
- Zdajemy sobie sprawę z tego że to dopiero początek .... 😊

# Wprowadzenie

Celem klasteryzacji powinno być odnalezienie naturalnego grupowania wzorców, cech, obrazów, itp... celem ich późniejszej „łatwiejszej obróbki”.

Klasteryzacja (grupowanie danych) jest jedną z metod nienadzorowanej analizy danych (w odróżnieniu od klasyfikacji).



Jej celem jest podział danych na klastry (grupy), tak aby każdy z nich zawierał możliwie „podobne do siebie” elementy (homogeniczność w grupach), a jednocześnie, poszczególne klastry powinny być jak najbardziej zróżnicowane pomiędzy sobą (heterogeniczność pomiędzy grupami).

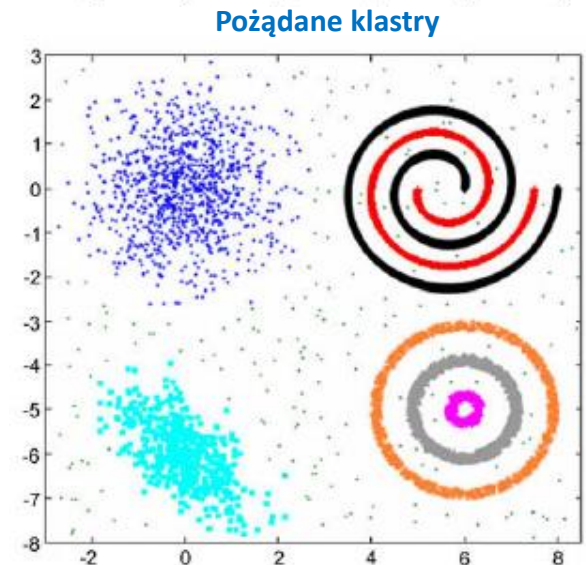
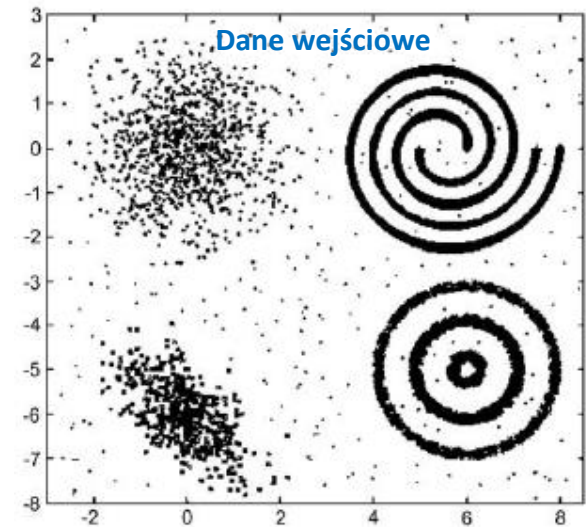
# Wprowadzenie

Podobieństwo grup danych może być opisane na różne sposoby, w zależności od przyjętej miary.

Grupy danych mogą być reprezentowane na różne sposoby, najczęściej poprzez centralny punkt w przestrzeni danych.

W zależności od sposobu reprezentacji i przyjętej miary grupy mogą wyglądać zupełnie inaczej.

W procesie grupowania danych nie posiadamy danych wzorcowych tak więc proces ten jest uczeniem „bez nauczyciela”.



Jain, A.K. Data clustering: 50 years beyond K-means. Pattern Recognition Lett. (2009),

# Wprowadzenie

---

W zagadnieniach klasyfikacji często operujemy pojęciami:

- **obiekty** np. wina, kwiaty, pacjenci, samochody, klienci banków, użytkownicy portali społecznościowych, teksty, obrazy, dźwięki, geny ...
- **cechy** np. kolor, woń, smak; długość, kolor, długość płatków kwiatowych; wyniki badań medycznych, parametry samochodów; upodobania ludzkie, rodzaje wykonywanych operacji finansowych; wzorce graficzne; słowa kluczowe ...

# Wprowadzenie

Przyjmujemy, że dane podlegające grupowaniu zapisywane są w postaci macierzy  $X$  zawierającej  $M$  obiektów, składających się z numerycznych wartości opisujących obiekty (każda po  $N$  cech).

Zakładając że grupy danych reprezentowane są przez ich środki, celem algorytmu grupowania danych jest uzyskanie  $c$  wektorów  $v_i = [v_{i1}, \dots, v_{iN}]$ ,  $i=1, \dots, c$  będących reprezentantami poszczególnych grup w przestrzeni danych.

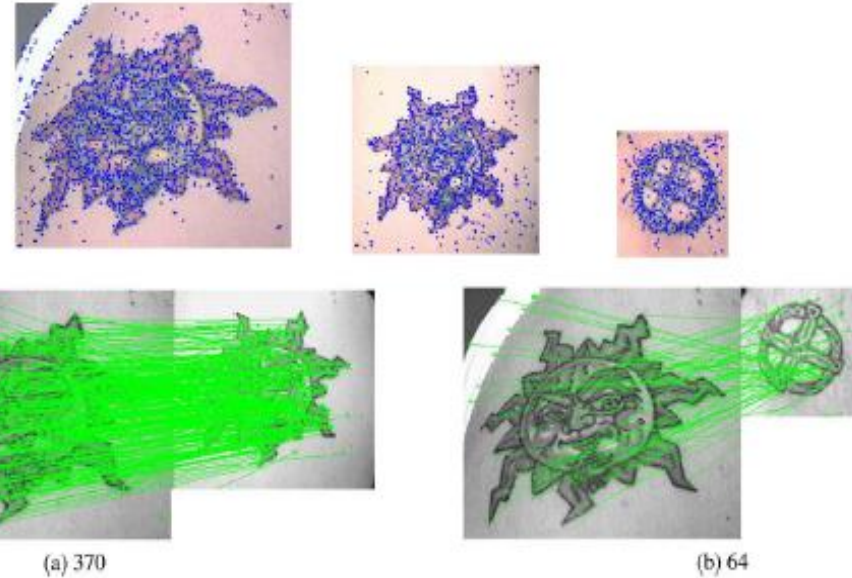
W problemie grupowania 100 obiektów do 5ciu różnych grup, mamy możliwość podziału tych danych na **6.57 e<sup>67</sup>** różnych sposobów !!!!!!!

Celem jest zatem znalezienie efektywnego algorytmu grupowania danych pozwalającego na znalezienie optymalnego (suboptymalnego) podziału tych danych bez konieczności analizy wszystkich możliwych kombinacji.



# Wprowadzenie

- Przy założeniu 1 mln obrazów i czasie porównywania pary obrazów ok. 10 ms., otrzymamy czas realizacji zapytania ok. 30 godzin;
- jednak aplikacje porównujące (wyszukujące) tekst działają znacznie szybciej: 10 miliardów dokumentów tekstowych jest przeszukiwana przez Google w ok. 0.1 sekundy !!!



(a) a pair of similar images has 370 matching key points;  
(b) a pair of different images has 64 matching key points.  
The green lines show the matching key-points between the images (Lee et al., 2008).

Jain, A.K. Data clustering: 50 years beyond K-means. Pattern Recognition Lett. (2009),

# Metody klasteryzacji

- grupujące (ostre, rozmyte, posybilistyczne);
- hierarchiczne

# Ostre i rozmyte podziały danych

## Ostre grupowanie danych

Celem ostrego grupowania danych jest podział danych na  $c$  grup  $A_i$ , tak aby:

$$\bigcup_{i=1}^c A_i = X \quad (1)$$

$$A_i \cap A_j = \emptyset, \quad 1 \leq i \neq j \leq c \quad (2)$$

$$\emptyset \subset A_i \subset X, \quad 1 \leq i \leq c \quad (3)$$

*Interpretacja:*

- (1) – zbiór wszystkich grup zawiera wszystkie wektory danych, a każdy obiekt należy dokładnie do jednej grupy;
- (2) – grupy danych są rozłączne;
- (3) – żadna z grup nie jest pusta ani nie zawiera całego zbioru danych  $X$ .

Aby podzielić dane na  $c$  grup, wygodnie jest przyjąć macierz podziału  $U$  o wymiarach  $cxM$ , zawierającą stopnie przynależności  $u_{ik}$   $k$ -tej danej do  $i$ -tej grupy  $k=1, \dots, M; i=1, \dots, c$ .

# Ostre i rozmyte podziały danych

## Ostre grupowanie danych

*Definicja:*

Niech  $X = [x_1, \dots, x_M]$  będzie zbiorem skończonym i  $c, 2 \leq c < M$ , liczbą całkowitą. Przestrzeń ostrego podziału danych zbioru  $X$  oznaczamy następująco:

$$Z_H = \left\{ U \in R^{c \times M} \mid \mu_{ik} \in \{0,1\}, \forall i, k; \sum_{i=1}^c \mu_{ik} = 1, \forall k; 0 < \sum_{k=1}^M \mu_{ik} < M, \forall i \right.$$

Powyższy podział zakłada, że obiekt należy wyłącznie do jednej grupy i nie istnieją puste grupy ani też grupy zawierające wszystkie obiekty.

# Ostre i rozmyte podziały danych

---

## Nieostre grupowanie danych

Wadą grupowania ostrego jest konieczność przydziału danych do którejś z grup, nawet jeśli „intuicyjnie” dane te do żadnej z grup nie powinny przynależeć.

Istnieją dwie grupy metod nieostrego podziału danych:

- rozmyte (probabilistyczne);
- posybilistyczne.

W obu metodach obiekty mogą należeć do dowolnej ilości grup z różnym stopniem przynależności do danej grupy (z zakresu  $[0 - 1]$ ).

W podziale rozmytym dodatkowo wymaga się aby suma stopni przynależności danego obiektu do każdej z  $c$  grup wynosiła 1.

# Ostre i rozmyte podziały danych

## Rozmyte grupowanie danych

*Definicja:*

Niech  $X = [x_1, \dots, x_M]$  będzie zbiorem skończonym i  $c, 2 \leq c < M$ , liczbą całkowitą. Przestrzeń rozmytego podziału danych zbioru  $X$  oznaczamy następująco:

$$Z_F = \left\{ U \in R^{c \times M} \mid \mu_{ik} \in [0,1], \forall i, k; \sum_{i=1}^c \mu_{ik} = 1, \forall k; 0 < \sum_{k=1}^M \mu_{ik} < M, \forall i \right\}$$

Powyższy podział zakłada, że obiekt może należeć jednocześnie do wszystkich grup, z pewnym stopniem przynależności.

Dodatkowo suma wszystkich stopni przynależności musi być równa 1.

Ponadto żadna z grup nie może być pusta ani nie zawierać całego zbioru danych  $X$ .

# Ostre i rozmyte podziały danych

## Posybilistyczne grupowanie danych

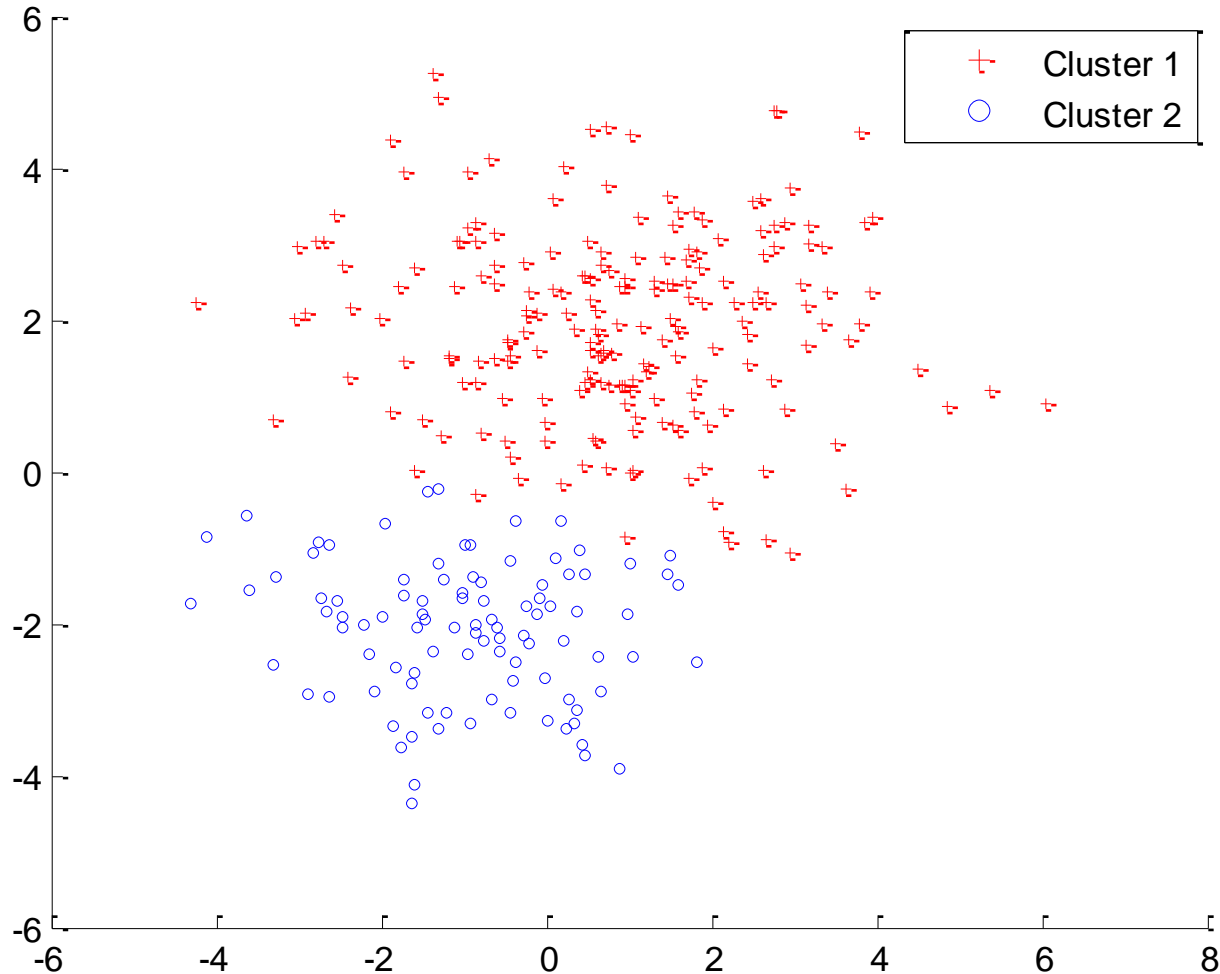
*Definicja:*

Niech  $X = [x_1, \dots, x_M]$  będzie zbiorem skończonym i  $c, 2 \leq c < M$ , liczbą całkowitą. Przestrzeń posybilistycznego podziału danych zbioru  $X$  oznaczamy następująco:

$$Z_P = \left\{ U \in R^{c \times M} \mid \mu_{ik} \in [0,1], \forall i, k; \forall k, \exists i, \mu_{ik} > 0; 0 < \sum_{k=1}^M \mu_{ik} < M, \forall i \right\}$$

# Ostre i rozmyte podziały danych

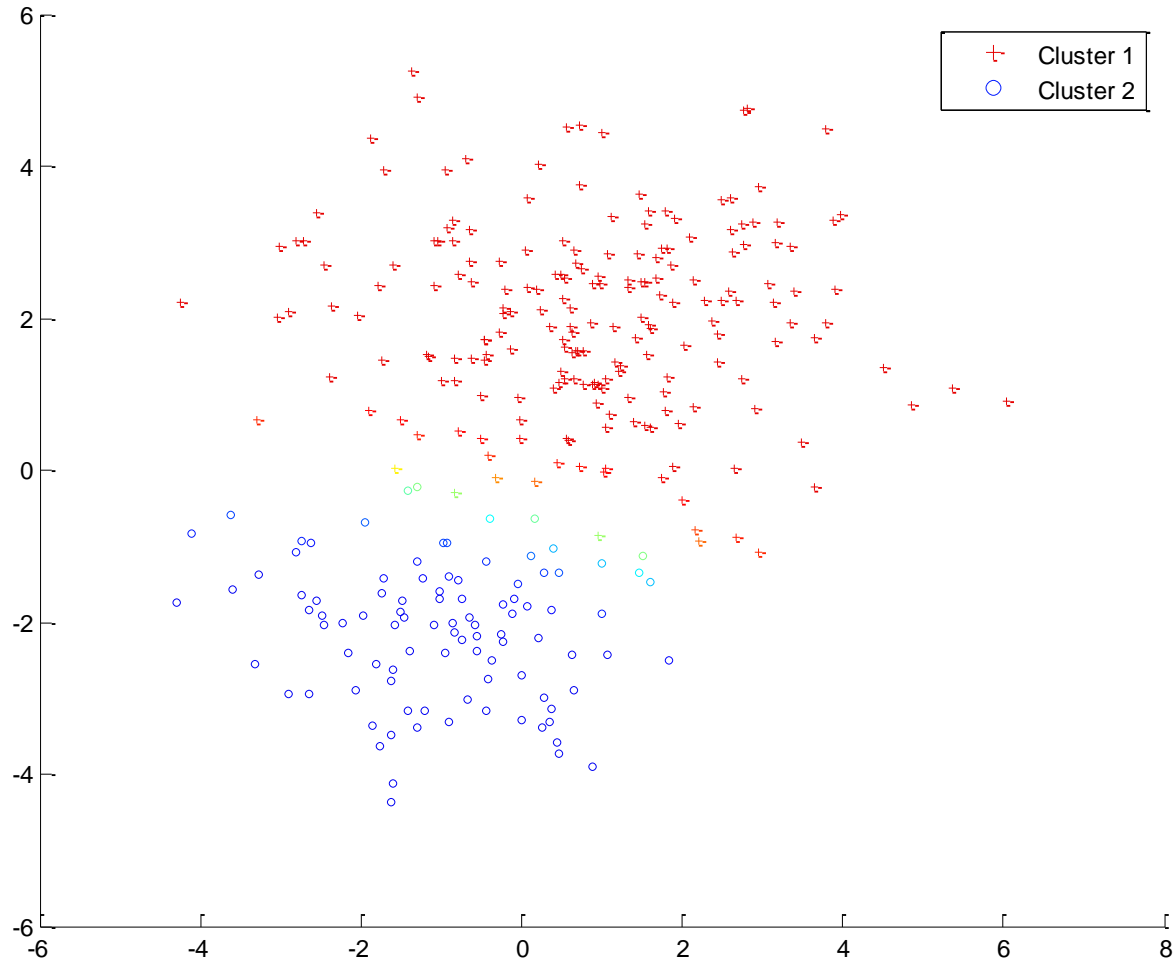
## Ostre grupowanie danych





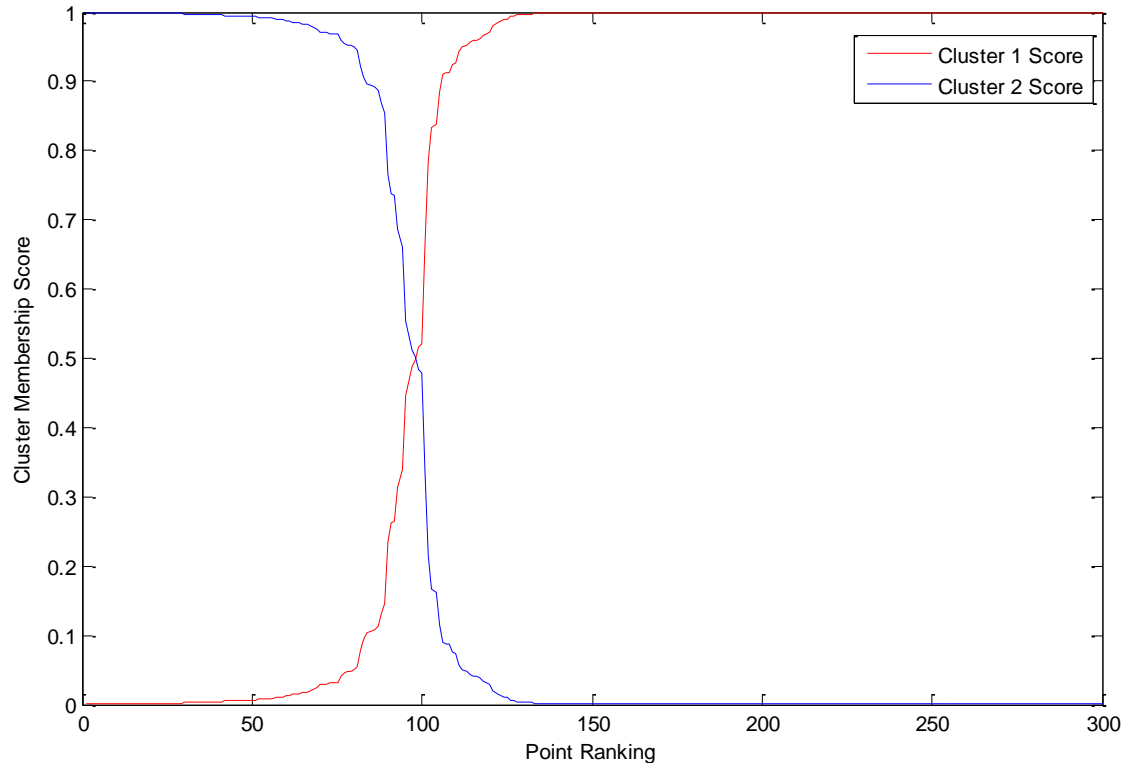
# Ostre i rozmyte podziały danych

## Nieostre grupowanie danych



# Ostre i rozmyte podziały danych

## Nieostre grupowanie danych



## Miary odległości

Niezmiernie istotnym czynnikiem wpływającym na wynik podziału danych jest sposób obliczania odległości pomiędzy obiektami zbioru danych.

W przypadku grupowania danych mierzymy odległości w przestrzeni cech.

Najczęściej stosowaną miarą odległości jest miara euklidesowa, interpretowana jako geometryczna odległość między dwoma punktami w przestrzeni  $X$ .

Rozważmy dwa punkty o współrzędnych  $x_s = [x_1, x_2, \dots, x_m]$  oraz  $x_t = [x_1, x_2, \dots, x_m]$  i policzmy odległości pomiędzy nimi zgodnie z normami:

- Euklidesowa;
- Euklidesowa ważona (odchyleniem standardowym);
- Manhattan (city block);
- Mahalanobis;
- Czebyszewa;
- Hamminga;
- Cosine.

# Miary odległości

## Miara euklidesowa

$$d = \sqrt{\sum_{j=1}^n (x_s - x_t)^2}$$

W zapisie wektorowym:

$$d = [(x_s - x_t)(x_s - x_t)']^{\frac{1}{2}}$$

Miara Euklidesowa jest uogólnieniem metryki Minkowskiego dla  $r = 2$ :

$$d = \left( \sum_{j=1}^n |x_{sj} - x_{tj}|^r \right)^{\frac{1}{r}}$$

Matlab

`D = pdist(X,'euclidean')`  
gdzie: D – zbiór danych

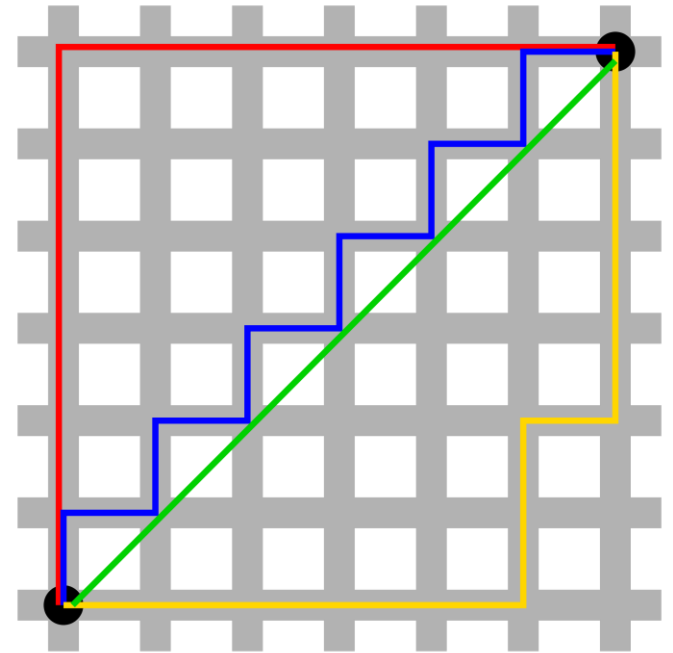
# Miary odległości

## Miara Manhattan (city block)

Miara Manhattan (city block) jest miarą przy  $r=1$ :

$$d = \sum_{j=1}^n |x_{sj} - x_{tj}|$$

Wszystkie odległości według miary city block wynoszą 12, natomiast odległość Euklidesowa (zielona) wynosi:  $6 \times \sqrt{2} \approx 8.48$ , i jest najkrótszą możliwą ścieżką.



źródło:

[http://en.wikipedia.org/wiki/Taxicab\\_geometry](http://en.wikipedia.org/wiki/Taxicab_geometry)

Matlab

```
D = pdist(X, 'cityblock')  
gdzie: D – zbiór danych
```

# Miary odległości

## *Miara Hamminga*

W przypadku zmiennych binarnych miara Manhattan staje się miarą Hamminga i określa liczbę bitów o którą różnią się dwa ciągi bitów. Ciągi te mogą reprezentować np. czarno-białe obrazy.

Matlab

```
D = pdist(X,'hamming')  
gdzie: D – zbiór danych
```

# Miary odległości

## Miara Mehalanobisa

Miary Minkowskiego są podatne na różnice w wielkości poszczególnych zmiennych (te większe „dominują” nad mniejszymi).

Rozwiązaniem jest wprowadzenie wag (macierz A):

$$d = \left[ (x_s - x_t) A (x_s - x_t)' \right]^{1/2}$$

Gdy macierz A:

$$A = \Sigma^{-1}$$

przy czym  $\Sigma$  jest macierzą diagonalną (nxn) odchyłeń standardowych, otrzymujemy miarę **Euklidesową ustandaryzowaną**:

Gdy macierz A:

$$A = R^{-1}$$

przy czym R jest macierzą (nxn) kowariancji danych, otrzymujemy miarę **Mehalanobisa**.

### Matlab

D = pdist(X,'mahalanobis')

D = pdist(X,'seuclidean')

gdzie: D – zbiór danych

# Algorytmy grupowania danych

## Algorytm HCM (Hard k-Means)

Algorytm HCM dzieli jednoznacznie dane  $X$  na  $c$  grup. W trakcie realizacji algorytmu obliczamy odległość pomiędzy każdym wektorem:

$$x_k \in R^n, \quad k = 1, \dots, M$$

i środkiem grupy:

$$v_i, \quad i = 1, \dots, c$$

Środek grupy jest średnią położenia wszystkich obiektów należących do tej grupy. Przynależność do grupy opisujemy za pomocą macierzy:

$$U = [\mu_{ik}] \in Z_H$$

Elementami macierzy  $U$  są zera i jedynki określające przynależność obiektu  $x_k$  do  $i$ -tej grupy.

Inicjalizacja algorytmu polega na wyborze liczby grup  $c$  i ustaleniu początkowego położenia ich środków np. losowo.



# Algorytmy grupowania danych

## Algorytm HCM (Hard K-Means)

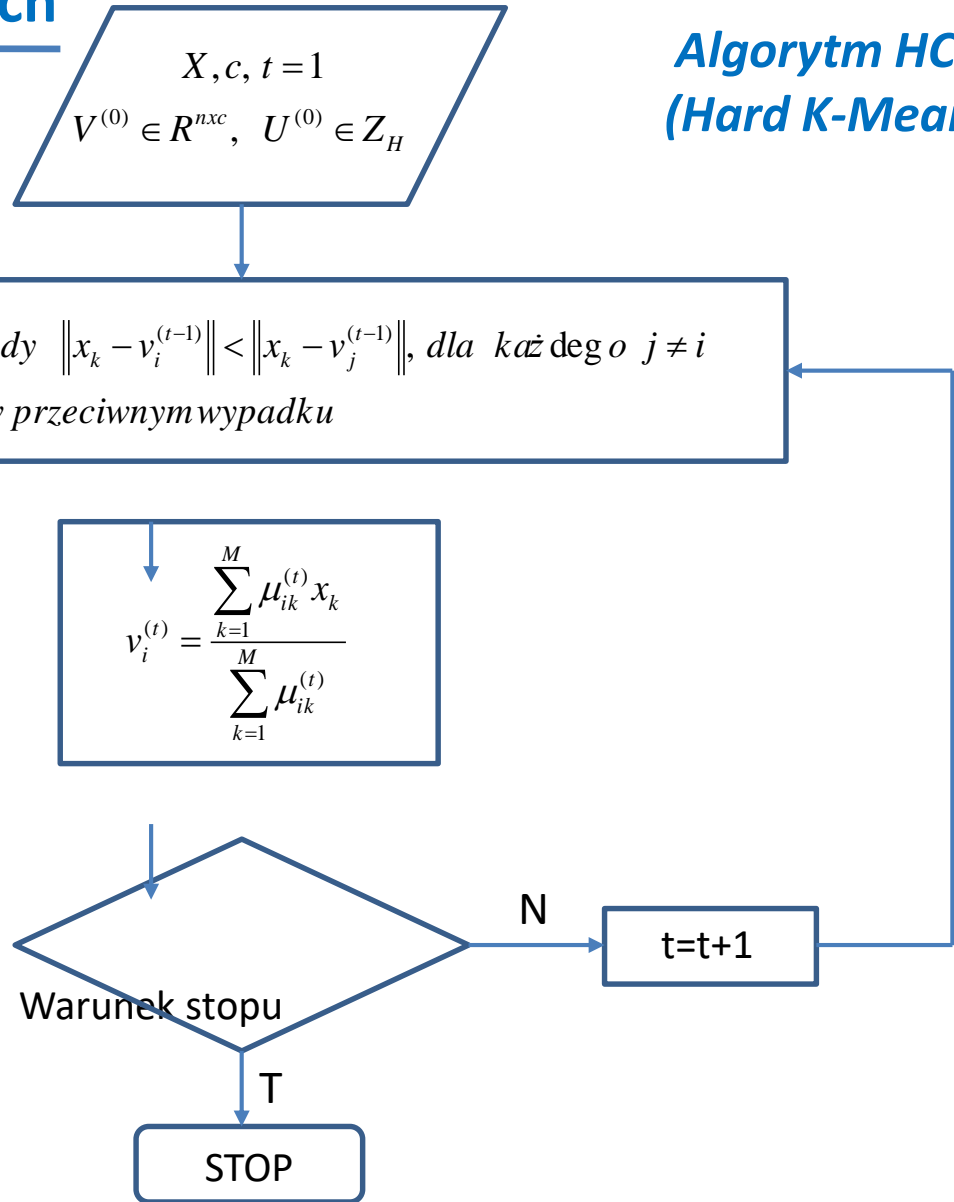
Warunek stopu:

$$\|U^{(t+1)} - U^{(t)}\| < \varepsilon$$

lub:

$$\|V^{(t+1)} - V^{(t)}\| < \varepsilon$$

Algorytm może dać różne wyniki  
w zależności od początkowego  
położenia środków grup



# Algorytmy grupowania danych

---

## *Algorytm k-medoids*

Algorytm *k-medoids* jest modyfikacją algorytmu *k-means*, polegającą na tym, że wybór reprezentantów klastrów (medodoidów) jest ograniczony do istniejących elementów grup.

Stosuje się go w sytuacjach w których nie możliwe jest obliczenie średniej zbioru lub gdy wartość ta jest trudna do interpretacji. Przykładem takiego problemu są zbiory danych kategoriycznych (np. lingwistycznych).

# Algorytmy grupowania danych

## *Algorytm FCM (Fuzzy C-Means), znany również jako Soft k-means*

Algorytm FCM pozwala przypisać te same obiekty do różnych grup z odpowiednimi stopniami przynależności.

Wszystkie grupy mają ten sam kształt, zależny od przyjętej normy, bo algorytm nie ma możliwości dostosowywania macierzy  $A$  do istniejących danych.

Algorytm minimalizuje kryterium:

$$J(X; U, V) = \sum_{i=1}^c \sum_{k=1}^M (\mu_{ik})^m \|x_k - v_i\|_A^2$$

$$U = [\mu_{ik}] \in Z_F$$

$U$  jest macierzą podziału zbioru  $X$ , natomiast  $V$  jest wektorem środków, które mają być określone w wyniku działania algorytmu.

$$V = [v_1, v_2, \dots, v_c], \quad v_i \in R^n, \quad i = 1, \dots, c$$

# Algorytmy grupowania danych

## Algorytm FCM (Fuzzy C-Means)

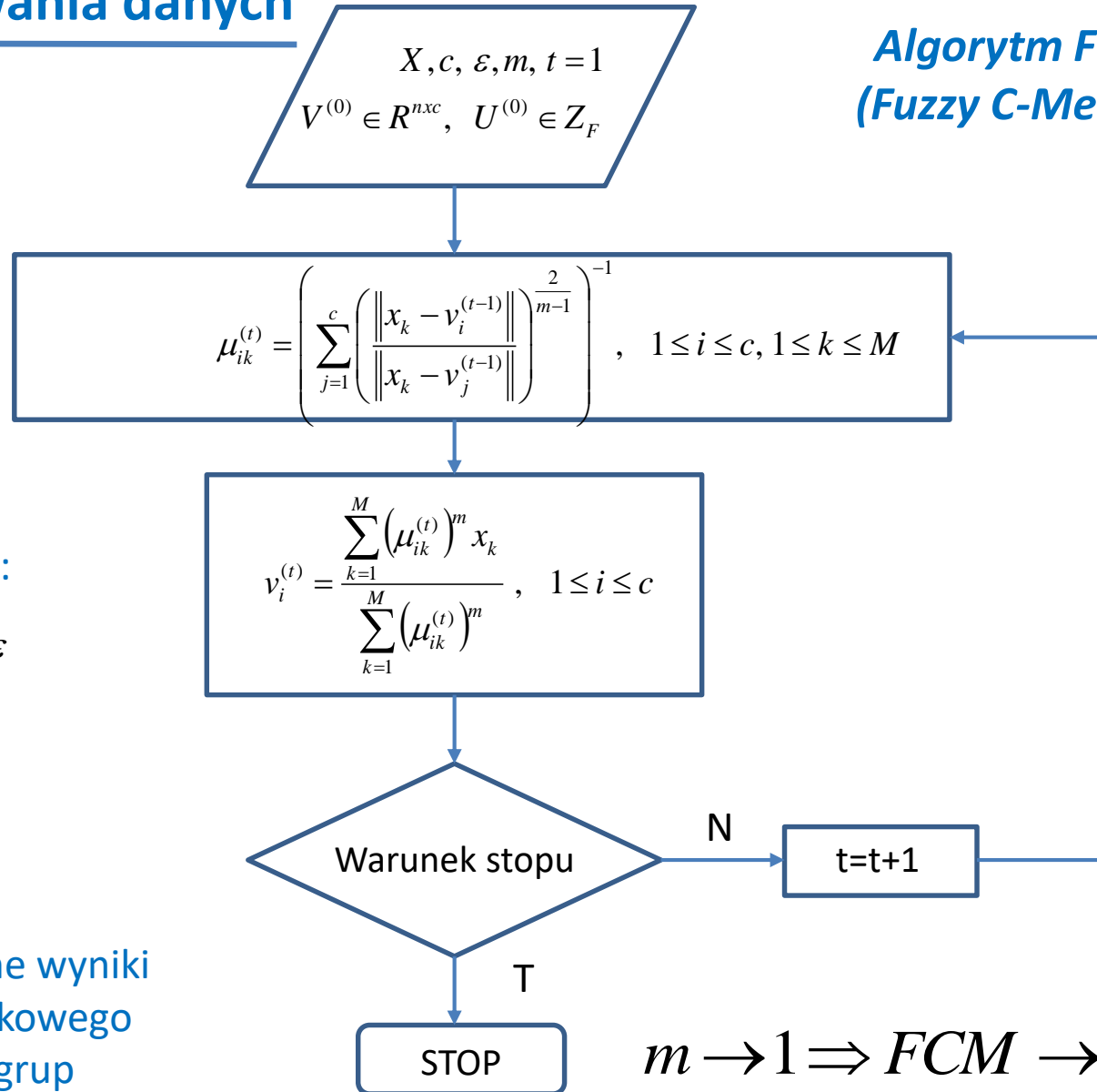
Warunek stopu:

$$\|U^{(t+1)} - U^{(t)}\| < \varepsilon$$

lub:

$$\|V^{(t+1)} - V^{(t)}\| < \varepsilon$$

Algorytm może dać różne wyniki  
w zależności od początkowego  
położenia środków grup



$$m \rightarrow 1 \Rightarrow FCM \rightarrow HCM$$

# Algorytmy grupowania danych

## Algorytm PCM (*Possibilistic C-Means*)

Algorytm FCM zakłada że suma stopni przynależności danego obiektu do każdej z grup jest zawsze równa 1.

Może to spowodować niepożądane przesunięcia środków w przypadku występowania pojedynczych „przypadkowych” danych (outliers).

Rezygnacja z tego ograniczenia prowadzi do PCM.

Algorytm minimalizuje kryterium:

$$J(X;U,V) = \sum_{i=1}^c \sum_{k=1}^M (\mu_{ik})^m \|x_k - v_i\|_A^2 + \sum_{i=1}^c \eta_i \sum_{k=1}^M (1 - \mu_{ik})^m$$

Drugi człon wymusza jak największe stopnie przynależności.

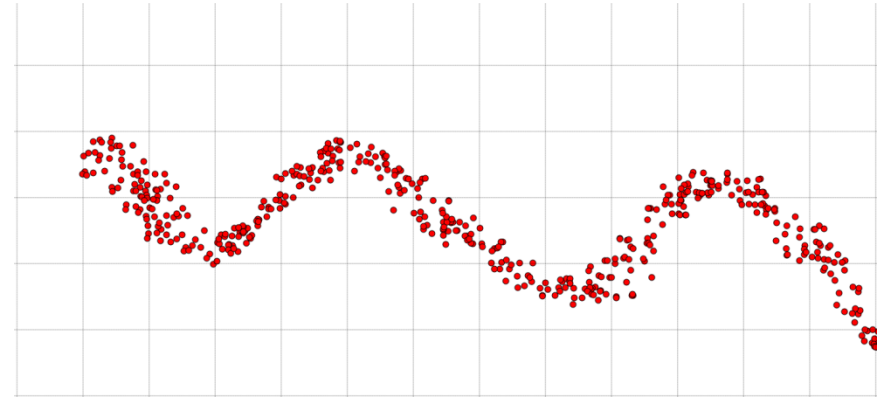
$\eta_i (>0)$  określa tzw. szerokość wynikowego rozkładu posybilistycznego.

Niewłaściwa inicjalizacja algorytmu może spowodować że wszystkie stopnie przynależności będą sobie równe, więc najczęściej najpierw stosujemy FCM.

# Algorytmy grupowania danych

## Gaussian Mixture Models (GMM)

Algorytm opisuje klastry poprzez modele opisane rozkładem Gaussa o odpowiedniej średniej i kowariancji



## Rozkład Gaussa

$$N(x / \bar{x}, \Sigma) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp \left\{ -\frac{1}{2} (x - \bar{x})^T \Sigma^{-1} (\bar{x} - \mu) \right\}$$

# Algorytmy grupowania danych

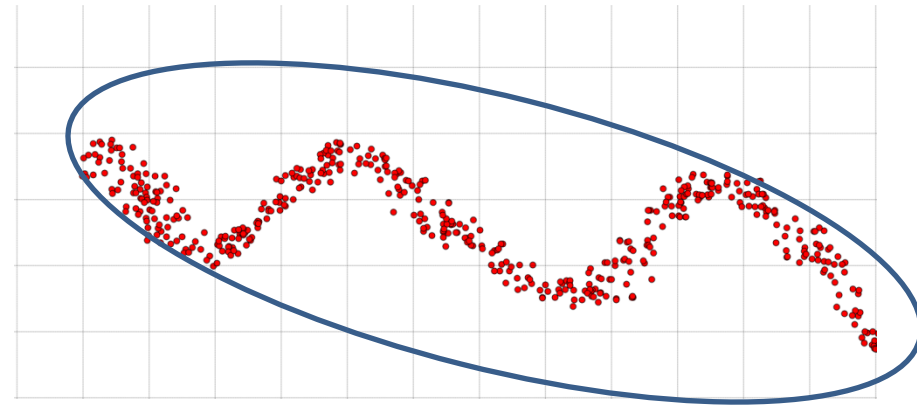
## Gaussian Mixture Models (GMM)

Algorytm opisuje klastry poprzez modele opisane rozkładem Gaussa o odpowiedniej średniej i kowariancji

### Rozkład Gaussa

$$N(x / \bar{x}, \Sigma) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp \left\{ -\frac{1}{2} (x - \bar{x})^T \Sigma^{-1} (\bar{x} - \mu) \right\}$$

Model Gaussa



# Algorytmy grupowania danych

## Gaussian Mixture Models (GMM)

Superpozycja lokalnych modeli Gaussa

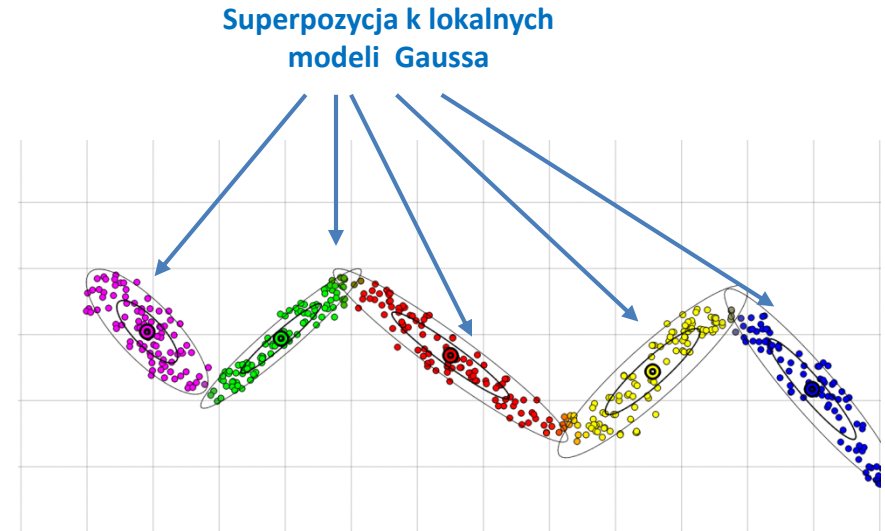
$$p(x) = \sum_{k=1}^K \alpha_k N(x / \bar{x}_k, \Sigma_k)$$

normalizacja modelu:

$$\sum_{k=1}^K \alpha_k = 1 \quad 0 \leq \alpha_k \leq 1$$

$\alpha_k$  można interpretować jako prawdopodobieństwo przynależności do danego modelu lokalnego:

$$p(x) = \sum_{k=1}^K p(k) p(x / k)$$





## Metody klasteryzacji

---

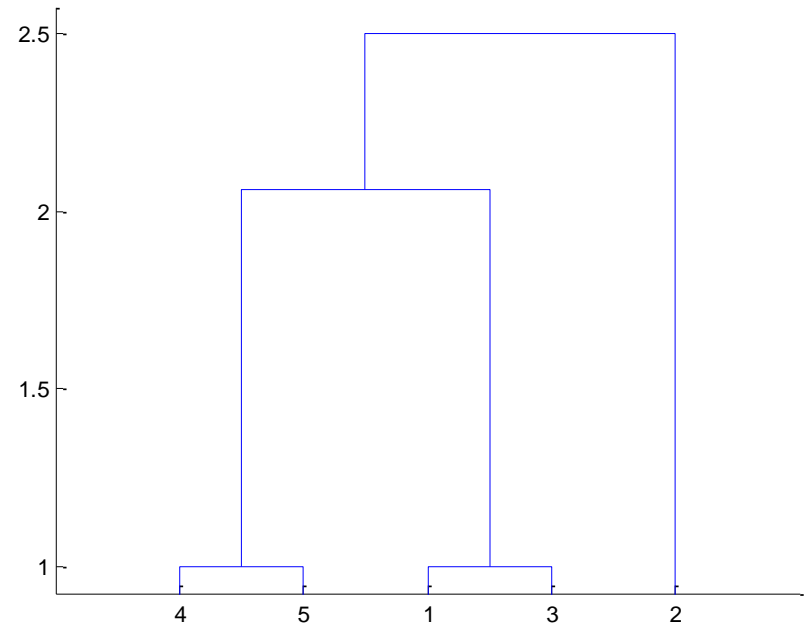
# Klasteryzacja hierarchiczna (hierarchical clustering)

# Klasteryzacja hierarchiczna

Klasteryzacja hierarchiczna jest metodą grupowania danych w różnej skali poprzez tworzenie drzewa klastrów zwanego dendrogramem.

Drzewo nie jest zwyczajnym zbiorem klastrów a raczej wielopoziomową strukturą hierarchiczną, w której to klastry jednego poziomu tworzą (po połączeniu) klastery na poziomie wyższym.

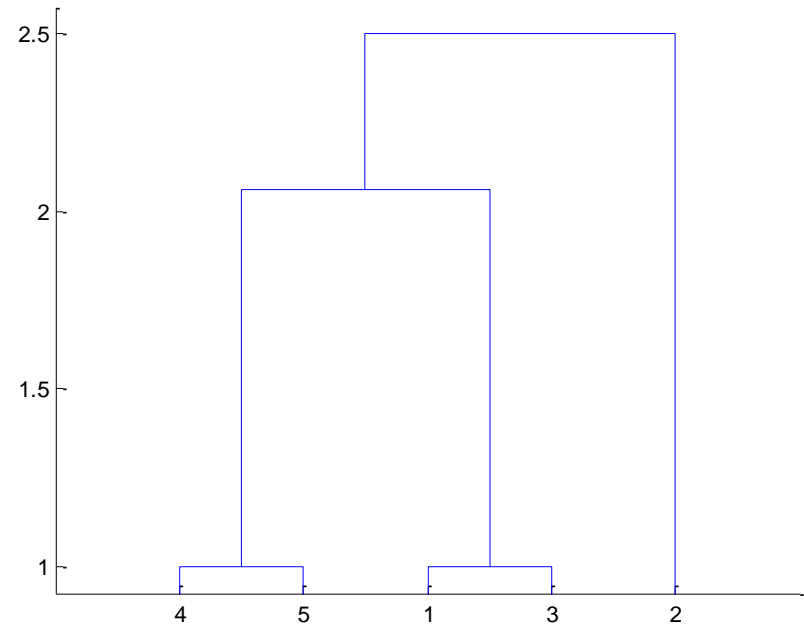
Dzięki temu możemy uzyskać odpowiedni do naszych potrzeb stopień klasteryzacji.



# Klasteryzacja hierarchiczna

Można wyróżnić dwa sposoby klasteryzacji hierarchicznej:

- klasteryzacja skupiająca – zaczynamy od pojedynczych elementów stanowiących klastry, a następnie w kolejnych krokach łączymy klastry na kolejnych poziomach, aż do uzyskania jednej grupy skupiającej wszystkie obiekty;
- klasteryzacja dzieląca – zakładamy że wszystkie dane należą do jednego klastra, a w kolejnych krokach dokonujemy podziału klastra na kolejne, aż do momentu gdy każdy z elementów będzie stanowił pojedynczy klaster.

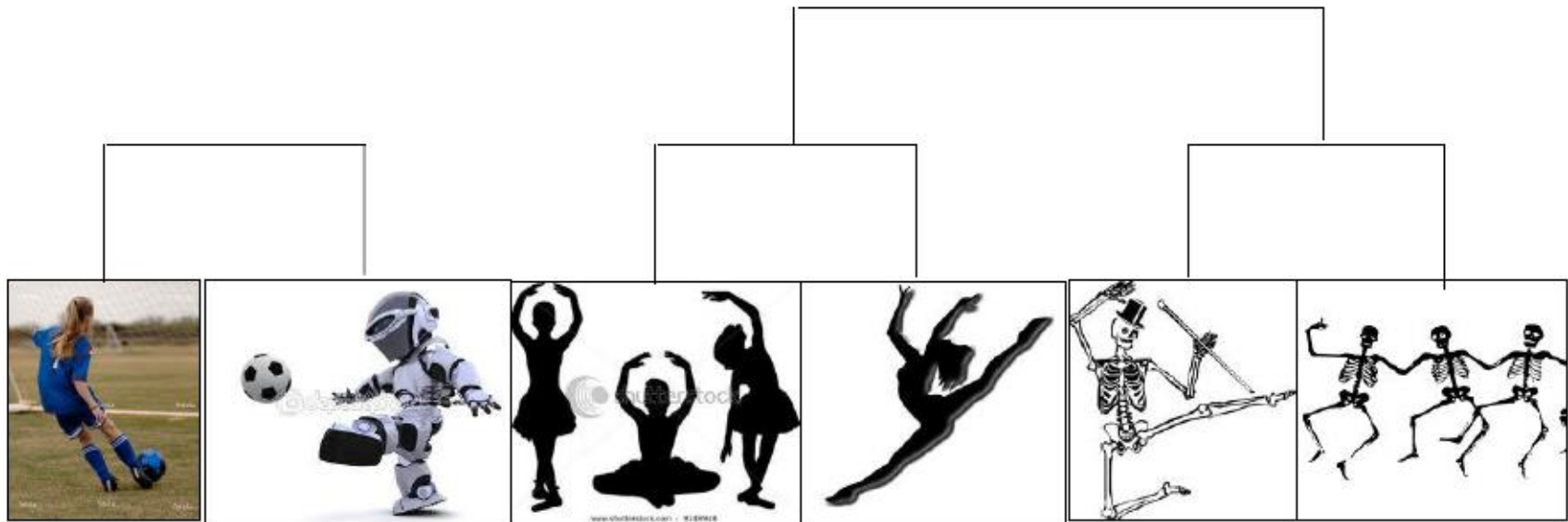


# Klasteryzacja hierarchiczna



źródło: Lecture Notes: Applied Machine Learning. LASA Laboratory, EPFL CH-1015 Lausanne, Switzerland.

# Klasteryzacja hierarchiczna



źródło: Lecture Notes: Applied Machine Learning. LASA Laboratory, EPFL CH-1015 Lausanne, Switzerland.

# Klasteryzacja hierarchiczna

## Algorytm klasteryzacji hierarchicznej przy użyciu Matlab Statistic Toolbox (skrajnie uproszczony!!):

1. Znajdź miary podobieństwa (odległości wedle przyjętej metryki) pomiędzy wszystkimi parami elementów danego zbioru danych.  
(Matlab: ***pdist*** – wiele miar podobieństwa);

2. Zgrupuj dane w binarne, hierarchiczne drzewo klastrów (łączenie „bliskich” elementów/klastrów )

W tym kroku grupujemy „bliskie” elementy zbioru przy użyciu polecenia *linkage*.

Polecenie to wykorzystuje miary podobieństwa obliczone w poprzednim kroku (*pdist*).

Nowo powstałe grupy poddawane są kolejnym „łączeniom” (na podstawie miar podobieństwa)

w kolejne, większe klastry aż do momentu gdy zostanie utworzone pełne drzewo.

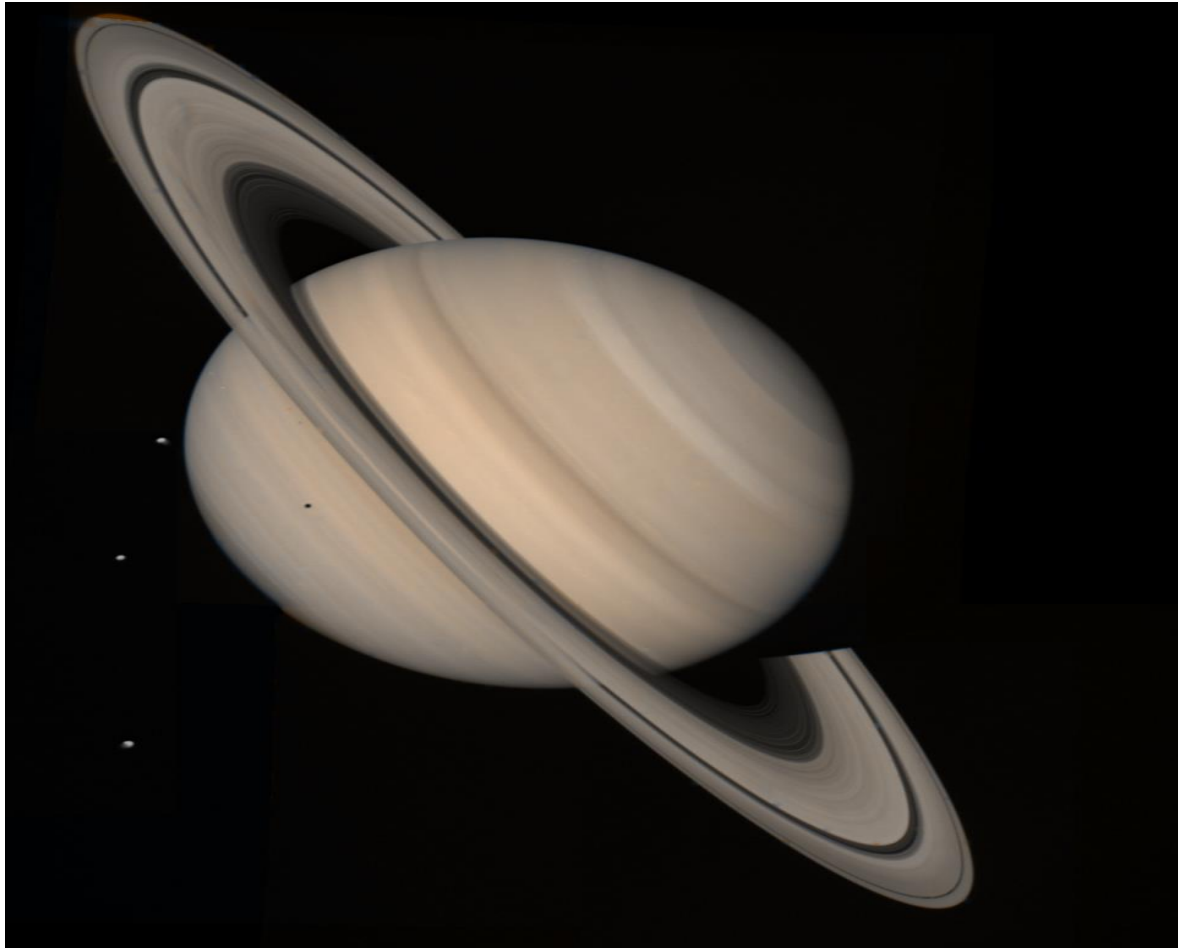
(Matlab: ***linkage***);

3. Zdecyduj w którym miejscu „potniesz” drzewo na klastry.

W tym kroku wykorzystując polecenie *cluster*, odcinamy gałęzie drzewa i przypisujemy elementy poniżej cięcia do jednego klastra. Funkcja *cluster*, umożliwia tworzenie klastrów poprzez detekcję naturalnego podziału danych oraz poprzez arbitralne narzucanie ilości klastrów. (Matlab: ***cluster***);

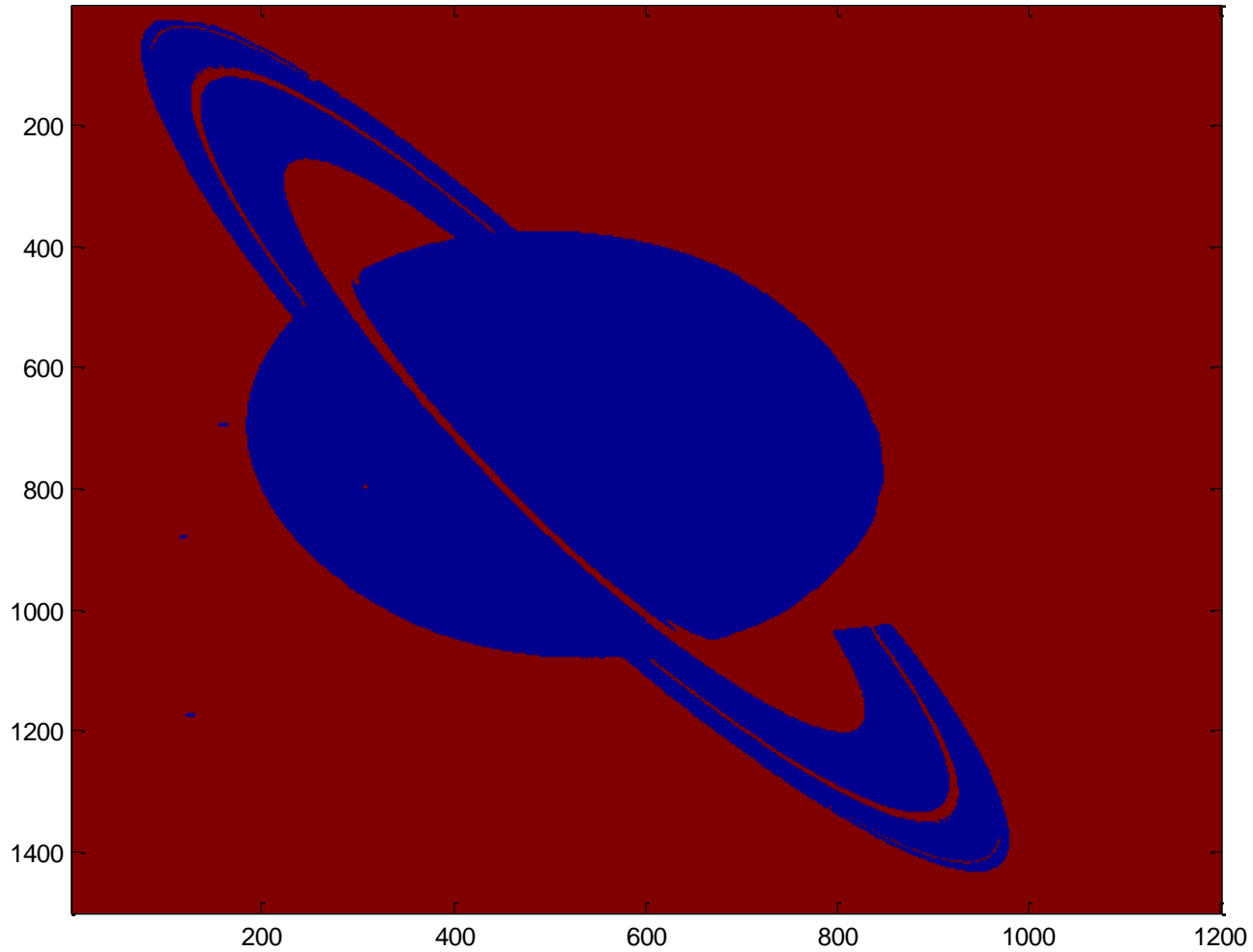
## Przykłady

---



# Przykłady

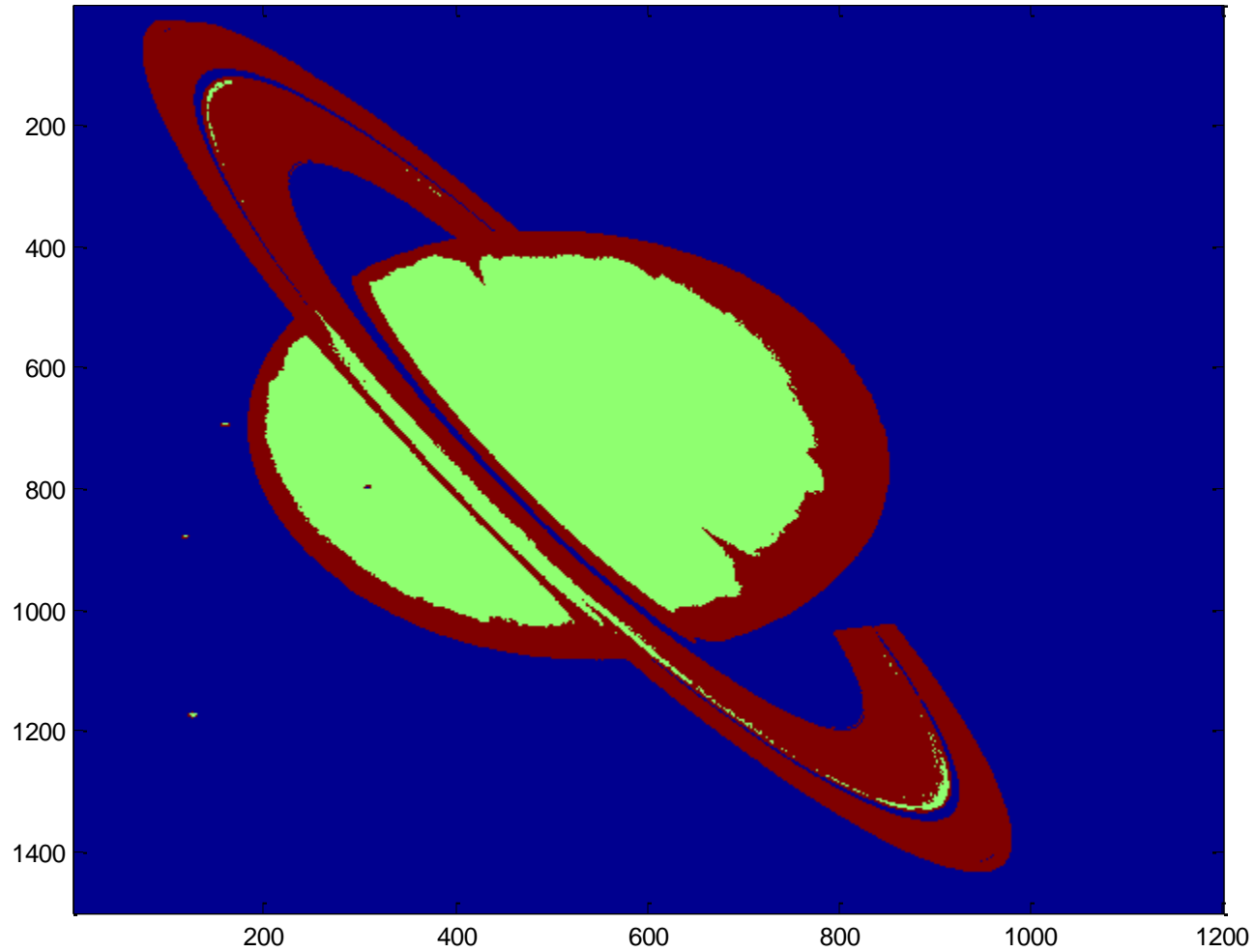
## 2 klastry





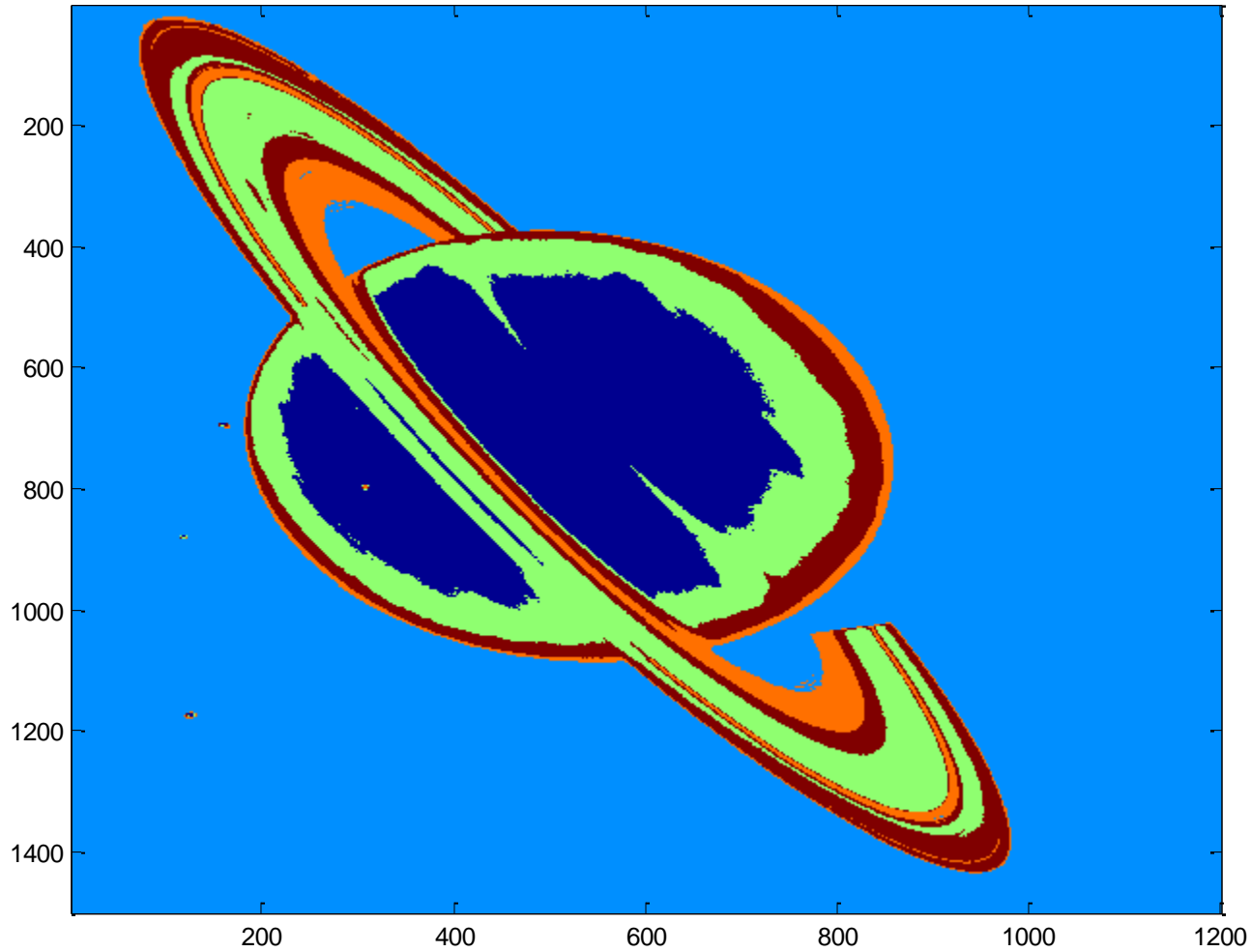
# Przykłady

## 3 klastry



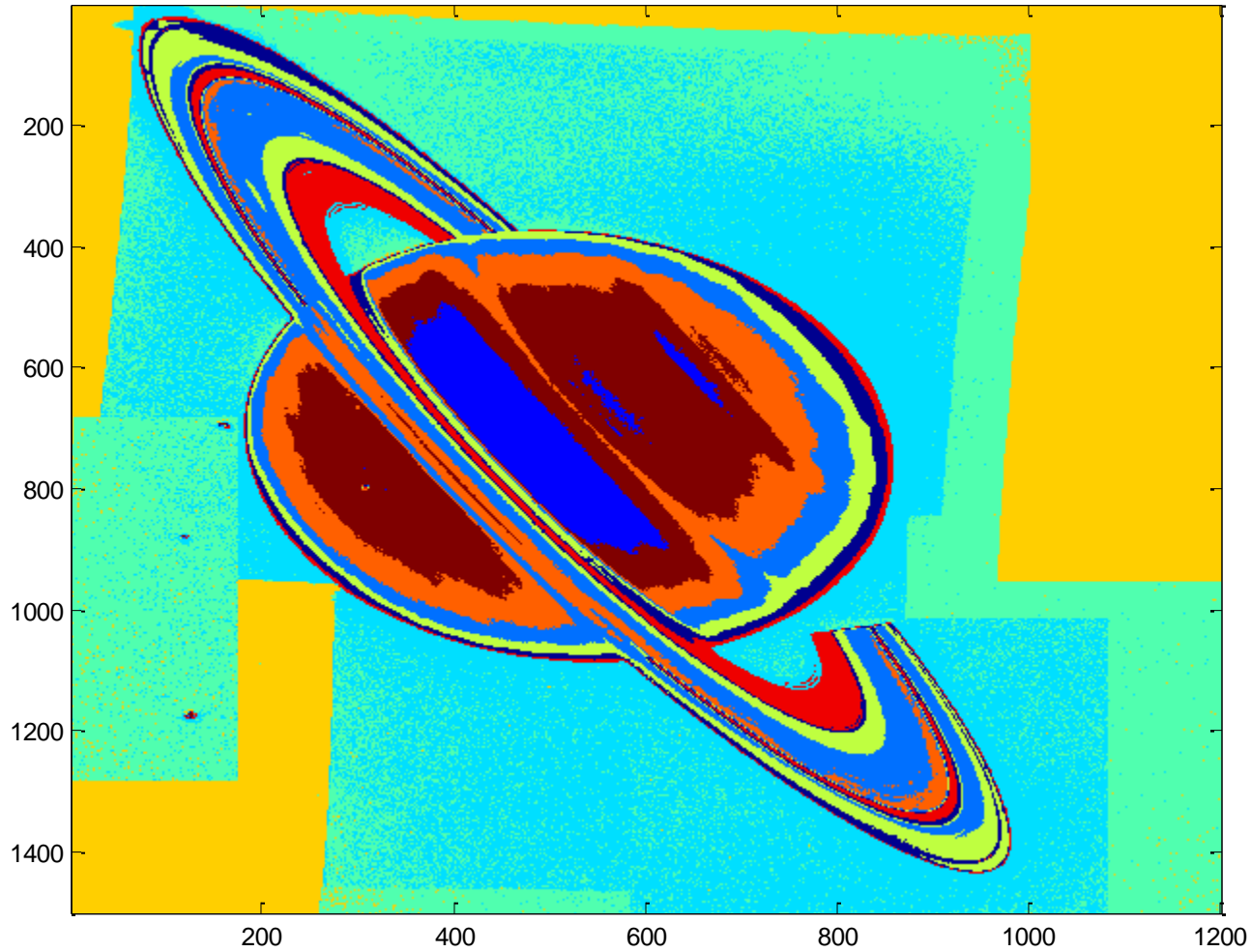
# Przykłady

## 5 klastrów



# Przykłady

## 10 klastrów

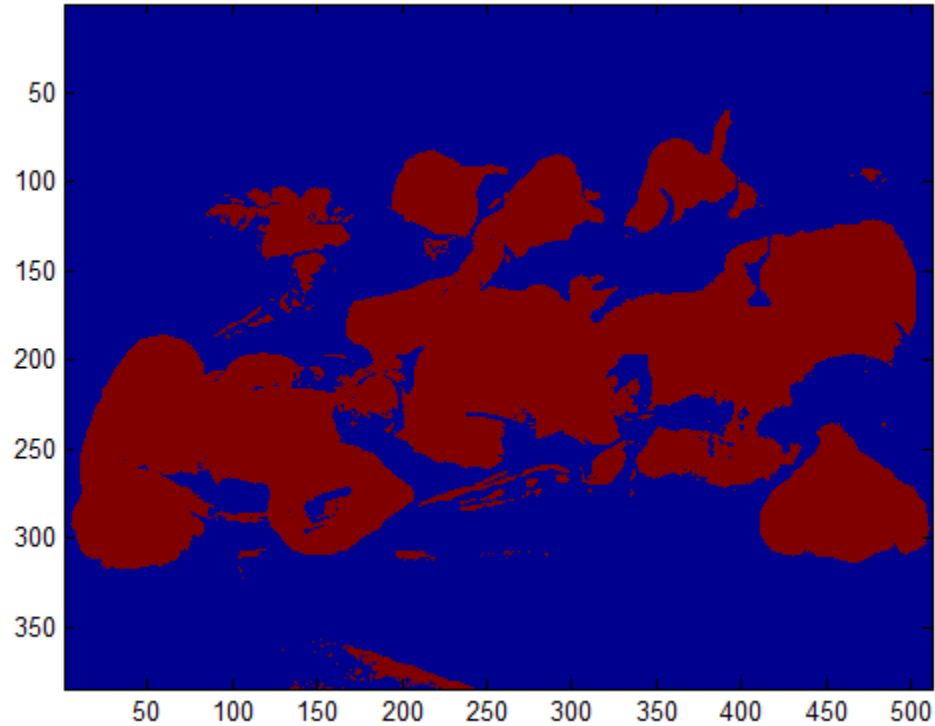


# Przykłady



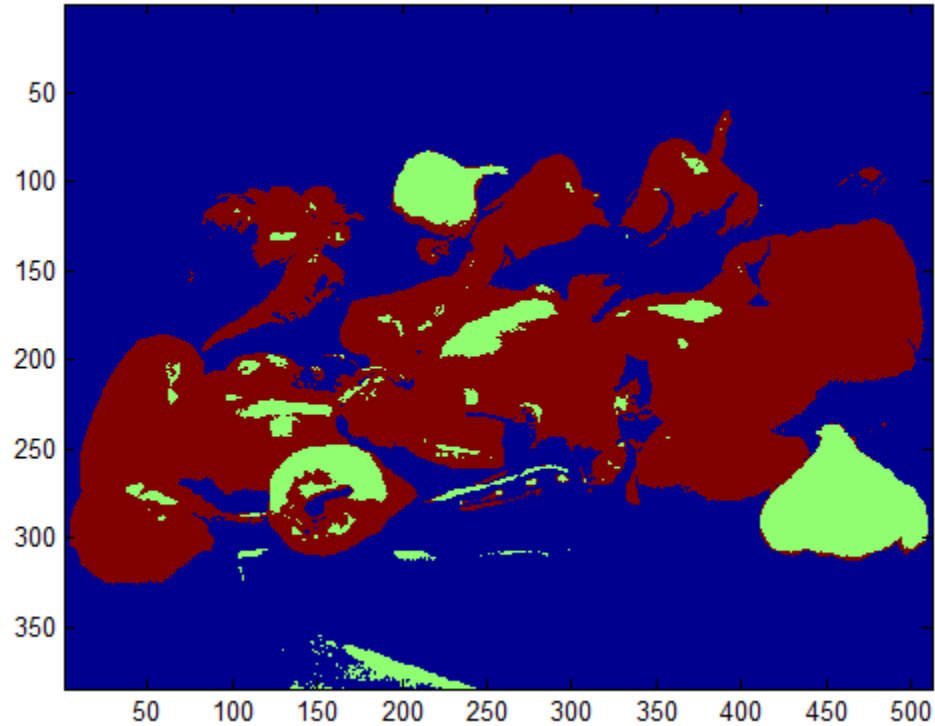
# Przykłady

## 2 klastry



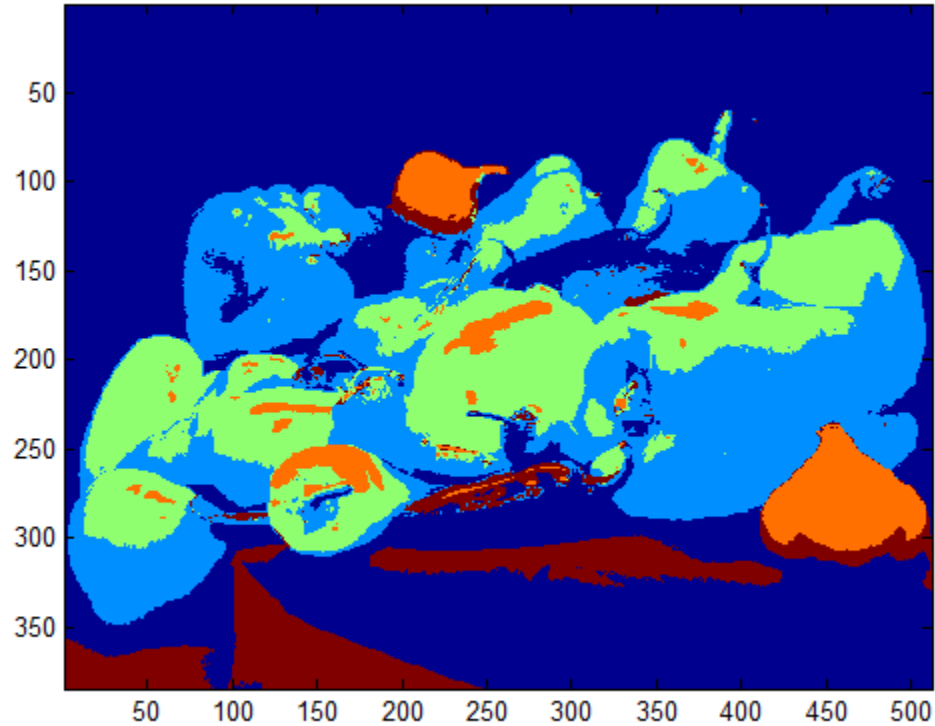
# Przykłady

## 3 klastry



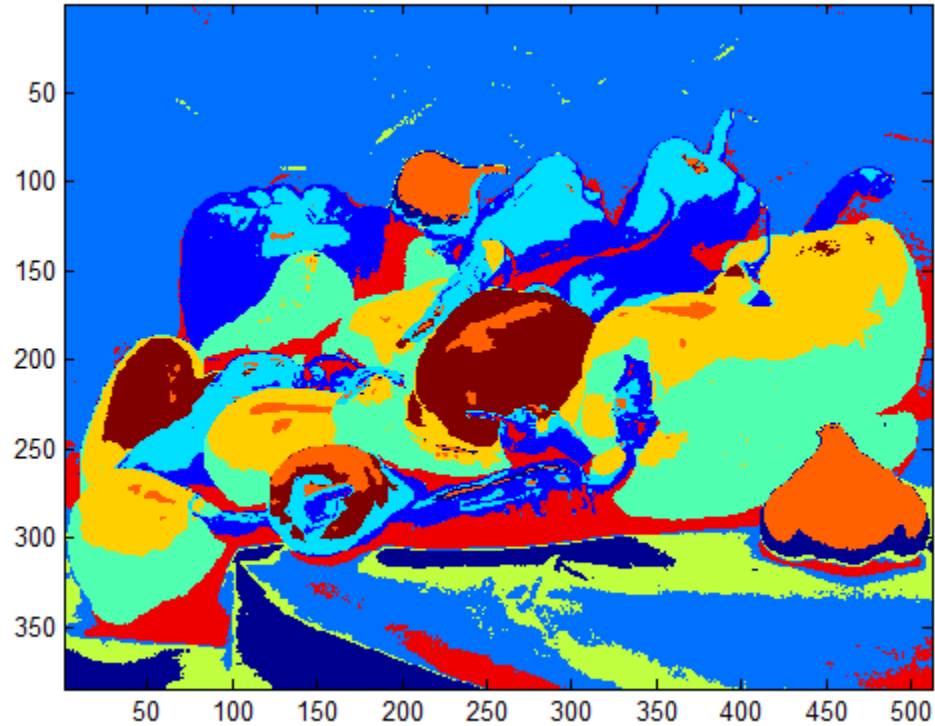
# Przykłady

## 5 klastrów



# Przykłady

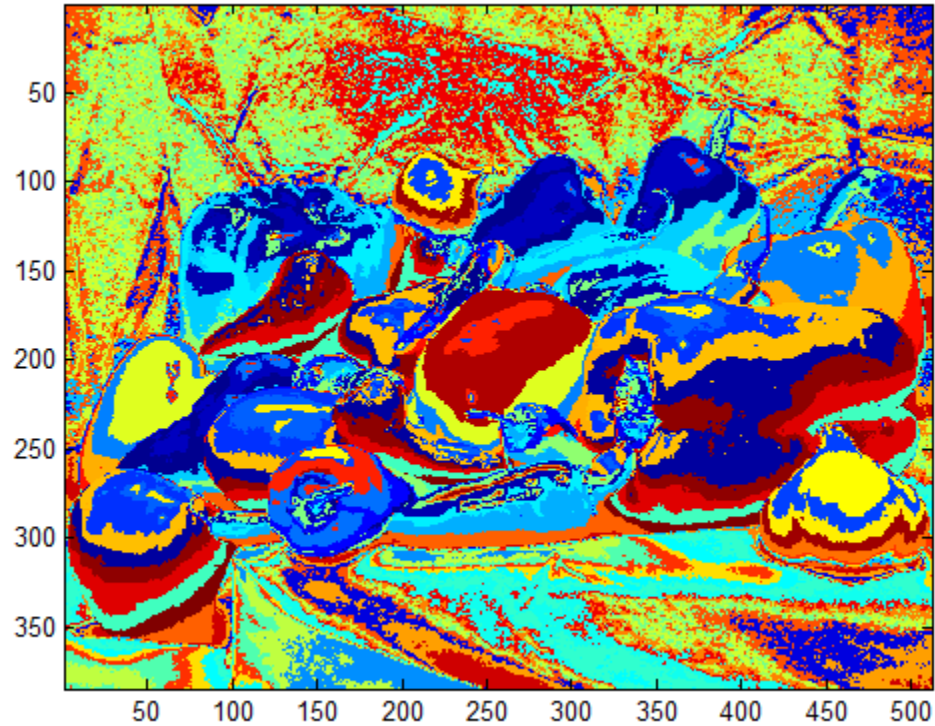
## 10 klastrów





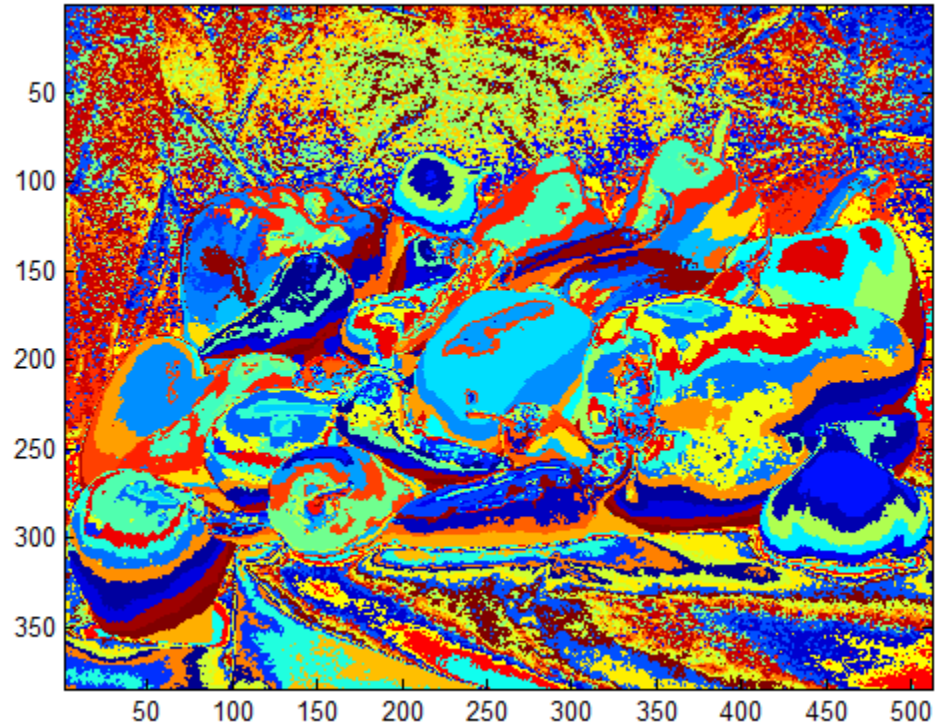
# Przykłady

## 50 klastrów



# Przykłady

## 70 klastrów



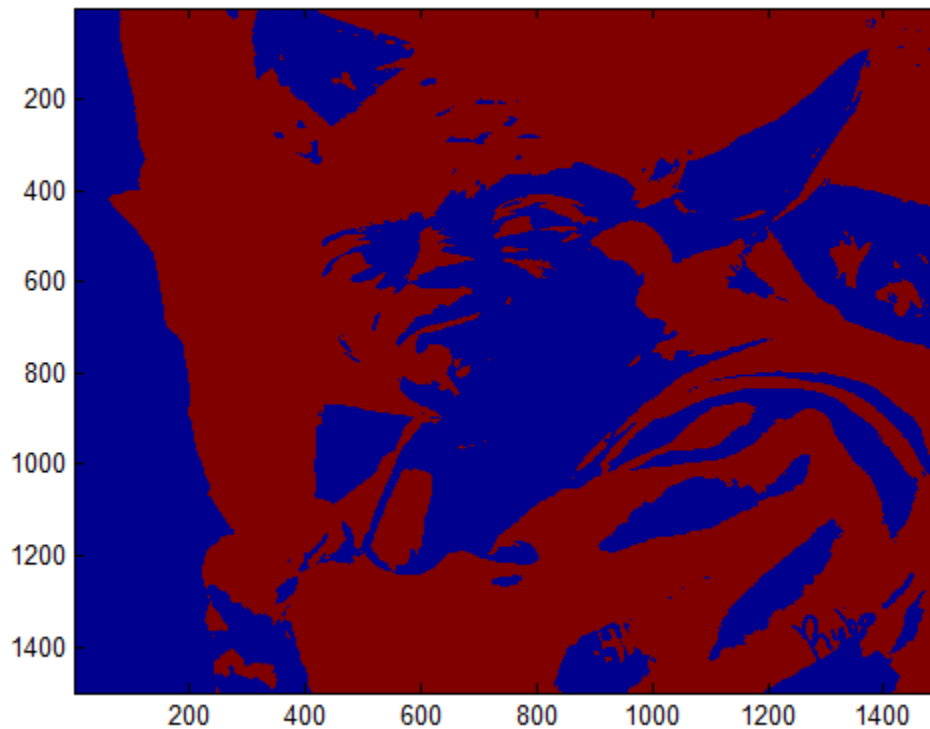
## Przykłady

---



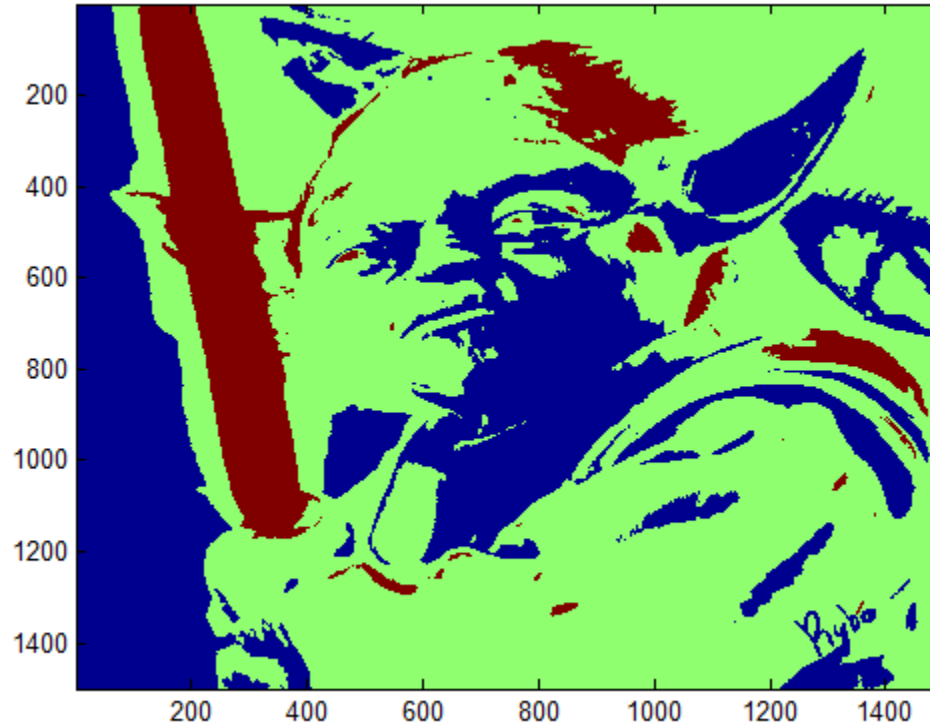
# Przykłady

## 2 klastry



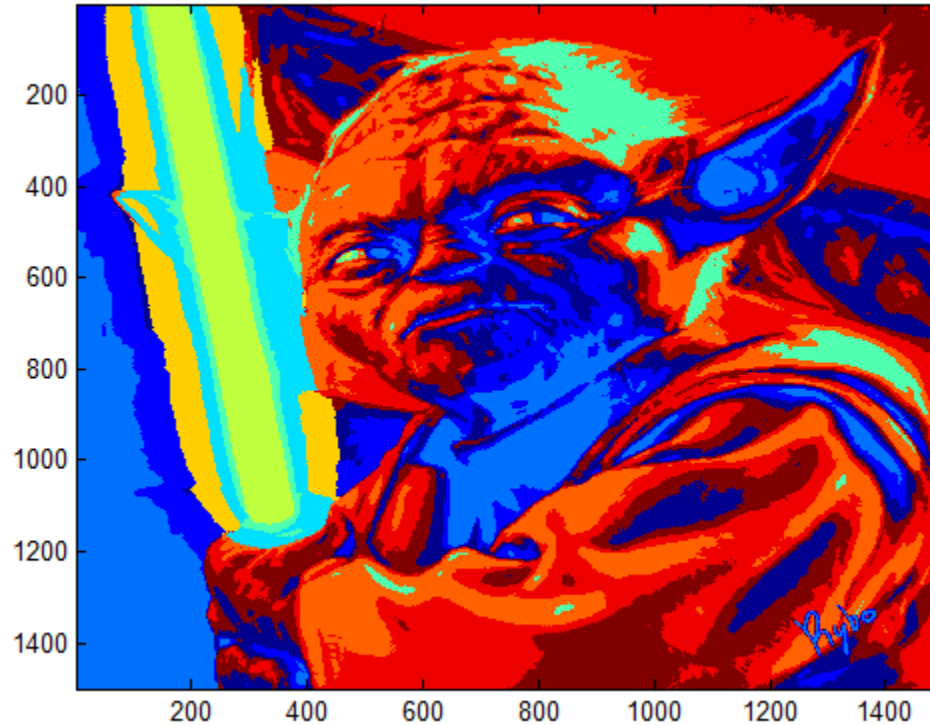
# Przykłady

## 3 klastry



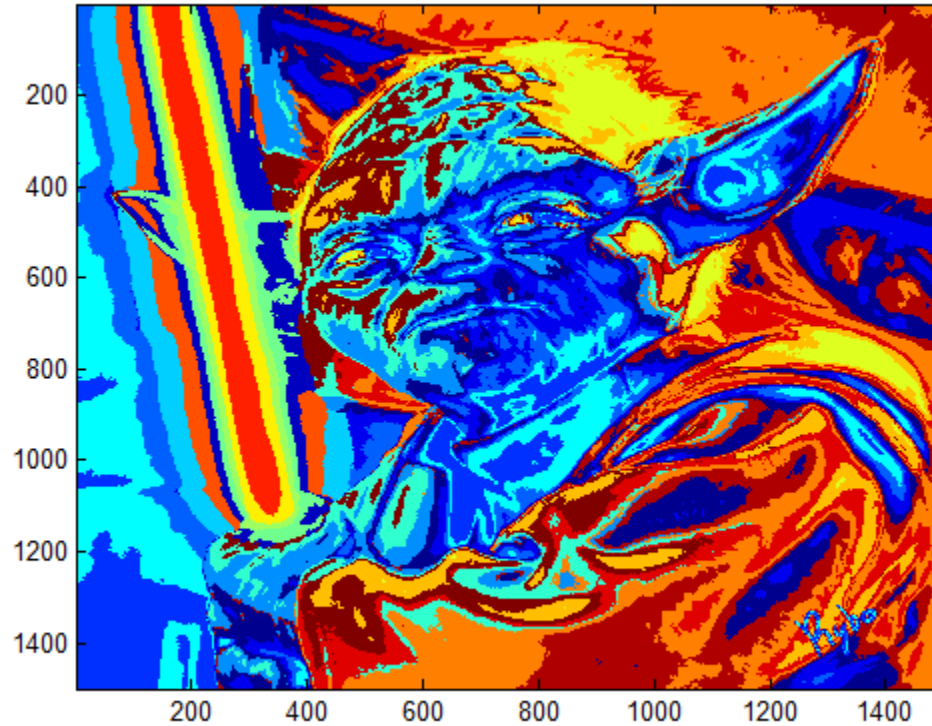
# Przykłady

## 10 klastrów



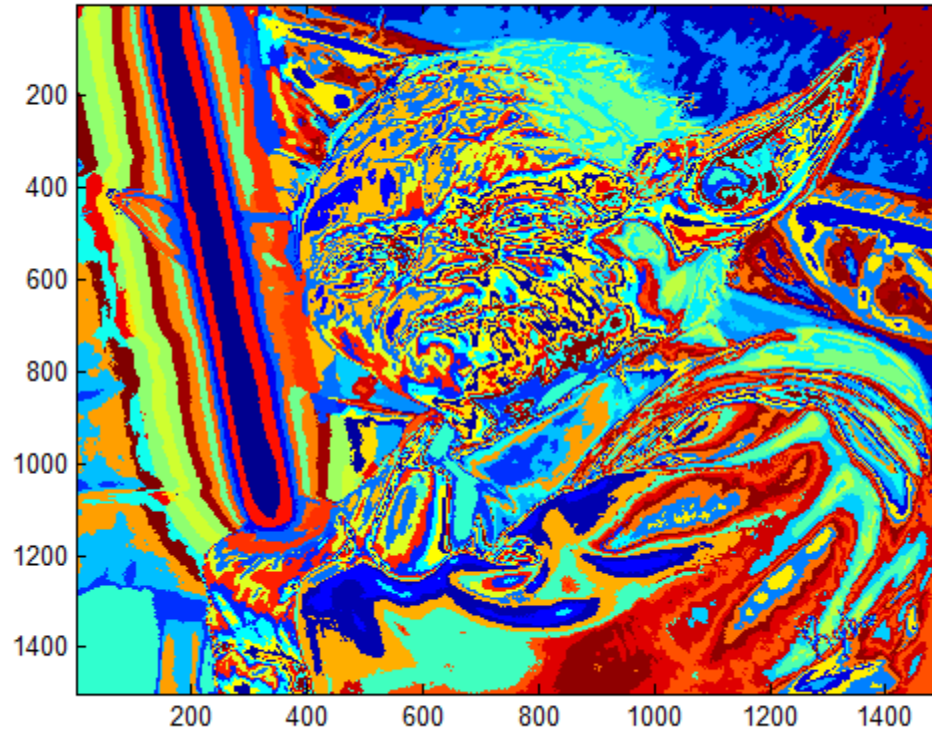
# Przykłady

## 20 klastrów



# Przykłady

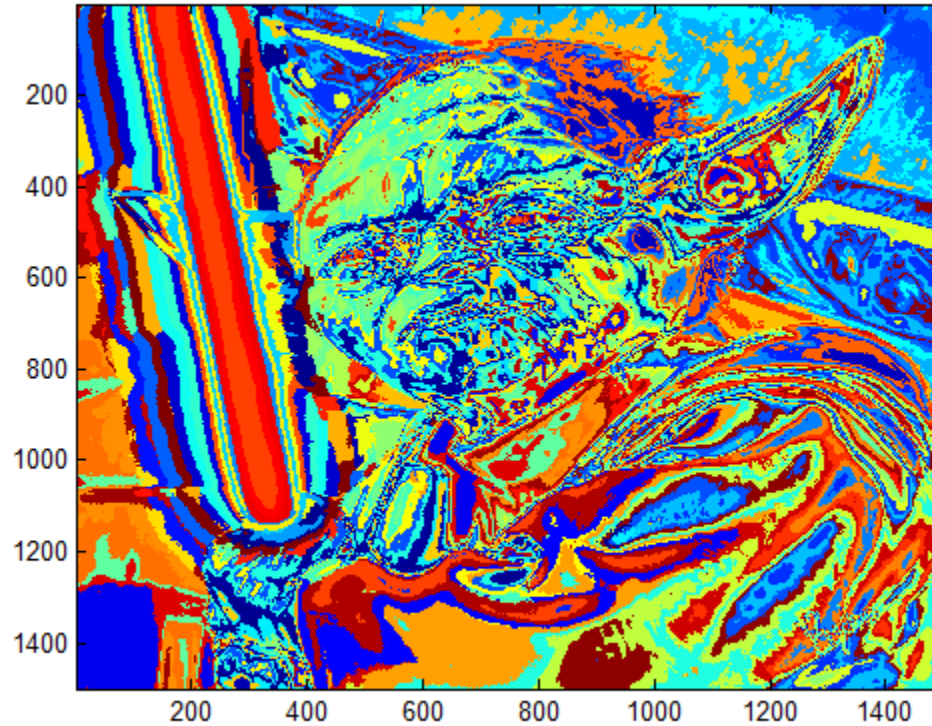
## 50 klastrów





# Przykłady

## 70 klastrów



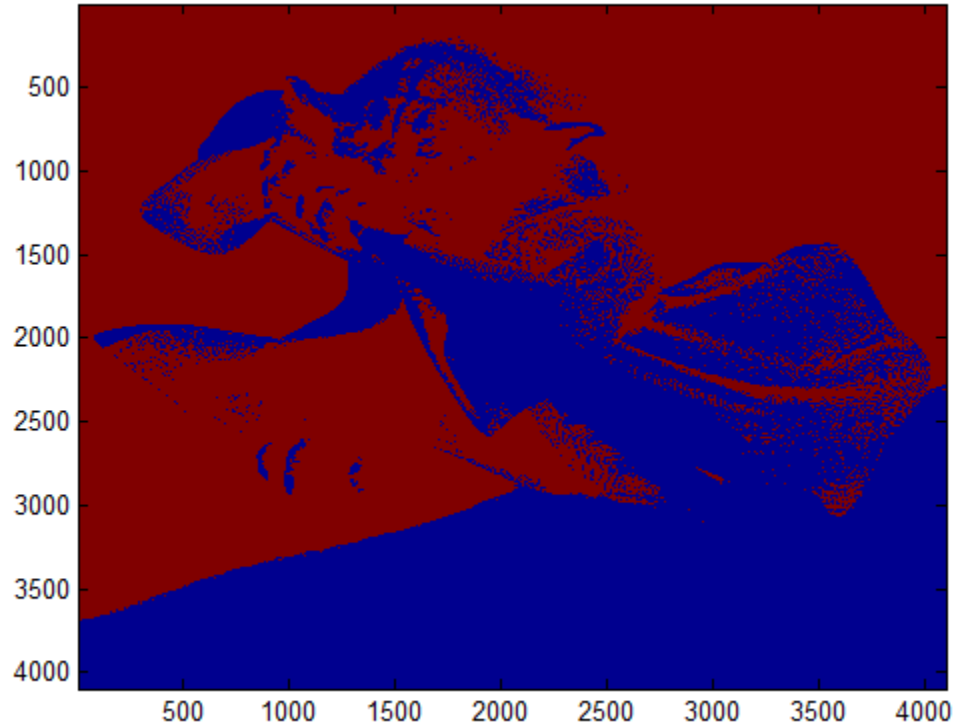
## Przykłady

**Rozmiar:  
4096x4096 !!!**



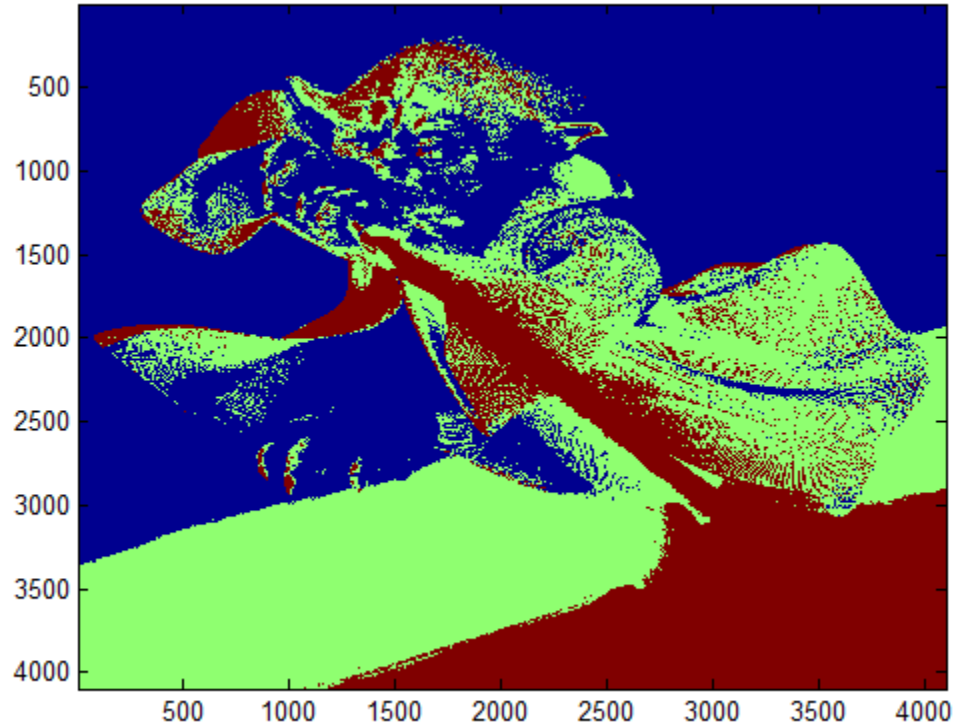
# Przykłady

## 2 klastry



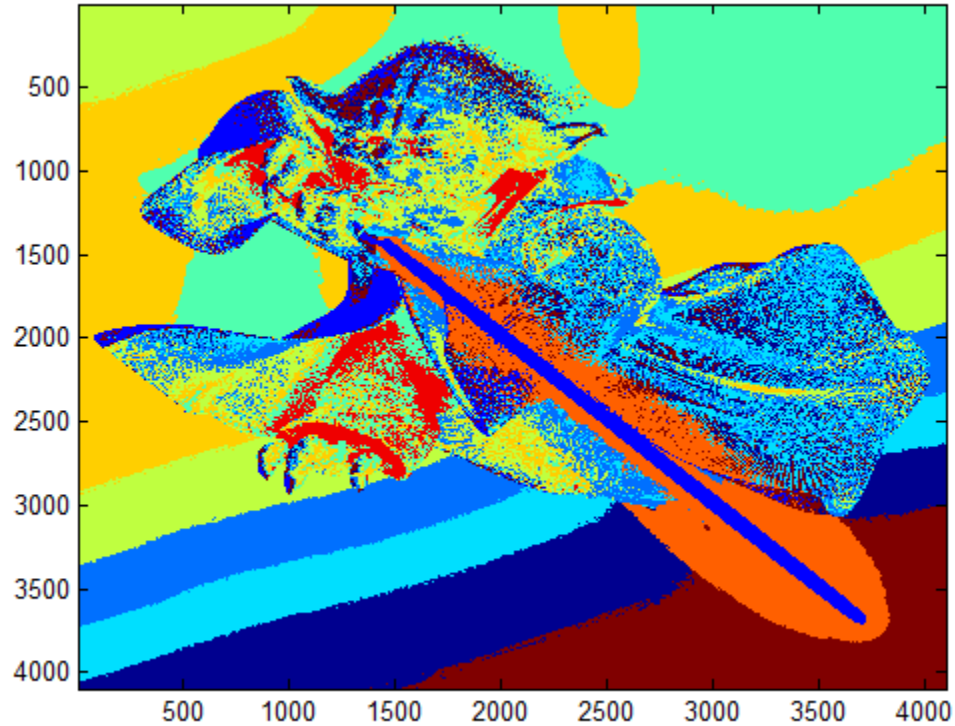
# Przykłady

## 3 klastry



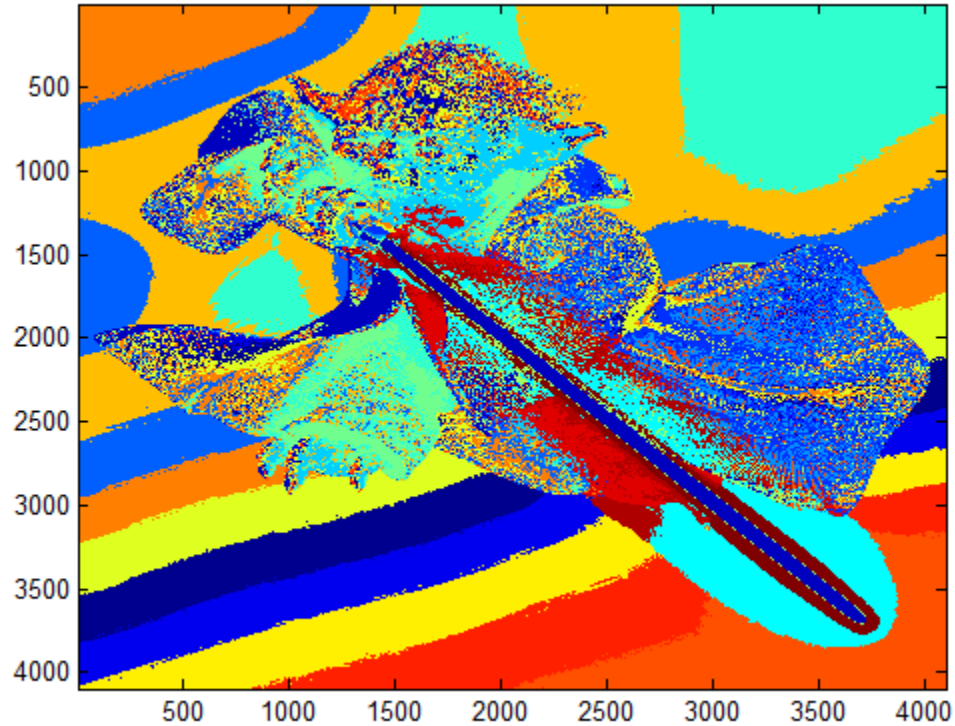
# Przykłady

## 10 klastrów



# Przykłady

## 20 klastrów



# Dziękuję za uwagę

