



# Detekce pohybu v dopravní úloze

---

**Cíl cvičení: Detekce pohybu v obraze, aktualizace modelu prostředí a demonstrace na modelu dopravní situace**

## 1 Teoretický úvod

### 1.1 Analýza pohybu

Analýza pohybu je oblastí počítačového vidění používanou ve velkém množství odvětví, od robotiky až po medicínu. Abychom mohli analyzovat pohyb, musíme si určit, co to z hlediska počítačového vidění je.

Při snímání objektu mohou nastat 4 případy:

1. kamera i objekt jsou v klidu,
2. kamera je v klidu, objekt se pohybuje,
3. kamera se pohybuje, objekt je v klidu,
4. kamera i objekt se pohybují.

Je jasné, že analyzovat pohyb je možné v případech 2 - 4.

Při pořízení snímku kamerou ztrácíme projekci na čip informaci o třetím rozměru a o informaci o změně polohy v čase nelze mluvit vůbec. Proto je pro analýzu pohybu použita sekvence snímků. Tehdy se mluví o tzv. dynamickém obraze. Jednotlivé snímky jsou zpracovávány jako statické a pak se mezi nimi vzájemně hledá informace o pohybu.

Mezi nejjednodušší úlohy pohybové analýzy patří detekce pohybu, tj. najít ty části obrazu, v nichž k pohybu dochází.

## 1.2 Rozdílové metody

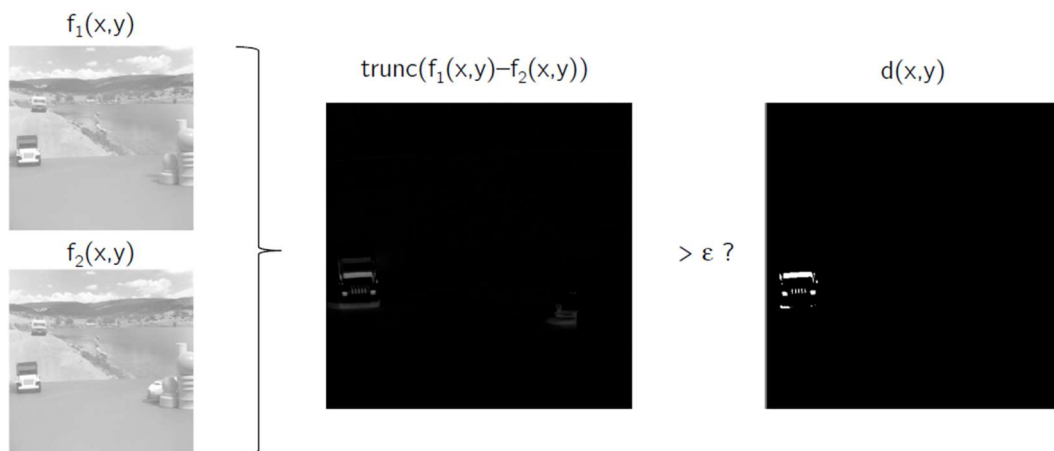
Rozdílové metody patří mezi jednoduché metody detekce pohybu. Jsou založeny na výpočtu rozdílu snímku ze sekvence. Pokud se jasové hodnoty odpovídajících si pixelů liší o více než je stanovený práh  $\epsilon$ , je na těchto pixelech detekován pohyb. Takto vzniklé snímky se nazývají rozdílové. Pro demonstraci je možné uvažovat sekvenci snímků  $\{f_1, f_1, \dots, f_n\}$ , pořázených v časech  $\{t_1, t_1, \dots, t_n\}$ . Z této sekvence je pak možné vytvořit následující rozdílové snímky:

- jednostranný
- oboustranný
- kumulovaný

**Jednostranný rozdílový snímek** – lze jej vypočítat podle vztahu:

$$d(x, y) = \begin{cases} 0 & \text{trunc}(f_1(x, y) - f_2(x, y)) < \epsilon \\ 1 & \text{trunc}(f_1(x, y) - f_2(x, y)) \geq \epsilon \end{cases}$$

Ze vztahu vyplývají i nedostatky jednostranného rozdílového snímku: vyhodnocuje pohyb jen v místech, kde  $f_1(x, y) > f_2(x, y)$  a nenesení informaci o směru pohybu.

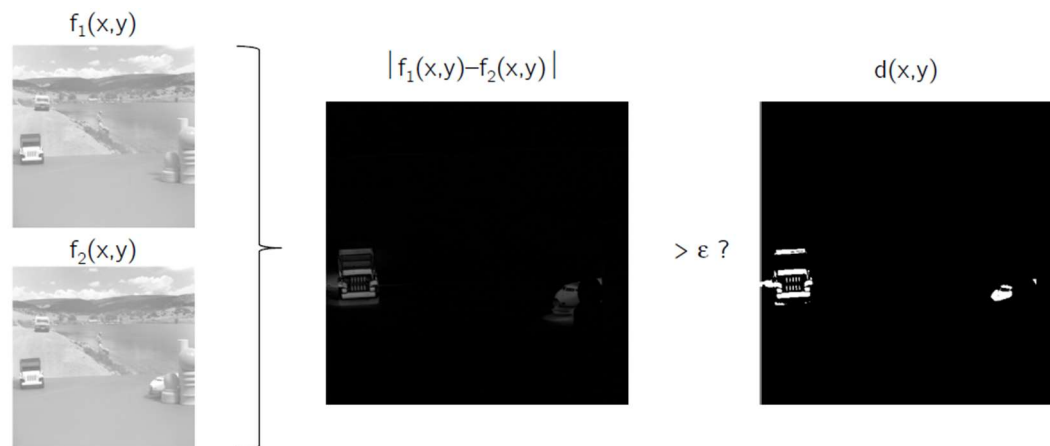


Obr. 1: Jednostranný rozdílový snímek

**Oboustranný rozdílový snímek** - definiční vztah pro oboustranný rozdílový snímek je následující:

$$d(x, y) = \begin{cases} 0 & |trunc(f_1(x, y) - f_2(x, y))| < \epsilon \\ 1 & |trunc(f_1(x, y) - f_2(x, y))| \geq \epsilon \end{cases}$$

Oboustranný rozdílový snímek odstraňuje první z nevýhod jednostranného rozdílového snímku a přináší záměnnost operandů  $f_1$  a  $f_2$ . Nicméně, stále nenese informaci o směru pohybu.

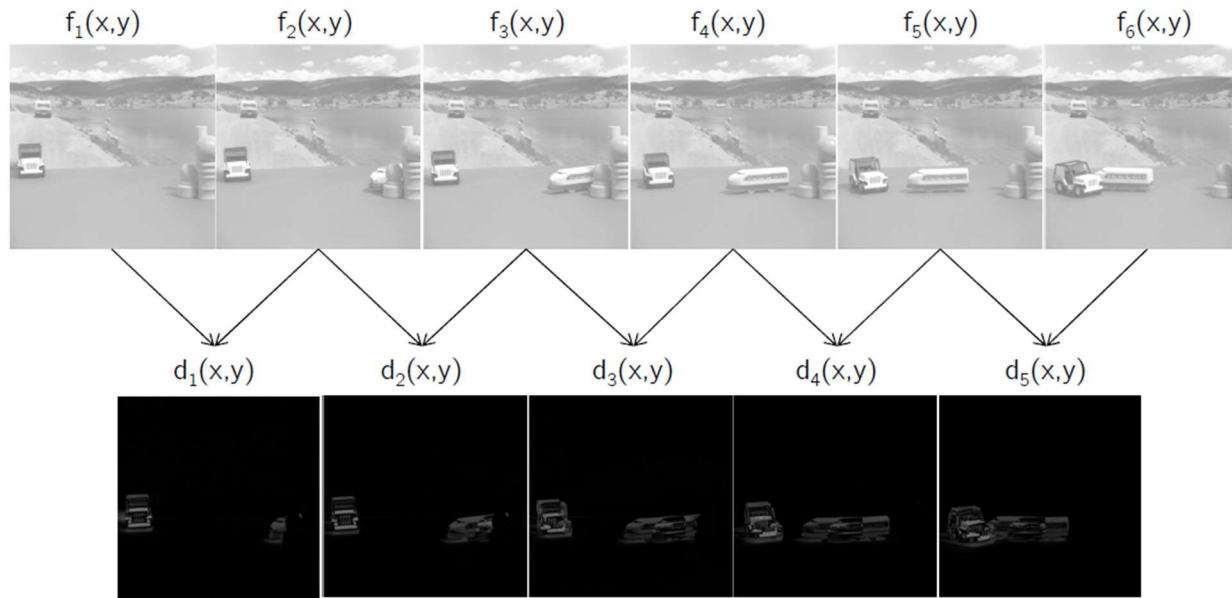


**Obr. 2: Oboustranný rozdílový snímek**

**Kumulovaný rozdílový snímek** - je vážený součet jednostranných nebo oboustranných rozdílových snímků v sekvenci. Přebírá tak jejich vlastnosti a navíc nese informaci o směru pohybu. Jeho definiční vztah:

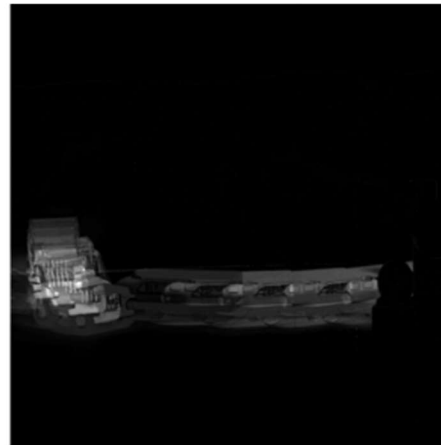
$$d_{kum}(x, y) = \sum_{i=1}^{N-1} \omega_i \cdot d_i(x, y)$$

Pro průměrovaný snímek je  $\omega = \frac{1}{N-1}$ .



Obr. 3: Kumulovaný rozdílový snímek – ukázka výpočtu

$$\sum_{i=1}^{N-1} w_i \cdot d_i(x,y) =$$



Obr. 4: Kumulovaný rozdílový snímek

### 1.3 Tvorba modelu prostředí

Statický model lze sestavit jako:

1. obraz čistého prostředí bez pohybujících se objektů,
2. průměrný/mediánový snímek série obrazů prostředí,
3. použít první snímek sekvence (naivní řešení).

Dynamický model  $b(x, y)$  je na začátku sekvence sestaven jako statický model a průběžně aktualizován podle iterativního algoritmu:

1. vypočti rozdílový snímek  $d_1(x, y)$  posledních dvou obrazů sekvence  $f_i(x, y)$  a  $f_{i-1}(x, y)$ .
2. pokud snímek  $d_1(x, y)$  vykazuje významnou změnu, scéna je dynamická, model prostředí se neaktualizuje, pokračuj výpočtem pro další snímek sekvence od bodu 1.
3. scéna je statická: vypočti rozdílový snímek  $d_2(x, y)$  obrazu  $f_i(x, y)$  a modelu  $b(x, y)$ .
4. pokud snímek  $d_2(x, y)$  nevykazuje významné lokální změny (scéna je statická bez objektů) aktualizuj model  $b(x, y)$ , např. pomocí lineárního zapomínání  $b(x, y) = \alpha \cdot f_i(x, y) + (1 - \alpha) \cdot b(x, y)$ , kde  $\alpha = 0,01$ .
5. pokračuj výpočtem pro další snímek sekvence od bodu 1.

## 2 Ovládání modelu

Model lze ovládat manuálně nebo automaticky. Režim lze zvolit páčkovým přepínačem na černé krabičce. Manuální ovládání se využívá pro ovladače zapojeného do vhodného bloku (kanál 1 ovládá černé vozidlo, kanál 2 červené vozidlo a kanál 3 modré policejní vozidlo).

Automatický režim naslouchá sériové komunikaci. Defaultně v automatickém režimu mají všechna autíčka rychlost 08. Pro komunikaci po sériové lince využijte program Herkules, záložku Serial, Name COM3, Boud 9600, Data size 8, Parity none, Handshake off, Mode free. Tlačítko **open** pro otevření komunikace.

Model přijímá příkaz s potvrzovací sekvencí. Potvrzovací sekvence se posílá hexadecimálně LFCR (0x0A0D). Je vhodné využít dvou řádků, do jednoho napsat příkaz do druhého potvrzovací sekvenci a u potvrzovací sekvence zaškrtnout HEX. Tabulka příkazů a možností:

Příkaz	Možnosti	Popis
CXSY	X – číslo auta (0 - 3), A pro všechny YY – rychlost auta (00 - 15)	Ovládání rychlosti auta
TSXPY	X – číslo semaforu (0 - 1), A pro všechny Y – program semaforu (-1 - 3)	Ovládání jednoho semaforu: -1 – semafor vypnut 0 – semafor červená 1 – semafor červená a oranžová 2 – semafor zelená 3 – semafor oranžová
TCPY	Y – program semaforů (-1 - 7)	Ovládání semaforů na křižovatce: -1 – semafor vypnut 0 – semafor 1 červená, semafor 2 červená 1 – semafor červená a oranžová, semafor 2 červená 2 – semafor zelená, semafor 2 červená 3 – semafor oranžová, semafor 2 červená
TAE		Zapnutí automatického ovládání semaforu
TAD		Vypnutí automatického ovládání semaforu
H		Nápověda

### 3 Seznam vybavení

1. 2x kamera Manta MG125B
2. model dopravní úlohy
3. laboratorní zdroj

### 4 Návod

1. Zapněte PC a spusťte na ploše program Vimba Viewer.
2. V programu nastavte vhodně kameru. Můžete do kamery nahrát konfigurační soubor *camera\_settings.xml*.
3. Program vypněte, zapněte MATLAB a otevřete předpřipravený skript.
4. Pomocí příkazu *gigecamlist* vyhledejte připojenou kameru a případně upravte nastavení kamery v záhlaví skriptu.
5. Ovládání modelu je popsáno v sekci 2.
6. Zvolte si vhodnou rozdílovou metodu.
7. Určete práh pro binarizaci rozdílového snímku (například z histogramu). Určete práh pro detekci pohybu.
8. Doplňte kód pro detekci pohybu.
9. Vhodně zobrazte průběh z kamery a snímek s detekovaným pohybem.
10. Rozšiřte funkčnost algoritmu o druhou kameru a realizujte úsekové měření rychlosti.
11. Pro jednu kameru řešte úlohu detekce pohybu s dynamickou aktualizací modelu prostředí.
12. Zkuste funkčnost algoritmů s jiným modelem autíčka. Případné zhoršení/zlepšení vysvětlete.
13. Výsledky prezentujte vyučujícímu. Zařízení vypněte a uklidte pracoviště.