

# Hledání úseček a kružnic s využitím Houghovy transformace při zpracování obrazu v LabView

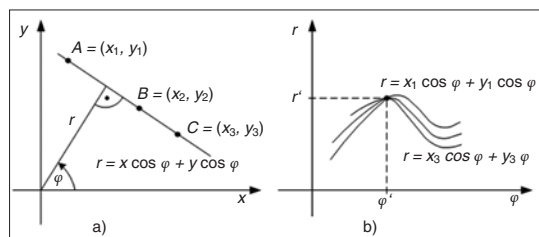
Jaroslav Vlach

Při zpracování obrazové informace se používá množství metod, jejichž základy byly položeny již koncem 50. let minulého století a jejichž rozvoj nastal zejména v posledních 25 letech. V tomto článku jsou uvedeny některé metody a postupy zpracování obrazu sejmутého kamerou. Pro příklad je uvedena realizace těchto algoritmů s použitím nástrojů programového prostředí LabView. Článek volně navazuje na [10].

There are a lot of methods used for image processing. Their grounds were found in the end of 50th years and have been developed particularly in the last 25 years. In this article, some methods for processing of images taken by camera are explained. As example, implementation of these algorithms with using of LabView tool is described. The article follows freely [10].

## 1. Úvod

Při zpracování obrazové informace v průmyslu se nejčastěji pracuje s dvourozměrnou projekcí obrazu snímané reality a jeho interpretací v podobě obrazové funkce  $f(x, y)$ . Každý prvek o souřadnici  $(x, y)$  nese informaci o jednom diskretním prvku obrazu – pixelu (z angl. *picture element*). S určitou mírou zjednodušení budou v dalším popisu uvažovány scény nasnímané jasovou interpretací,



Obr. 1. K výkladu Houghovy transformace pro hledání úseček

takže každý prvek obrazu je popsán funkční hodnotou obrazové funkce  $f(x, y)$  ve tvaru celého čísla, které vyjadřuje velikost jasu. Množinu funkčních hodnot obrazové funkce lze pak vyjádřit dvourozměrným polem – maticí, s níž lze dále pracovat pomocí matematických funkcí pro práci s maticemi.

Při zpracování obrazu se pracuje s funkcemi a nástroji umožňujícími s dostatečnou přesností a rychlostí vykonat potřebné operace s obrazovou funkcí  $f(x, y)$ . Základní operace jsou např. prahování, hledání hran, filtrace nebo vyhlazení. Cílem práce je použít takovou operaci, s jejíž pomocí je dosaženo požadované nebo hledané vlastnosti obrazu.

Často se při práci s obrazem pracuje jen v určité vybrané oblasti obrazu, ve specifikovaném okolí zvoleného bodu (např. 4-okolí, 8-okolí, okolí 3 × 3 apod.) nebo s ohraničenou oblastí zájmu – ROI (z angl. *Region Of Interest*).

## 2. Segmentace

Segmentace je jedním z nejdůležitějších kroků analýzy obrazu. Jde o postup, kterým se v obraze vybere určitá část chápaná jako objekt. Obraz se tedy rozdělí do částí, které korespondují s konkrétními objekty v obraze. Informaci o rozdělení obrazu do jednotlivých segmentů využívají algoritmy zpracování obrazu, které se snaží porozumět obsahu obrazu.

Pro segmentaci obrazu existuje mnoho segmentačních algoritmů. Nejpoužívanější (a nejnázornější) jsou metody založené na prahování (angl. *threshold*). Příkladem je prahování podle úrovně jasu, které patří k nejjednodušším a také nejstarším segmentačním metodám. Prahování je funkce, která transformuje vstupní obraz  $f$  na výstupní binární obraz  $g$  podle rozhodovacího vztahu:

$$g(i, j) = 1 \text{ pro } f(i, j) \geq T$$

$$g(i, j) = 0 \text{ pro } f(i, j) < T \quad (1)$$

kde  $T$  je prahová hodnota (práh).

Jestliže  $g(i, j) = 1$ , bod  $(i, j)$  je obrazovým prvkem objektu, a jestliže  $g(i, j) = 0$ , obrazovým prvkem je pozadí.

Kromě metod prahování lze pro segmentaci obrazu dále využívat metody založené na detekci významných hran v obraze (angl. *edge-based*). Lokální hrany jsou detekovány pomocí hranových detektorů na základě rozdílu hodnot jasu sousedních bodů (pixelů). Hranový detektor je algoritmus, který vyhledává množinu hran (bodů, pixelů) v obraze.

Dalšími metodami jsou metody založené na hledání oblastí v obraze (angl. *region-based*). Jestliže lze identifikovat hrany, měly by tyto teoreticky ohraničovat oblasti. Kontury oblastí však mohou být porušeny, nemusí ohraničovat celou oblast. Není také zaručeno, že hranice oblastí nalezené metodou detekce hran budou stejné jako ty nalezené metodou hledání oblastí. Pro složitější úlohy segmentace lze použít další metody založené na statistické analýze obrazových dat, tzv. znalostní metody (angl. *knowledge-based*), a další.

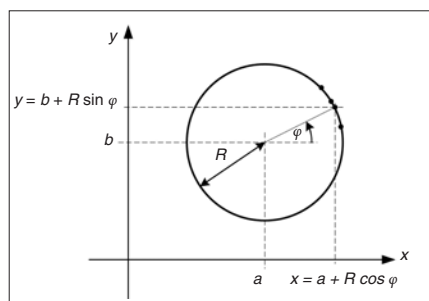
V reálných obrazech je často třeba pracovat s liniemi, které jsou přerušeny (z důvodu šumu, po prahování nebo detekci hran apod.). V těchto případech je třeba použít metody, které si umějí poradit s chybějícími pixely.

Jednou z používaných metod pro hledání jednoduchých útvarů, jako je úsečka, elipsa či kružnice, je Houghova transformace. Princip metody publikoval P. V. C. Hough v roce 1959 a patentoval ji v roce 1962; v roce 1972 zobecněnou metodu publikovali R. O. Duda a P. E. Hart [3]. Významné použití této transformace je při analýze textu, kdy lze s její pomocí nalézt řádek textu v obraze.

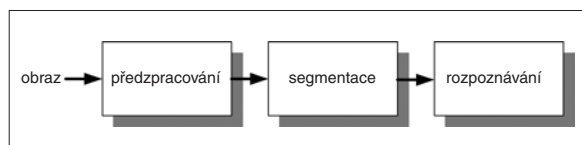
Úlohu pro Houghovu transformaci je možné formulovat jako hledání takové podmnožiny bodů v obraze, která co nejvíce odpovídá části přímky – úsečky. Přímka s vyznačenými body  $A, B$  a  $C$  je znázorněna na obr. 1a. Každý bod na přímce je popsán dvěma souřadnicemi, např.  $A = (x_1, y_1)$ . Rovnice přímky se vyjádří v polárních souřadnicích:

$$r = x \cos \varphi + y \sin \varphi \quad (2)$$

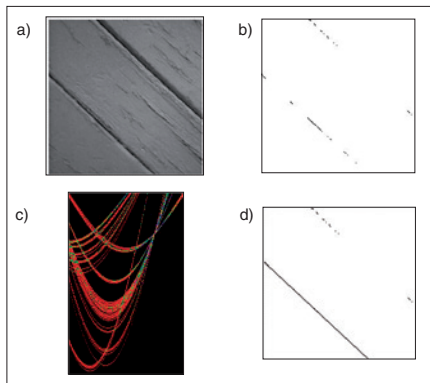
kde  $r$  je délka normály od přímky k počátku souřadnic,



Obr. 2. K výkladu Houghovy transformace pro hledání kružnice



Obr. 3. Základní kroky zpracování obrazu



Obr. 4. Příklad Houghovy transformace pro hledání úsečky: a) původní vstupní obraz, b) vstupní obraz po prahování ( $T = 35$ ), c) promítnutí akumulátoru  $A(\varphi_i, r_j)$  do roviny  $(\varphi, r)$ , d) obraz po prahování s vloženou nalezenou úsečkou

$\varphi$  úhel mezi normálou a osou  $x$ .

Je patrné, že pro bod  $A$  přejde z rovnice (2) na tvar:

$$r = x_1 \cos \varphi + y_1 \sin \varphi \quad (3)$$

Podobné vztahy je možné vyjádřit i pro další body  $B$  a  $C$  na přímce. Poté, po zobrazení křivek popisujících jednotlivé body  $A$ ,  $B$  a  $C$  v souřadné soustavě  $(\varphi, r)$  se zjistí, že se protínají v jednom bodě o souřadnicích  $(\varphi', r')$ , jak je znázorněno na obr. 1b.

Obvykle se Houghova transformace implementuje tak, že obraz se diskretizuje v rasteru  $M \times N$ . Každý prvek tohoto prostoru bude pak tvořen dvojicí souřadnic  $(\varphi_i, r_j)$ , kde  $i = 1, 2, \dots, M$  a  $j = 1, 2, \dots, N$ . Algoritmus Houghovy transformace pro hledání úsečky v obraze lze popsat např. takto:

1. vstupem je binární obraz  $f$ , zajímaví nás hodnoty obrazové funkce  $f(x_k, y_k) = 1$ , kterých je celkem  $K$ ,
2. vytvoříme pole  $A$  o velikosti  $M \times N$  (budeme mu říkat akumulátor) a na počátku je vynulujeme:  $A(\varphi_i, r_j) = 0$  pro všechna  $i = 1, 2, \dots, M$  a  $j = 1, 2, \dots, N$  (zvolíme vhodné dělení, např.  $\varphi_i = i \cdot \pi / M$ ,  $r_j = j(r_{\max} - r_{\min}) / N$ ),
3. nastavíme počítadlo  $j = 1$ ,
4. nastavíme počítadlo  $i = 1$ ,
5. pro každý pixel  $(x_k, y_k)$ , kde  $k = 1, 2, \dots, K$ , jehož hodnota jasu  $f(x_k, y_k) = 1$ , vypočteme hodnotu:  $r_j = x_k \cos \varphi_i + y_k \sin \varphi_i$ ,
6. inkrementujeme hodnotu v akumulátoru:  $A'(\varphi_i, r_j) = A(\varphi_i, r_j) + 1$ ,
7. opakujeme pro všechna další  $i = 2 \dots M$  od kroku 5,
8. opakujeme pro všechna další  $j = 2 \dots N$  od kroku 4,
9. nyní, po průchodu celým obrazem jsou v akumulátoru  $A(\varphi_i, r_j)$  hodnoty  $n_{ij}$ , které určují počet nalezených bodů ležících na přímce dané parametry  $(\varphi_i, r_j)$ ,
10. největší hodnota  $n_{ij}$  (tj. maximum všech hodnot) určuje parametry  $(\varphi_i, r_j)$  přímky, na které se nachází nejvíce bodů v obraze.

Obdobný postup lze modifikovat pro hledání parametrů hranic objektů, které je možné popsat analytickou rovnicí. Příkladem je hledání kružnice popsané rovnicí:

$$(x - a)^2 + (y - b)^2 = R^2 \quad (4)$$

Každý bod na kružnici o poloměru  $R$  a se středem v bodě  $(a, b)$  lze popsat podle obr. 2 souřadnicemi:

$$\begin{aligned} x &= a + R \cos \varphi \\ y &= b + R \sin \varphi \end{aligned} \quad (5)$$

Bude-li se v obraze hledat bod ležící na kružnici s daným poloměrem  $R$ , vypočítá se tedy jeho souřadnice a zjistí se hodnoty parametrů  $a$  a  $b$  podle vztahu:

$$\begin{aligned} a &= x - R \cos \varphi \\ b &= y - R \sin \varphi \end{aligned} \quad (6)$$

Všechny body se stejnou hodnotou parametrů  $a$  a  $b$  budou pak ležet na dané kružnici.

Mějme nyní obraz diskretizován v rasteru  $M \times N$ , každý prvek tohoto prostoru bude tvořen dvojicí souřadnic  $(x_i, y_j)$ , kde  $i = 1, 2, \dots, M$  a  $j = 1, 2, \dots, N$ . Algoritmus Houghovy transformace pro hledání kružnice v obraze lze popsat např. takto:

1. vstupem je binární obraz  $f$ , zajímaví nás hodnoty obrazové funkce  $f(x_k, y_k) = 1$ , kterých je celkem  $K$ ,
2. zvolíme hodnotu  $R$  poloměru hledané kružnice,
3. vytvoříme pole  $A$  o velikosti  $M \times N$  (budeme mu říkat akumulátor) a na počátku je vynulujeme:  $A(x_i, y_j) = 0$  pro všechna  $i = 1, 2, \dots, M$  a  $j = 1, 2, \dots, N$ ,
4. nastavíme počítadlo  $j = 1$ ,
5. nastavíme počítadlo  $i = 1$ ,
6. nastavíme  $\varphi_m = 0$ ,
7. pro každý pixel  $(x_k, y_k)$ , kde  $k = 1, 2, \dots, K$ , jehož hodnota jasu  $f(x_k, y_k) = 1$ , vypočteme hodnoty  $a$  a  $b$ :  $a = x_k - R \cos \varphi_m$ ,  $b = y_k - R \sin \varphi_m$ ,
8. jestliže bude logický výraz  $(a > 0) \& (a < M) \& (b > 0) \& (b < N) = 1$ , inkrementujeme hodnotu v akumulátoru:  $A'(a, b) = A(a, b) + 1$ ,
9. opakujeme pro všechna  $\varphi_m$  do hodnoty  $2\pi$  od kroku 7,
10. opakujeme pro všechna další  $i = 2 \dots M$  od kroku 6,

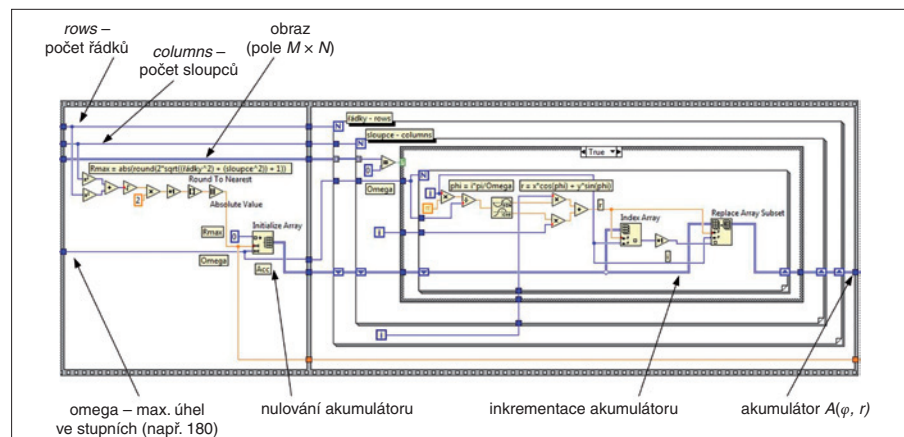
11. opakujeme pro všechna další  $j = 2 \dots N$  od kroku 5,
12. nyní, po průchodu celým obrazem jsou v akumulátoru  $A(x_i, y_j)$  hodnoty  $n_{ij}$ , které určují počet nalezených kružnic se středem v bodě  $(x_i, y_j)$  o poloměru  $R$ ,
13. největší hodnota  $n_{ij}$  (tj. maximum všech hodnot) je středem hledané kružnice v obraze.

Lze tušit, že pro dosažení vysoké míry úspěšnosti bude zřejmě záležet na vhodné úpravě vstupního obrazu (resp. vstupní obrazové funkce  $f$ ), která v již uvedených algoritmech není řešena. Významným krokem je prahování (viz vztah (1)) a dalším hledání hran v obraze. Nejčastějšími metodami pro detekci hran jsou:

- metody založené na hledání maximálních derivací pomocí operátorů (Robertův, Prewittův, Sobelův), významným nástrojem pro aplikaci těchto metod je konvoluce,
- metody založené na hledání průchodu druhých derivací nulou, angl. *zero-crossing* – ZC, příkladem je Marrův-Hildrethův operátor a Cannyho hranový detektor,
- metody založené na lokální aproximaci obrazové funkce parametrickým modelem, např. polynomem dvou proměnných (Harralick).

V roce 1980 zveřejnil David Marr společně s Ellen Hildrethovou [6] teorii popisující matematický model detekce skokových hran při neurofyziologickém měření na sítnici oka. Základem teorie je hledání polohy hrany v obraze v místě průchodu druhé derivace obrazové funkce nulou. První derivace obrazové funkce má v místě hrany své lokální maximum, druhá derivace v místě hrany protíná nulovou hodnotu (odtud *zero-crossing*).

Významné místo mezi moderními metodami hledání hran v obraze zaujímá Cannyho hranový detektor, který publikoval John Canny v roce 1986 [1]. Není bez zajímavosti, že tato práce vznikla na stejném pracovišti jako Marrova teorie. Základní myšlenkou je představa, že skokovou hranu lze hledat filtrem (hledání nejlepší impulzní funkce filtru).



Obr. 5. Algoritmus Houghovy transformace pro hledání úsečky v prostředí LabView

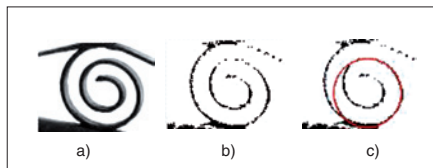
Detektor je optimální pro skokové hrany, když jsou splněna tři kritéria:

- detekční kritérium – významné hrany nesmí být přehlédnuty a na jednu hranu by neměly být vícenásobné odezvy,
- lokalizační kritérium – rozdíl skutečnou a nalezenou polohou hrany má být minimální,
- požadavek jedné odezvy – detektor nesmí reagovat na jednu hranu v obraze vícenásobně pro zašuměné a nehladké hrany.

Principem Cannyho hranového detektoru je syntéza z odezev detektoru v různých měřítkách. Algoritmus lze potom popsat např. v bodech:

1. najdi přibližné směry gradientu,
2. pro každý pixel najdi derivaci ve směru gradientu pomocí „optimální“ masky spojující vyhlazení a derivaci,
3. najdi lokální maxima těchto derivací,
4. hranové body získáš prahováním s hysterezí,
5. proved' syntézu hran získaných pro různé veliké vyhlazení (málokdy se používá).

Řešení konkrétní úlohy zpracování obrazu je možné obvykle rozdělit do tří základních kroků (viz obr. 3): vstupní obraz je podroben předzpracování (např. diskretizace, prahování),



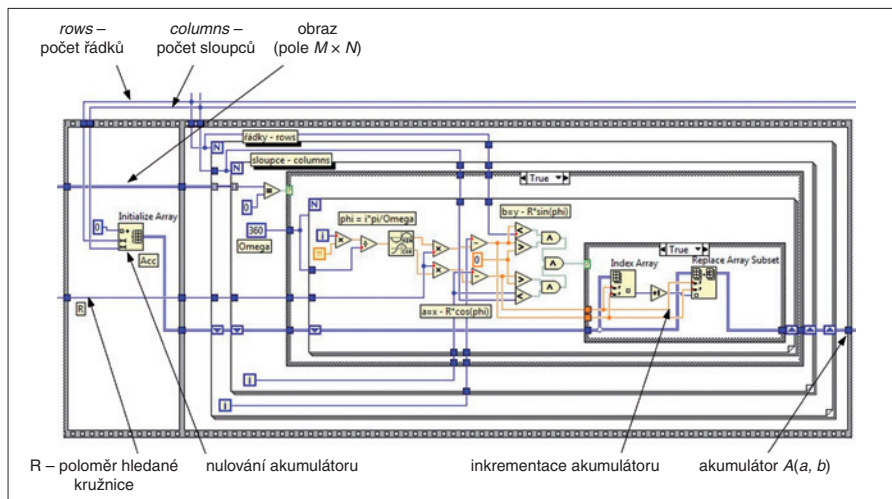
Obr. 6. Příklad Houghovy transformace pro hledání kružnice: a) původní vstupní obraz, b) obraz po prahování ( $T = 35$ ), c) obraz po prahování s vloženou nalezenou kružnicí ( $R = 29$ )

následně segmentaci (hledání oblastí a objektů v obraze) a nakonec vlastnímu rozpoznávání obrazu. Pro každou část lze použít několik metod, algoritmů a konkrétních předem vytvořených postupů v daném programovém prostředí. Vždy je však třeba počítat s tím, že každá úloha může mít své specifické vlastnosti, které vedou k množství změn, úprav a doladění konkrétních postupů a algoritmů. Jinak se bude přistupovat k úloze počítání objektů v obraze, jinak k úloze posuzování shody tvaru se vzorem (angl. *pattern matching*), jinak k úloze měření rozměru snímaného objektu v jednom směru, jinak k úloze měření ve více směrech a jinak k úloze posuzování povrchových či podpovrchových vlastností objektu v obraze. Ukazuje se také, že velmi důležitou problematikou je optimální nasvícení scény.

### 3. Ukázky řešení v prostředí LabView

Nyní pro názornost uvedme příklad řešení některých úloh v programovém prostředí LabView. Pro zájemce je zde odkaz na publikaci [9] seznamující se základy programování v tomto prostředí.

V první úloze se bude ve vstupním obraze hledat úsečka. Program realizuje algoritmus



Obr. 7. Algoritmus Houghovy transformace pro hledání kružnice o poloměru  $R$

Houghovy transformace pro hledání linie. Pro jednoduchost se bude vstupní obraz (jde o šedotónový obraz s osmibitovým rozlišením, kde hodnota 0 odpovídá černé a hodnota 255 bílé barvě pixelu) upravovat pouze prahováním, což nemusí vždy vést k dobrému výsledku. Na obr. 4a je vstupní obraz a na obr. 4b obraz po prahování s prahovou hodnotou  $T = 35$ . Na obr. 4c je znázorněno promítnutí akumulátoru  $A(\phi_i, r_j)$  do roviny  $(\phi, r)$ . Poznamenejme, že oproti obr. 1d se počátek souřadného systému nachází vlevo nahoře (osa  $\phi$  je zde otočena o  $180^\circ$ ). Na obr. 4d je znázorněn obraz po prahování s vloženou úsečkou nalezenou za použití Houghovy transformace.

Řešení v prostředí LabView je graficky velmi podobné jednotlivým krokům z obr. 3. Zde bude předpokládáno, že vstupní obraz je již diskretizován do šedotónové podoby. Dále byla pro názornost použita segmentace prahováním podle předpisu podle vztahu (1). Vlastní rozpoznávání v uváděném případě realizuje algoritmus Houghovy transformace (obr. 5).

V další úloze je úkolem ve vstupním obraze nalézt kružnici daného poloměru. V tomto případě je realizován algoritmus Houghovy transformace pro hledání kružnice. Opět se bude pracovat se vstupním obrazem s osmibitovým šedotónovým rozlišením, který se bude segmentovat prahováním. Na obr. 6a je uveden vstupní obraz, na obr. 6b obraz po prahování s prahovou hodnotou  $T = 35$  a na obr. 6c je obraz s vloženou nalezenou kružnicí (se zvolenou hodnotou  $R = 29$ ).

Při realizaci v prostředí LabView bude opět předpokládáno, že vstupní obraz je již diskretizován do šedotónové podoby a rovněž se bude segmentovat prahováním. Vlastní rozpoznávání v uváděném případě realizuje algoritmus Houghovy transformace pro hledání kružnice o poloměru  $R$  (obr. 7).

### 4. Závěr

Hlavním cílem článku bylo seznámit čtenáře s použitím Houghovy transformace v praxi a s její realizací v prostředí LabView.

V příloženém seznamu odkazů lze získat další informace a podněty k práci.

#### Literatura:

- [1] CANNY, J.: *A Computational Approach to Edge Detection*. In: IEEE Trans. Pattern Analysis and Machine Intelligence, 1986, s. 679–698. Dostupné na <<http://www.icst.pku.edu.cn/course/ImageProcessing/2009/resource/canny1986.pdf>>.
- [2] DOBEŠ, M.: *Zpracování obrazu a algoritmy v C#*. BEN Praha, 2008, ISBN 978-80-7300-233-6.
- [3] DUDA, R. O. – HART, P. E.: *Use of the Hough Transformation to Detect Lines and Curves in Pictures*. In: Comm. ACM, January, 1972, Vol. 15, s. 1–15. Dostupné na <<http://www.ai.sri.com/pubs/files/tn036-duda71.pdf>>.
- [4] HLAVÁČ, V. – ŠONKA, M.: *Počítačové vidění*. GRADA, Praha, 1992.
- [5] HLAVÁČ, V. – SEDLÁČEK, M.: *Zpracování signálů a obrazu*. Skripta FEL ČVUT, Praha, 2007, ISBN 978-80-01-03110-0.
- [6] MARR, D. – HILDRETH, E.: *Theory of Edge Detection*. Proceedings of the Royal Society of London. Series B, Biological Sciences, Feb. 29, 1980, Vol. 207, No. 1167, s. 187–217. Dostupné na <<http://www.cim.mcgill.ca/~langer/646/PAPERS/MarrHildreth80.pdf>>.
- [7] SONKA, M. – HLAVAC, V. – BOYLE, R.: *Image Processing, Analysis, and Machine Vision*. Thomson Learning, Toronto 2008. Dostupné na <<http://www.icaen.uiowa.edu/~dip/LECTURE/lecture.html>> (odkaz na části staršího vydání).
- [8] VLACH, J.: *Začínáme s LabView*. Sdělovací technika, 4/2008, s. 20–21, ISSN 0035-9942.
- [9] VLACH, J. a kol.: *Začínáme s LabView*. BEN Praha, 2008, ISBN 978-80-7300-245-9.
- [10] VLACH, J.: *Základy zpracování obrazu v prostředí LabView*. Automatizace, 52, č. 1, s. 40–41, ISSN 0005-125X.

#### Webové odkazy:

National Instruments – odkaz <[www.ni.com/czech](http://www.ni.com/czech)>

Ing. Jaroslav Vlach