

# OBJECT TRACKING IN VIDEO

**Peter Beťko**

Bachelor Degree Programme (3), FIT BUT  
E-mail: xbetko00@stud.fit.vutbr.cz

Supervised by: Adam Herout  
E-mail: herout@fit.vutbr.cz

## ABSTRACT

This document describes experiments with object tracking in video. The tracker is based on Particle Filter algorithm and adds many useful features, such as dynamic change of tracking parameters, solution of object overlay and evaluation of tracking quality.

## 1. ÚVOD

Cieľom mojej práce je experimentovať so sledovaním objektov vo videu (ďalej len trackovanie). Implementácia je v jazyku C++ s využitím knižnice OpenCV (Open Source Computer Vision), ktorá vďaka množstvu vstavaných funkcií výrazne uľahčuje tvorbu aplikácií v oblasti počítačového videnia.

Za samotný algoritmus bol zvolený Particle filter. Medzi jeho nesporné výhody patrí možnosť trackovať nelineárne sa pohybujúce objekty, minimálna náchylnosť na zmeny pozadia a pohyb kamery, v niektorých prípadoch tiež zmeny natočenia sledovaného objektu a intuitívnosť fungovania.

Základný algoritmus bol doplnený o rôzne optimalizácie a vylepšenia, ako napríklad dynamická zmena parametrov trackovania, detekcia a riešenie prekrývania či nástroj na ohodnotenie fungovania algoritmu.

## 2. ROZBOR

Základnou myšlienkou algoritmu Particle filter je aproximovať funkciu hustoty pravdepodobnosti výskytu objektu súborom vážených častíc (tzv. particle) [1]. Ohodnocovacou funkciou sa každej častici priradí váha, ktorá určuje ako presne zodpovedá referenčnému vzorku. Následne sa vyberie najlepšia, ktorá bude zobrazená. Na základe takto ohodnotených častíc dôjde k ich prevzorkovaniu (resampling). K tomu sa využíva Bayesov vzorec. Čím väčšiu váhu má daná častica, tým viac nových bude vygenerovaných na jej pozícií. Poloha (prípadne aj rozmery a natočenie) takto prevzorkovaných častíc sa ešte upraví pričítaním šumu, ktorý bol náhodne vytvorený v rozmedzí danom vstupnými parametrami programu. Spolu s počtom častíc je to veľmi dôležitý faktor ovplyvňujúci úspešnosť algoritmu.

## 2.1. VÝBER NAJLEPŠEJ ČASTICE

Momentálne sú implementované štyri metódy na výber najlepšej častice. Prvou je odčítanie hodnôt pixelov pre každú farebnú zložku RGB podľa vzorca (1):

$$\|arr1 - arr2\| = \sum_{x,y} (arr1_{x,y} - arr2_{x,y})^2 \quad (1)$$

Ďalšie dve metódy vypočítajú šedotónový resp. farebný histogram pre časticu a referenčný snímok a následne ich porovnajú metódou Chi-square podľa vzorca (2):

$$d(H_1, H_2) = \sum_i \frac{(H_1(i) - H_2(i))^2}{H_1(i) + H_2(i)} \quad (2)$$

Posledný spôsob je kombináciou predchádzajúcich. Obrázok sa rozdelí všeobecne na  $M \times N$  častí a tie sú následne ohodnocované pomocou histogramov.

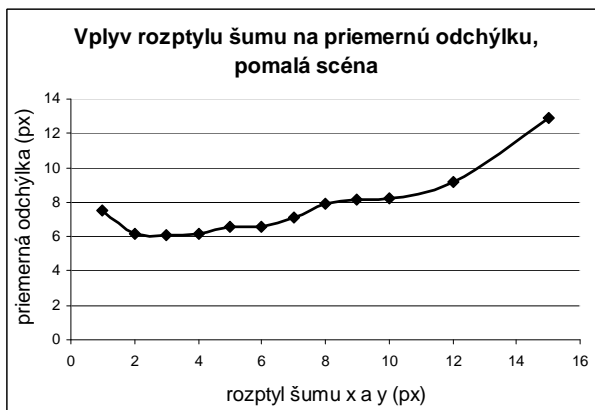
## 2.2. TESTOVANIE KVALITY TRACKOVANIA

Základnou myšlienkou je ručne vyznačiť trajektóriu pohybu objektu a následne ju porovnať s vypočítanou trajektóriou. V prvej fáze užívateľ vyberie sledovaný objekt a každých  $N$  snímok znovu vyznačí jeho polohu. Všetko prebieha interaktívne, použitím myši. Program takto získané súradnice polohy objektu zapíše do súboru. Pri ďalšom spustení prebehne bez zásahu užívateľa samotné trackovanie. Program si zo súboru načíta súradnice, v ktorých sa mal objekt v danom čase nachádzať a porovná ich so súradnicami, ktoré sa vypočítali programom. Výstupom je priemerná odchýlka (vzdialenosť v pixeloch), o ktorú sa algoritmus odchýlil od ideálnej trajektórie.

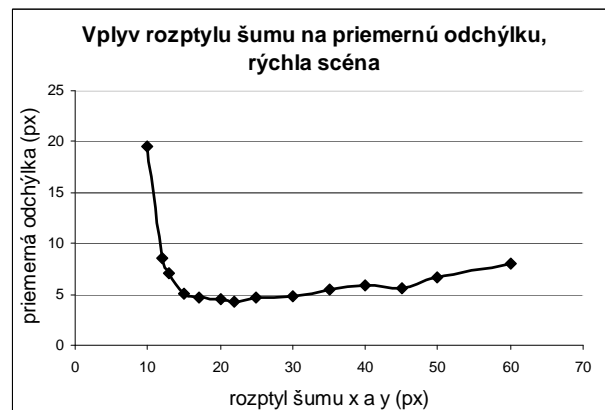
## 2.3. VPLYV ROZPTYLU ŠUMU NA KVALITU TRACKOVANIA

Užívateľ pomocou prepínačov zadá požadovaný rozptyl náhodného šumu pre x-ové ( $r_x$ ) a y-ové ( $r_y$ ) súradnice. Následne sa podľa normálneho (Gaussového) rozloženia pravdepodobnosti s parametrami  $\text{Normal}(0, r_x)$  resp.  $\text{Normal}(0, r_y)$  vygeneruje náhodný šum, ktorý sa pripočíta k súradniciam každej častice. Čím väčší rozptyl šumu sa nastaví, tým budú častice rozptýlené do väčšieho priestoru a zároveň schopné trackovať rýchlejší pohyb.

Výsledky dvoch experimentov sú na nasledujúcich grafoch. Na y-ovú os je vynesená odchýlka od ideálnej trajektórie, čiže nižšia hodnota značí lepší výsledok.



Graf a)



Graf b)

Z grafov vidno, že optimálna hodnota rozptylu šumu je pre každú scénu jedinečná. Pre graf a) je to rozmedzie od 2 do 4, pre rýchlu scénu v grafe b) od 15 do 25. Príliš nízka hodnota šumu spôsobí stratenie sledovaného objektu (graf b) rozptyl 12 a menší), naopak príliš vysoký šum negatívne vplyva na presnosť.

#### 2.4. DYNAMICKÁ ZMENA ROZPTYLU ŠUMU A POČTU ČASTÍC

V prípade že rýchlosť sledovaného objektu nie je konštantná, je potreba parametre meniť dynamicky. Rozdiel súradníc sledovaného objektu z predchádzajúceho a aktuálneho snímku, spolu s ohodnotením najlepšej častice sa predajú funkcií, ktorá parametre nastaví na optimálne hodnoty. Pomalší pohyb resp. lepšie ohodnotenie častice znamená nastavenie nižšieho rozptylu šumu a počtu generovaných častíc. Pre rýchlejší pohyb je to naopak.

#### 2.5. PREKRÝVANIE OBJEKTOV

Program je schopný vysporiadať sa s javom, keď je sledovaný objekt dočasne prekrytý iným objektom. Algoritmus popisuje nasledujúci pseudokód:

```
1  IF(ohodnotenie najlepšej častice < prahová hodnota){
2      Spočítaj priemernú zmenu polohy objektu  $\Delta p$  z predchádzajúcich snímkov
3      Zvýš počet častíc na maximálnu hodnotu
4      WHILE (ohodnotenie najlepšej častice < prahová hodnota){
5          Posuň stred generovaných častíc o  $\Delta p$ 
6          Zvýš rozptyl šumu pre x-ové a y-ové súradnice polohy častíc
7          Prejdi na ďalší snímok
8          Vygeneruj a ohodnoť nové častice}}
```

### 3. ZÁVER

Zo zatiaľ implementovaných metód na ohodnotenie častíc vychádza najefektívnejšie sčítavanie rozdielov jednotlivých pixelov. Nižšia úspešnosť riešení s histogramami je pravdepodobne spôsobená menej kvalitnými testovacím videami.

Rozumný pomer kvality trackovania a doby behu aplikácie sa dá dosiahnuť použitím dynamickej zmeny parametrov. V niektorých scénach sa týmto spôsobom podarilo dosiahnuť viac ako 30% zrýchlenie aplikácie a to pri zachovaní kvality trackovania.

Významnou vlastnosťou programu je aj odolnosť voči prekrývaniu sa objektov. Vo väčšine testov algoritmus po znovuobjavení sa objektu úspešne pokračoval v trackovaní ďalej.

### LITERATÚRA

- [1] Musa, Z.B., Watada, J.: Motion Tracking Using Particle Filter, In: Knowledge-Based Intelligent Information and Engineering Systems, Volume 5179/2010, Heidelberg, 2010, (s. 119)
- [2] Li, P., Wang, H.: Object Tracking with Particle Filter Using Color Information, In: Computer Vision/Computer Graphics Collaboration Techniques, Volume 4418/2007, Heidelberg, 2010 (s. 534)