Analiza i przetwarzanie obrazów i wideo, wykład 13

Analiza wideo - śledzenie obiektów, predykcja i filtrowanie ruchu

J. Sas

AI-Tech, 2022/2023

Fundusze Europejskie Polska Cyfrowa
Rzeczpospolita Polska
Unia Europejska Europejski Fundusz Rozwoju Regionalnego

Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Rozwoju Regionalnego Program Operacyjny Polska Cyfrowa na lata 2014-2020,
Oś Priorytetowa nr 3 "Cyfrowe kompetencje społeczeństwa" Działanie nr 3.2 "Innowacyjne rozwiązania na rzecz aktywizacji cyfrowej"

Agenda:

- Metoda meanshift w analizie wideo
- Śledzenie obiektów w wideo z metodą meanshift
- Filtrowanie i predykcja ruchu w wideo
- Proste metody predykcji ruchu: filtry alfa-beta, alfa-beta-gamma
- Filtr Kalmana: intuicja i podstawy formalne

Segmentacja metodą *meanshift*

- **Idea**: klastry (segmenty) stanowią grupy podobnych (w przestrzeni cech) pikseli
- Atrybuty pikseli można traktować jako realizacje zmiennej losowej o skomplikowanym rozkładzie
- Liczne grupy podobnych pikseli tworzą gęste skupiska w przestrzeni cech odpowiadające obszarom gdzie gęstość prawdopodobieństwa jest wysoka
- Wektor cech najlepiej reprezentujący klaster znajduje się w centrum skupiska - tam gdzie gęstość w przestrzeni cech jest najwyższa
- Utożsamimy słowa wizualne w lokalnych maksimach gęstości z reprezentantami segmentów
- Każdy piksel "przypisujemy" do bliskiego mu centrum klastra
- Piksele "przypisane" do centrum określonego klastra tworzą segment

Segmentacja *meanshift* - - problemy realizacji

- Jaką wybrać przestrzeń cech?
  - atrybuty przestrzeni barw,
  - cechy teksturowe (np. cechy Tamury - patrz: wcześniejsze wykłady)
  - histogramy z cech pikseli w otoczeniu,
  - współrzędne położenia,
  - ...
- Jak znajdować maksima gęstości?
  - nieparametryczna estymacja gęstości w każdym punkcie przestrzeni cech - nieefektywne w przypadku wielowymiarowych wektorów cech
- Jak przypisywać piksele do znalezionych maksimów jasności?
  - najbliższe maksimum (np. jak w k-means) - o ile jawnie znaleźliśmy lokalne maksima gęstości (wymagana estymacja pdf - problemy j/w)
  - *hill climbing*

Segmentacja *meanshift* - - problemy realizacji

- Jaką wybrać przestrzeń cech?
  - atrybuty przestrzeni barw,
  - cechy teksturowe (np. cechy Tamury - patrz: wcześniejsze wykłady)
  - histogramy z cech pikseli w otoczeniu,
  - współrzędne położenia,
  - ...
- Jak znajdować maksima gęstości?
  - nieparametryczna estymacja gęstości w każdym punkcie przestrzeni cech - nieefektywne w przypadku wielowymiarowych wektorów cech
- Jak przypisywać piksele do znalezionych maksimów jasności?
  - najbliższe maksimum (np. jak w k-means) - o ile jawnie znaleźliśmy lokalne maksima gęstości (wymagana estymacja pdf - problemy j/w)
  - *hill climbing*

Segmentacja *meanshift* - - problemy realizacji

- Jaką wybrać przestrzeń cech?
  - atrybuty przestrzeni barw,
  - cechy teksturowe (np. cechy Tamury - patrz: wcześniejsze wykłady)
  - histogramy z cech pikseli w otoczeniu,
  - współrzędne położenia,
  - ...
- Jak znajdować maksima gęstości?
  - nieparametryczna estymacja gęstości w każdym punkcie przestrzeni cech - nieefektywne w przypadku wielowymiarowych wektorów cech
- Jak przypisywać piksele do znalezionych maksimów jasności?
  - najbliższe maksimum (np. jak w k-means) - o ile jawnie znaleźliśmy lokalne maksima gęstości (wymagana estymacja pdf - problemy j/w)
  - *hill climbing*

Segmentacja *meanshift* – własności

- Zaproponowane w Fukunaga K., Hostetler L.D.: The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition. IEEE Transac on Information Theory, 1975
- Wykorzystuje klasteryzację wokół maksimów gęstości
- Nie wymaga jawnego estymowania pdf w każdym punkcie przestrzeni cech
- Nie wymaga jawnego znajdowania lokalnych maksimów gęstości
- Niebezpośrednio stosuje *hill climbing*
- ...

Segmentacja *meanshift* – własności

- ...
- Niewiele parametrów do apriorycznego ustawienia (r, h)
- Własności segmentacji można kształtować przez odpowiedni dobór cech pikseli
- Implementacja może być łatwo zrównoleglona (niezależne przetwarzanie każdego wektora cech występującego w obrazie)
- Nie udowodniono zbieżności do wyników metody z jawną estymacją pdf i znajdowaniem maksimów dla ogólnego przypadku, ale wyniki zadowalające w praktyce
- Wystarczająco efektywna do praktycznych zastosowań

Segmentacja *meanshift* – elementy metody

Hill climbing - podobieństwa:

- Startując z zadanego punktu w przestrzeni cech chcemy dotrzeć do sąsiedniego (tzn "położonego blisko") maksimum lokalnego
- Gdyby była dostępna estymowana gęstość prawdopodobieństw wyliczona w każdym punkcie zdyskredytowanej przestrzeni cech:
  - iteracyjnie przesuwany się do sąsiednich pikseli wybierając w każdym kroku sąsiada o największej wartości pdf
  - ...dopóki nie osiągniemy piksela nie mającego sąsiadów o większej pdf - osiągnięto maksimum lokalne
- Przy braku estymaty pdf - losowo wybieramy sąsiednie punkty x_i, $i = 1, ..., N$ i przesuwamy się do tego z nich x_{i*}, dla którego $f(x_{i*})$ maksymalne

Segmentacja *meanshift* – elementy metody

Hill climbing - podobieństwa:

- Startując z zadanego punktu w przestrzeni cech chcemy dotrzeć do sąsiedniego (tzn "położonego blisko") maksimum lokalnego
- Gdyby była dostępna estymowana gęstość prawdopodobieństw wyliczona w każdym punkcie zdyskredytowanej przestrzeni cech:
  - iteracyjnie przesuwany się do sąsiednich pikseli wybierając w każdym kroku sąsiada o największej wartości pdf
  - ...dopóki nie osiągniemy piksela nie mającego sąsiadów o większej pdf - osiągnięto maksimum lokalne
- Przy braku estymaty pdf - losowo wybieramy sąsiednie punkty x_i, $i = 1, ..., N$ i przesuwamy się do tego z nich x_{i^*}, dla którego $f(x_{i^*})$ maksymalne

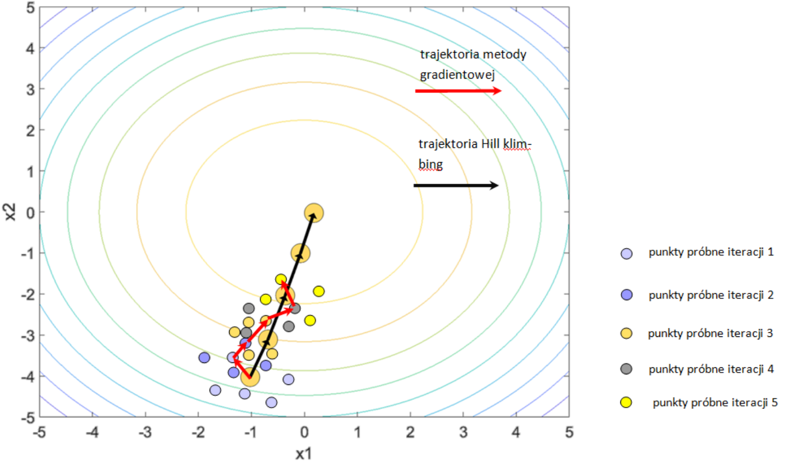
Segmentacja *meanshift* – elementy metody

Hill climbing - podobieństwa:

- Startując z zadanego punktu w przestrzeni cech chcemy dotrzeć do sąsiedniego (tzn "położonego blisko") maksimum lokalnego
- Gdyby była dostępna estymowana gęstość prawdopodobieństw wyliczona w każdym punkcie zdyskredytowanej przestrzeni cech:
  - iteracyjnie przesuwany się do sąsiednich pikseli wybierając w każdym kroku sąsiada o największej wartości pdf
  - ...dopóki nie osiągniemy piksela nie mającego sąsiadów o większej pdf - osiągnięto maksimum lokalne
- Przy braku estymaty pdf - losowo wybieramy sąsiednie punkty $x_i, i = 1, ..., N$ i przesuwamy się do tego z nich x_{i^*}, dla którego $f(x_{i^*})$ maksymalne

Segmentacja *meanshift* – elementy metody

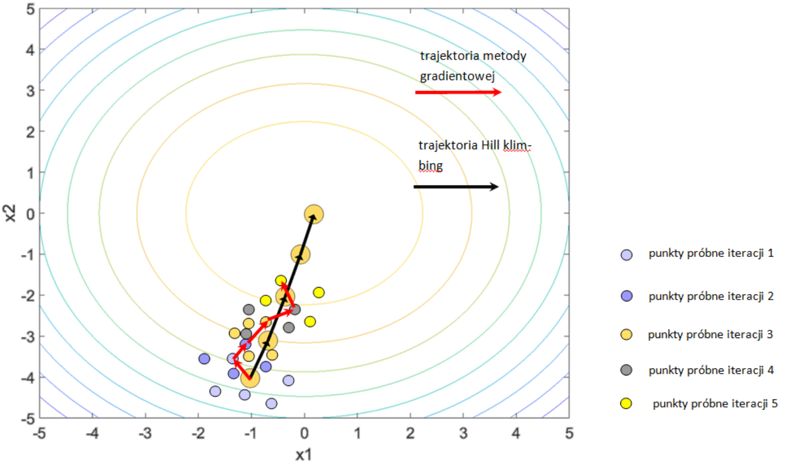
Rysunek 1: *Hill climbing vs. optymalizacja gradientowa*



Źródło: Opracowano na podstawie materiałów M. Przewoźniczka - za zgodą autora

Metoda *meanshift* stosuje technikę przesuwania w kierunku średniej

Rysunek 1: *Hill climbing vs. optymalizacja gradientowa*



Źródło: Opracowano na podstawie materiałów M. Przewoźniczka - za zgodą autora

Metoda *meanshift* stosuje technikę przesuwania w kierunku średniej

Segmentacja *meanshift* – elementy metody

Przesuwanie średniej:

- $p(x)$ - liczba pikseli w obrazie, których atrybuty/cechy są równe x
- $K_r(x)$ - średnia ważona punktów w otoczeniu punktu x o promieniu r

$$K_r(x) = \sum_{x':|x-x'|\leq r; p(x')>0} w(x,x')p(x'); \quad w(x,x') = exp\frac{|x-x'|}{h} \tag{1}$$

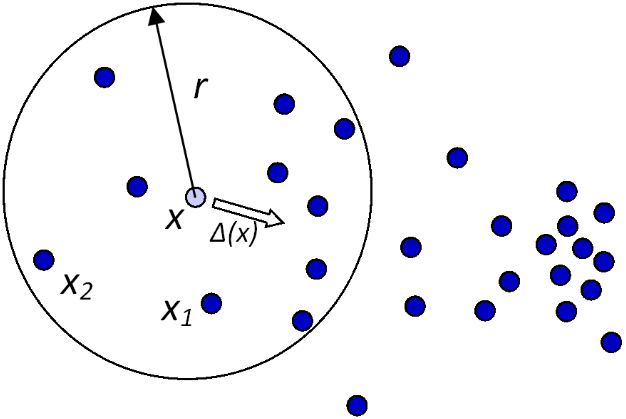
$w(x,x')$ - funkcja wagowa (jądro)

- $\Delta(x)$ - wektor przesunięcia punktu x

$$\Delta(x) = \frac{K_r(x)}{\sum_{x':|x-x'|\leq r; p(x')>0} w(x,x')} - x \tag{2}$$

Przesuwanie średniej:

Rysunek 2: *Ilustracja zasady przesuwania średniej - średnia przesunięta względem środka w kierunku obszaru o większej koncentracji wektorów cech*



Segmentacja *meanshift* – elementy metody

Przesuwanie średniej:

- Punkt x iteracyjnie przesuwany o wektor $\Delta(x_i)$ gdzie x_i pozycja osiągnięta w i-tym kroku; $x_i, i = 1, 2, \ldots$ ma tendencję do podążania do lokalnego maksimum gęstości
- Iteracyjne przesuwanie punktu kończy się gdy $|\Delta(x_i)| < \epsilon$
- Jeśli przestrzeń cech jest zdyskretyzowana, to wszystkie punkty w przestrzeni cech "podążają" do skończonej liczby punktów koncentracji
- Wszystkie piksele, których wektory cech "dotarły" do tego samego punktu koncentracji przypisywane są do tego samego segmentu

Śledzenie z wykorzystaniem techniki *meanshift*

Geneza:

- Oryginalna technika *meanshift* - opracowana jako metoda segmentacji (patrz: poprzednie wykłady dot. segmentacji)
- Technika ta adaptowana tutaj do znajdowania przesunięcia widoku obiektu pomiędzy sąsiednimi klatkami sekwencji wideo

Śledzenie z wykorzystaniem techniki *meanshift*

Geneza:

- Oryginalna technika *meanshift* - opracowana jako metoda segmentacji (patrz: poprzednie wykłady dot. segmentacji)
- Technika ta adaptowana tutaj do znajdowania przesunięcia widoku obiektu pomiędzy sąsiednimi klatkami sekwencji wideo

Założenia:

- niewielka zmienność cech wizualnych śledzonego widoku obiektu pomiędzy klatkami
- stosunkowo niewielkie przemieszczenia obiektu pomiędzy kolejnymi klatkami
- możliwość wyznaczenia konturu obiektu (lub jego przybliżenia np. prostokątem otaczającym) w pierwszej klatce - innymi metodami
- ...

Śledzenie z wykorzystaniem techniki *meanshift*

Założenia - c.d:

- ...
- Fragment obrazu pokrywające widok przemieszczającego się obiektu - opisany wektorem cech $F(O)$
  - histogram kolorów,
  - histogram gradientów,
  - cechy kształtu,
  - inne cechy teksturowe...
- Cechy wyliczone w obszarze obiektu różnią się znacząco od cech wyznaczonych tą samą metodą dla tła na którym porusza się obiekt - tzn obiekt wizualnie odróżnialny od tła
- Przyjmujemy miarę podobieństwa $S(F(O_1), F(O_2))$:
  - np. przy L1-znormalizowanych wektorach cech mogą to być miary podobieństwa rozkładów prawdopodobieństwa

Założenia - c.d:

- ...
- Fragment obrazu pokrywające widok przemieszczającego się obiektu - opisany wektorem cech $F(O)$
  - histogram kolorów,
  - histogram gradientów,
  - cechy kształtu,
  - inne cechy teksturowe...
- Cechy wyliczone w obszarze obiektu różnią się znacząco od cech wyznaczonych tą samą metodą dla tła na którym porusza się obiekt - tzn obiekt wizualnie odróżnialny od tła
- Przyjmujemy miarę podobieństwa $S(F(O_1), F(O_2))$:
  - np. przy L1-znormalizowanych wektorach cech mogą to być miary podobieństwa rozkładów prawdopodobieństwa

Założenia - c.d:

- ...
- Fragment obrazu pokrywające widok przemieszczającego się obiektu - opisany wektorem cech $F(O)$
  - histogram kolorów,
  - histogram gradientów,
  - cechy kształtu,
  - inne cechy teksturowe...
- Cechy wyliczone w obszarze obiektu różnią się znacząco od cech wyznaczonych tą samą metodą dla tła na którym porusza się obiekt - tzn obiekt wizualnie odróżnialny od tła
- Przyjmujemy miarę podobieństwa $S(F(O_1), F(O_2))$:
  - np. przy L1-znormalizowanych wektorach cech mogą to być miary podobieństwa rozkładów prawdopodobieństwa

Ogólna zasada:

- W klatce początkowej wyznaczamy obszar interesującego nas obiektu (zbiór pikseli) $O(t_0)$
- W obszarze O wyliczamy wartość wektora cech poruszającego się obiektu $F(O(t_0))$,
- Niech $O(t_i)$ oznacza obszar obiektu ustalony w i-tej klatce t_i
- Przechodzimy do następnej klatki: t_{i+1}
- Przyjmujemy dopuszczalny zbiór wektorów przemieszczeń obiektu pomiędzy klatkami: V - mogą to być np. wektory o składowych całkowitoliczbowych i ograniczonej długości:

$$V = \{(d_x, d_y) : d_x, d_y \in \{-d_{max}, d_{max}\}\} \tag{3}$$

- Niech $O_{\rightarrow v}(t_{i+1})$ oznacza obszar O z klatki t_i przesunięty o wektor v w klatce t_{i+1}
- Wyznaczamy dla klatki t_{i+1} pole podobieństwa (**UWAGA! uproszczenie w stosunku do oryginalnej metody**):

Ogólna zasada:

- ...
- Wyznaczamy dla klatki t_{i+1} pole podobieństwa:

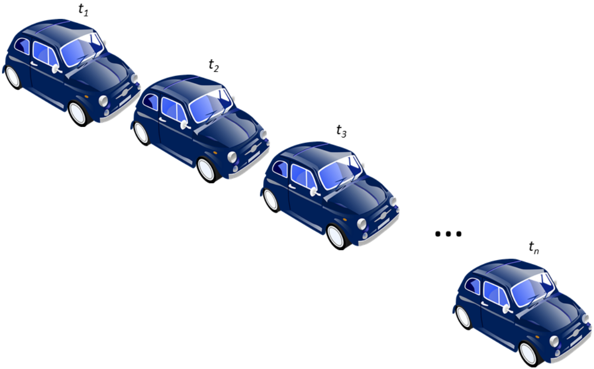
$$P(x, y) = S(F(O(t_i), F(O_{\rightarrow(x,y)}(t_{i+1}))) \tag{4}$$

- Poszukujemy maksimum w polu podobieństwa, niech będzie ono na pozycji (x^*, y^*)
- Przesuwamy pozycję obszaru $O(t_i)$ o wektor (x^*, y^*) i przyjmujemy, że jest to nowa pozycja śledzonego obiektu:

$$O(t_{i+1}) = O(t_i)_{\rightarrow(x^*,y^*)} \tag{5}$$

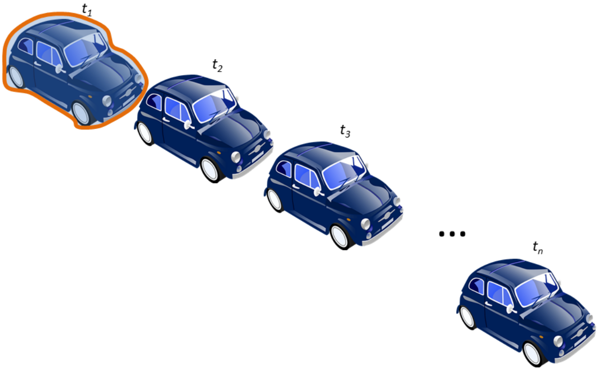
- Znajdujemy pozycję obiektu w kolejnych klatkach powtarzając iteracyjnie powyższe czynności

Śledzenie *meanshift* - przykład



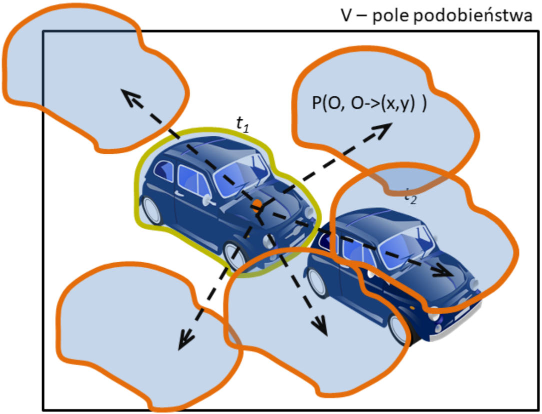
Analizujemy sekwencję klatek z poruszającym się obiektem

Śledzenie *meanshift* - przykład

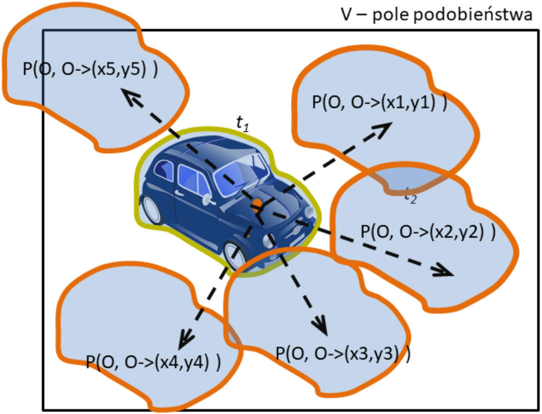


W początkowej klatce znajdujemy kontur obiektu do śledzenia

Śledzenie *meanshift* - przykład

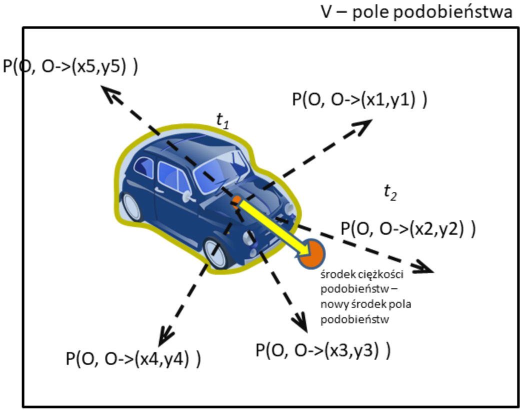


Wyznaczamy losowo pozycje w polu podobieństwa



Wyliczamy podobieństwa w ustalonych pozycjach

Śledzenie *meanshift* - przykład



- Przenosimy centrum pola podobieństwa do wyznaczonego środka ciężkości
- Ponawiamy iterację aż przesunięcie $< \epsilon$

Śledzenie z wykorzystaniem techniki *meanshift*

Technika znajdowania przesunięcia:

- Wyznaczanie maksimum globalnego w polu podobieństwa nie zawsze daje dobre rezultaty:
  - odlegle maksimum globalne może odpowiadać innemu podobnemu obiektowi położonemu w sąsiedztwie śledzonego,
  - może występować wiele lokalnych maksimów zbliżonych do globalnego
- W praktyce dobre rezultaty osiąga się znajdując "bliskie" (tzn, odpowiadające niewielkiemu przesunięciu) maksimum lokalne
- Do znalezienia maksimum lokalnego stosujemy metodę *meanshift*:
  - otaczamy aktualną pozycję (x, y) w polu podobieństwa kołem o promieniu r i wyliczamy środek ciężkości podobieństw $P(x', y')$ dla (x', y') w obszarze kołą r
  - przesuwamy pozycję (x, y) do wyznaczonego środka ciężkości
  - kontynuujemy dopóki wyliczony środek ciężkości jest odległy o więcej niż ϵ od środka koła
- Uzyskaną metodą meanshift pozycję uznajemy za wektor przesunięcia obiektu pomiędzy klatkami

Śledzenie *meanshift* - rozszerzenia

- Założenie o niezmienności (w szczególności niezmienności rozmiarów) śledzonego obiektu jest ograniczające - np. w sytuacji gdy obiekt przybliża się lub oddala względem obserwatora
- **Rozwiązanie** - zastosowanie dodatkowego wymiaru w polu podobieństwa odpowiadającego skali
  - rozważamy współczynniki skali:
    - $(s < 1)$ - pomniejszanie/oddalanie obiektu
    - $(s > 1)$ - powiększanie/przybliżanie obiektu
  - wykonujemy *meanshift* w przestrzenie trójwymiarowego pola podobieństwa (x, y, s)
  - zamiast wyliczać środek ciężkości w obszarze koła wyliczamy go w 3D w obszarze kuli o promieniu r
  - osiagnięty przez *meanshift* punkt wyznacza przesunięcie środka obszaru obiektu i zmianę jego skali
  - pozostała część algorytmu śledzenia - bez zmian
- Analogiczna technika dla uwzględnienia innych transformacji obiektu (obrotów, skalowania niejednorodnego) - ale zwiększanie wymiaru przestrzeni podobieństwa znacząco zwiększa nakłady obliczeniowe

Śledzenie *meanshift* - rozszerzenia

- Założenie o niezmienności (w szczególności niezmienności rozmiarów) śledzonego obiektu jest ograniczające - np. w sytuacji gdy obiekt przybliża się lub oddala względem obserwatora
- **Rozwiązanie** - zastosowanie dodatkowego wymiaru w polu podobieństwa odpowiadającego skali
  - rozważamy współczynniki skali:
    - $(s < 1)$ - pomniejszanie/oddalanie obiektu
    - $(s > 1)$ - powiększanie/przybliżanie obiektu
  - wykonujemy *meanshift* w przestrzenie trójwymiarowego pola podobieństwa (x, y, s)
  - zamiast wyliczać środek ciężkości w obszarze koła wyliczamy go w 3D w obszarze kuli o promieniu r
  - osiagnięty przez *meanshift* punkt wyznacza przesunięcie środka obszaru obiektu i zmianę jego skali
  - pozostała część algorytmu śledzenia - bez zmian
- Analogiczna technika dla uwzględnienia innych transformacji obiektu (obrotów, skalowania niejednorodnego) - ale zwiększanie wymiaru przestrzeni podobieństwa znacząco zwiększa nakłady obliczeniowe

Śledzenie *meanshift* - rozszerzenia

- Założenie o niezmienności (w szczególności niezmienności rozmiarów) śledzonego obiektu jest ograniczające - np. w sytuacji gdy obiekt przybliża się lub oddala względem obserwatora
- **Rozwiązanie** - zastosowanie dodatkowego wymiaru w polu podobieństwa odpowiadającego skali
  - rozważamy współczynniki skali:
    - $(s < 1)$ - pomniejszanie/oddalanie obiektu
    - $(s > 1)$ - powiększanie/przybliżanie obiektu
  - wykonujemy *meanshift* w przestrzenie trójwymiarowego pola podobieństwa (x, y, s)
  - zamiast wyliczać środek ciężkości w obszarze koła wyliczamy go w 3D w obszarze kuli o promieniu r
  - osiagnięty przez *meanshift* punkt wyznacza przesunięcie środka obszaru obiektu i zmianę jego skali
  - pozostała część algorytmu śledzenia - bez zmian
- Analogiczna technika dla uwzględnienia innych transformacji obiektu (obrotów, skalowania niejednorodnego) - ale zwiększanie wymiaru przestrzeni podobieństwa znacząco zwiększa nakłady obliczeniowe

Śledzenie *meanshift* - rozszerzenia

- Założenie o niezmienności (w szczególności niezmienności rozmiarów) śledzonego obiektu jest ograniczające - np. w sytuacji gdy obiekt przybliża się lub oddala względem obserwatora
- **Rozwiązanie** - zastosowanie dodatkowego wymiaru w polu podobieństwa odpowiadającego skali
  - rozważamy współczynniki skali:
    - $(s < 1)$ - pomniejszanie/oddalanie obiektu
    - $(s > 1)$ - powiększanie/przybliżanie obiektu
  - wykonujemy *meanshift* w przestrzenie trójwymiarowego pola podobieństwa (x, y, s)
  - zamiast wyliczać środek ciężkości w obszarze koła wyliczamy go w 3D w obszarze kuli o promieniu r
  - osiagnięty przez *meanshift* punkt wyznacza przesunięcie środka obszaru obiektu i zmianę jego skali
  - pozostała część algorytmu śledzenia - bez zmian
- Analogiczna technika dla uwzględnienia innych transformacji obiektu (obrotów, skalowania niejednorodnego) - ale zwiększanie wymiaru przestrzeni podobieństwa znacząco zwiększa nakłady obliczeniowe

Śledzenie z wykorzystaniem technik głębokiego uczenia

Użyteczne techniki:

- **I** - sieci do segmentacji instancyjnej (ludzie, pojazdy, zwierzęta, inne obiekty poruszające się)
- **II** - sieci do oceny podobieństwa obrazów (np. wykorzytujące architekturę GAN)
- **III** - sieci uczone bezpośrednio do wyznaczania pola ruchu dla dwóch (lub większej liczby) sąsiadujących ze sobą klatek sekwencji animacji

Zarys metametody (jednej z wielu możliwych):

- Wykorzystując sieć typu I rozpoznaj w klatce początkowej obiekt zainteresowania i znajdź jego przybliżony kontur (usuń tło)
- Wykonaj dwa poprzednie kroki dla kolejnej klatki sekwencji - otrzymamy być może wiele segmentów zawierających obiekty zainteresowania - należy utożsamić segmenty pomiędzy klatkami
- Oceń podobieństwo pomiędzy segmentami znalezionymi a kolejnych klatkach - wykorzystaj przemieszczenia i sieci typu II
- Dołącz pozycję obiektu w klatce t_{i+i} utożsamionego z obiektem w klatce t_i do jego ścieżki ruchu

Co może pójść nie tak?

Co może pójść nie tak?

- detektor obiektów zwraca niedokładne położenia
- śledzony obiekt zmienia postać (reprezentację wizualną)
- śledzony obiekt zostaje tymczasowo przysłonięty
- śledzony obiekt znika z pola widzenia / pojawia się nowy obiekt

Predykcja ruchu: podstawy

Predykcja ruchu obiektu - próba ustalenia pozycji obiektu na następnych klatkach sygnału wideo na podstawie informacji i jego położeniu w poprzednich klatkach

- Maksymalne wykorzystanie posiadanej informacji o pozycji obiektu,
- Ograniczenie analizy przyszłych klatek tylko do pewnych ich fragmentów,
- Rozwiązanie problemów częściowego lub całościowego przesłonięcia śledzonych obiektów (problem interpolacji położenia obiektu).
- Metody predykcji ruchu obiektu obejmują zagadnienia:
    - **estymacji** parametrów ruchu,
    - **filtracji** pozyskanych próbek.

Predykcja ruchu: metody elementarne

Estymacja pozycji $\hat{x}_t$ obiektu w czasie t

$$\hat{x}_t = x_{t-1} + v_{t-1}. \tag{6}$$

Aktualizacja prędkości v_t obiektu w czasie t

$$v_t = x_t - x_{t-1}. \tag{7}$$

- Obserwujemy nieznaną pozycję obiektu (x_t) - obserwacja nie odpowiada rzeczywistej pozycji - jest obarczona błędem losowym,
- Estymacja pozycji na podstawie aktualnej prędkości.
- Pomiar prędkości na podstawie mierzonych pozycji.
- Metoda bardzo czuła na błędy pomiaru.
- Tylko estymacja, brak filtracji.

Predykcja ruchu: estymacja i filtracja

- Estymacja ruchu nie bierze pod uwagę błędów pomiaru.
- Połączenie metod **estymacji** i **filtracji**.
- Filtr *uczy* się trajektorii ruchu na podstawie obserwacji.
- Odporność filtra na błędy pomiarowe.
- Przykładowe filtry:
  - filtr *alfa–beta*,
  - filtr *alfa–beta–gamma*,
  - filtr Kalmana.

Predykcja ruchu: filtr *alfa–beta*

- Bardzo prosty rekurencyjny filtr.
- Model filtra - dwa parametry (zmienne stanu):
  - położenie obiektu $\hat{x}_t$,
  - prędkość obiektu $\hat{v}_t$.
- Prędkość obiektu jest:
  - **lokalnie stała**,
  - aktualizowana na podstawie błędów estymacji.
- Filtr ma dwa parametry:
  - $\alpha \in \langle 0, 1 \rangle$ – odpowiedzialny za aktualizację pozycji,
  - $\beta \in \langle 0, 1 \rangle$ – odpowiedzialny za aktualizację prędkości.
- Intuicyjnie - odpowiada zmieszaniu estymacji z poprzedniego kroku z wynikiem pomiaru

Predykcja ruchu: filtr *alfa–beta*

Estymacja parametrów filtra

$$\hat{x}_t = \hat{x}_{t-1} + \hat{v}_{t-1}, \quad \hat{v}_t = \hat{v}_{t-1}.$$

Wyznaczenie błędu estymacji pozycji

$$\hat{r}_t = x_t - \hat{x}_t$$

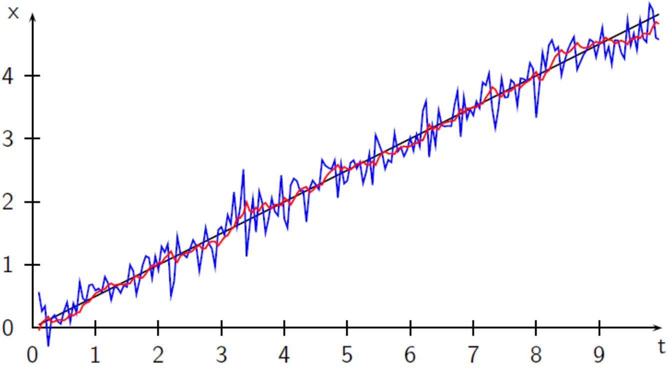
Aktualizacja parametrów filtra

$$\hat{x}_t = \hat{x}_t + \alpha \hat{r}_t, \quad \hat{v}_t = \hat{v}_t + \beta \hat{r}_t,$$

Sprowadza się do elementarnej estymacji przy parametrach:

- $\alpha = 1$,
- $\beta = 0$.

Filtr *alfa–beta* – ruch jednostajny

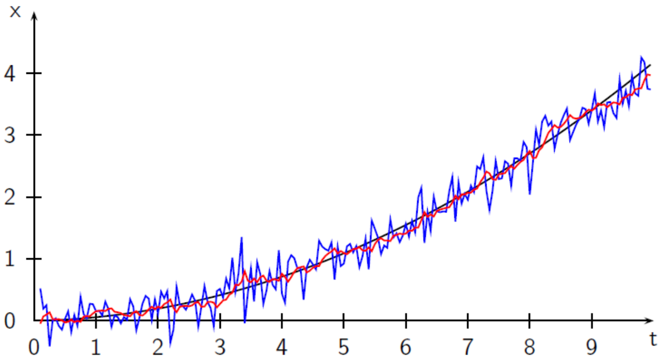


$$MAE(dane, pomiar) = 0,194 \tag{8}$$

$$MAE(dane, ab(pomiar)) = 0,070 \tag{9}$$

MAE - mean absolute error
dane - wartość rzeczywista (w praktyce nieznana)
ab(dane) - wynik zastosowania filtra do wyniku obserwacji

Filtr *alfa–beta* – ruch jednostajnie przyspieszony



$$MAE(dane, pomiar) = 0,194 \quad (10)$$

$$MAE(dane, ab(pomiar)) = 0,072 \quad (11)$$

Predykcja ruchu: filtr *alfa–beta–gamma*

- Rozszerzenie filtra *alfa–beta*
- Model filtra stanowią **trzy** parametry:
  - położenie obiektu $\hat{x}_t$,
  - prędkość obiektu $\hat{v}_t$,
  - przyspieszenie obiektu $\hat{a}_t$.
- Prędkość obiektu jest:
  - zmienna, zależna od przyspieszenia,
  - aktualizowana na podstawie błędów estymacji.
- Przyspieszenie obiektu jest:
  - **lokalnie stałe**,
  - aktualizowane na podstawie błędów estymacji.
- Filtr ma trzy parametry:
  - $\alpha \in \langle 0, 1 \rangle$ – odpowiedzialny za aktualizację pozycji,
  - $\beta \in \langle 0, 1 \rangle$ – odpowiedzialny za aktualizację prędkości.
  - $\gamma \in \langle 0, 1 \rangle$ – odpowiedzialny za aktualizację przyspieszenia.
- Możliwe jest budowanie filtrów wyższych rzędów.

Predykcja ruchu: filtr *alfa–beta–gamma*

Estymacja parametrów filtra

$$\hat{x}_t = \hat{x}_{t-1} + \hat{v}_{t-1} + \frac{1}{2}\hat{a}_{t-1}, \quad \hat{v}_t = \hat{v}_{t-1} + \hat{a}_{t-1}, \quad \hat{a}_t = \hat{a}_{t-1}.$$

Wyznaczenie błędu estymacji pozycji

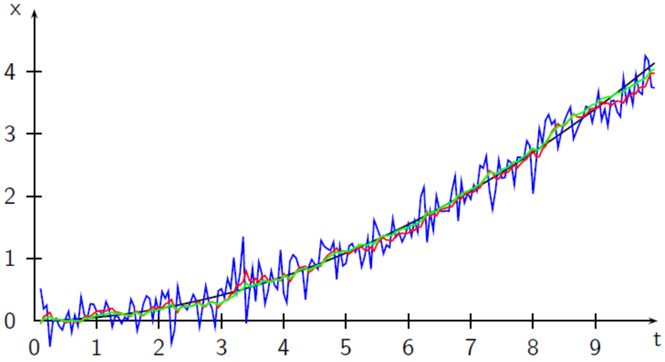
$$\hat{r}_t = x_t - \hat{x}_t.$$

Aktualizacja parametrów filtra

$$\hat{x}_t = \hat{x}_t + \alpha\hat{r}_t, \quad \hat{v}_t = \hat{v}_t + \beta\hat{r}_t, \quad \hat{a}_t = \hat{a}_t + \gamma\hat{r}_t,$$

- $1/2a_{t-1}$ - w pierwszym wzorze wynika z uwzględnienia prędkości średniej w przedziale $(t-1, t)$

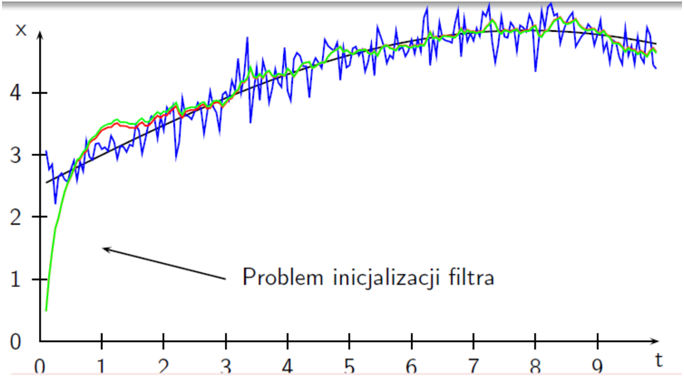
Filtr *alfa–beta–gamma* – ruch jednostajnie przyspieszony



$$MAE(dane, pomiar) = 0,194 \tag{12}$$

$$MAE(dane, abg(pomiar)) = 0,042 \tag{13}$$

Filtr *alfa–beta–gamma* – inna trajektoria (sin)



$$MAE(dane, pomiar) = 0,194 \tag{14}$$

$$MAE(dane, abg(pomiar)) = 0,132 \tag{15}$$

$$MAE(dane, ab(pomiar)) = 0,144 \tag{16}$$

Filtr Kalmana

- Zaproponowany przez Rudolfa E. Kalmana w 1960r. w obszarze analizy sygnałów
- Cel: **iteracyjne** wyznaczanie ewoluującego w czasie stanu obiektu na podstawie **nieprecyzyjnych (zaszumionych)** obserwacji

Filtr Kalmana

- Zaproponowany przez Rudolfa E. Kalmana w 1960r. w obszarze analizy sygnałów
- Cel: **iteracyjne** wyznaczanie ewoluującego w czasie stanu obiektu na podstawie **nieprecyzyjnych (zaszumionych)** obserwacji

Składniki:

- Obiekt opisany swym **wektorem stanu** (np. współrzędne pozycji, wektora prędkości i przyspieszenia dla poruszającego się punktu)

Filtr Kalmana

- Zaproponowany przez Rudolfa E. Kalmana w 1960r. w obszarze analizy sygnałów
- Cel: **iteracyjne** wyznaczanie ewoluującego w czasie stanu obiektu na podstawie **nieprecyzyjnych (zaszumionych)** obserwacji

Składniki:

- Obiekt opisany swym **wektorem stanu** (np. współrzędne pozycji, wektora prędkości i przyspieszenia dla poruszającego się punktu)
- Znany **model ewolucji** obiektu - równania pozwalające na wyliczenie przyszłego stanu obiektu na podstawie stanu obecnego
- Model opisany jest przekształceniem liniowym z dodatkowym zaburzeniem losowym (szum procesowy)

Filtr Kalmana

- Zaproponowany przez Rudolfa E. Kalmana w 1960r. w obszarze analizy sygnałów
- Cel: **iteracyjne** wyznaczanie ewoluującego w czasie stanu obiektu na podstawie **nieprecyzyjnych (zaszumionych)** obserwacji

Składniki:

- Obiekt opisany swym **wektorem stanu** (np. współrzędne pozycji, wektora prędkości i przyspieszenia dla poruszającego się punktu)
- Znany **model ewolucji** obiektu - równania pozwalające na wyliczenie przyszłego stanu obiektu na podstawie stanu obecnego
- Model opisany jest przekształceniem liniowym z dodatkowym zaburzeniem losowym (szum procesowy)
- Dysponujemy metodą **obserwacji** stanu obiektu w kolejnych chwilach - również zaburzonych losowo

Filtr Kalmana

- Obiekt opisany swym **wektorem stanu** (np. współrzędne pozycji, wektora prędkości i przyspieszenia dla poruszającego się punktu)
- Znany **model ewolucji** obiektu - równania pozwalające na wyliczenie przyszłego stanu obiektu na podstawie stanu obecnego
- Model opisany jest przekształceniem liniowym z dodatkowym zaburzeniem losowym (szum procesowy)
- Dysponujemy metodą **obserwacji** stanu obiektu w kolejnych chwilach - również zaburzonych losowo (tzn. obserwacje obiektu będącego w takim samym stanie w dwóch różnych chwilach czasowych nie muszą być sobie równe)

Filtr Kalmana

- Obiekt opisany swym **wektorem stanu** (np. współrzędne pozycji, wektora prędkości i przyspieszenia dla poruszającego się punktu)
- Znany **model ewolucji** obiektu - równania pozwalające na wyliczenie przyszłego stanu obiektu na podstawie stanu obecnego
- Model opisany jest przekształceniem liniowym z dodatkowym zaburzeniem losowym (szum procesowy)
- Dysponujemy metodą **obserwacji** stanu obiektu w kolejnych chwilach - również zaburzonych losowo (tzn. obserwacje obiektu będącego w takim samym stanie w dwóch różnych chwilach czasowych nie muszą być sobie równe)
- *PS: Obserwacje nie muszą bezpośrednio dotyczyć zmiennych stanu obiektu, ale są z nimi związane przekształceniem liniowym*

Filtr Kalmana

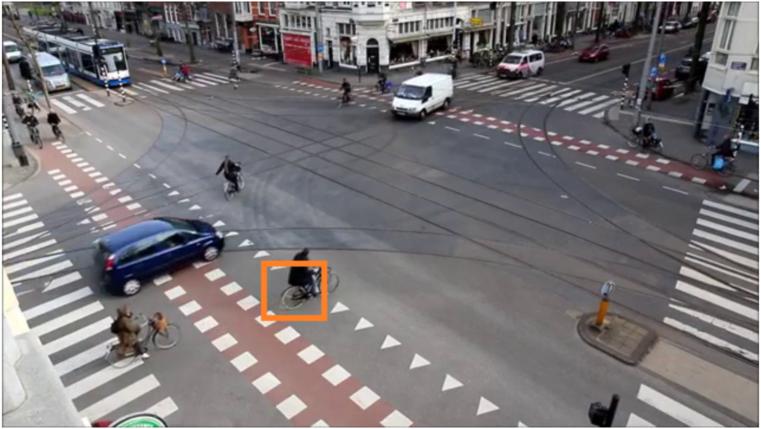
- Obiekt opisany swym **wektorem stanu** (np. współrzędne pozycji, wektora prędkości i przyspieszenia dla poruszającego się punktu)
- Znany **model ewolucji** obiektu - równania pozwalające na wyliczenie przyszłego stanu obiektu na podstawie stanu obecnego
- Model opisany jest przekształceniem liniowym z dodatkowym zaburzeniem losowym (szum procesowy)
- Dysponujemy metodą **obserwacji** stanu obiektu w kolejnych chwilach - również zaburzonych losowo (tzn. obserwacje obiektu będącego w takim samym stanie w dwóch różnych chwilach czasowych nie muszą być sobie równe)
- *PS: Obserwacje nie muszą bezpośrednio dotyczyć zmiennych stanu obiektu, ale są z nimi związane przekształceniem liniowym*
- Filtr Kalmana pozwala estymować nieznany stan obiektu w sposób **optymalny**, tzn. tak aby zminimalizować wartość oczekiwaną błędu średniokwadratowego pomiędzy stanem faktycznym a estymowanym

Rysunek 3: ***Wykryte*** *położenie obiektu w chwili* t_1

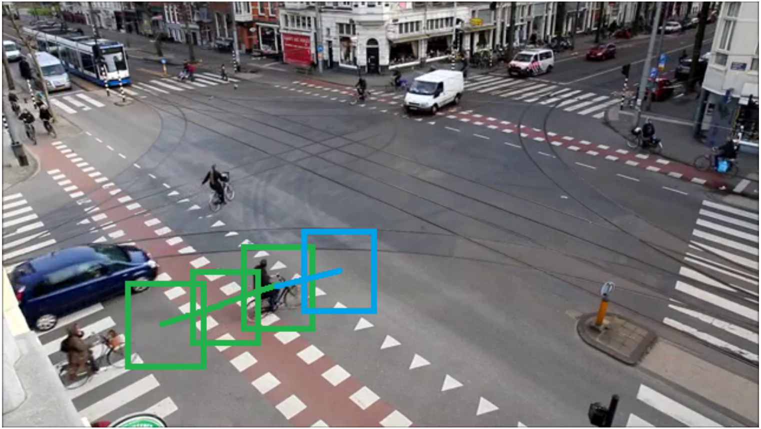
Filtr Kalmana - koncepcja

Rysunek 4: ***Wykryte*** *położenie obiektu w chwili* t_2

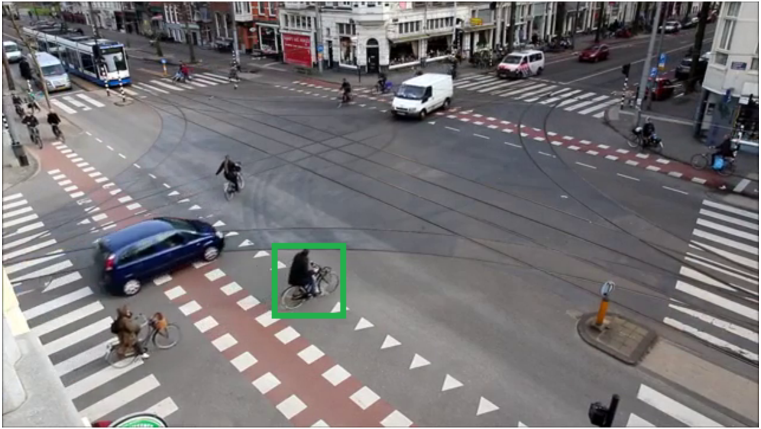
Rysunek 5: ***Wykryte*** *położenie obiektu w chwili* t_3



Rysunek 6: ***Wykryte*** *położenie obiektu w chwili t_4 - niedokładność detektora*



Rysunek 7: ***Predykcja*** *położenia obiektu w chwili* t_4



Rysunek 8: ***Faktyczne*** *położenie obiektu w chwili t_4 – nieznane!*

Filtr Kalmana - koncepcja

W co wierzyć?

Skoro błędna może być zarówno predykcja następnego położenia obiektu, jak i obserwacja tego położenia?

Wprowadźmy oznaczenia:

- x_k – faktyczny stan obiektu w chwili t_k – nieznany!
- z_k – wektor obserwacji pozyskanej w chwili t_k
- $x_{k+1} = Ax_k + w_k; w_k \sim N(0, s_k^W)$ – model procesu
- $z_k = Hx_k + v_k; v_k \sim N(0, s_k^V)$ – związek obserwacji ze stanem
- x_k^- – estymowany stan obiektu w chwili t_k przed uwzględnieniem obserwacji
- x_k^+ – estymowany stan obiektu po uwzględnieniu obserwacji

Zawsze zakładamy, że błąd procesowy w_k i błąd obserwacji v_k są nieskorelowane.

Filtr Kalmana - koncepcja

W co wierzyć?

Skoro błędna może być zarówno predykcja następnego położenia obiektu, jak i obserwacja tego położenia?

Wprowadźmy oznaczenia:

- x_k – faktyczny stan obiektu w chwili t_k – nieznany!
- z_k – wektor obserwacji pozyskanej w chwili t_k
- $x_{k+1} = Ax_k + w_k$; $w_k \sim N(0, s_k^W)$ – model procesu
- $z_k = Hx_k + v_k$; $v_k \sim N(0, s_k^V)$ – związek obserwacji ze stanem
- x_k^- – estymowany stan obiektu w chwili t_k przed uwzględnieniem obserwacji
- x_k^+ – estymowany stan obiektu po uwzględnieniu obserwacji

Zawsze zakładamy, że błąd procesowy w_k i błąd obserwacji v_k są nieskorelowane.

Filtr Kalmana - koncepcja

W co wierzyć?

Skoro błędna może być zarówno predykcja następnego położenia obiektu, jak i obserwacja tego położenia?

Wprowadźmy oznaczenia:

- x_k – faktyczny stan obiektu w chwili t_k – nieznany!
- z_k – wektor obserwacji pozyskanej w chwili t_k
- $x_{k+1} = Ax_k + w_k$; $w_k \sim N(0, s_k^W)$ – model procesu
- $z_k = Hx_k + v_k$; $v_k \sim N(0, s_k^V)$ – związek obserwacji ze stanem
- x_k^- – estymowany stan obiektu w chwili t_k przed uwzględnieniem obserwacji
- x_k^+ – estymowany stan obiektu po uwzględnieniu obserwacji

Zawsze zakładamy, że błąd procesowy w_k i błąd obserwacji v_k są nieskorelowane.

Filtr Kalmana - koncepcja

W co wierzyć?

Skoro błędna może być zarówno predykcja następnego położenia obiektu, jak i obserwacja tego położenia?

Wprowadźmy oznaczenia:

- x_k – faktyczny stan obiektu w chwili t_k – nieznany!
- z_k – wektor obserwacji pozyskanej w chwili t_k
- $x_{k+1} = Ax_k + w_k; w_k \sim N(0, s_k^W)$ – model procesu
- $z_k = Hx_k + v_k; v_k \sim N(0, s_k^V)$ – związek obserwacji ze stanem
- x_k^- – estymowany stan obiektu w chwili t_k przed uwzględnieniem obserwacji
- x_k^+ – estymowany stan obiektu po uwzględnieniu obserwacji

Zawsze zakładamy, że błąd procesowy w_k i błąd obserwacji v_k są nieskorelowane.

Filtr Kalmana - koncepcja

W co wierzyć?

Skoro błędna może być zarówno predykcja następnego położenia obiektu, jak i obserwacja tego położenia?

Wprowadźmy oznaczenia:

- x_k – faktyczny stan obiektu w chwili t_k – nieznany!
- z_k – wektor obserwacji pozyskanej w chwili t_k
- $x_{k+1} = Ax_k + w_k; w_k \sim N(0, s_k^W)$ – model procesu
- $z_k = Hx_k + v_k; v_k \sim N(0, s_k^V)$ – związek obserwacji ze stanem
- x_k^- – estymowany stan obiektu w chwili t_k przed uwzględnieniem obserwacji
- x_k^+ – estymowany stan obiektu po uwzględnieniu obserwacji

Zawsze zakładamy, że błąd procesowy w_k i błąd obserwacji v_k są nieskorelowane.

Filtr Kalmana - koncepcja

W co wierzyć?

Skoro błędna może być zarówno predykcja następnego położenia obiektu, jak i obserwacja tego położenia?

Wprowadźmy oznaczenia:

- x_k – faktyczny stan obiektu w chwili t_k – nieznany!
- z_k – wektor obserwacji pozyskanej w chwili t_k
- $x_{k+1} = Ax_k + w_k; w_k \sim N(0, s_k^W)$ – model procesu
- $z_k = Hx_k + v_k; v_k \sim N(0, s_k^V)$ – związek obserwacji ze stanem
- x_k^- – estymowany stan obiektu w chwili t_k przed uwzględnieniem obserwacji
- x_k^+ – estymowany stan obiektu po uwzględnieniu obserwacji

Zawsze zakładamy, że błąd procesowy w_k i błąd obserwacji v_k są nieskorelowane.

Filtr Kalmana - koncepcja

W co wierzyć?

Skoro błędna może być zarówno predykcja następnego położenia obiektu, jak i obserwacja tego położenia?

Wprowadźmy oznaczenia:

- x_k – faktyczny stan obiektu w chwili t_k – nieznany!
- z_k – wektor obserwacji pozyskanej w chwili t_k
- $x_{k+1} = Ax_k + w_k; w_k \sim N(0, s_k^W)$ – model procesu
- $z_k = Hx_k + v_k; v_k \sim N(0, s_k^V)$ – związek obserwacji ze stanem
- x_k^- – estymowany stan obiektu w chwili t_k przed uwzględnieniem obserwacji
- x_k^+ – estymowany stan obiektu po uwzględnieniu obserwacji

Zawsze zakładamy, że błąd procesowy w_k i błąd obserwacji v_k są nieskorelowane.

Filtr Kalmana - koncepcja

W co wierzyć?

Skoro błędna może być zarówno predykcja następnego położenia obiektu, jak i obserwacja tego położenia?

Wprowadźmy oznaczenia:

- x_k – faktyczny stan obiektu w chwili t_k – nieznany!
- z_k – wektor obserwacji pozyskanej w chwili t_k
- $x_{k+1} = Ax_k + w_k; w_k \sim N(0, s_k^W)$ – model procesu
- $z_k = Hx_k + v_k; v_k \sim N(0, s_k^V)$ – związek obserwacji ze stanem
- x_k^- – estymowany stan obiektu w chwili t_k przed uwzględnieniem obserwacji
- x_k^+ – estymowany stan obiektu po uwzględnieniu obserwacji

Zawsze zakładamy, że błąd procesowy w_k i błąd obserwacji v_k są nieskorelowane.

Filtr Kalmana - przypadek uproszczony

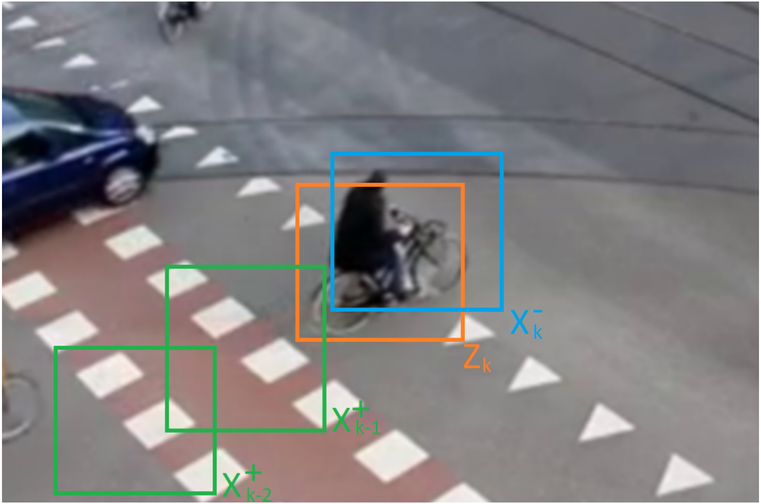
Oznaczenia:

- x_k – faktyczny stan obiektu w chwili t_k – nieznany!
- z_k – wektor obserwacji pozyskanej w chwili t_k
- $x_{k+1} = Ax_k + w_k; w_k \sim N(0, s_k^W)$ – model procesu
- $z_k = Hx_k + v_k; v_k \sim N(0, s_k^V)$ – związek obserwacji ze stanem
- x_k^- – estymowany stan obiektu w chwili t_k przed uwzględnieniem obserwacji
- x_k^+ – estymowany stan obiektu po uwzględnieniu obserwacji

Zawsze zakładamy, że błąd procesowy w_k i błąd obserwacji v_k są nieskorelowane.

Dla uproszczenia (chwilowo) założymy, że:

- składowe wektora błędu procesowego i składowe wektora błędu obserwacji są nieskorelowane - s_k^W, s_k^V sprowadzamy do postaci wektorów wariancji
- obserwujemy **bezpośrednio** (ale z błędem v_k) składowe wektora stanu - macierz H jest jednostkowa



Rysunek 9: *Framework: na podstawie estymacji historycznych (x_{k-1}^+, x_{k-2}^+) wyznaczamy predykcję x_k^-, a równolegle dokonujemy pomiaru z_k*

Filtr Kalmana - przypadek uproszczony

- Przy upraszczającym założeniu, że obserwujemy (z błędem losowym) wektor stanu uwzględnienie obserwacji - zmieszanie w optymalnych proporcjach estymacji z modelu $x_k^- = Ax_{k-1}^+$ z obserwacjami z_{k+1}:

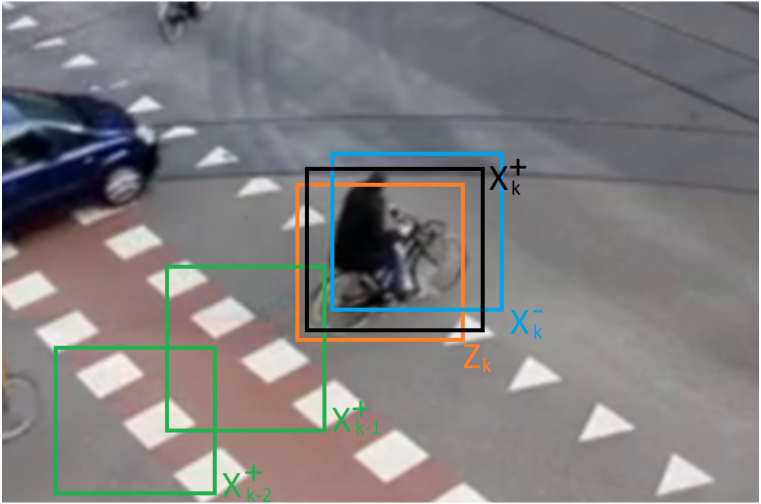
$$x_k^+ = (1 - K_k)x_k^- + K_k z_k \tag{17}$$

$$x_k^+ = x_k^- + K_k(z_k - x_k^-) \tag{18}$$

gdzie:

  - x_k^- – estymacja apriori (na podstawie poprzedniej estymacji i modelu)
  - x_k^+ – estymacja aposteriori (po uwzględnieniu obserwacji)
  - K_k – **wzmocnienie Kalmana** (*Kalman gain*), odpowiada za proporcję, z jaką łączymy estymacje i obserwacje

Filtr Kalmana - przypadek uproszczony



Rysunek 10: *Z odpowiednio ważonego połączenia estymacji i obserwacji uzyskujemy estymację a posteriori* x_k^+

Filtr Kalmana - przypadek uproszczony

- Przy upraszczającym założeniu, że obserwujemy (z błędem losowym) wektor stanu uwzględnienie obserwacji - zmieszanie w optymalnych proporcjach estymacji z modelu $x_k^- = Ax_{k-1}^+$ z obserwacjami z_{k+1}:

$$x_k^+ = (1 - K_k)x_k^- + K_k z_k x_k^+ = x_k^- + K_k(z_k - x_k^-) \tag{19}$$

gdzie:

- x_k^- – estymacja apriori (na podstawie poprzedniej estymacji i modelu)
- x_k^+ – estymacja aposteriori (po uwzględnieniu obserwacji)
- K_k – **wzmocnienie Kalmana** (*Kalman gain*), odpowiada za proporcję, z jaką łączymy estymacje i obserwacje

Problem: jak optymalnie wyznaczyć **wzmocnienie Kalmana** K_k

- **Problem:** jak optymalnie wyznaczyć **wzmocnienie Kalmana** K_k
- **Intuicyjnie** - uzależnić wagi w mieszaniu od wariancji obserwacji i wariancji estymatora z modelu:

$$s_k^- = E\{(x_k - x_k^-)^2\}, \quad s_k^z = E\{(x_k - z_k)^2\} = (s_k^V)^2 \tag{20}$$

  - większa wariancja - mniejsza waga
  - mniejsza wariancja - większa waga, np:

$$(y_1 : s_1), (y_2 : s_2) \rightarrow y = \frac{s_2}{s_1 + s_2} y_1 + \frac{s_1}{s_1 + s_2} y_2 \tag{21}$$

Filtr Kalmana - przypadek uproszczony

Zarys algorytmu:

Wyznacz x_0^-, s_0^-
for k = 0 to k=T **do**
    Odczytaj z_k
    $x_k^- = Ax_{k-1}^+$
    Wyznacz K_k
    $x_k^+ = x_k^- + K_k(z_k - x_k^-)$
    $x_{k+1}^- = x_k^+$
end for

Inicjalizacja:

- Niestosowne przyjęcie wartości początkowych może spowodować niedokładną estymację w początkowych krokach działania filtru
- $x_0^- = z_0$ obserwacja w pierwszym kroku (możliwe gdy bezpośrednio obserwujemy zmienne stanu zakłócone szumem pomiaru v_k)
- $x_0^- = E\{X\}$ - średnia wartość wektora stanu
- $s_0^- = D^2W$ - wariancja szumu procesowego

Filtr Kalmana - przypadek uproszczony

Zarys algorytmu:

```
Wyznacz x_0^-, s_0^-
for k = 0 to k=T do
    Odczytaj z_k
    x_k^- = A x_{k-1}^+
    Wyznacz K_k
    x_k^+ = x_k^- + K_k(z_k - x_k^-)
    x_{k+1}^- = x_k^+
end for
```

Inicjalizacja:

- Niestosowne przyjęcie wartości początkowych może spowodować niedokładną estymację w początkowych krokach działania filtru
- $x_0^- = z_0$ obserwacja w pierwszym kroku (możliwe gdy bezpośrednio obserwujemy zmienne stanu zakłócone szumem pomiaru v_k)
- $x_0^- = E\{X\}$ - średnia wartość wektora stanu
- $s_0^- = D^2 W$ - wariancja szumu procesowego

Filtr Kalmana - przypadek uproszczony

Wyznaczanie wzmocnienia Kalmana:

$$x_k^+ = x_k^- + K_k(z_k - x_k^-) \tag{22}$$

- Dążymy do zminimalizowania wartości oczekiwanej błędu średniokwadratowego:

$$s_k^+ = E\{(x_k - x_k^+)^2\} \rightarrow min \tag{23}$$

$$s_k^- = E\{(x_k - x_k^-)^2\} \tag{24}$$

$$E\{(x_k - x_k^+)^2\} = E\{[(x_k - x_k^-) - K_k(z_k - x_k^-)]^2\} \tag{25}$$

- Zastępujemy $z_k = x_k + v_k$

$$... = E\{[(x_k - x_k^-) - K_k(x_k + v_k - x_k^-)]^2\} \tag{26}$$

- Rozwijamy kwadrat dwumianu

$$... = E\{[(x_k - x_k^-)^2 - 2K_k(x_k - x_k^-)(x_k + v_k - x_k^-) + K_k^2(x_k + v_k - x_k^-)]^2\} \tag{27}$$

Filtr Kalmana - przypadek uproszczony

Wyznaczanie wzmocnienia Kalmana:

$$x_k^+ = x_k^- + K_k(z_k - x_k^-) \tag{22}$$

- Dążymy do zminimalizowania wartości oczekiwanej błędu średniokwadratowego:

$$s_k^+ = E\{(x_k - x_k^+)^2\} \to min \tag{23}$$

$$s_k^- = E\{(x_k - x_k^-)^2\} \tag{24}$$

$$E\{(x_k - x_k^+)^2\} = E\{[(x_k - x_k^-) - K_k(z_k - x_k^-)]^2\} \tag{25}$$

- Zastępujemy $z_k = x_k + v_k$

$$... = E\{[(x_k - x_k^-) - K_k(x_k + v_k - x_k^-)]^2\} \tag{26}$$

- Rozwijamy kwadrat dwumianu

$$... = E\{[(x_k - x_k^-)^2 - 2K_k(x_k - x_k^-)(x_k + v_k - x_k^-) + K_k^2(x_k + v_k - x_k^-)]^2\} \tag{27}$$

Wyznaczanie wzmocnienia Kalmana:

$$x_k^+ = x_k^- + K_k(z_k - x_k^-) \tag{22}$$

- Dążymy do zminimalizowania wartości oczekiwanej błędu średniokwadratowego:

$$s_k^+ = E\{(x_k - x_k^+)^2\} \rightarrow min \tag{23}$$

$$s_k^- = E\{(x_k - x_k^-)^2\} \tag{24}$$

$$E\{(x_k - x_k^+)^2\} = E\{[(x_k - x_k^-) - K_k(z_k - x_k^-)]^2\} \tag{25}$$

- Zastępujemy $z_k = x_k + v_k$

$$... = E\{[(x_k - x_k^-) - K_k(x_k + v_k - x_k^-)]^2\} \tag{26}$$

- Rozwijamy kwadrat dwumianu

$$... = E\{[(x_k - x_k^-)^2 - 2K_k(x_k - x_k^-)(x_k + v_k - x_k^-) + K_k^2(x_k + v_k - x_k^-)]^2\} \tag{27}$$

Filtr Kalmana - przypadek uproszczony

$$\ldots = E\{[(x_k - x_k^-)^2 - 2K_k(x_k - x_k^-)(x_k + v_k - x_k^-) + K_k^2(x_k + v_k - x_k^-)]^2\}$$

- Wartość oczekiwana sumy zmiennych losowych $==$ suma wartości oczekiwanych tych zmiennych:

$$\begin{aligned}\ldots = & E\{(x_k - x_k^-)^2\} + \\ & 2K_k E\{(x_k - x_k^-)(x_k + v_k - x_k^-)\} + \\ & K_k^2 E\{(x_k + v_k - x_k^-)^2\}\end{aligned}$$

$$\begin{aligned}\ldots = & E\{(x_k - x_k^-)^2\} - \\ & 2K_k E\{(x_k - x_k^-)v_k\} - 2K_k E\{(x_k - x_k^-)(x_k - x_k^-)\} + \\ & K_k^2(E\{(x_k - x_k^-)^2\} - 2E\{(x_k - x_k^-)v_k\} + E\{v_k^2\})\end{aligned}$$

- Ale v_k i $x_k - x_k^-$ niekorelowane więc $E\{(x_k - x_k^-)v_k = 0\}$

$$\ldots = s_k^- - 2K_k s_k^- + K_k^2 s_k^- + K_k^2 s_k^V \qquad (29)$$

Filtr Kalmana - przypadek uproszczony

$$\ldots = E\{[(x_k - x_k^-)^2 - 2K_k(x_k - x_k^-)(x_k + v_k - x_k^-) + K_k^2(x_k + v_k - x_k^-)]^2\}$$

- Wartość oczekiwana sumy zmiennych losowych == suma wartości oczekiwanych tych zmiennych:

$$\begin{aligned}\ldots =& E\{(x_k - x_k^-)^2\}+\\ & 2K_k E\{(x_k - x_k^-)(x_k + v_k - x_k^-)\}+\\ & K_k^2 E\{(x_k + v_k - x_k^-)^2\}\end{aligned}$$

$$\begin{aligned}\ldots =& E\{(x_k - x_k^-)^2\}-\\ & 2K_k E\{(x_k - x_k^-)v_k\} - 2K_k E\{(x_k - x_k^-)(x_k - x_k^-)\}+\\ & K_k^2(E\{(x_k - x_k^-)^2\} - 2E\{(x_k - x_k^-)v_k\} + E\{v_k^2\})\end{aligned}$$

- Ale v_k i $x_k - x_k^-$ niekorelowane więc $E\{(x_k - x_k^-)v_k = 0\}$

$$\ldots = s_k^- - 2K_k s_k^- + K_k^2 s_k^- + K_k^2 s_k^V \tag{29}$$

Filtr Kalmana - przypadek uproszczony

Ostatecznie wartość oczekiwana kwadratu błędu aposteriori:

$$E\{(x_k - x_k^+)^2\} = s_k^+ = s_k^- - 2K_k s_k^- + K_k^2 s_k^- + K_k^2 s_k^V \tag{30}$$

Poszukujemy takiego K_k, które zminimalizuje błąd w k-tym kroku:

$$\frac{\partial s_k^+}{\partial K_k} = -2s_k^- + 2K_k s_k^- + 2K_k s_k^V = 0 \tag{31}$$

Po oczywistych przekształceniach:

$$\frac{\partial s_k^+}{\partial K_k} = -2s_k^- + 2K_k s_k^- + 2K_k s_k^V = 0 \rightarrow K_k = \frac{s_k^-}{s_k^- + s_k^V} \tag{32}$$

Do wyznaczania K_k potrzebujemy:

- wariancji szumu pomiaru: $s_k^V = s^V = E\{v_k^2\}$ - zakładamy, że wynika z zastosowanego sposobu pomiaru i jest znana
- wartości oczekiwanej kwadratu błędu apriori s_k^- - musi być wyznaczania z kroku na krok

Filtr Kalmana - przypadek uproszczony

- Kwadrat błędu estymaty apriori może być efektywnie wyznaczany iteracyjnie.
- Dążymy do wyznaczenia:
  - s_k^+ na podstawie s_k^-
  - s_{k+1}^- na podstawie s_k^+

$$
\begin{aligned}
s_k^+ &= E\{[x_k^+ - x_k]^2\} && (33)\\
&= E\{[x_k^- + K_k(z_k - x_k^-) - x_k]^2\} && (34)\\
&= E\{[x_k^- + K_k(x_k + v_k - x_k^-) - x_k]^2\} && (35)\\
&= E\{[(1 - K_k)x_k^- + K_k(x_k + v_k) - x_k]^2\} && (36)\\
&= E\{[((1 - K_k)x_k^- - (1 - K_k)x_k + K_k v_k]^2\} && (37)\\
&= E\{[(1 - K_k)(x_k^- - x_k) + K_k v_k]^2\} && (38)
\end{aligned}
$$

Ponownie wykorzystujemy nieskorelowanie $(x_k^- - x_k)$ i v_k oraz własności wartości oczekiwanej zmiennych losowych:

$$\begin{aligned} s_k^+ &= E\{[(1-K_k)(x_k^- - x_k) + K_k v_k]^2\} && (39) \\ &= (1-K_k)^2 E\{(x_k^- - x_k)^2\} + K_k^2 E\{v_k^2\} && (40) \\ &= (1-K_k)^2 s_k^- + K_k^2 s^V && (41) \end{aligned}$$

Uwzględniając, że: $K_k = \frac{s_k^-}{s_k^- + s_k^V}$

$$\begin{aligned} s_k^+ &= (1-K_k)^2 s_k^- + K_k^2 s^V && (42) \\ &= (1 - \frac{s_k^-}{s_k^- + s^V})^2 s_k^- + \frac{(s_k^-)^2}{(s_k^- + s^V)^2} s^V + && (43) \\ &= \frac{(s^V)^2 s_k^-}{(s_k^- + s^V)^2} + \frac{(s_k^-)^2 s^V}{(s_k^- + s^V)^2} && (44) \\ &= \frac{s^V s_k^- (s_k^- + s^V)}{(s_k^- + s^V)^2} = \frac{s^V s_k^-}{s_k^- + s^V} = (1-K_k) s_k^- && (45) \end{aligned}$$

Podsumowanie - co należy zapamiętać:

- Śledzenie ruchu obiektów w sekwencjach wideo jest często wykorzystywane do rozpoznawania sytuacji i schematów zachowania obiektów
- Aby analizować trajektorię ruchu należy utożsamić segmenty obrazujące ten sam obiekt w kolejnych klatkach
- Adaptacja metody *meanshift* pozwala śledzić obiekt w sekwencji klatek
- W *meanshift* dokonujemy przesunięć próbnych obszaru konturu obiektu w polu podobieństwa
- Wyliczanie środka ciężkości pola podobieństwa pozwala uniknąć wpadania w fałszywe minima lokalne pola podobieństwa
- Metody filtrowania i predykcji ruchu pozwalają wygładzić trajektorię śledzonego obiektu
- W filtrowaniu i predykcji uwzględnia się parametry stanu opisane położeniem, prędkością i przyspieszeniem obiektu

Materiały dodatkowe

Meanshift

Wyjaśnienie metody meanshift z zastosowaniem w segmentacji obrazów: https://www.youtube.com/watch?v=PCNz_zttmtA

Filtr Kalmana

Szczegółowe wyjaśnienie filtru Kalmana: https://www.youtube.com/playlist?list=PLX2gX-ftPVXU3oUFNATxGXY90AULiqnWT
Gorąco zachęcam do obejrzenia przynajmniej pierwszych trzech lekcji!

Literatura uzupełniająca:

1. Sonka M., Hlavac V., Boyle R.: *Image Processing, Analysis, and Machine Vision, Fourth Edition*, Cengage Learning, 2015, (rozdz. 16.4-16.6)
2. Bradski G., Kaehler A.: *Learning OpenCV*, 2008, (rodz. 10 - Tracking and motion)
3. Forsyth D., Ponce j.: *Computer Vision - A Modern Approach*, 2012, (rozdz 11.3)
4. Kumar-Jatoth R., Gopisetty S.: *Performance Analysis of Alpha Beta Filter, Kalman Filter and Meanshift for Object Tracking in Video Sequences*, 2015
5. Saho K., Masugi M: *Performance analysis of alpha-beta-gamma tracking filters using position and velocity measurements*, EURASIP Journal on Advances in Signal Processing, 2015