

Analiza i przetwarzanie obrazów i wideo, wykład 12

Analiza sekwencji wideo. Separacja tła. Przepływ optyczny.

J. Sas

AI-Tech, 2022/2023

Agenda:

- Podstawowe problemy analizy wideo
- Metody wykrywania ruchu
- Pole ruchu i przepływ optyczny
- Geometryczne metody wykorzystania pola ruchu
- Wyznaczanie przepływu optycznego metodą Lucasa-Kanade

- Sygnał o charakterze trójwymiarowym (czasoprzestrzennym):
 - oś współrzędnej X ,
 - oś współrzędnej Y ,
 - **oś czasu T** .
- Oś czasu T ma charakter **dyskretny**, ale stanowi wycinek z ciągłej osi czasu.
- Ciągłość sygnału wideo w pojedynczym *ujęciu*.
 - możliwość analizy zmian pomiędzy kolejnymi chwilami czasu (ujęciami strumienia),
 - możliwość analizy dynamiki zmian,
 - możliwość predykcji zmian.

Pozyskanie/poszerzenie informacji o statycznych elementach: obrazu:

- usunięcie zmiennego w czasie szumu urządzenia akwizycyjnego (filtrowanie 3D),
- wyeliminowanie przemieszczających się elementów obrazu (odtworzenie statycznego tła sceny),
- wyszukanie najlepszego (w sekwencji obrazów) widoku interesującego nas elementu sceny;

Pozyskanie informacji o elementach ruchomych:

- wykrycie ruchu,
- separacja istotnych i nieistotnych elementów poruszających się,
- dynamiczne śledzenie ruchu obiektu (np. jako obszaru do ustawienia ostrości)
- określenie ścieżek ruchu (trajektorii poruszających się obiektów),
- rozpoznawanie specyficznych schematów występującego ruchu - rozpoznawanie akcji i zachowań.
- rozpoznawanie stanu obiektu na podstawie cech ruchu (np. stanu emocjonalnego osoby, dysfunkcji ruchowych, stanów chorobowych)

- Olbrzymia objętość danych (np. Full HD):
 - pojedyncza klatka : $1920 \times 1080 \times 3 = 6075kB$,
 - 24 klatek / sek : $6075kB \times 24 \frac{1}{s} = 145800kb/s \approx 142MB/s$,
 - 1 godzina : $142MB/s \times 3600s \approx 499GB$.
- Konieczność stosowania kompresji sygnału:
 - kompresja poszczególnych klatek obrazu,
 - kompresja ruchu (oś czasu).

Pojęcia podstawowe: cyfrowa reprezentacja sygnału

- Olbrzymia ilość metod i standardów kompresji sygnału wideo (na potrzeby transmisji i przechowywania):
 - Windows Media Video (zamkniętoźródłowy),
 - MPEG-4, część 2/H.263, DivX – przestarzałe,
 - AVC (MPEG-4, H.264) – nieco przestarzałe, ale nadal bardzo popularne,
 - HEVC (H.265) – *state of the art*,
 - VP9 (otwarty ale opracowywany przez Google),
 - ...
- Ważne: nie myl *kontenera* wideo (.mp4, .mkv, .avi, ...) z formatem kompresji (wyżej).
- Analiza na podstawie:
 - sekwencji obrazów odtworzonych z postaci skompresowanej,
 - cech wyliczanych bezpośrednio z danych skompresowanych.

- Analiza sygnału wideo to połączenie metod:
 - klasycznej analizy obrazów,
 - **analizy (wykrywania) ruchu,**
 - **predykcji ruchu,**
 - **rozpoznawania sekwencji.**
- Analiza i predykcja ruchu służą do:
 - polepszenia jakości wyników analizy (korekcja na podstawie poprzednich wyników, np. problem chwilowego zanikania śledzonego obiektu),
 - przyspieszenia działania metod (analiza tylko części danych zamiast pełnego strumienia).

- Analiza sygnału wideo zazwyczaj odbywa się **on-line**:
 - problem wydajności metod,
 - stosowanie bardzo wydajnych metod analizy obrazu,
 - stosowanie specjalizowanego sprzętu, zwykle o architekturze potokowej (np. z użyciem FPGA),
 - analiza tylko fragmentów obrazu, a nie całości (np. tylko poruszających się fragmentów),
 - wykorzystanie informacji o ruchu do określenia obszarów zainteresowania.

- Możliwe zachowania kamery:
 - całkowicie nieruchoma kamera,
 - kamera o bardzo niewielkich zmianach pozycji (np. wynikających z niestabilności jej umocowania),
 - kamera w ruchu.
- Możliwe zachowania tła ujęcia:
 - całkowicie statyczne tło,
 - nieruchome tło ze zmianami intensywności oświetlenia,
 - bardzo niewielki ruch tła,
 - dynamicznie zmieniające się tło.
- Dla każdego z przypadków stosowane są inne metody.

Wykrywanie ruchu: przykład



- Brak wykrycia ruchu na części obrazu:
 - *przyczyna*: nieskuteczny algorytm detekcji ruchu (odejmowania tła),
 - *przyczyna*: zbyt podobne analizowane fragmenty obrazu,
 - *skutek*: brak wykrycia istotnych poruszających się obiektów lub podział obiektu na części.
- Wykrycie nieistniejącego ruchu:
 - *przyczyna*: artefakty związane z rejestracją obrazu (szum),
 - *skutek*: wykrycie ruchu "nieistotnego" (zmiana niewielkich detali),
 - *przyczyna*: zbyt gwałtowny ruch kamery dla danej metody,
 - *skutek*: pojawianie się chwilowej informacji o ruchu.

Wykrywanie ruchu: typowe problemy

- Podzielenie ruchomego obiektu na wiele segmentów:
 - *przyczyna*: brak wykrycia ruchu dla części obiektu, niepoprawne wyznaczenie kierunku ruchu,
 - *skutek*: bardzo utrudnione śledzenie obiektów.
- Połączenie ruchomych obiektów w jeden segment:
 - *przyczyna*: różne ruchome obiekty się częściowo przesłaniają,
 - *skutek*: utrata istotnej informacji.
- Gubienie śledzonego obiektu:
 - *przyczyna*: zbyt duża różnica na obrazie w stosunku do modelu,
 - *przyczyna*: zbyt szybki ruch obiektu,
 - *skutek*: chwilowe zaprzestanie obserwacji obiektu,
 - *skutek*: niepoprawne *wysokopoziomowe* wnioskowanie,

Wykrywanie ruchu: metoda odejmowania tła

Model tła - zestaw danych pozwalający na wyznaczenie w obrazie elementów statycznych

```
1: model  $\leftarrow$  capture()
2: loop
3:   frame  $\leftarrow$  capture()
4:   delta  $\leftarrow$   $\Delta(\textit{model}, \textit{frame})$ 
5:   motion  $\leftarrow$   $\emptyset$ 
6:   for all fragment  $\in$  delta do
7:     if  $S(\textit{fragment}) > t$  then
8:       motion  $\leftarrow$  motion  $\cup$  {fragment}
9:     end if
10:  end for
11:  model  $\leftarrow$  frame {Opcjonalnie}
12: end loop
```

- Najprostsza stosowana metoda.
- **Statyczny** model tła:
 - model nie podlega zmianom,
 - bardzo wrażliwy na zmiany oświetlenia i ruch kamery,
 - zmiany tła powodują zafałszowanie wyników.
- **Dynamiczny** model tła:
 - modelem jest przedostatnia lub kilka przedostatnich klatek,
 - niewrażliwy na powolny ruch kamery i zmiany oświetlenia,
 - nie jest w stanie dokładnie określić obszarów ruchu,
 - niektóre ruchome elementy mogą nie zostać rozpoznane.

Wykrywanie ruchu: dynamiczny model tła

```
1: model  $\leftarrow$  initialize(capture())
2: loop
3:   frame  $\leftarrow$  capture()
4:   delta  $\leftarrow$   $\Delta$ (model, frame)
5:   motion  $\leftarrow$   $\emptyset$ 
6:   for all fragment  $\in$  delta do
7:     if S(fragment) > t then
8:       motion  $\leftarrow$  motion  $\cup$  {fragment}
9:     end if
10:  end for
11:  model  $\leftarrow$  update(model, frame)
12: end loop
```

- Szkielet metody wykrywania ruchu z modelem tła,
- Model tła podlega dynamicznym zmianom – *update*,
- Różnica pomiędzy modelem i obrazem – Δ ,

Wykrywanie ruchu: różnica pomiędzy modelem i obrazem

bezwzględna różnica

$$\Delta(M_t, I_t) = |I_t - M_t|. \quad (1)$$

względna różnica

$$\Delta(M_t, I_t) = \frac{|I_t - M_t|}{M_t}. \quad (2)$$

znormalizowana różnica

$$\Delta(M_t, I_t) = \frac{|I_t - M_t - \mu|}{\sigma}, \quad (3)$$

gdzie:

μ – wartość średnia różnic $|I_t - M_t|$ dla wszystkich pikseli,

σ – odchylenie std różnic $|I_t - M_t|$ dla wszystkich pikseli.

- jeśli większość obrazu stanowi tło σ reprezentuje naturalną zmienność tła
- μ - redukuje globalne zmiany (np. jasności) w obrazie

Wykrywanie ruchu: metoda mediany

Idea algorytmu

- Zapamiętywane jest N ostatnich klatek.
- Dla każdego piksela wyznaczana jest **mediana**:

$$m_t(x, y) = \text{median}[i_{t-1}(x, y), i_{t-2}(x, y), \dots, i_{t-N}(x, y)].$$

- Mediany stanowią **model tła**.

Własności:

- **Model tła** podlega zmianom, budowany jest **dynamicznie**:
- Jedna z najpowszechniej stosowanych metod - prosta i skuteczna.
- Duży narzut obliczeniowy przy liczeniu mediany.
- Duży narzut pamięciowy ze względu na konieczność pamiętania n klatek.

Wykrywanie ruchu: metoda mediany

Idea algorytmu

- Zapamiętywane jest N ostatnich klatek.
- Dla każdego piksela wyznaczana jest **mediana**:

$$m_t(x, y) = \text{median}[i_{t-1}(x, y), i_{t-2}(x, y), \dots, i_{t-N}(x, y)].$$

- Mediany stanowią **model tła**.

Własności:

- **Model tła** podlega zmianom, budowany jest **dynamicznie**:
- Jedna z najpowszechniej stosowanych metod - prosta i skuteczna.
- Duży narzut obliczeniowy przy liczeniu mediany.
- Duży narzut pamięciowy ze względu na konieczność pamiętania n klatek.

Własności c.d.:

- **Kluczowe założenie:** w sekwencji przynajmniej w połowie z N klatek przez piksel (x,y) widoczne tło.
- Jeśli rozrzut jasności tła jest niewielki, to poruszające się obiekty nie zaburzają wyznaczenia jasności tła
 - mediana mniejsza od największego elementu z pikseli tła - bo powyżej maksymalnego w najgorszym przypadku mniej niż $N/2$ pikseli z poruszającego się obiektu
 - mediana większa od najmniejszego - bo poniżej mniej niż $N/2$ pikseli z poruszającego się obiektu
- Ze względu na powyższe - metoda lepsza od uśredniania n poprzedzających klatek (uśrednianie uśrednia również piksele poruszającego się obiektu)

Wykrywanie ruchu: metoda aproksymowanej mediany

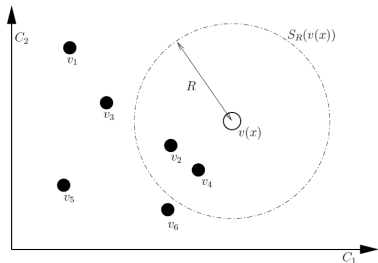
- Szybki wariant metody mediany.
- Nie ma konieczności zapamiętywania wszystkich klatek.
- Nie ma potrzeby liczenia mediany dla każdego piksela.
- Wartość mediany dla każdego piksela jest **aproksymowana**:

$$m_t(x, y) = m_{t-1}(x, y) + \begin{cases} 1 & \text{if } m_{t-1}(x, y) < i_{t-1}(x, y) \\ 0 & \text{if } m_{t-1}(x, y) = i_{t-1}(x, y) \\ -1 & \text{if } m_{t-1}(x, y) > i_{t-1}(x, y) \end{cases}$$

- Złożoność obliczeniowa $O(x_r \times y_r)$, (x_r, y_r - rozdzielczość obrazu)
- ... podczas gdy dla pełnej mediany: $O(x_r \times y_r \times n)$, n - liczba klatek analizowanych wstecz

Wykrywanie ruchu: metoda ViBE

- ViBe – **V**isual **B**ackground **E**xtractor
- Założenie - w sąsiedztwie każdego piksela obrazu statycznego znajdują się piksele o podobnych atrybutach
- Model tła opisany jest zbiorem próbek N (np. 20) (a nie rozkładem)
- Wykrywanie ruchu (lub raczej zmienności w obrazie) traktowane jest jako problem klasyfikacji:

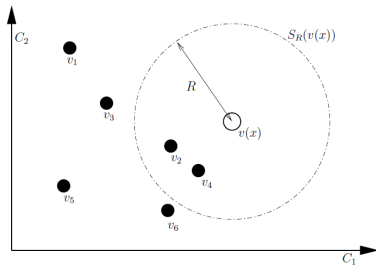


Źródło: Barnich O., M. Van Droogenbeek M..
ViBe : A universal background subtraction
algorithm for video sequences. IEEE Transactions
on Image Processing, 2011

- Piksel należy do tła jeśli w promieniu r (np. 20 dla jasności) w przestrzeni atrybutów znaleziono co najmniej t (np. 2) próbek w modelu tła dla pozycji odpowiadającej pikselowi.

Wykrywanie ruchu: metoda ViBE

- Inicjalizacja modelu tła z jednej klatki:
 - inicjalizacja powinna być możliwie szybka
 - powinna być łatwa możliwość powtórnej inicjalizacji
 - rozwiązanie: wartości atrybutów losowo wybranych pikseli z sąsiedztwa
- Aktualizacja modelu tła – wymiana losowego piksela
- Z każdą klatką można modyfikować tylko część modeli (tutaj: "model" dotyczy jednej pozycji (piksela) w obrazie)



Źródło: Barnich O., M. Van Droogenbeek M..
ViBe : A universal background subtraction
algorithm for video sequences. IEEE Transactions
on Image Processing, 2011

Inicjalizacja:

- Dla każdego piksela wybieramy sąsiedztwo o określonym promieniu R w przestrzeni (x, y) i z niego wybieramy losowo N pikseli - wektory ich cech tworzą model piksela.
- Sąsiedztwo o promieniu R można zastąpić 8-sąsiedztwem (tylko bezpośredni sąsiedzi)
- Niektóre piksele z otoczenia mogą występować wielokrotnie w modelu (ze względu na losowanie ze zwracaniem do puli)
- Pełny model piksela można uzyskać już z pojedynczej ramki.
- Podczas losowania można odrzucać wylosowane piksele jeśli znacznie odbiegają od piksela centralnego (np. jeśli leży on w pobliżu krawędzi różniących się znacznie obszarów)

Inicjalizacja:

- Dla każdego piksela wybieramy sąsiedztwo o określonym promieniu R w przestrzeni (x, y) i z niego wybieramy losowo N pikseli - wektory ich cech tworzą model piksela.
- Sąsiedztwo o promieniu R można zastąpić 8-sąsiedztwem (tylko bezpośredni sąsiedzi)
- Niektóre piksele z otoczenia mogą występować wielokrotnie w modelu (ze względu na losowanie ze zwracaniem do puli)
- Pełny model piksela można uzyskać już z pojedynczej ramki.
- Podczas losowania można odrzucać wylosowane piksele jeśli znacznie odbiegają od piksela centralnego (np. jeśli leży on w pobliżu krawędzi różniących się znacznie obszarów)

Inicjalizacja:

- Dla każdego piksela wybieramy sąsiedztwo o określonym promieniu R w przestrzeni (x, y) i z niego wybieramy losowo N pikseli - wektory ich cech tworzą model piksela.
- Sąsiedztwo o promieniu R można zastąpić 8-sąsiedztwem (tylko bezpośredni sąsiedzi)
- Niektóre piksele z otoczenia mogą występować wielokrotnie w modelu (ze względu na losowanie ze zwracaniem do puli)
- Pełny model piksela można uzyskać już z pojedynczej ramki.
- Podczas losowania można odrzucać wylosowane piksele jeśli znacznie odbiegają od piksela centralnego (np. jeśli leży on w pobliżu krawędzi różniących się znacznie obszarów)

Inicjalizacja:

- Dla każdego piksela wybieramy sąsiedztwo o określonym promieniu R w przestrzeni (x, y) i z niego wybieramy losowo N pikseli - wektory ich cech tworzą model piksela.
- Sąsiedztwo o promieniu R można zastąpić 8-sąsiedztwem (tylko bezpośredni sąsiedzi)
- Niektóre piksele z otoczenia mogą występować wielokrotnie w modelu (ze względu na losowanie ze zwracaniem do puli)
- Pełny model piksela można uzyskać już z pojedynczej ramki.
- Podczas losowania można odrzucać wylosowane piksele jeśli znacznie odbiegają od piksela centralnego (np. jeśli leży on w pobliżu krawędzi różniących się znacznie obszarów)

Inicjalizacja:

- Dla każdego piksela wybieramy sąsiedztwo o określonym promieniu R w przestrzeni (x, y) i z niego wybieramy losowo N pikseli - wektory ich cech tworzą model piksela.
- Sąsiedztwo o promieniu R można zastąpić 8-sąsiedztwem (tylko bezpośredni sąsiedzi)
- Niektóre piksele z otoczenia mogą występować wielokrotnie w modelu (ze względu na losowanie ze zwracaniem do puli)
- Pełny model piksela można uzyskać już z pojedynczej ramki.
- Podczas losowania można odrzucać wylosowane piksele jeśli znacznie odbiegają od piksela centralnego (np. jeśli leży on w pobliżu krawędzi różniących się znacznie obszarów)

Modyfikacja:

- Losowo wybieramy piksele do modyfikacji modelu (w szczególności wszystkie - **czasochłonne i niestabilne**)
- Dla wybranego piksela losujemy próbki z otoczenia do wymiany (w szczególności - tylko jeden)

Strategia konserwatywna:

Wymieniamy tylko takie próbki, których piksele nie zostały zakwalifikowane jako piksele poruszającego się obiektu (foreground). **Wada:** Pewne piksele nigdy nie staną się zmodyfikowanym tłem (tzw. *ghost effect*, np. obszar widoku samochodu, który odjechał z miejsca parkowania)

Modyfikacja:

- Losowo wybieramy piksele do modyfikacji modelu (w szczególności wszystkie - **czasochłonne i niestabilne**)
- Dla wybranego piksela losujemy próbki z otoczenia do wymiany (w szczególności - tylko jeden)

Strategia ślepa:

Wymieniamy próbki losowo - bez względu czy zostały zakwalifikowane jako foreground. **Wada:** Brak wykrywania powoli poruszających się obiektów, które stopniowo stają się tłem.

Modyfikacja:

- Losowo wybieramy piksele do modyfikacji modelu (w szczególności wszystkie - **zasochłonne i niestabilne**)
- Dla wybranego piksela losujemy próbki z otoczenia do wymiany (w szczególności - tylko jeden)

Częściowe rozwiązanie poprzedniego problemu:

Wykorzystanie licznego zbioru próbek wymienianych w długim oknie czasowym

- foreground staje się tłem po długim czasie
- piksele modelu znakujemy czasem losujemy piksel do wymiany tak aby "stare" piksele były bardziej prawdopodobne

Modyfikacja:

- Dla każdej klatki wybieramy tylko niewielki podzbiór pikseli do modyfikacji modeli tła
 - modyfikacja tylko niewielkiej części pikseli stabilizuje model tła
 - inercję modelu możemy kontrolować ustalając prawdopodobieństwo wybrania piksela do modyfikacji
- W podstawowym procesie zamiany próbek stanowiących model piksela stosujemy taktykę konserwatywną - modyfikujemy tylko te modele pikseli, które zostały zakwalifikowane w tej klatce jako piksele tła
- Dla uniknięcia blokady możliwości uaktualniania mapy tła (mapa tła - macierz o wymiarach obrazu z zaznaczonymi pikselami tła) stosujemy **dyfuzję przestrzenną** - losowe wstrzykiwanie próbek z modelu dla pewnego piksela do modeli pikseli sąsiednich

Modyfikacja:

- Dla każdej klatki wybieramy tylko niewielki podzbiór pikseli do modyfikacji modeli tła
 - modyfikacja tylko niewielkiej części pikseli stabilizuje model tła
 - inercję modelu możemy kontrolować ustalając prawdopodobieństwo wybrania piksela do modyfikacji
- W podstawowym procesie zamiany próbek stanowiących model piksela stosujemy taktykę konserwatywną - modyfikujemy tylko te modele pikseli, które zostały zakwalifikowane w tej klatce jako piksele tła
- Dla uniknięcia blokady możliwości uaktualniania mapy tła (mapa tła - macierz o wymiarach obrazu z zaznaczonymi pikselami tła) stosujemy **dyfuzję przestrzenną** - losowe wstrzykiwanie próbek z modelu dla pewnego piksela do modeli pikseli sąsiednich

Modyfikacja:

- Dla każdej klatki wybieramy tylko niewielki podzbiór pikseli do modyfikacji modeli tła
 - modyfikacja tylko niewielkiej części pikseli stabilizuje model tła
 - inercję modelu możemy kontrolować ustalając prawdopodobieństwo wybrania piksela do modyfikacji
- W podstawowym procesie zamiany próbek stanowiących model piksela stosujemy taktykę konserwatywną - modyfikujemy tylko te modele pikseli, które zostały zakwalifikowane w tej klatce jako piksele tła
- Dla uniknięcia blokady możliwości uaktualniania mapy tła (mapa tła - macierz o wymiarach obrazu z zaznaczonymi pikselami tła) stosujemy **dyfuzję przestrzenną** - losowe wstrzykiwanie próbek z modelu dla pewnego piksela do modeli pikseli sąsiednich

Wykrywanie ruchu metodą ViBE - zarys pseudokodu

- $S[y, x]$ - tablica zbiorów próbek dla pikseli,
- $I[y, x]$ - klatka sekwencji wideo,
- $rand(a, b)$ - generator liczb losowych z przedziału (a, b)
- $Rand(A, B)$ - generator całkowitych liczb losowych ze zbioru (A, B)
- p_R - prawdopodobieństwo wymiany próbki w modelu dla piksela
- p_N - prawdopodobieństwo dyfuzji próbki do sąsiednich modeli

```
1: for  $x = 0$  to  $width - 1$  do
2:   for  $y = 0$  to  $height - 1$  do
3:      $n = 0$ 
4:     for all  $s \in S[y, x]$  do
5:       if  $|s - I[y, x]| < r$  then
6:          $n = n + 1$ 
7:       end if
8:     end for
9:     if  $n < N_{min}$  then
10:       $FB[y, x] = foreground$ 
11:    else
12:       $FB[y, x] = background$ 
13:      {Losowa wymiana próbki w modelu:}
14:      if  $rand(0, 1) \leq p_R$  then
15:         $S[y, x][Rand(0, M - 1)] =$ 
16:         $I[y + Rand(-R, R), x + Rand(-R, R)]$ 
17:      end if
18:      {Dyfuzja losowa:}
19:      if  $rand(0, 1) \leq p_N$  then
20:         $d_x, d_y = Rand(-1, 1)$ 
21:         $S[y + d_y, x + d_x][Rand(0, M - 1)] = I[y, x]$ 
22:      end if
23:    end if
24:  end for
25: end for
```

Wykrywanie ruchu metodą ViBE - cechy

- Mała wrażliwość na zakłócenia typu szumowego - model jak zbiór próbek z zaszumionego sąsiedztwa,
- Tolerancja ruchu małych obiektów (np. liście na wietrze) - ze względu na losowy charakter próbek pobranych z bliskiego otoczenia,
- Tolerancja powolnych ruchów kamery (bo model piksela zawiera próbki z sąsiedztwa - przesuwanie obrazu o ok. 1 px/kl. (panoramowanie - ang. panning) nie powoduje masowych błędów fałszywej detekcji
- ...

- ...
- Tolerancja stopniowych zmian oświetlenia - ze względu na stopniową wymianę próbek w modelu
- Nagłe zmiany oświetlenia powodują tylko chwilowe fałszywe wykrycie ruchu
- Obiekty poruszające się po zatrzymaniu stopniowo stają się elementem tła.
- Bardziej szczegółowe omówienie metody można znaleźć w artykule [1].

Cel:

Scalenie zbiorów pikseli wykrytych poprzednio opisanymi metodami (regiony zmian/ruchu)

- Oznacz piksele sklasyfikowane jako nie-tło.
- Utwórz z tych pikseli komponenty spójności.
- Zastosuj operacje morfologiczne (typowo: domknięcie oraz dylatacja).
- Usuń komponenty spójności o niewielkich rozmiarach.
- Utwórz prostokąty otaczające (*bounding boxes*) dla otrzymanych obszarów.
- Przyjmij otrzymane prostokąty otaczające jako regiony ruchu.

Opis ruchu dla całego obrazu

- **Przepływ optyczny** (ang. *optical flow*) - pole (tablica) wektorów wyznaczających kierunek i szybkość zmian obrazu w czasie.
- **Pole ruchu** (ang. *motion field*) - tablica wektorów opisujących lokalną szybkość i kierunek **rzeczywistego** ruchu elementów obrazu.

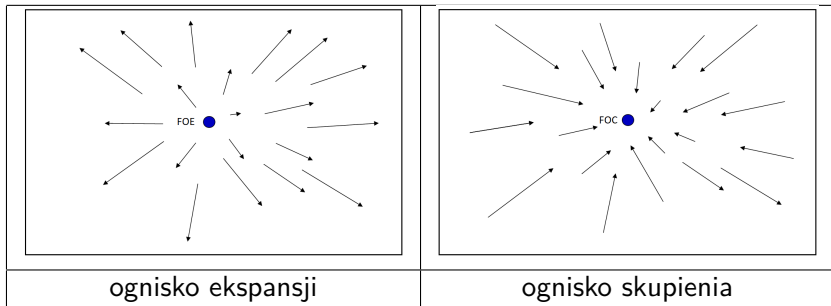
przepływ optyczny \neq pole ruchu

np. przepływ optyczny = 0.0 dla wnętrza poruszających się obiektów o jednorodnych atrybutach

- atrybuty wizualne są stałe we wnętrzu ruchomego obiektu
- idealne pole ruchu powinno wskazywać ruch również dla wnętrza rzutu poruszającego się obiektu

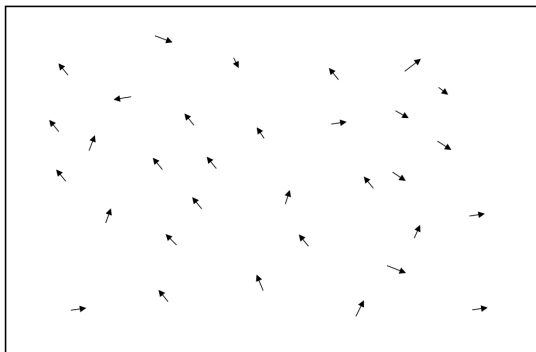
Opis ruchu dla całego obrazu

- **Ognisko** (ang. *focus*) - punkt w polu ruchu będący (w przybliżeniu) punktem wspólnych prostych zawierających wektory pola ruchu.
 - **Ognisko ekspansji** (*FOE - focus of expansion*) - jeśli wektory pola ruchu skierowane są **OD** ogniska ekspansji.
 - **Ognisko skupienia** (*FOC - focus of contraction*) - jeśli wektory pola ruchu skierowane są **DO** ogniska ekspansji.



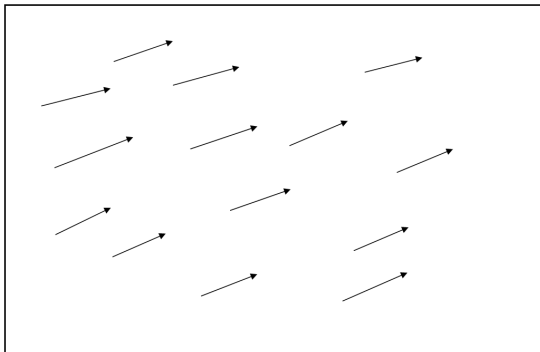
- Rozpoznawania przyczyny zmian w obrazie
 - ruch kamery - prostopadle do osi obiektywu (ang. *panning*)
 - ruch kamery wzdłuż osi
 - zmiana kąta widzenia (FOV)
 - ruch obiektów w polu widzenia
 - zmiana oświetlenia
- Lokalizowanie poruszających się obiektów (w przestrzeni sceny)
- Określanie kierunku poruszających się obiektów
- Określanie szybkości poruszających się obiektów
- Separacja obiektów poruszających się w różnych kierunkach

Rysunek 1: *Brak ewidentnych cech poruszających się obiektów/ruchu kamery*



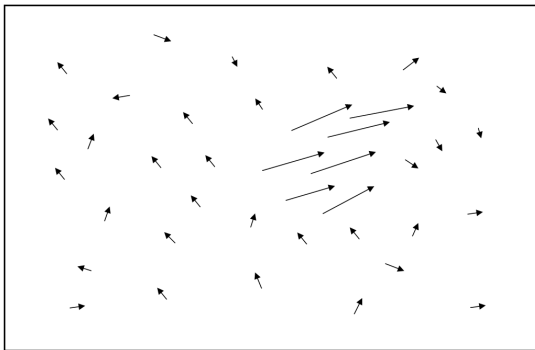
- Wektory ruchu o niewielkiej długości
- Kierunki wektorów ruchu rozłożone równomiernie i losowo

Rysunek 2: *Ruch kamery - panoramowanie (ang. panning)*



- Wektory ruchu o znacznej długości
- Długości wektorów ruchu porównywalne w obszarze całego pola ruchu
- Kierunki wszystkich wektorów skupione wokół kierunku dominującego

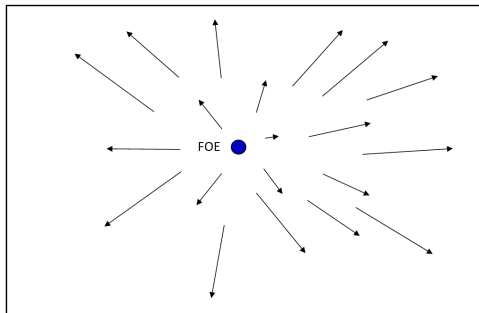
Rysunek 3: *Przemieszczający się pojedynczy obiekt*



- W większości obszaru pola ruchu - wektory ruchu o nieznaczej długości i o kierunkach równomiernie rozproszonych losowo
- W niewielkim/niewielkich obszarach pola ruchu wektory ruchu o znacznej długości zorientowane jednorodnie, o zbliżonych długościach

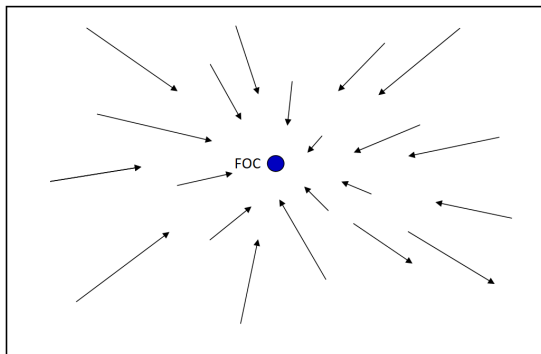
Pole ruchu - rozpoznawanie charakteru ruchu

Rysunek 4: *Ruch kamery w przód wzdłuż osi lub zmniejszanie FOV*



- Kierunki wektorów ruchu przechodzą przez obszar w centrum pola ruchu (FOE)
- Wektory ruchu zwrócone do krawędzi pola ruchu
- Długość wektorów ruchu zależna od odległości od FOE
- Znajdowanie FOE - np. metodą zbliżoną do transformacji Hougha:
 - każdy wektor "głosuje" na punkty wzdłuż prostej, która go zawiera
 - istnienie FOC/FOE - wyraźne maksimum istnieje i znajduje się w pobliżu środka obrazu

Rysunek 5: *Ruch kamery w tył wzdłuż osi lub zwiększanie FOV*

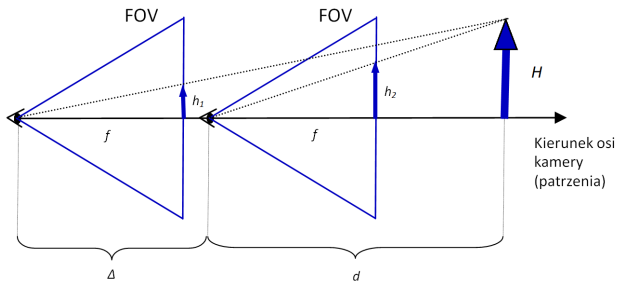


- Kierunki wektorów ruchu przechodzą przez obszar w centrum pola ruchu (FOC)
- Wektory ruchu zwrócone do środka obrazu
- Długość wektorów ruchu zależna od odległości od FOC

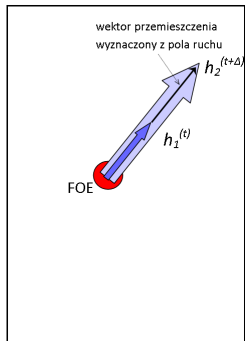
Pole ruchu - zastosowanie do przewidywania kolizji

Rozważamy sytuację statycznej sceny i kamery poruszającej się w przód wzdłuż jej osi

Rysunek 6: Zmiany pozycji punktu obiektu na rzutni przy przemieszczaniu kamery w przód

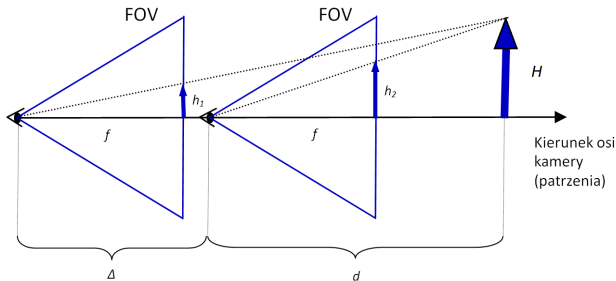


Rysunek 7: Widok na rzutni (równoważnie: w polu ruchu)



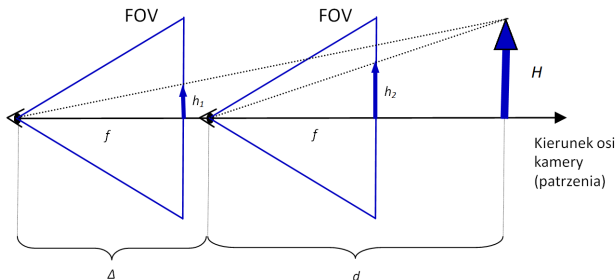
Pole ruchu - zastosowanie do przewidywania kolizji

Rysunek 8: Zmiany pozycji punktu obiektu na rzutni przy przemieszczaniu kamery w przód



- Wektor ruchu wyznaczony na podstawie obrazów pozyskanych w czasie t_0 i $t_0 + \Delta_T$ jest $h_2 - h_1$
- **Cel:** wyznaczyć czas do kolizji z obiektem widzianym jako h_1 i h_2 przy założeniu, że kamera (lub obiekt) poruszają się ruchem jednostajnym
- Dla osiągnięcia celu wystarczy wyznaczyć stosunek d/Δ

Pole ruchu - zastosowanie do przewidywania kolizji



$$\frac{H}{h_2} = \frac{d}{f}; \quad \frac{H}{h_1} = \frac{d + \Delta}{f} \quad \rightarrow \quad H = \frac{h_1(d + \Delta)}{f} \quad (4)$$

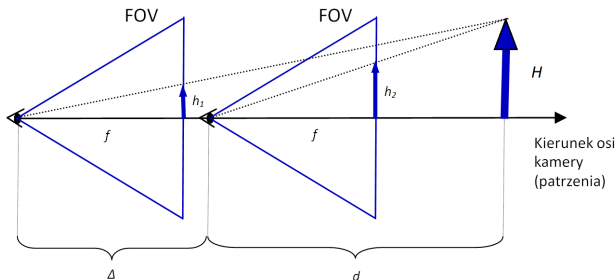
eliminujemy nieznane H wstawiając do pierwszej proporcji:

$$\frac{h_1(d + \Delta)}{fh_2} = \frac{d}{f} \quad (5)$$

eliminujemy niekoniecznie znane f (ogniskowa - odległość od obiektywu do rzutni)

$$\frac{d + \Delta}{d} = \frac{h_2}{h_1} \quad (6)$$

Pole ruchu - zastosowanie do przewidywania kolizji



$$\frac{H}{h_2} = \frac{d}{f}; \quad \frac{H}{h_1} = \frac{d + \Delta}{f} \quad \rightarrow \quad H = \frac{h_1(d + \Delta)}{f} \quad (4)$$

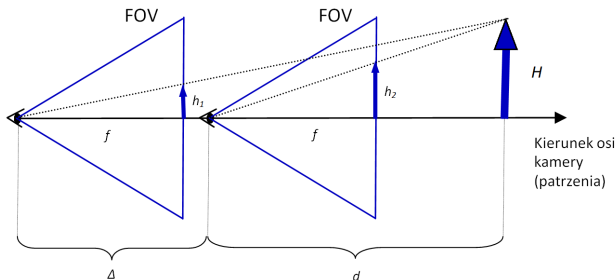
eliminujemy nieznane H wstawiając do pierwszej proporcji:

$$\frac{h_1(d + \Delta)}{fh_2} = \frac{d}{f} \quad (5)$$

eliminujemy niekoniecznie znane f (ogniskowa - odległość od obiektywu do rzutni)

$$\frac{d + \Delta}{d} = \frac{h_2}{h_1} \quad (6)$$

Pole ruchu - zastosowanie do przewidywania kolizji



$$\frac{H}{h_2} = \frac{d}{f}; \quad \frac{H}{h_1} = \frac{d + \Delta}{f} \quad \rightarrow \quad H = \frac{h_1(d + \Delta)}{f} \quad (4)$$

eliminujemy nieznane H wstawiając do pierwszej proporcji:

$$\frac{h_1(d + \Delta)}{fh_2} = \frac{d}{f} \quad (5)$$

eliminujemy niekoniecznie znane f (ogniskowa - odległość od obiektywu do rzutni)

$$\frac{d + \Delta}{d} = \frac{h_2}{h_1} \quad (6)$$

Pole ruchu - zastosowanie do przewidywania kolizji

$$\frac{d + \Delta}{d} = \frac{h_2}{h_1} \quad (7)$$

po prostych przekształceniach:

$$(d + \Delta)h_1 = dh_2; \quad dh_1 + \Delta h_1 = dh_2; \quad d(h_2 - h_1) = \Delta h_1 \quad (8)$$

$$\frac{d}{\Delta} = \frac{h_2 - h_1}{h_1} \quad (9)$$

Przy założeniu jednostajności ruchu wykorzystujemy proporcje przebytych odległości i czasu:

$$\frac{t_C}{\Delta_T} = \frac{d}{\Delta} = \frac{h_2 - h_1}{h_1} \quad \longrightarrow \quad t_C = \Delta_T \frac{h_2 - h_1}{h_1} \quad (10)$$

- Δ_T - odstęp czasowy pomiędzy dwoma klatkami sekwencji wideo (zwykle znany)
- t_C - czas pozostały do kolizji

Pole ruchu - zastosowanie do przewidywania kolizji

$$\frac{d + \Delta}{d} = \frac{h_2}{h_1} \quad (7)$$

po prostych przekształceniach:

$$(d + \Delta)h_1 = dh_2; \quad dh_1 + \Delta h_1 = dh_2; \quad d(h_2 - h_1) = \Delta h_1 \quad (8)$$

$$\frac{d}{\Delta} = \frac{h_2 - h_1}{h_1} \quad (9)$$

Przy założeniu jednostajności ruchu wykorzystujemy proporcje przebytych odległości i czasu:

$$\frac{t_C}{\Delta_T} = \frac{d}{\Delta} = \frac{h_2 - h_1}{h_1} \quad \longrightarrow \quad t_C = \Delta_T \frac{h_2 - h_1}{h_1} \quad (10)$$

- Δ_T - odstęp czasowy pomiędzy dwoma klatkami sekwencji wideo (zwykle znany)
- t_C - czas pozostały do kolizji

Pole ruchu - zastosowanie do przewidywania kolizji

$$\frac{d + \Delta}{d} = \frac{h_2}{h_1} \quad (7)$$

po prostych przekształceniach:

$$(d + \Delta)h_1 = dh_2; \quad dh_1 + \Delta h_1 = dh_2; \quad d(h_2 - h_1) = \Delta h_1 \quad (8)$$

$$\frac{d}{\Delta} = \frac{h_2 - h_1}{h_1} \quad (9)$$

Przy założeniu jednostajności ruchu wykorzystujemy proporcje przebytych odległości i czasu:

$$\frac{t_C}{\Delta_T} = \frac{d}{\Delta} = \frac{h_2 - h_1}{h_1} \quad \longrightarrow \quad t_C = \Delta_T \frac{h_2 - h_1}{h_1} \quad (10)$$

- Δ_T - odstęp czasowy pomiędzy dwoma klatkami sekwencji wideo (zwykle znany)
- t_C - czas pozostały do kolizji

W podobny sposób przy znajomości:

- f - ogniskowej obiektywu
- Δ_T - odstępu czasowego pomiędzy kolejnymi klatkami sekwencji wideo ($1/fps$)
- Δ/Δ_T - prędkości poruszania się kamery
- rozmiaru fizycznego pojedynczego piksela (potrzebne do wyznaczenia fizycznych rozmiarów h_1, h_2 - odległości punktu od środka rzutni)

można wyznaczyć inne interesujące dane:

- d - odległość kamery do obiektu
- H - rzeczywista wielkość obiektu (odległość punktu od osi kamery)
- ...

- Przepływ optyczny (ang. *optical flow*) opisuje ruch fragmentów obrazu.
- Konieczne założenia pozwalające łatwo zidentyfikować fragmenty odpowiadające temu samemu obiektowi w kolejnych klatkach
- Problem precyzji wyznaczania pola ruchu:
 - nie jest możliwe dla pojedynczego piksela,
 - wymaga przyjęcia dodatkowych założeń (spójność przestrzenna),
 - może być konieczna analiza dla różnych sk
- Ze względu na ograniczenia precyzji i efektywność obliczeniową - konieczność rozważenia ograniczenia liczby analizowanych obszarów:
 - metody *gęste* (wektor ruchu wyznaczony dla każdego piksela)
 - metody *rzadkie* (tylko dla niektórych pikseli) wyznaczania przepływu.

- Przepływ optyczny (ang. *optical flow*) opisuje ruch fragmentów obrazu.
- Konieczne założenia pozwalające łatwo zidentyfikować fragmenty odpowiadające temu samemu obiektowi w kolejnych klatkach
- Problem precyzji wyznaczania pola ruchu:
 - nie jest możliwe dla pojedynczego piksela,
 - wymaga przyjęcia dodatkowych założeń (spójność przestrzenna),
 - może być konieczna analiza dla różnych sk
- Ze względu na ograniczenia precyzji i efektywność obliczeniową - konieczność rozważenia ograniczenia liczby analizowanych obszarów:
 - metody *gęste* (wektor ruchu wyznaczony dla każdego piksela)
 - metody *rzadkie* (tylko dla niektórych pikseli) wyznaczania przepływu.

- Przepływ optyczny (ang. *optical flow*) opisuje ruch fragmentów obrazu.
- Konieczne założenia pozwalające łatwo zidentyfikować fragmenty odpowiadające temu samemu obiektowi w kolejnych klatkach
- Problem precyzji wyznaczania pola ruchu:
 - nie jest możliwe dla pojedynczego piksela,
 - wymaga przyjęcia dodatkowych założeń (spójność przestrzenna),
 - może być konieczna analiza dla różnych sk
- Ze względu na ograniczenia precyzji i efektywność obliczeniową - konieczność rozważenia ograniczenia liczby analizowanych obszarów:
 - metody *gęste* (wektor ruchu wyznaczony dla każdego piksela)
 - metody *rzadkie* (tylko dla niektórych pikseli) wyznaczania przepływu.

- ...
- Przykładowe metody wyznaczania przepływu:
 - metoda śledzenia bloków,
 - **metoda Lucasa–Kanade** (podejście lokalne),
 - metoda piramidowa Lucasa–Kanade (lokalne + wiele skal),
 - metoda Horna–Schuncka (podejście globalne, jeden wektor ruch dla obrazu lub dużego jego fragmentu, dodatkowy czynnik regularyzacji minimalizujący zmiany pomiędzy sąsiadującym fragmentami - długość gradientu wektora ruchu),

Założenia:

- Stałość oświetlenia - fragmenty odpowiadające temu samemu detalowi w kolejnych klatkach wideo nie różnią się znacznie jasnością
- Spójność przestrzenna - sąsiadujące piksele poruszają się w tym samym (zbliżonym kierunku)
- Stałość czasowa - fragmenty odpowiadające temu samemu obiektowi nie przemieszczają się znacznie pomiędzy kolejnymi klatkami (powolny ruch, wektor przemieszczenia o niewielkiej długości)

Wykrywanie ruchu: równania przepływu optycznego

W poniższych formułach $I(x, y)$ oznacza średnią jasność obrazu w niewielkim (np. kołowym) otoczeniu punktu (x, y)

Zależność jasności dwóch obrazów

- Jeśli fragment obrazu (obiekt) przesunął się pomiędzy klatkami t i $t + 1$ o wektor (u, v) to:

$$I(x + u, y + v, t + 1) = I(x, y, t), \quad (11)$$

gdzie:

- x, y, t – współrzędna punktu w czasoprzestrzeni,
 - u, v – przesunięcie punktu pomiędzy czasem t i $t + 1$.
- Przyjmując model ciągłej czasoprzestrzeni z szeregu Taylora wyznaczamy przybliżenie:

$$I(x + u, y + v, t + 1) \approx I(x, y, t) + uI_x(x, y, t) + vI_y(x, y, t) + I_t(x, y, t), \quad (12)$$

gdzie:

- I_x, I_y, I_t – pochodne cząstkowe obrazu-funkcji $I(x, y, t)$ po x, y oraz t .

Wykrywanie ruchu: równania przepływu optycznego

W poniższych formułach $I(x, y)$ oznacza średnią jasność obrazu w niewielkim (np. kołowym) otoczeniu punktu (x, y)

Zależność jasności dwóch obrazów

- Jeśli fragment obrazu (obiekt) przesunął się pomiędzy klatkami t i $t + 1$ o wektor (u, v) to:

$$I(x + u, y + v, t + 1) = I(x, y, t), \quad (11)$$

gdzie:

- x, y, t – współrzędna punktu w czasoprzestrzeni,
 - u, v – przesunięcie punktu pomiędzy czasem t i $t + 1$.
- Przyjmując model ciągłej czasoprzestrzeni z szeregu Taylora wyznaczamy przybliżenie:

$$I(x + u, y + v, t + 1) \approx I(x, y, t) + uI_x(x, y, t) + vI_y(x, y, t) + I_t(x, y, t), \quad (12)$$

gdzie:

- I_x, I_y, I_t – pochodne cząstkowe obrazu-funkcji $I(x, y, t)$ po x, y oraz t .

Różnica pomiędzy dwoma obrazami

Wektor ruchu (u, v) minimalizuje różnicę pomiędzy obrazami tzn.:

$$I(x + u, y + v, t + 1) \approx I(x, y, t) \quad (13)$$

Ponieważ:

$$I(x + u, y + v, t + 1) \approx I(x, y, t) + \\ uI_x(x, y, t) + vI_y(x, y, t) + I_t(x, y, t),$$

... to różnica bliska zeru:

$$uI_x(x, y, t) + vI_y(x, y, t) + I_t(x, y, t) = 0. \quad (14)$$

$$uI_x(x, y, t) + vI_y(x, y, t) = -I_t(x, y, t), \quad (15)$$

(u, v) – można interpretować jako wektor prędkości ($\Delta t = 1$) punktu.

Różnica pomiędzy dwoma obrazami

Wektor ruchu (u, v) minimalizuje różnicę pomiędzy obrazami tzn.:

$$I(x + u, y + v, t + 1) \approx I(x, y, t) \quad (13)$$

Ponieważ:

$$I(x + u, y + v, t + 1) \approx I(x, y, t) + \\ uI_x(x, y, t) + vI_y(x, y, t) + I_t(x, y, t),$$

... to różnica bliska zeru:

$$uI_x(x, y, t) + vI_y(x, y, t) + I_t(x, y, t) = 0. \quad (14)$$

$$uI_x(x, y, t) + vI_y(x, y, t) = -I_t(x, y, t), \quad (15)$$

(u, v) – można interpretować jako wektor prędkości ($\Delta t = 1$) punktu.

Różnica pomiędzy dwoma obrazami

Wektor ruchu (u, v) minimalizuje różnicę pomiędzy obrazami tzn.:

$$I(x + u, y + v, t + 1) \approx I(x, y, t) \quad (13)$$

Ponieważ:

$$I(x + u, y + v, t + 1) \approx I(x, y, t) + \\ ul_x(x, y, t) + vl_y(x, y, t) + I_t(x, y, t),$$

... to różnica bliska zeru:

$$ul_x(x, y, t) + vl_y(x, y, t) + I_t(x, y, t) = 0. \quad (14)$$

$$ul_x(x, y, t) + vl_y(x, y, t) = -I_t(x, y, t), \quad (15)$$

(u, v) – można interpretować jako wektor prędkości ($\Delta t = 1$) punktu.

Przepływ optyczny – metoda Lucasa–Kanade

- Podstawowe równanie ruchu piksela p (zapis macierzowy):

$$I_t(p) + \begin{bmatrix} I_x(p) & I_y(p) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = 0,$$

- **Cel:** wyznaczyć wektor przesunięcia (u, v) dla każdego piksela.
- Jedno równanie, dwie niewiadome.
- Założenie - sąsiednie piksele mają identyczny (lub bardzo zbliżony) wektor ruchu.
- Do rozwiązywania formalnie wystarcza tylko dwa piksele (poruszające się w sposób spójny)
 - może okazać się, że dostaniemy równania liniowo zależne
- Ze względu na szумы i niepełne spełnienie założeń, w/w równość zwykle nie jest spełniona dokładnie
- ...

- Podstawowe równanie ruchu piksela p (zapis macierzowy):

$$I_t(p) + \begin{bmatrix} I_x(p) & I_y(p) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = 0,$$

- ...
- Można wziąć nie dwa ale więcej sąsiadujących pikseli
- Sąsiedztwa o wielkości 3×3 lub większe.
- Możliwość sformułowania większej liczby równań - znajdowanie wektora minimalizującego stopień niespełnienia równań.

Przepływ optyczny: metoda Lucasa–Kanade

Początkowy układ równań

$$\mathbf{A}\mathbf{s} = \mathbf{b}, \mathbf{s} = (u, v)^T, \quad (16)$$

Zwykle otrzymany układ równań sprzeczny (więcej równań niż niewiadomych):

$$\begin{bmatrix} l_x(p_1) & l_y(p_1) \\ l_x(p_2) & l_y(p_2) \\ \vdots & \vdots \\ l_x(p_n) & l_y(p_n) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} l_t(p_1) \\ l_t(p_2) \\ \vdots \\ l_t(p_n) \end{bmatrix}, \quad (17)$$

Rozwiązanie metodą minimalizacji błędu średniokwadratowego

Problem minimalizacji $\min \|\mathbf{A}\mathbf{s} - \mathbf{b}\|^2$ rozwiązywany standardowo:
 $(\mathbf{A}^T \mathbf{A})\mathbf{s} = \mathbf{A}^T \mathbf{b}.$

$$\begin{bmatrix} \sum_{i=1}^n l_x l_x & \sum_{i=1}^n l_x l_y \\ \sum_{i=1}^n l_x l_y & \sum_{i=1}^n l_y l_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum_{i=1}^n l_t l_x \\ \sum_{i=1}^n l_t l_y \end{bmatrix}, \quad (18)$$

Przepływ optyczny: metoda Lucasa–Kanade

Początkowy układ równań

$$\mathbf{A}\mathbf{s} = \mathbf{b}, \mathbf{s} = (u, v)^T, \quad (16)$$

Zwykle otrzymany układ równań sprzeczny (więcej równań niż niewiadomych):

$$\begin{bmatrix} l_x(p_1) & l_y(p_1) \\ l_x(p_2) & l_y(p_2) \\ \vdots & \vdots \\ l_x(p_n) & l_y(p_n) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} l_t(p_1) \\ l_t(p_2) \\ \vdots \\ l_t(p_n) \end{bmatrix}, \quad (17)$$

Rozwiązanie metodą minimalizacji błędu średniokwadratowego

Problem minimalizacji $\min \|\mathbf{A}\mathbf{s} - \mathbf{b}\|^2$ rozwiązywany standardowo:
 $(\mathbf{A}^T \mathbf{A})\mathbf{s} = \mathbf{A}^T \mathbf{b}.$

$$\begin{bmatrix} \sum_{i=1}^n l_x l_x & \sum_{i=1}^n l_x l_y \\ \sum_{i=1}^n l_x l_y & \sum_{i=1}^n l_y l_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum_{i=1}^n l_t l_x \\ \sum_{i=1}^n l_t l_y \end{bmatrix}, \quad (18)$$

Rozwiązanie metodą minimalizacji błędu średniokwadratowego

Problem minimalizacji $\min |\mathbf{A}\mathbf{s} - \mathbf{b}|^2$ rozwiązywany standardowo:
 $(\mathbf{A}^T \mathbf{A})\mathbf{s} = \mathbf{A}^T \mathbf{b}$.

$$\begin{bmatrix} \sum_{i=1}^n l_x l_x & \sum_{i=1}^n l_x l_y \\ \sum_{i=1}^n l_x l_y & \sum_{i=1}^n l_y l_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum_{i=1}^n l_t l_x \\ \sum_{i=1}^n l_t l_y \end{bmatrix}, \quad (19)$$

Finalna postać równania

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n l_x l_x & \sum_{i=1}^n l_x l_y \\ \sum_{i=1}^n l_x l_y & \sum_{i=1}^n l_y l_y \end{bmatrix}^{-1} \begin{bmatrix} -\sum_{i=1}^n l_t l_x \\ -\sum_{i=1}^n l_t l_y \end{bmatrix}. \quad (20)$$

Podsumowanie - co należy zapamiętać:

- Podstawą analizy wideo jest wykrywanie ruchu
- Ruch obserwowany w sekwencji wideo może być istotny i nieistotny
- Skuteczne metody wykrywania ruchu wykrywają ruch istotny i ignorują nieistotny
- Podstawowe metody wykrywania ruchu to: metoda odejmowania tła, metoda odejmowania mediany i aproksymowanej mediany
- Przepływ optyczny opisuje kierunki ruchu obiektów w obrazie
- Nawet proste metody analizy geometrycznej ruchu w obrazie pozwalają wyznaczyć użyteczne atrybuty ruchu obiektów lub kamery
- Przepływ optyczny może być wyznaczony wieloma opisanymi w literaturze metodami, metoda Lucasa-Kanade jest jedną z częściej stosowanych

Literatura uzupełniająca:

1. Barnich O., M. Van Droogenbeek M.: *ViBe : A universal background subtraction algorithm for video sequences*. IEEE Transactions on Image Processing, 2011
2. Sonka M., Hlavac V., Boyle R.: *Image Processing, Analysis, and Machine Vision, Fourth Edition*, Cengage Learning, 2015 (rozdz. 16.1, 16.2)
3. Szeliski R.: *Computer Vision: Algorithms and Applications 2nd Edition*, 2021, rodz. 9
4. Santoyo-Morales J., Hasimoto-Beltran R.: *Video Background Subtraction in Complex Environments*, Journ. of Appl. Research and Technology, 2014
5. Kulchandani J.S., Dangarwala J.: *Moving object detection: Review of recent research trends.*, Proc. of. Int. Conf. on Pervasive Computing, 2015