

# Detektory oblastí

Karel Horák



Rozvrh přednášky:

1. Úvod.
2. Shi&Tomasi.
3. FAST.

# Detektory oblastí

Karel Horák



Rozvrh přednášky:

- 1. Úvod.**
2. Shi&Tomasi.
3. FAST.

# Úvod

- ▶ Detektory významných bodů (rohů - corner features) a oblastí (blob features):
  - ▶ Moravcův operátor
  - ▶ Harrisův detektor
  - ▶ Shi&Tomasi (GFT = Good Features to Track)
  - ▶ FAST (Features from Accelerated Segment Test)
  - ▶ BRISK (Binary Robust Invariant Scalable Keypoints)
  - ▶ FREAK (Fast Retina Keypoint)
  - ▶ HOG (Histogram of Oriented Gradients)
- ▶ GLOH (Gradient Location and Orientation Histogram)
- ▶ BRIEF (Binary Robust Independent Elementary Features)
- ▶ ORB (Oriented FAST and Rotated BRIEF)
- ▶ MSER (Maximally Stable Extremal Regions)
- ▶ SIFT (Scale-Invariant Feature Transform)
- ▶ SURF (Speeded-Up Robust Features)

# Úvod

## ► Literatura chronologicky:

- [1] Moravec, H. P. Towards Automatic Visual Obstacle Avoidance. 1977. (Moravec<sup>2</sup>)
- [2] Harris, C., Stephens, M. A Combined Corner and Edge Detector. 1988. (Harris<sup>2</sup>)
- [3] Shi, J., Tomasi, C. Good Features to Track. 1994. (Shi-Tomasi<sup>1</sup>)
- [4] Matas, J., et al. Robust wide baseline stereo from maximally stable extremal regions. 2002. (MSER<sup>2</sup>)
- [5] Lowe, D. G. Distinctive Image Features from Scale-Invariant Keypoints. 2004. (SIFT<sup>2</sup>)
- [6] Mikolajczyk, K., Schmid, C. A performance evaluation of local descriptors. 2005. (GLOH<sup>3</sup>)
- [7] Rosten, E., Drummond, T. Fusing Points and Lines for High Performance Tracking. 2005. (FAST<sup>1</sup>)
- [8] Dalal, N., Triggs, B. Histograms of oriented gradients for human detection. 2005. (HOG<sup>3</sup>)
- [9] Bay, H., Ess, A., Tuytelaars, T., Gool, L. Van. SURF:Speeded Up Robust Features. 2008. (SURF<sup>2</sup>)
- [10] Calonder, M., et al. BRIEF:Binary Robust Independent Elementary Features. 2010. (BRIEF<sup>3</sup>)
- [11] Leutenegger, S., Chli, M., Siegwart, R. BRISK: Binary Robust Invariant Scalable Keypoints. 2011. (BRISK<sup>3</sup>)
- [12] Rublee, E., et al. ORB: An efficient alternative to SIFT or SURF. 2011. (ORB<sup>3</sup>)
- [13] Alahi, A., Ortiz, R., Vandergheynst, P. FREAK: Fast Retina Keypoint. 2012. (FREAK<sup>3</sup>)

► <sup>1</sup> probráno v tomto materiálu

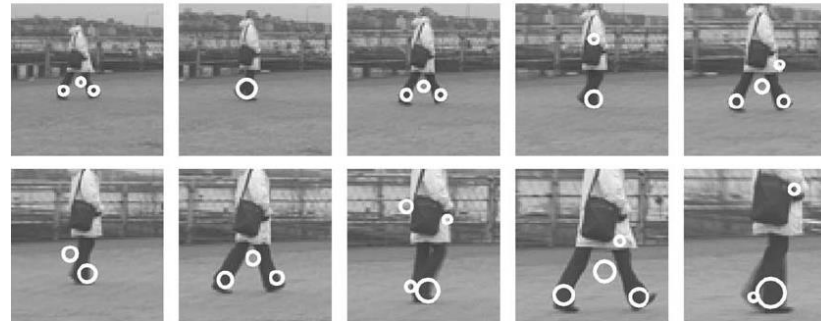
► <sup>2</sup> probráno v jiném materiálu

► <sup>3</sup> neprobráno

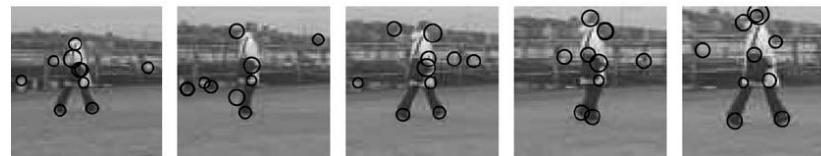
# Úvod

- Co od významných bodů očekáváme?
- Stabilita – nezávislost detekce na obecné afinní transformaci (translace, rotace, zkosení, měřítko),

*Spatio-temporal interest points*



*Spatial interest points*



- Opakovatelnost – detekce stejného znaku ve dvou a více různých snímcích stejné scény



# Detektory a deskriptory oblastí

Karel Horák



Rozvrh přednášky:

1. Úvod.
- 2. Shi&Tomasi.**
3. FAST.
4. MSER.
5. SIFT.
6. SURF.

# Shi&Tomasi

- ▶ Shi&Tomasi (někdy odkazováno jako Kanade&Tomasi)
- ▶ Nejde o prostý detektor významných bodů na jednom statickém obrazu (popř. dvou po sobě jdoucích obrazech pro nalezení korespondencí), ale o komplexní techniku pro sledování trajektorie významných bodů ve videosekvenci.
- ▶ Nový pojem – kvalita významného bodu: ne každý významný bod ve statickém obrazu je významný v sekvenci obrazů pro sledování.
- ▶ Důvod? Detekovaný významný bod nemusí nutně reprezentovat fyzický bod ve scéně.

Given Sequence



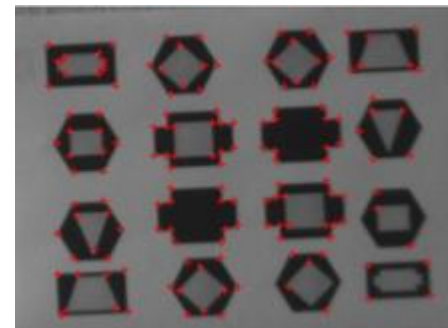
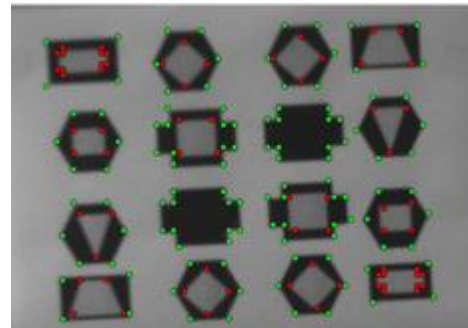
Antenna and mirror support bar create a feature which does not correlate to a real-world feature



# Shi&Tomasi

- ▶ Významné body stabilní nejen ve statickém obraze, ale i dalších obrazech sekvence jsou pak ad-hoc nazývány Good Features to Track (GFT).
- ▶ Dvě třídy úloh, kde lze GFT použít:
  - 1) sledování pohybujícího se objektu ve scéně - kamera fixní nebo plovoucí (vektory pohybu se sčítají)
  - 2) rekonstrukce 3D prostoru/objektu při pohybující se kameře
- ▶ Metoda Shi&Tomasi může být rozdělena na dvě fáze:
  1. detekce významných bodů ve statickém obraze (používá podobný formalismus jako Harris)
  2. filtrace GFT
- ▶ Filosofie metody: pro každý sledovaný znak/rys (feature) je v průběhu videosekvence monitorována jeho kvalita pomocí tzv. nepodobnosti = rozdíl vzhledu znaku v aktuálním snímku a vzhledu původního znaku (např. z prvního snímku sekvence) -> pokud nepodobnost překročí práh, je daný znak z dalšího sledování vypuštěn

- ▶ Testovací obrazec s rohy typu L, T, X a Y:
- ▶ 180 bodů = 112 vnější (zelená)  
+ 70 vnitřní (červená)
- ▶ Obraz vpravo = odezva Shi-Tomasi





# Shi&Tomasi

► Ad 1. detekce významných bodů:

► Podobná matematika jako u Harrise – symetrická matice 2x2 kvadrátů gradientů:

$$C(x, y) = \begin{bmatrix} \sum_w I_x^2(u, v) & \sum_w I_x(u, v) \cdot I_y(u, v) \\ \sum_w I_x(u, v) \cdot I_y(u, v) & \sum_w I_y^2(u, v) \end{bmatrix}$$

► Podmínky na významný bod:

a) matice musí být nad úroveň šumu (šum je v podstatě nespolehlivý významný bod)

b) matice musí být dobře podmíněná (viz podmíněnost matice)

Ad a) – obě vlastní čísla  $\lambda_1$  a  $\lambda_2$  matice musí být dostatečně velká

Ad b) – obě vlastní čísla matice se navzájem nesmí lišit o několik řádů (pak by to nebyl bidirekcionál jako např. roh nebo pepř&sůl vzor, ale hrana nesoucí informaci o pohybu pouze v jednom směru – unidirekcionál)

► Prakticky navíc platí úvaha – pokud je menší z vlastních čísel nad úroveň šumu (podmínka 1), je matice zpravidla zároveň i dobře podmíněná (podmínka 2), protože kvůli maximální přípustné jasové hodnotě pixelů (např. 255) nemůže být hodnota většího vlastního čísla libovolně vysoká.

# Shi&Tomasi

► Ad 1. detekce významných bodů:

► Z toho tedy vyplývá, že jako GFT uznáme okno kolem bodu, pro nějž platí:

$$\min(\lambda_1, \lambda_2) > \lambda$$

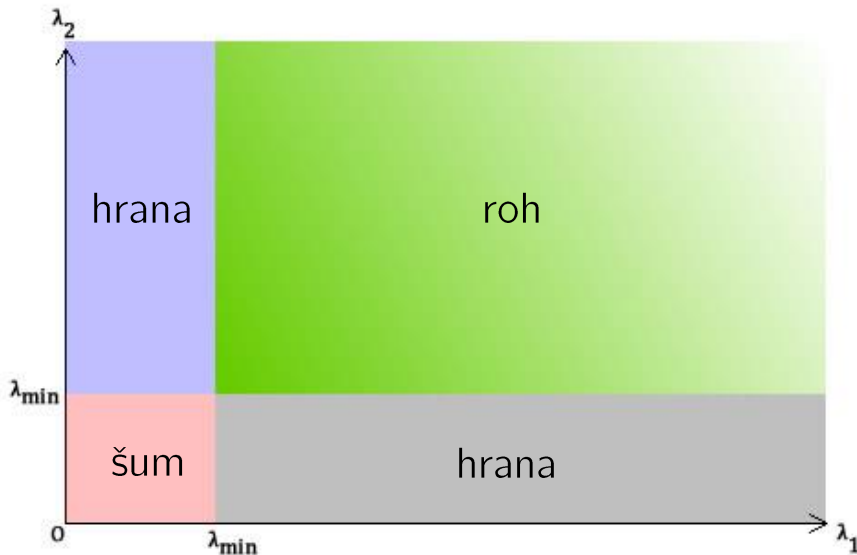
► kde  $\lambda$  je předdefinovaný práh.

► Rozdíl v implementaci Shi&Tomasi vs. Harris:

$$H(\lambda_1, \lambda_2) = \lambda_1 \cdot \lambda_2 - \kappa \cdot (\lambda_1 + \lambda_2)^2 = \det(C) - \kappa \cdot \text{trace}(C)^2$$

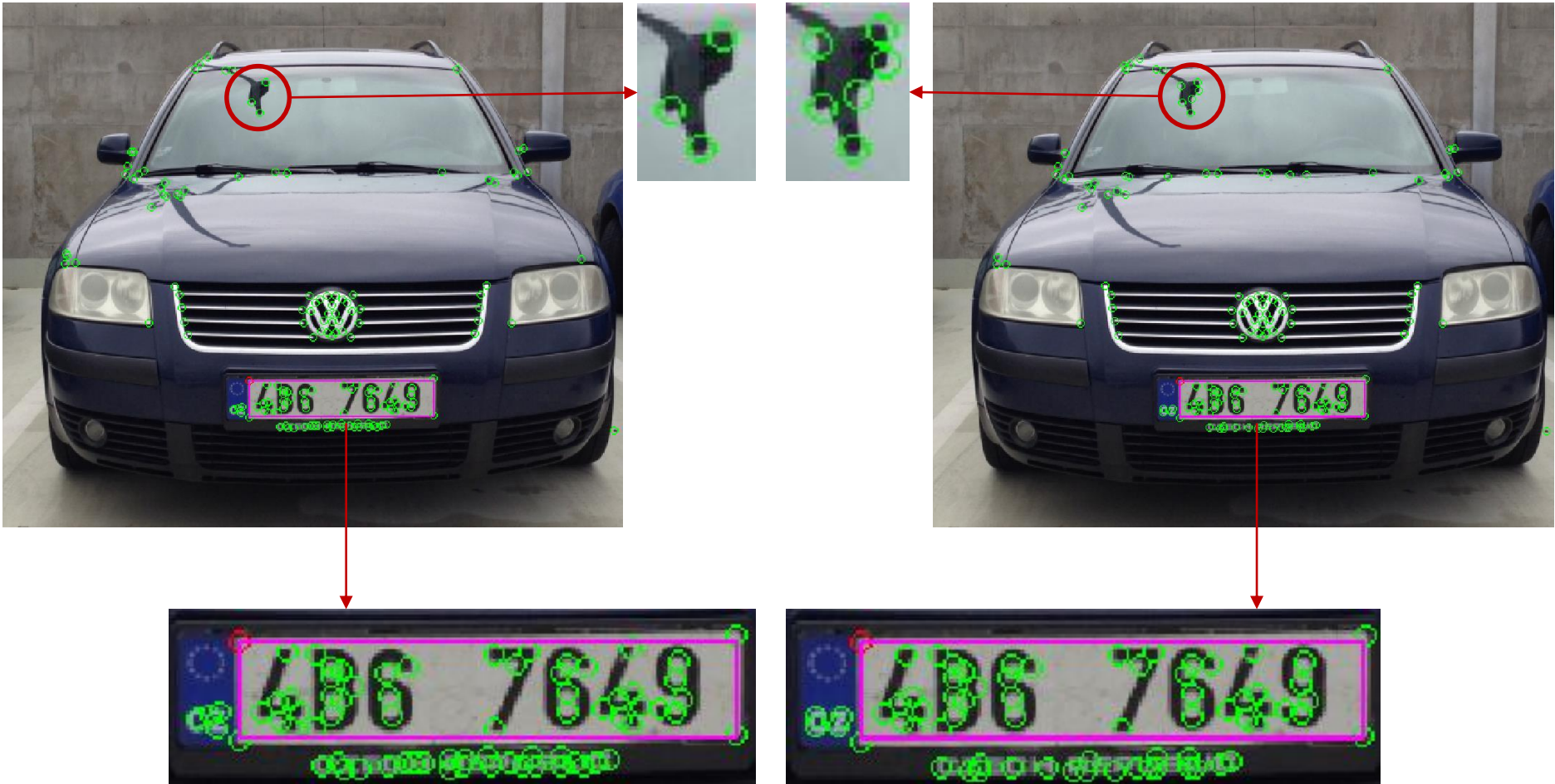


Narozdíl od Shi&Tomasi není nutné vůbec počítat vlastní čísla matice – stačí vyjádřit determinant a stopu.



# Shi&Tomasi

► Detekční rozdíl Shi&Tomasi (vlevo) vs. Harris:



# Shi&Tomasi

- Ad 2. Filtrace GFT:

- Předpoklad pro změnu scény/obrazu za čas  $\tau$ :

$$I(x, y, t + \tau) = I(x - \xi(x, y, t, \tau), y - \eta(x, y, t, \tau))$$

- Slovy: změnu obrazové funkce uvažujeme pouze jako translaci v ose x a ose y danou funkcí  $\delta$ :

$$\delta = (\xi, \eta)$$

- tj. vůbec nepředpokládáme změnu jasu pixelů – stín, překrytí, odraz/lom světla apod. – to je **extrémně silný předpoklad**, který nás záhy zastaví (\*).

- Posun  $\delta$  je aproximován pomocí afinní transformace (rotace, měřítko, zkosení, translace):

$$\delta = D\mathbf{x} + \mathbf{d}$$

$$D = \begin{bmatrix} d_{xx} & d_{xy} \\ d_{yx} & d_{yy} \end{bmatrix}$$

deformační matice  
(afinní transformace)

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$$

polohový vektor

$$\mathbf{d} = \begin{bmatrix} d_x \\ d_y \end{bmatrix}$$

vektor translace  
(střed okna znaku)

- Pozn.: afinita = spřízněnost, podobnost (afinní transformace nemění topologii tj. blízké body jsou transformovány opět na blízké body)

# Shi&Tomasi

► Ad 2. Filtrace GFT:

► Pak lze snadno vyvodit, že změna polohy bodu  $\mathbf{x}$  z obrazu I bude v obrazu J dána:

$$J(A\mathbf{x} + \mathbf{d}) = I(\mathbf{x})$$

► kde  $A = \mathbf{1} + D$  ( $\mathbf{1} = 2 \times 2$  jednotková matice)


► Sledování bodu  $\mathbf{x}$  lze pak chápat jako výpočet šesti parametrů (4 v D, dva v  $\mathbf{d}$ ) v každé iteraci – tento odhad parametrů se děje z hodnot v určitém okolí (okně)  $W$  kolem bodu  $\mathbf{x}$ .

► Problém 1 – velikost okna: u příliš malého okna se špatně/nepřesně počítají parametry afinní transformace, u velkého je zase výpočet dlouhý.

► Problém 2 – ideální model: na reálných datech není rovnice nahoře splnitelná a výpočet šesti parametrů je velmi nepřesný a nestabilní. (\*)

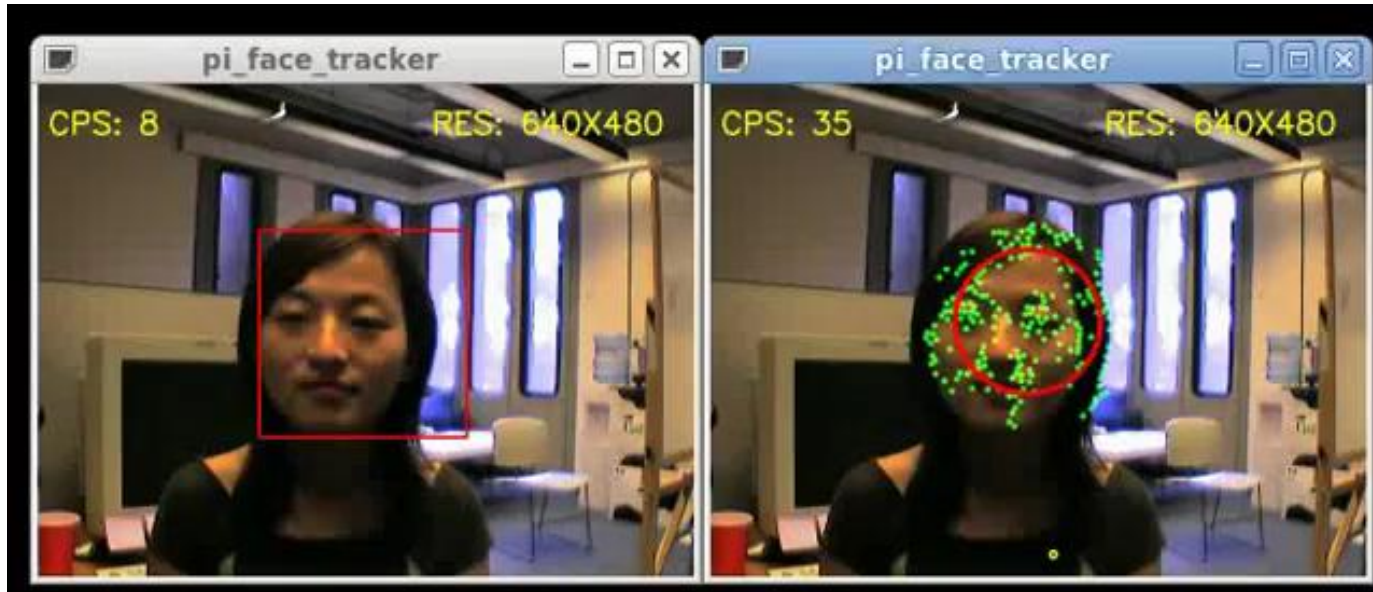
► Proto se úloha odhadu pohybu předefinuje jako minimalizace odchylky (nepodobnosti) J a I:

$$K = \iint_W [J(A\mathbf{x} + \mathbf{d}) - I(\mathbf{x})]^2 w(\mathbf{x}) d\mathbf{x}$$


 váhová funkce  
(binární nebo Gauss)

# Shi&Tomasi

- Demontrace rozdílu použití Haarových příznaků a GFT pro detekci obličeje ve videu:



↑  
Haarovy příznaky

↑  
Good Features to Track + optický tok

pozn.: na začátku je potřeba jednou nalézt objekt (obličej)

- Stejným způsobem pracuje sledování libovolného objektu, u nějž známe GFT.

# Shi&Tomasi

- Ukázka: detektor rohu Shi-Tomasi – implementace v OpenCV 3.0 beta:

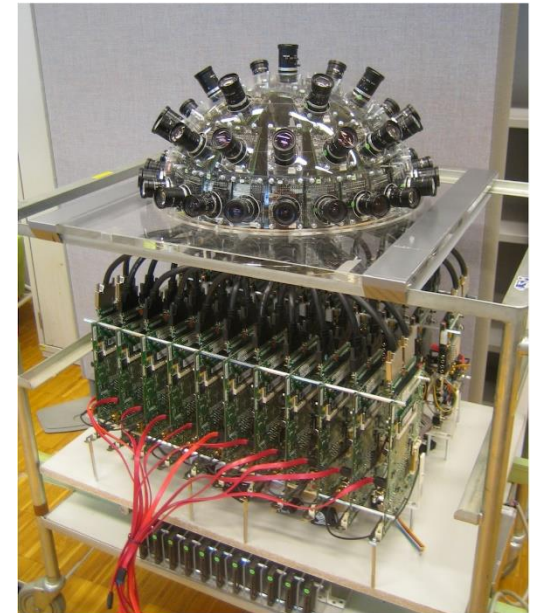
The image shows a screenshot of a Linux desktop environment. On the left, a terminal window is open, displaying the prompt `jhair@jhair: ~/Documents/Opencv_tutorial` and the command `python opencv_tutorial_85x48.py`. The terminal output is mostly black, indicating a video recording. On the right, the SimpleScreenRecorder application window is open, showing recording settings. The window title is "SimpleScreenRecorder" and it is in "Recording" mode. The "Start recording" button is highlighted. The "Enable recording hotkey" checkbox is checked, and the hotkey is set to "Ctrl + R". The "Enable sound notifications" checkbox is unchecked. The "Information" section shows: Total time: 0:00:00, FPS in: 0.00, FPS out: 0.00, Size in: 1373x871, Size out: ?, File name: ?, File size: 0 B, and Bit rate: 0 bps. The "Preview" section shows a "Preview frame rate" of 10 and a note: "Note: Previewing requires extra CPU time (especially at high frame rates)". The "Log" section shows: "[PageRecord::StartPage] Starting page ..." and "[PageRecord::StartPage] Started page.". At the bottom, there are buttons for "Cancel recording" and "Save recording".

# Shi&Tomasi

- Ukázka: výpočet optického toku v úloze všesměrového (omnidirekcionálního) vidění – významné body pro korespondenci jsou získány pomocí Shi-Tomasi detektoru.



- Pro zajímavost – jak vypadá omnidirekcionální kamera?



- Courtesy of: Allied Vision Technologies / Point Grey / EPFL



# Detektory oblastí

Karel Horák

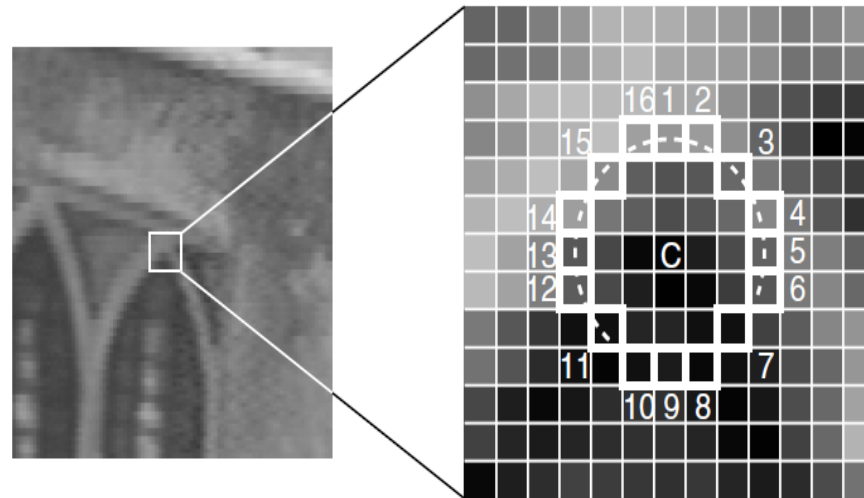


Rozvrh přednášky:

1. Úvod.
2. Shi&Tomasi.
- 3. FAST.**

# FAST

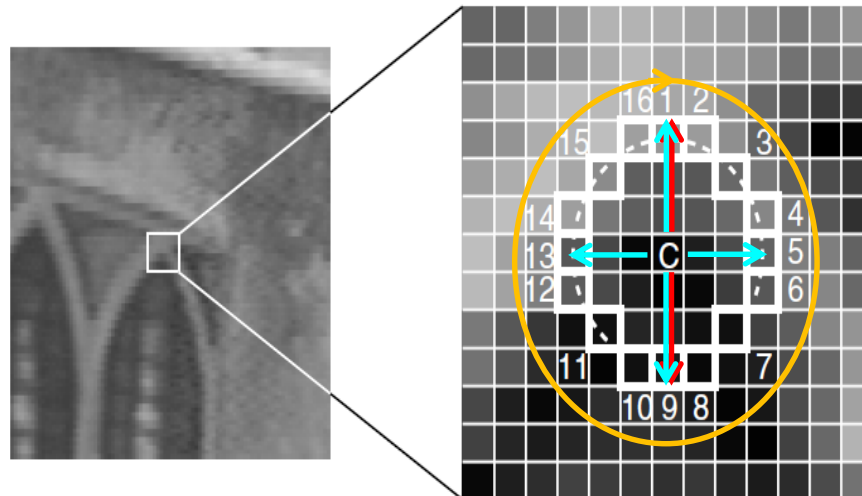
- FAST = Feature from Accelerated Segment Test
- Patří do třídy algoritmů AST (Accelerated Segment Test), které jsou orientovány na nízkou výpočetní náročnost.
- Princip: pro určení, zda bod C je významným bodem je zkoumána rozdílnost jasových hodnot 16 pixelů ležících na kružnici o poloměru 3 kolem bodu C.



- Pokud jsou jasové hodnoty určitého počtu pixelů  $n \in \langle 1, 16 \rangle$  současně všechny větší nebo současně všechny menší o předem daný práh  $t$ , je bod C prohlášen za významný.
- Obvykle se jako hodnota oddělující výraznou hranu od rohu uvádí  $n = 12$  (hledáme tedy  $n \geq 12$ ).

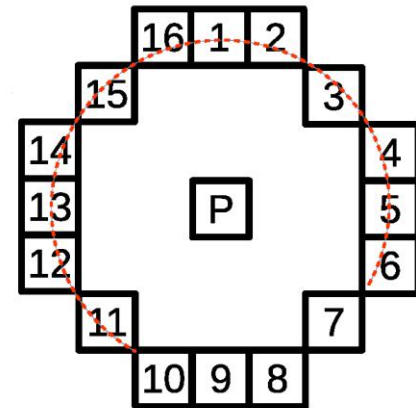
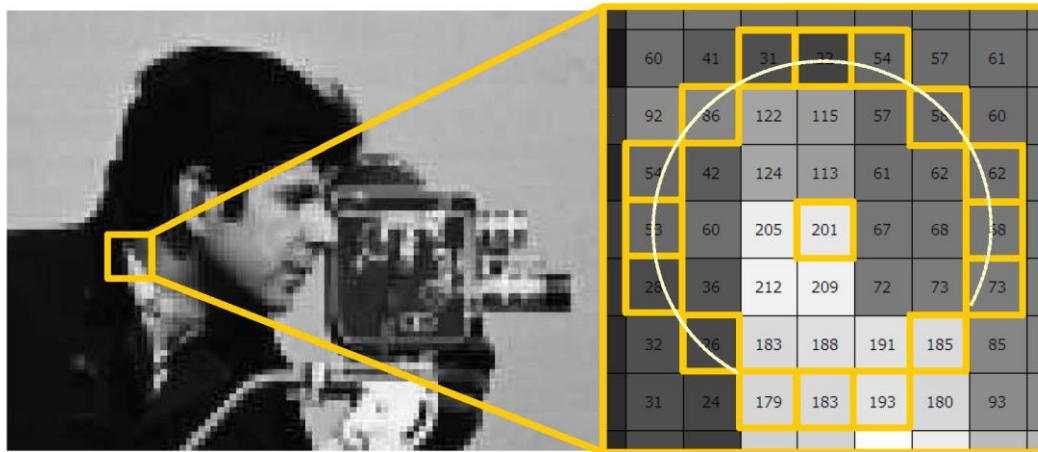
# FAST

- ▶ Algoritmus musí být rychlý tj. záleží na pořadí testování rozdílů jasových hodnot pixelů  $P_1$  až  $P_{16}$  a postup je samozřejmě deterministický.
- ▶ Pomocí experimentu s rozhodovacím stromem ID3 byla nalezena výpočetně optimální sekvence testování pixelů (nejkratší strom).
- ▶ Kroky algoritmu (k následujícímu kroku se přechází pouze při splnění předchozího - bod C tedy může být významným bodem pouze tehdy, projdeme-li úspěšně všechny body algoritmu):
  1. jasová hodnota alespoň jednoho pixelu z ( $P_1, P_9$ ) leží mimo interval  $\langle I_C-t, I_C+t \rangle$
  2. jasové hodnoty alespoň tří pixelů z ( $P_1, P_5, P_9, P_{13}$ ) leží mimo interval  $\langle I_C-t, I_C+t \rangle$
  3. porovnání ostatních hodnot  $\langle P_1:P_{16} \rangle$  s C a testování podmínky  $n \geq 12$



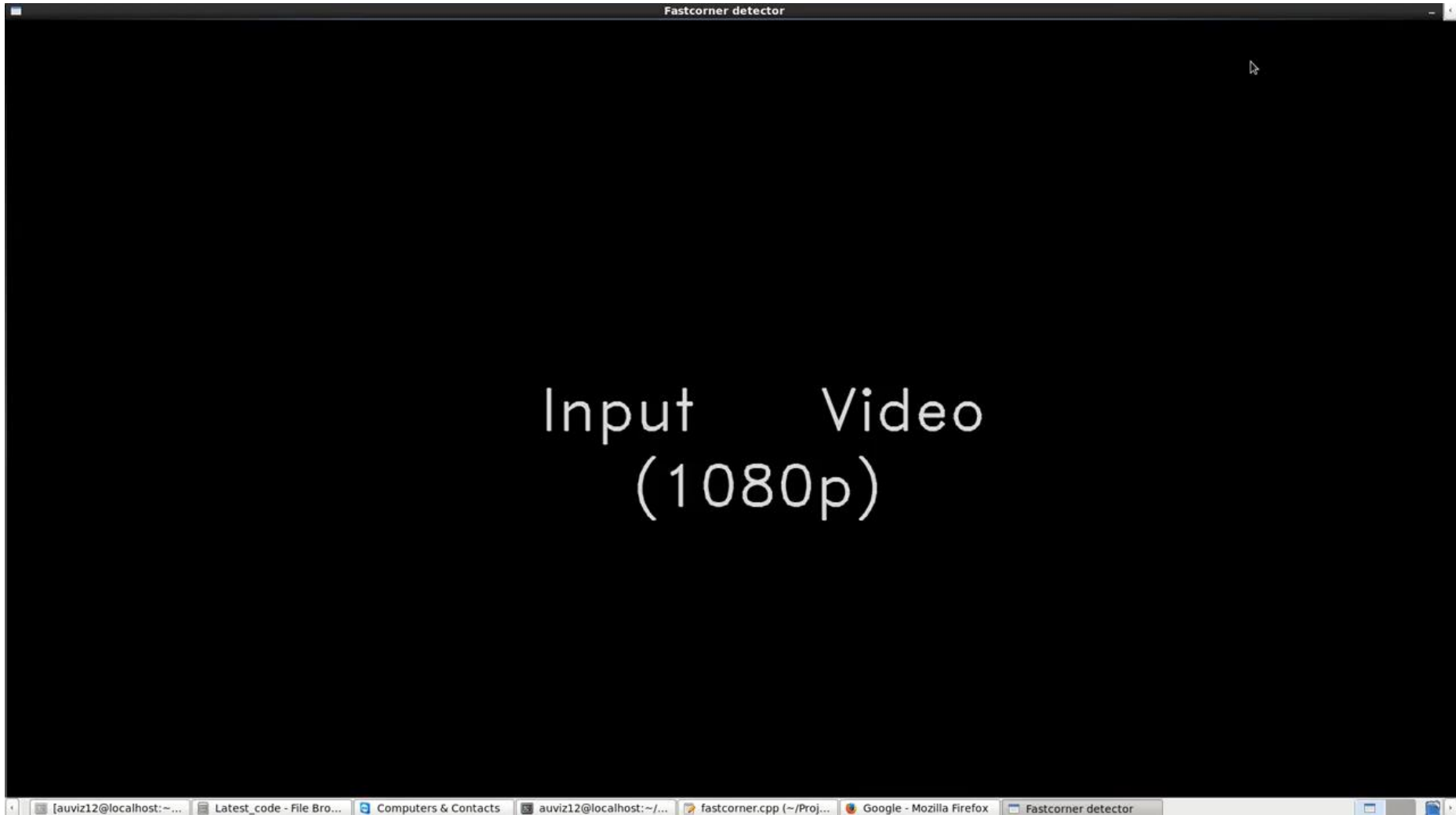
# FAST

• K algoritmu:



# FAST

- Ukázka – detekce významných bodů v dopravě:



# FAST

► Poznámky k rychlosti FAST:

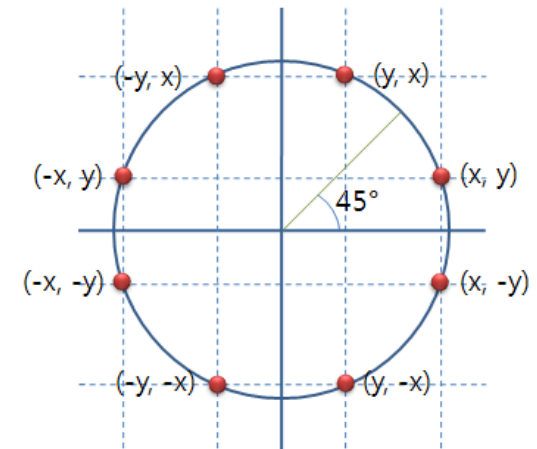
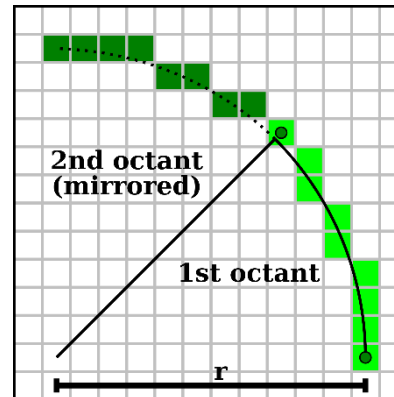
► Poloměr  $r=3$  je specificky používaný u algoritmu FAST a je pro něj stanoveno optimální pořadí testování pixelů (viz předchozí slide), nicméně teoreticky i prakticky lze implementovat i jiné  $r$ .

► Pro výpočet souřadnic  $(x,y)$  bodů ležících na kružnici se zpravidla používá Bresenhamův algoritmus:

1. ve spodní polovině prvního kvadrantu (první oktant) spočítej pro všechna přípustná  $x$  odpovídající  $y$
2. zrcadlením hodnot (záměnou souřadnic  $x$  a  $y$ ) do druhého oktantu kompletuj první kvadrant
3. postupnou záměnou znamének souřadnic  $x$  a  $y$  kompletuj 2., 3. a 4. kvadrant

$$f_{circle}(x, y) = (x - x_c)^2 + (y - y_c)^2 - r^2$$

$$f_{circle}(x, y) = \begin{cases} < 0 & \text{inside the circle boundary} \\ = 0 & \text{on the circle boundary} \\ > 0 & \text{outside the circle boundary} \end{cases}$$



► FAST-ER modifikace: FAST + Enhanced Repeatability, větší opakovatelnost

# FAST

- Srovnání výpočetní náročnosti různých detektorů významných bodů (video 640x480 / 30 FPS)

Detector	Training set		Test set	
	Pixel rate (MPix/s)	%	MPix/s	%
FAST $n = 9$	188	4.90	179	5.15
FAST $n = 12$	158	5.88	154	5.98
Original FAST ( $n = 12$ )	79.0	11.7	82.2	11.2
FAST-ER	75.4	12.2	67.5	13.7
SUSAN	12.3	74.7	13.6	67.9
Harris	8.05	115	7.90	117
Shi-Tomasi	6.50	142	6.50	142
DoG	4.72	195	5.10	179