



CENTER FOR
MACHINE PERCEPTION



CZECH TECHNICAL
UNIVERSITY IN PRAGUE

DIPLOMOVÁ PRÁCE

Adaptivní modelování a sledování objektů ve videosekvencích

Jan Konečný

jan.konecny@gmail.com, svobodat@fel.cvut.cz, zimmerk@cmp.felk.cvut.cz

CTU–CMP–2008–03

18. ledna 2008

Lze získat na
ftp://cmp.felk.cvut.cz/pub/cmp/articles/zimmerk/konecny_msc2008.pdf

Školitel: Tomáš Svoboda, Karel Zimmermann

Práce je podpořena Akademií věd České Republiky projektem
1ET101210407.

Research Reports of CMP, Czech Technical University in Prague, No. 3, 2008

Published by

Centrum strojového vnímání, Katedra kybernetiky
Fakulta elektrotechnická ČVUT
Technická 2, 166 27 Praha 6

fax: (02) 2435 7385, tel: (02) 2435 7637, www: <http://cmp.felk.cvut.cz>

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: Jan Konečný

Obor: Technická kybernetika

Název tématu: Adaptivní modelování a sledování objektů ve videosekvencích

Zásady pro vypracování:

Jednou z hlavních oblastí výzkumu v počítačovém vidění je sledování objektů ve videosekvencích [1]. Objekt je obvykle popsán konstantním modelem, který umožňuje v každém snímku určovat jeho polohu. Pokud objekt změní svoji podobu (např. změna tvaru úst), jeho model už není použitelný a dojde k selhání sledovacího procesu.

1. Seznamte se s problematikou sledování a modelování a navrhnete metodu 2D sledování, která se umí adaptovat na pomalé změny podoby objektu a metodu ověřte na reálných datech. Využijte metody sledování v reálném čase pomocí lineárního regresoru [2, 3], která je v současnosti vyvíjena na katedře kybernetiky.
2. Tuto metodu rozšířte o schopnost adaptace na změny podoby sledovaného objektu tak, aby byla stále využitelná v reálném čase.

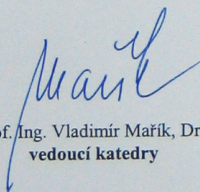
Seznam odborné literatury:

- [1] Baker, S. and Matthews, I.: *Lucas-Kanade 20 Years On: A Unifying Framework*. International Journal of Computer Vision, 2004.
- [2] Matas, J.; Zimmermann, K.; Svoboda, T.; Hilton, A.: *Learning Efficient Linear Predictors for Motion Estimation*. Proceedings of 5th Indian Conference on Computer Vision, Graphics and Image Processing, 2006.
- [3] Jurie, F.; Dhome, M.: *Real Time 3D Template Matching*. International Conference on Computer Vision, 2001.
- [4] Hartley, R. and Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

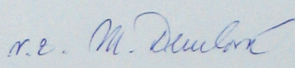
Vedoucí diplomové práce: Ing. Tomáš Svoboda, Ph.D.

Termín zadání diplomové práce: letní semestr 2005/2006 (změna zadání: 28.06.2007)

Termín odevzdání diplomové práce: leden 2008


prof. Ing. Vladimír Mařík, DrSc.
vedoucí katedry




doc. Ing. Boris Šimák, CSc.
děkan

V Praze dne 29.06.2007

Anotace

Navrhujeme metodu vhodnou pro sledování objektů, které mění svůj vzhled. Je založena na predikci pohybu pomocí tzv. lineárního prediktoru (LLiP), což je předem naučené zobrazení, které pozorovaným intenzitám přiřazuje pohyb objektu. Vzhledem k tomu, že se podoba objektu v jedné kameře může měnit (například změna tvaru nerigidního objektu), rozšiřujeme LLiP na parametrický lineární prediktor (dále jen PLLiP), který je parametrizován aktuálním vzhledem objektu. Při sledování objektu v obrázkové sekvenci PLLiP nejdříve zarovná body objektu s aktuálním snímkem a potom transformuje pozorovaný vzhled objektu do vektoru parametrů, které přímo ovlivní funkci PLLiPu v následujícím snímku. Experimentálně ukazujeme, že chyba PLLiP je menší než chyba LLiP, který nevyužívá parametrickou informaci o vzhledu objektu.

Annotation

We propose tracking method which is based on Linear Predictor (LLiP) suitable for objects with variable appearance. Linear Predictor is a learned linear mapping between observed intensities and motion. Since object appearance may change (e.g. due to non-rigid deformation) we propose Parametric Linear Predictor (PLLiP), which is adjusted by the appearance. PLLiP tracking algorithm first aligns the object with current image data and then estimates actual appearance parameters that adjust PLLiP for the next image. We experimentally show that PLLiP has lower motion estimation error than LLiP which does not use extended information of actual object appearance.

Prohlášení

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v příloženém seznamu.

V Praze dne 18. ledna 2008

podpis

Poděkování

Rád bych poděkoval:
Tomáši Svobodovi a Karlu Zimmermannovi za vedení diplomové práce.

Adaptivní modelování a sledování objektů ve videosekvencích

Jan Konečný

18. ledna 2008

Obsah

1	Úvod	3
2	Stav poznání	5
2.1	Sledování	5
2.2	Kanade Lucas Tracker (KLT)	5
2.3	Frederic Jurie - Lineární prediktor (LLiP)	9
2.3.1	Trénovací množina LLiP	9
2.3.2	Učení LLiP	11
2.3.3	Algoritmus sledování LLiP	12
3	Vlastní práce	13
3.1	Přínos práce	13
3.2	Sledování objektů měnících vzhled	14
3.3	Parametrický lineární prediktor (PLLiP)	15
3.4	Trénovací množina PLLiP	16
3.5	Učení PLLiPu	18
3.5.1	Učení pro jeden skalární vzhledový parametr	19
3.6	Odhad vzhledových parametrů	20
3.7	Výpočetní složitost PLLiP a LLiP	21
3.8	Sekvenční parametrický lineární prediktor (SPLLiP)	21
4	Experimenty	24
4.0.1	Predikce translace	24
4.0.2	Predikce vzhledového parametru	26
4.0.3	Zpětnovazební zapojení PLLiPů	29
4.0.4	Zhodnocení experimentů	32
5	Závěr	33
6	Přílohy	34
6.1	Příložené CD	34

6.1.1	Digitální verze diplomové práce	34
6.1.2	Zdrojové kódy matlabu	34
6.1.3	Materiály - sekvence obrázků	35

Kapitola 1

Úvod

Počítačové vidění je v době informačních technologií a automatizovaného řízení jedním z dynamicky se rozvíjejících oborů. Jeho využití najdeme v zabezpečovací technice při detekci osob v hlídaném prostoru, v řídicích procesech a v autonomních robotických systémech, v oblasti dopravy při rozpoznání kradených vozidel nebo jako součást radarových a mýtných systémů Obr.1.1¹. Další oblastí jsou bezpilotní letadla a bezposádkové vozidla, navigační a zaměřovací jednotky. Velké uplatnění nalézá v zábavním průmyslu, kde se zaznamenává lidský pohyb a mimika tváře pro vytvoření počítačových modelů, které jsou použity k produkci animovaných filmů nebo počítačových her. V lékařství pomáhá počítačové vidění diagnostikovat snímané orgány. Dále máme systémy pro rozpoznání a čtení textu. Počítačové vidění je souborem různorodých aplikací, které se z pohledu zpracování a využití obrazové informace navzájem liší.

Podstatnou úlohou v počítačovém vidění je hledání vzájemných korespon-



Obrázek 1.1: Rozpoznání SPZ vozidel, detekce lidských obličejů, 3D modelování

dencí mezi dvěma obrázky. Pokud se jedná o sekvenci obrázků, mluvíme o sledování. Tímto dokážeme vytvořit pohybový záznam pro rekonstrukci crash

¹Obrázky převzaty ze <http://cmp.felk.cvut.cz/demos/>

testu automobilu, interpretaci řeči hluchoněmých, automatický dohled osob ve věznicích nebo získání trajektorie plavající pokusné krysy, viz Obr.1.2². Některé z aplikací sledování nepotřebují rychlou odezvu, například off-line vyhodnocení deformačního testu automobilu, a některé rychlost přímo vyžadují, jako komunikační rozhraní mezi počítačem a člověkem. Právě u problémů sledování v reálném čase nás složitost algoritmů a nedostatečný výpočetní výkon nutí zdokonalovat, zrychlovat a zjednodušovat již známé algoritmy. V této práci se zabýváme lineárním prediktorem, který je náročný ve fázi učení a velmi rychlý v procesu sledování.



Obrázek 1.2: Sledování plavání krysy pro vědecké účely

²Obrázky převzaty ze <http://cmp.felk.cvut.cz/demos/>

Kapitola 2

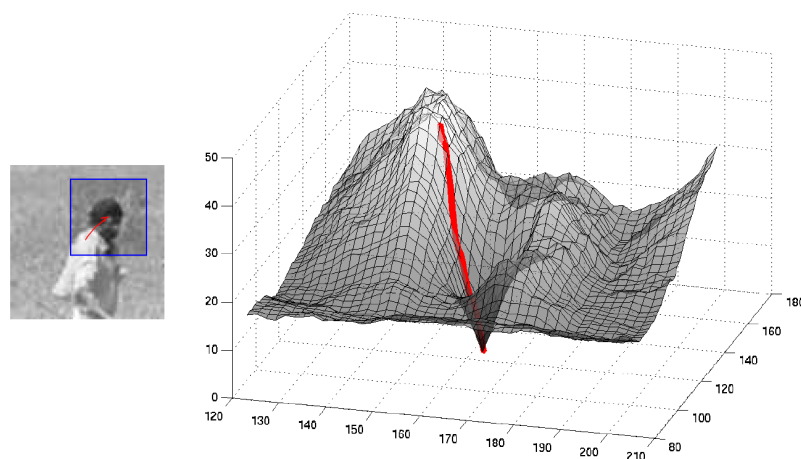
Stav poznání

2.1 Sledování

Sledováním rozumíme opakované určování pozice objektu v sekvenci obrázků [9]. Obrázkem \mathcal{I} chápeme zobrazení $\mathcal{I} : (\mathcal{N}^2) \rightarrow \mathcal{R}$ z množiny souřadnic do prostoru jasových intenzit. Sledovaný objekt je reprezentován množinou pixelů $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, kterou nazýváme opěrná množina, a množinou intenzit $\mathbf{I}(\mathbf{X}) = (\mathcal{I}(\mathbf{x}_1), \dots, \mathcal{I}(\mathbf{x}_n))^T$ pozorovaných na \mathbf{X} . Sledování je často zjednodušeno jako hledání transformace pixelů objektu, která je zobrazuje na co nejpravděpodobnější místo v aktuálním obrázku. Vzájemnou podobnost pixelů vyjadřujeme kriteriální funkcí. Ta může být reprezentována součtem kvadrátů odchylek intenzit, rozdílem histogramů intenzit [3], či jinak. Optimální transformaci získáme minimalizací této kriteriální funkce. Můžeme prohledávat přes všechny možné transformace, což je proces výpočetně náročný a pomalý. Další alternativou je využití toho, že mezi dvěma po sobě jdoucími snímky sekvence jsou pozice objektu tak blízko sebe, že lze správnou pozici nalézt gradientní optimalizací kriteriální funkce inicializované na poslední známé pozici.

2.2 Kanade Lucas Tracker (KLT)

KLT [7] se vyhýbá kompletnímu prohledání prostoru všech translací. Kriteriální funkce je vyjádřena součtem kvadrátů odchylek intenzit pixelů opěrné množiny dvou obrázků. Předpokládáme, že funkce je pro hledanou translaci v okolí sledované oblasti konvexní a opakovaně ji aproximujeme parabolou. Gradientní metoda v několika krocích spadá do jejího minima, Obr.2.1. Bohužel nemáme zaručeno, že se jedná o globální minimum, což je hlavní nevýhodou KLT [1]. Výhodou metody je rychlost predikované translace.



Obrázek 2.1: Metoda KLT gradientně minimalizuje kriteriální funkci [6]



Obrázek 2.2: Translace mezi vzorovým obrázkem \mathcal{I} (vlevo) a vstupním obrázkem \mathcal{J} (vpravo)

KLT [8] se pokusíme ukázat na jednoduchém příkladu translace \mathbf{t} v rovině kolmé na kameru. Uvažujeme dva vstupní obrázky, vzorový obrázek \mathcal{I} a testovaný obrázek \mathcal{J} , viz Obr.2.2. Budeme uvažovat vektor intenzit vzorového obrázku $\mathbf{I}(\mathbf{x})$ a testovaného obrázku $\mathbf{J}(\mathbf{x})$. Hledáme optimální translaci \mathbf{t}^* vstupního obrázku od vzorového, Obr.2.3

$$\mathbf{t}^* = \arg \min_{\mathbf{t}} \|\mathbf{I}(\mathbf{x} + \mathbf{t}) - \mathbf{J}(\mathbf{x})\|_2^2 = \quad (2.1)$$



Obrázek 2.3: Translace \mathbf{t} opěrné množiny vstupního obrázku \mathcal{I} od vzorového obrázku \mathcal{J}

$$= \arg \min_{\mathbf{t}} (\mathbf{I}(\mathbf{X} + \mathbf{t}) - \mathbf{J}(\mathbf{X}))^T (\mathbf{I}(\mathbf{X} + \mathbf{t}) - \mathbf{J}(\mathbf{X})) \quad (2.2)$$

Zavedeme aproximaci

$$\mathbf{I}(\mathbf{X} + \mathbf{t}) \approx \mathbf{I}(\mathbf{X}) + \mathbf{G}\mathbf{t} \quad (2.3)$$

$$\mathbf{G} = \frac{\delta \mathbf{I}(\mathbf{X})}{\delta \mathbf{x}} \quad (2.4)$$

kde \mathbf{G} ($n \times 2$) je matice gradientu opěrné množiny vzorového obrázku.

Dosazením do rovnice (2.2) dostaneme

$$\arg \min_{\mathbf{t}} \left[(\mathbf{I}(\mathbf{X}) - \mathbf{J}(\mathbf{X}) + \mathbf{G}\mathbf{t})^T (\mathbf{I}(\mathbf{X}) - \mathbf{J}(\mathbf{X}) + \mathbf{G}\mathbf{t}) \right] = \quad (2.5)$$

$$= \arg \min_{\mathbf{t}} \left[(\mathbf{I}(\mathbf{X}) - \mathbf{J}(\mathbf{X}))^T (\mathbf{I}(\mathbf{X}) - \mathbf{J}(\mathbf{X})) + 2\mathbf{t}\mathbf{G}^T (\mathbf{I}(\mathbf{X}) - \mathbf{J}(\mathbf{X})) + \mathbf{t}^T \mathbf{G}^T \mathbf{G} \mathbf{t} \right] \quad (2.6)$$

Derivujeme podle \mathbf{t} a pokládáme derivaci rovnu nule:

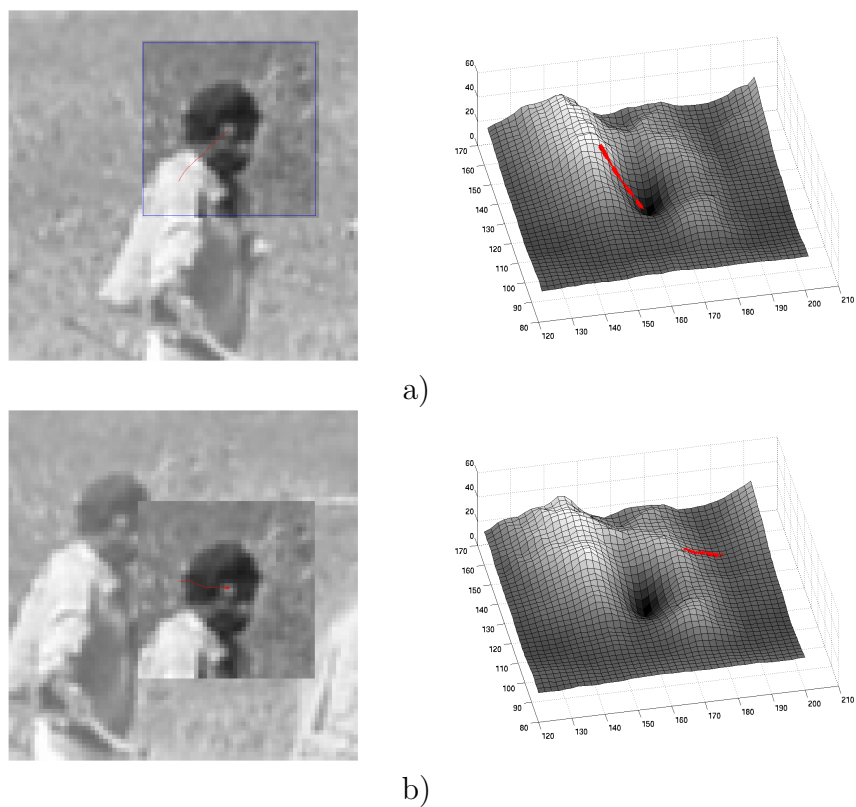
$$\frac{\delta}{\delta \mathbf{t}} : 2\mathbf{G}^T (\mathbf{I}(\mathbf{X}) - \mathbf{J}(\mathbf{X})) + 2\mathbf{G}^T \mathbf{G} \mathbf{t} = 0 \quad (2.7)$$

$$\mathbf{G}^T \mathbf{G} \mathbf{t} = \mathbf{G}^T (\mathbf{I}(\mathbf{X}) - \mathbf{J}(\mathbf{X})) \quad (2.8)$$

Dostáváme výsledný vztah pro optimální \mathbf{t}^*

$$\mathbf{t}^* = (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T (\mathbf{I}(\mathbf{X}) - \mathbf{J}(\mathbf{X})) = \mathbf{G}^+ (\mathbf{I}(\mathbf{X}) - \mathbf{J}(\mathbf{X})) \quad (2.9)$$

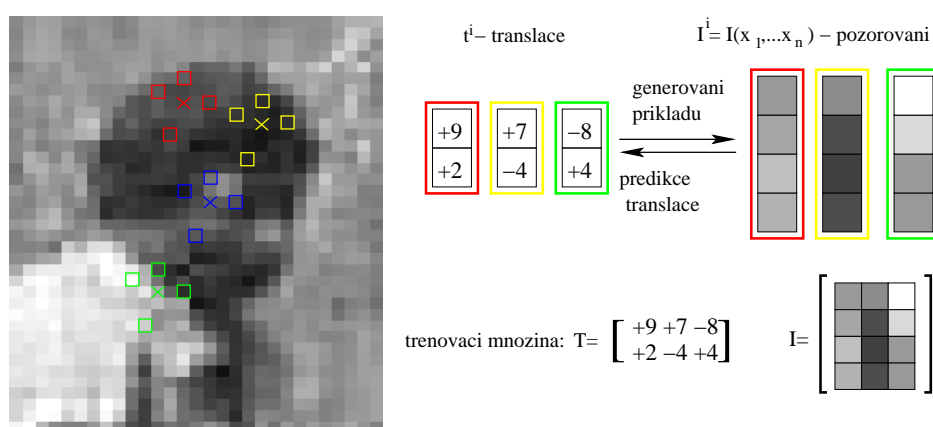
kde \mathbf{G}^+ je pseudoinverzní matice. Hodnota optimální translace se spočítá z vektoru odchylek intenzit vzorového a vstupního obrázku a gradientu intenzit vzorového obrázku. Vzhledem k tomu, že jsme $\mathbf{I}(\mathbf{X} + \mathbf{t})$ aproximovali lineární funkcí, nalezené řešení je nepřesné a proto výslednou translaci nalezneme iterováním rovnice (2.9). Z počáteční pozice tak spadáme do lokálního minima kriteriální funkce. Při větší vzdálenosti počáteční pozice se nemusí jednat o hledané globální minimum, viz Obr.2.4.



Obrázek 2.4: Gradientní minimalizace KLT spadající do a) globálního minima b) lokálního minima, v závislosti na počáteční pozici [6]

2.3 Frederic Jurie - Lineární prediktor (LLiP)

Frederic Jurie [4] se vyhýbá problémům spojených s minimalizací kriteriální funkce a navrhuje použít předem naučenou lineární funkci (tzv. lineární prediktor), která pro vstupní vektor pozorovaných intenzit $\mathbf{I}(\mathbf{X})$ přiřadí translaci \mathbf{t} , viz Obr.2.5. Funkci učí [5] na vytvořené trénovací množině. Trénovací příklady translací a odpovídajících vektorů intenzit lineárně prokládá metodou nejmenších čtverců. Výhodou metody je rychlost predikované transformace, nevýhodou je nutnost jejího učení.



Obrázek 2.5: Lineární prediktor přiřadí pozorování $\mathbf{I}(\mathbf{X})$ translaci \mathbf{t} [6]

Vektoru pozorovaných intenzit $\mathbf{I}(\mathbf{X})$ opěrné množiny \mathbf{X} budeme říkat pozorování. Pak *lineární prediktor* (LLiP převzato z anglické terminologie Learned Linear Predictor) si definujeme jako uspořádanou dvojici (\mathbf{H}, \mathbf{X}) , která každému pozorování $\mathbf{I}(\mathbf{X})$ přiřadí translaci

$$\mathbf{t} = \mathbf{H}\mathbf{I}(\mathbf{X}) \quad (2.10)$$

kde \mathbf{H} ($2 \times n$) obsahuje koeficienty lineární funkce a budeme ji nazývat prediktivní matice LLiP, a n je počet opěrných bodů.

2.3.1 Trénovací množina LLiP

Trénovací množina slouží k učení LLiPu. Mějme opěrnou množinu \mathbf{X} a obrázek \mathcal{I} . Vygenerujeme syntetické příklady, myšleno dvojice translací \mathbf{t}^i a pozorování $\mathbf{I}(\mathbf{X} + \mathbf{t}^i)$. Ty dáváme do matic \mathbf{T} , \mathbf{I} . Uspořádanou dvojici (\mathbf{I}, \mathbf{T}) nazveme trénovací množinou, kde

- $\mathbf{T} \dots (2 \times k)$ matice trénovacích translací po sloupcích složená z \mathbf{t}^i

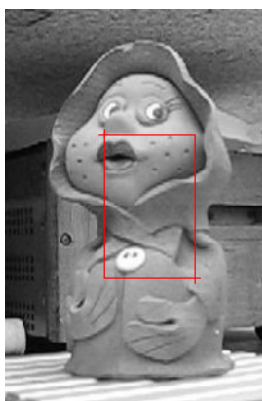
$$\begin{array}{c} \text{I(X)} \searrow \\ \boxed{\mathbf{t} = \mathbf{H}\mathbf{I}(\mathbf{X})} \\ \swarrow \text{t} \end{array}$$

Obrázek 2.6: Lineární prediktor LLiP si definujeme jako uspořádanou dvojici (\mathbf{H}, \mathbf{X}) , která každému pozorování $\mathbf{I}(\mathbf{X})$ přiřadí translaci $\mathbf{t} = \mathbf{H}\mathbf{I}(\mathbf{X})$

- $\mathbf{I} \dots (n \times k)$ matice trénovacích intenzit po sloupcích složená z $\mathbf{I}(\mathbf{x} + \mathbf{t}^i)$
- $k \dots$ počet trénovacích příkladů

Algoritmus vytvoření trénovací množiny LLiPu:

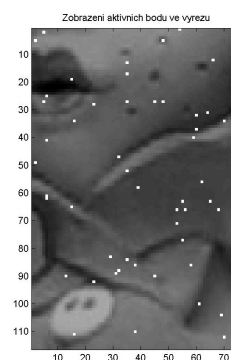
1. načti vzorový obrázek \mathcal{I} , Obr.2.7
2. definuj počet trénovacích příkladů k , opěrnou množinu \mathbf{X} , maximální rozsah translací t_{max}
3. vytvoř matici trénovacích translací \mathbf{T} , po sloupcích složenou z \mathbf{t}^i
4. vektory odpovídajících intenzit $\mathbf{I}(\mathbf{x} + \mathbf{t}^i)$ vlož jako i -té sloupce matice \mathbf{I} , Obr.2.8



a)

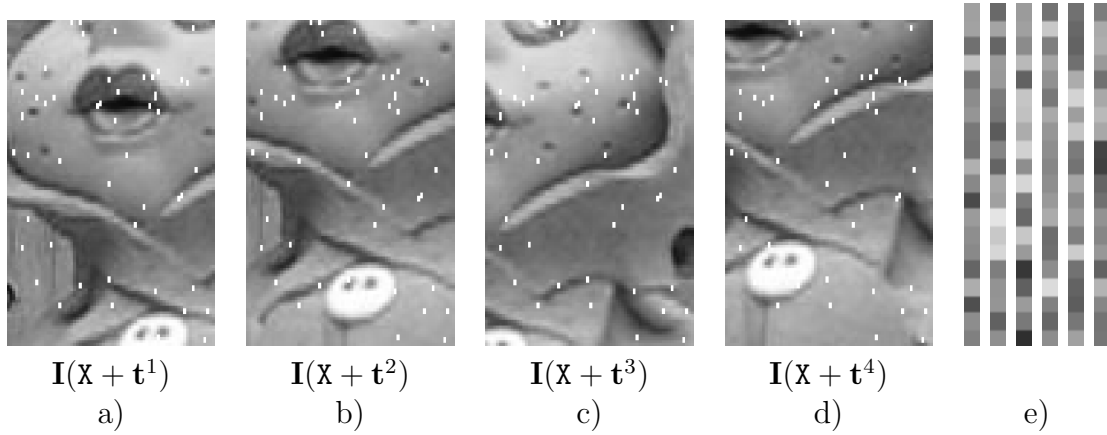


b)



c)

Obrázek 2.7: a) vzorový obrázek \mathcal{I} s červeně vyznačenou sledovanou oblastí, b) sledovaná oblast, c) bíle vyznačené opěrné body



Obrázek 2.8: a-d) Pro trénovací translace \mathbf{t}^i odpovídající pozorování $\mathbf{I}(\mathbf{X} + \mathbf{t}^i)$, e) vektory intenzit $\mathbf{I}(\mathbf{X} + \mathbf{t}^i)$ tvořící sloupce matice \mathbf{I}

2.3.2 Učení LLiP

Učením rozumíme nalezení matice \mathbf{H} , která minimalizuje součet kvadrátů chyby predikce na trénovací množině.

$$\mathbf{H}^* = \arg \min_{\mathbf{H}} \|\mathbf{H}\mathbf{I} - \mathbf{T}\|_F^2 \quad (2.11)$$

Zavedeme si značení \mathbf{h}_i^T ($1 \times n$) a \mathbf{t}_i^T ($1 \times k$) pro řádky matice \mathbf{H} a \mathbf{T} , pak $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2]^T$, $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2]^T$.

Pro každý řádek hledáme optimální \mathbf{h}_i^* , pro které platí

$$\mathbf{h}_i^* = \arg \min_{\mathbf{h}_i} \|\mathbf{h}_i^T \mathbf{I} - \mathbf{t}_i^T\|_2^2 = \arg \min_{\mathbf{h}_i} (\mathbf{h}_i^T \mathbf{I} - \mathbf{t}_i^T) (\mathbf{h}_i^T \mathbf{I} - \mathbf{t}_i^T)^T \quad (2.12)$$

Derivaci podle \mathbf{h}_i položíme rovnu nule:

$$\frac{\delta}{\delta \mathbf{h}_i} \left[(\mathbf{h}_i^T \mathbf{I} - \mathbf{t}_i^T) (\mathbf{h}_i^T \mathbf{I} - \mathbf{t}_i^T)^T \right] = 0 \quad (2.13)$$

$$\frac{\delta}{\delta \mathbf{h}_i} [\mathbf{h}_i^T \mathbf{I} \mathbf{I}^T \mathbf{h}_i - \mathbf{h}_i^T \mathbf{I} \mathbf{t}_i - \mathbf{t}_i^T \mathbf{I}^T \mathbf{h}_i + \mathbf{t}_i^T \mathbf{t}_i] = 2\mathbf{I} \mathbf{I}^T \mathbf{h}_i - 2\mathbf{I} \mathbf{t}_i = 0 \quad (2.14)$$

hledané \mathbf{h}_i^* vyjádříme jako

$$\mathbf{h}_i^* = (\mathbf{I} \mathbf{I}^T)^{-1} \mathbf{I} \mathbf{t}_i \quad (2.15)$$

Prediktivní matice \mathbf{H}^* je rovna:

$$\mathbf{H}^* = [\mathbf{h}_1, \mathbf{h}_2]^T \quad (2.16)$$

$$\mathbf{H}^* = \mathbf{T}\mathbf{I}^+ \quad (2.17)$$

kde \mathbf{I}^+ je pseudoinverzní matice

$$\mathbf{I}^+ = \mathbf{I}^T (\mathbf{I}\mathbf{I}^T)^{-1} \quad (2.18)$$

2.3.3 Algoritmus sledování LLiP

Metoda lineárního prediktoru je při sledování rychlá, protože pro predikci translace \mathbf{t} stačí jediné maticové násobení vektoru intenzit $\mathbf{I}(\mathbf{X})$ a prediktivní matice \mathbf{H} .

Algoritmus sledování objektu LLiPem v obrázkové sekvenci:

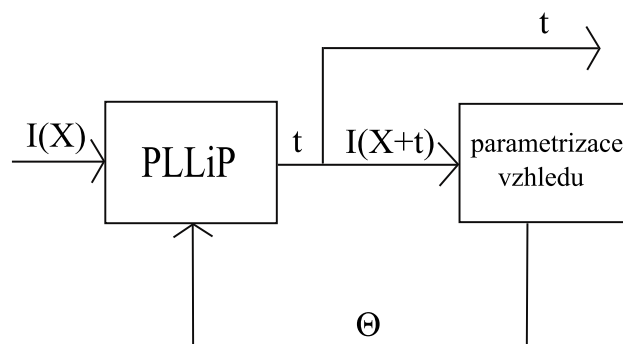
1. načti první obrázek \mathcal{S}^1 , inicializuj pozici objektu
2. přes všechny obrázky sekvence, $j=1$ až délka sekvence
3. LLiPem predikuj translaci $\mathbf{t}^j = \mathbf{H}\mathbf{I}^j(\mathbf{X}^j)$ z vektoru intenzit $\mathbf{I}^j(\mathbf{X}^j)$ přečteného z \mathcal{S}^j
4. aktualizuj pozici opěrné množiny $\mathbf{X}^{j+1} = \mathbf{X}^j + \mathbf{t}^j$
5. přejdi k dalšímu obrázku sekvence, $j = j + 1$ a jdi na bod 2

Kapitola 3

Vlastní práce

3.1 Přínos práce

Sledování objektů, které mění svůj vzhled způsobem, který nelze popsat jednoduchou obrazovou transformací jakou je například změna měřítky či rotace, je obtížné. Cílem této práce je rozšířit lineární prediktor na parametrický lineární prediktor (PLLiP, převzato z anglické terminologie Parametric Learned Linear Predictor), který je schopen adaptace na aktuální vzhled sledovaného objektu. PLLiP si zachovává rychlost predikce LLiPu a navíc umožňuje sledovat objekty měnící svůj vzhled.



Obrázek 3.1: Zpětnovazební zapojení PLLiP umožňující adaptaci na aktuální vzhled sledovaného objektu.

3.2 Sledování objektů měnících vzhled

Předpokládejme, že sledovaný objekt může měnit svůj vzhled způsobem, který je popsateľný nějakým vektorem parametrů Θ . Příkladem je lidské oko, Obr.3.2, u kterého při jeho otevírání a zavírání dochází ke změně vzhledu nerigidních částí. Skalární vzhledový parametr Θ můžeme volit jako míru otevřenosti oka v intervalu $(0, 1)$, pro zavřené oko $\Theta = 0$, polootevřené $\Theta = 0.5$ a otevřené $\Theta = 1$. Samozřejmě je možné vzhledový parametr volit jinak, například jako počet hnědých pixelů, kterých s otevírajícím se okem přibývá. Jiným příkladem změny vzhledu je otáčení hrnečku kolem osy, Obr.3.3. Zde můžeme vzhledový parametr Θ volit jako úhel otočení, $\Theta \in (0^\circ, 360^\circ)$. Hledáme libovolnou vzhledovou parametrizaci objektu, která je hladká, robustní a dostatečně diskriminativní. Hladkou rozumíme postupně měnící se v čase, robustní rozumíme odolnou proti zarovnanosti obrázku a diskriminativní rozumíme dobře vystihující vzhled objektu. Předpokládáme, že máme metodu, která rozumně obrázku přidělí vektor vzhledových parametrů Θ . Objekt budeme sledovat parametrickým lineárním prediktorem, který na základě Θ mění chování predikce, viz Obr.3.1.



Obrázek 3.2: Parametrizace vzhledu otevřenosti oka, $\Theta \in (0, 1)$



Obrázek 3.3: Parametrizace vzhledově proměnného otáčejícího hrnečku parametrem $\Theta \in (0^\circ, 360^\circ)$ vyjadřujícího úhel otočení

3.3 Parametrický lineární prediktor (PLLIP)

Parametrický lineární prediktor (PLLIP, Obr.3.4) si definujeme jako uspořádanou dvojici (\mathbf{H}, \mathbf{X}) , která každému pozorování $\mathbf{I}(\mathbf{X})$ a vektoru vzhledových parametrů Θ přiřadí translaci,

$$\mathbf{t} = (\mathbf{H}_0 + \Theta_1 \cdot \mathbf{H}_1 + \Theta_2 \cdot \mathbf{H}_2 + \dots + \Theta_q \cdot \mathbf{H}_q) \mathbf{I}(\mathbf{X}) \quad (3.1)$$

pro

$$\mathbf{H} = [\mathbf{H}_0 \quad \mathbf{H}_1 \quad \dots \quad \mathbf{H}_q] \quad (3.2)$$

$$\Theta = [\Theta_1 \quad \Theta_2 \quad \dots \quad \Theta_q] \quad (3.3)$$

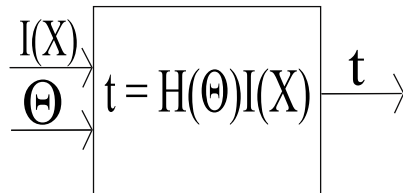
- \mathbf{H} ... $(2 \times (q + 1) n)$ prediktivní matice PLLIP
- \mathbf{H}_i ... $(2 \times n)$ dílčí matice \mathbf{H}
- Θ ... $(q \times 1)$ vektor vzhledových parametrů
- q ... počet parametrů popisující vzhled objektu

Pro další popis si zavedeme maticové zobrazení $g(\mathbf{H}, \Theta)$, resp. $\mathbf{H}(\Theta)$, které matici \mathbf{H} a vektoru Θ přiřadí:

$$g(\mathbf{H}, \Theta) = \mathbf{H}(\Theta) = [\mathbf{H}_0 + \Theta_1 \cdot \mathbf{H}_1 + \Theta_2 \cdot \mathbf{H}_2 + \dots + \Theta_q \cdot \mathbf{H}_q] \quad (3.4)$$

pak \mathbf{t} vyjádříme jako

$$\mathbf{t} = \mathbf{H}(\Theta) \mathbf{I}(\mathbf{X}) \quad (3.5)$$



Obrázek 3.4: PLLIP přiřadí každému pozorování $\mathbf{I}(\mathbf{X})$ a vektoru vzhledových parametrů Θ translaci \mathbf{t} , kde $\mathbf{t} = \mathbf{H}(\Theta) \mathbf{I}(\mathbf{X})$

3.4 Trénovací množina PLLiP

Trénovací množinu PLLiPu vytváříme obdobným způsobem jako u LLiPu. Mějme opěrnou množinu \mathbf{X} a trénovací sadu d obrázků rozdílného vzhledu objektu, viz Obr.3.5. Pro každý trénovací obrázek \mathcal{S}_j popsáný vzhledovými parametry Θ_j , vygenerujeme trénovací translace \mathbf{t}_j^i a k nim odpovídající vektory intenzit $\mathbf{I}_j(\mathbf{X} + \mathbf{t}^i)$. Ty dáváme do matic \mathbf{T} , \mathbf{I} , Θ . Uspořádanou trojici $(\mathbf{I}, \mathbf{T}, \Theta)$ nazveme trénovací množinou PLLiPu, kde

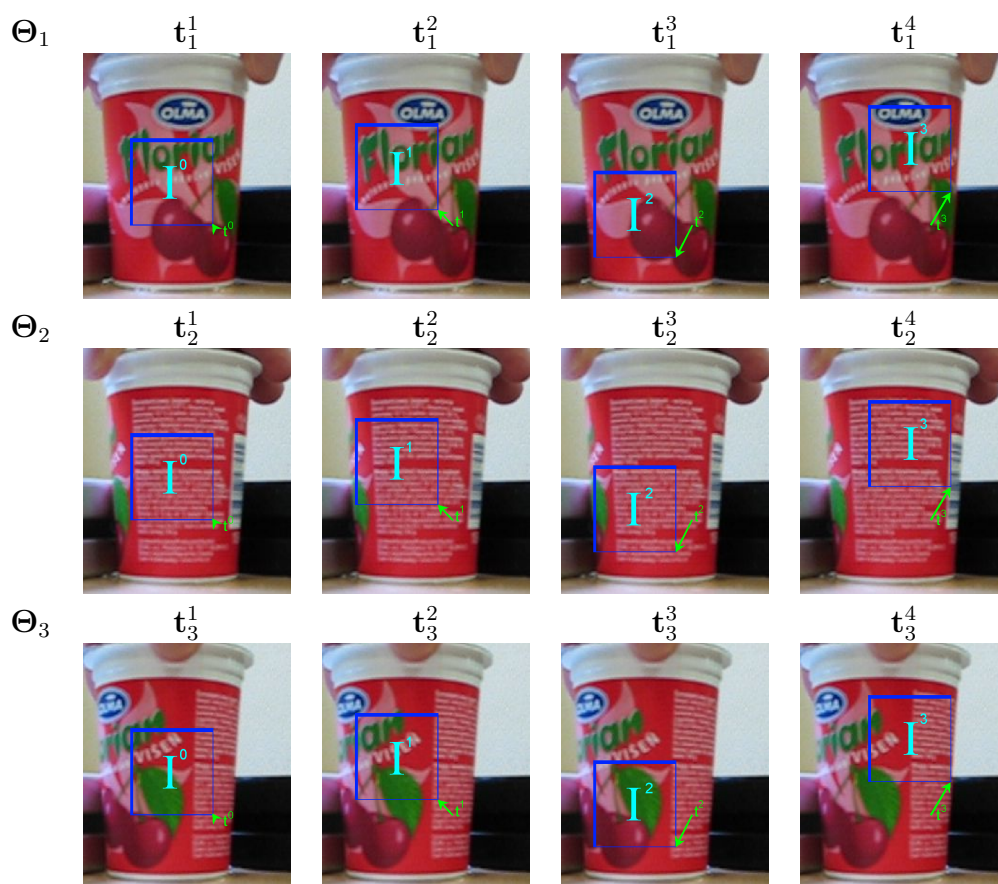
- $\mathbf{T} \dots (2 \times k)$ matice trénovacích translací
- $\mathbf{I} \dots (n \times k)$ matice trénovacích intenzit
- $\Theta \dots (q \times k)$ matice trénovacích vzhledových parametrů
- $k = p.d$ je počet trénovacích příkladů, kde p je počet trénovacích translací na jeden trénovací obrázek \mathcal{S}_j



Obrázek 3.5: Trénovací sada d obrázků různého vzhledu otáčejícího kelímku

Algoritmus vytvoření trénovací množiny PLLiPu, Obr.3.6

- definuj maximální rozsah translací t_{max} , opěrnou množinu \mathbf{X} , počet trénovacích translací p na jeden obrázek,
- přes všechny obrázky \mathcal{S}_j různého vzhledu z trénovací sady ($j = 1$ až d) popsané vektorem vzhledových parametrů Θ_j
 - vytvoř matici trénovacích translací \mathbf{T}_j ($2 \times p$)
 - přes všechny translace \mathbf{t}_j^i z \mathbf{T}_j , ($i=1$ až p)
 - vlož vektor intenzity $\mathbf{I}_j(\mathbf{X} + \mathbf{t}_j^i)$ do \mathbf{I}
 - vektor vzhledových parametrů Θ_j do Θ
 - trénovací translaci \mathbf{t}_j^i do \mathbf{T}
 - $i = i + 1$
 - $j = j + 1$



Obrázek 3.6: Pozorování $\mathbf{I}_j(\mathbf{X} + \mathbf{t}_j^i)$ odpovídající trénovacím translacím \mathbf{t}_j^i j -tého trénovacího obrázku popsaného vektorem vzhledových parametrů Θ_j

Vytvořené matice \mathbf{T} , Θ jsou tak ve tvaru

$$\mathbf{T} = [\mathbf{t}_1^1 \quad \mathbf{t}_1^2 \quad \dots \quad \mathbf{t}_1^p \quad \mathbf{t}_2^1 \quad \mathbf{t}_2^2 \quad \dots \quad \mathbf{t}_2^p \quad \dots \quad \mathbf{t}_d^1 \quad \mathbf{t}_d^2 \quad \dots \quad \mathbf{t}_d^p] \quad (3.6)$$

$$\Theta = [\Theta_1 \quad \Theta_1 \quad \dots \quad \Theta_1 \quad \Theta_2 \quad \Theta_2 \quad \dots \quad \Theta_2 \quad \dots \quad \Theta_d \quad \Theta_d \quad \dots \quad \Theta_d] \quad (3.7)$$

3.5 Učení PLLiPu

Učením rozumíme nalezení \mathbf{H} takové, která minimalizuje součet kvadrátu chyby na trénovací množině $(\mathbf{T}, \mathbf{I}, \Theta)$

$$\mathbf{I} = [\mathbf{I}^1 \quad \mathbf{I}^2 \quad \dots \quad \mathbf{I}^k], \mathbf{T} = [\mathbf{t}^1 \quad \mathbf{t}^2 \quad \dots \quad \mathbf{t}^k], \Theta = [\Theta^1 \quad \Theta^2 \quad \dots \quad \Theta^k] \quad (3.8)$$

Hledáme matici \mathbf{H}^* ,

$$\mathbf{H}^* = \arg \min_{\mathbf{H}} \sum_{i=1}^k \|g(\mathbf{H}, \Theta^i) \mathbf{I}^i - \mathbf{t}^i\|_2^2 = \quad (3.9)$$

$$= \arg \min_{\mathbf{H}} \sum_{i=1}^k \|[\mathbf{H}_0 + \Theta_1^i \cdot \mathbf{H}_1 + \Theta_2^i \cdot \mathbf{H}_2 + \dots + \Theta_q^i \cdot \mathbf{H}_q] \cdot \mathbf{I}^i - \mathbf{t}^i\|_2^2 \quad (3.10)$$

Což můžeme vyjádřit jako

$$\mathbf{H}^* = \arg \min_{\mathbf{H}} \sum_{i=1}^k \left\| \left[\mathbf{H}_0 \quad \mathbf{H}_1 \quad \dots \quad \mathbf{H}_q \right] \begin{bmatrix} \mathbf{I}^i \\ \Theta_1^i \mathbf{I}^i \\ \vdots \\ \Theta_q^i \mathbf{I}^i \end{bmatrix} - \mathbf{t}^i \right\|_2^2 = \quad (3.11)$$

$$= \arg \min_{\mathbf{H}} \left\| \left[\mathbf{H}_0 \quad \mathbf{H}_1 \quad \dots \quad \mathbf{H}_q \right] \begin{bmatrix} \begin{bmatrix} \mathbf{I}^1 \\ \Theta_1^1 \mathbf{I}^1 \\ \vdots \\ \Theta_q^1 \mathbf{I}^1 \end{bmatrix} & \begin{bmatrix} \mathbf{I}^2 \\ \Theta_1^2 \mathbf{I}^2 \\ \vdots \\ \Theta_q^2 \mathbf{I}^2 \end{bmatrix} & \dots & \begin{bmatrix} \mathbf{I}^k \\ \Theta_1^k \mathbf{I}^k \\ \vdots \\ \Theta_q^k \mathbf{I}^k \end{bmatrix} \end{bmatrix} - \begin{bmatrix} \mathbf{t}^1 & \mathbf{t}^2 & \dots & \mathbf{t}^k \end{bmatrix} \right\|_F^2 \quad (3.12)$$

Minimalizujeme frobeniovu normu matice odchylek celé trénovací množiny. Matici intenzit a vzhledových parametrů si označíme \mathbf{A}

$$\mathbf{A} = \begin{bmatrix} \begin{bmatrix} \mathbf{I}^1 \\ \Theta_1^1 \mathbf{I}^1 \\ \vdots \\ \Theta_q^1 \mathbf{I}^1 \end{bmatrix} & \begin{bmatrix} \mathbf{I}^2 \\ \Theta_1^2 \mathbf{I}^2 \\ \vdots \\ \Theta_q^2 \mathbf{I}^2 \end{bmatrix} & \dots & \begin{bmatrix} \mathbf{I}^k \\ \Theta_1^k \mathbf{I}^k \\ \vdots \\ \Theta_q^k \mathbf{I}^k \end{bmatrix} \end{bmatrix} \quad (3.13)$$

a dostáváme vztah

$$\mathbf{H}^* = \arg \min_{\mathbf{H}} \|\mathbf{H}\mathbf{A} - \mathbf{T}\|_F^2 \quad (3.14)$$

Jedná se o minimalizaci frobeniovy normy matice, která je ve stejné formě, jako tomu bylo při učení LLiPu. Odvození hledané optimální matice \mathbf{H}^* je shodné, viz 2.3.2, a dostáváme tak vztah pro prediktivní matici PLLiPu

$$\mathbf{H} = \mathbf{T}\mathbf{A}^T (\mathbf{A}\mathbf{A}^T)^{-1} = \mathbf{T}\mathbf{A}^+ \quad (3.15)$$

kde \mathbf{A}^+ je pseudoinverzní matice.

3.5.1 Učení pro jeden skalární vzhledový parametr

Pro vyjádření vzhledu sledovaného objektu si často vystačíme s jedním skalárním vzhledovým parametrem Θ , jako tomu je u otáčejícího kelímku, Obr.3.7, kde jím může být úhel otočení $\Theta \in (0^\circ, 360^\circ)$. Pak hledáme H^* ,

$$H^* = \arg \min_H \left\| \begin{bmatrix} H_0 & H_1 \end{bmatrix} \begin{bmatrix} \begin{bmatrix} \mathbf{I}^1 \\ \Theta^1 \mathbf{I}^1 \end{bmatrix} & \begin{bmatrix} \mathbf{I}^2 \\ \Theta^2 \mathbf{I}^2 \end{bmatrix} & \cdots & \begin{bmatrix} \mathbf{I}^k \\ \Theta^k \mathbf{I}^k \end{bmatrix} \end{bmatrix} - \begin{bmatrix} \mathbf{t}^1 & \mathbf{t}^2 & \cdots & \mathbf{t}^k \end{bmatrix} \right\|_F^2 \quad (3.16)$$

pro

$$A = \begin{bmatrix} \begin{bmatrix} \mathbf{I}^1 \\ \Theta^1 \mathbf{I}^1 \end{bmatrix} & \begin{bmatrix} \mathbf{I}^2 \\ \Theta^2 \mathbf{I}^2 \end{bmatrix} & \cdots & \begin{bmatrix} \mathbf{I}^k \\ \Theta^k \mathbf{I}^k \end{bmatrix} \end{bmatrix} \quad (3.17)$$

je hledané optimální H^* rovno

$$H^* = \mathbf{T}A^T (\mathbf{A}A^T)^{-1} = \mathbf{T}A^+ \quad (3.18)$$

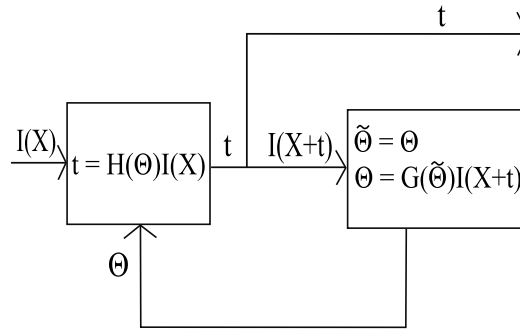


Obrázek 3.7: Parametrizace vzhledu objektu skalárním parametrem Θ , $\Theta \in (0^\circ, 360^\circ)$ vyjadřujícího úhel otočení

3.6 Odhad vzhledových parametrů

Ve zpětnovazební smyčce dle Obr.3.1 potřebujeme správně odhadnout aktuální vzhledové parametry Θ sledovaného objektu. Protože se jedná o ad hoc vzhledovou parametrizaci, můžeme použít například PCA [2], klasifikaci metodou nejbližšího souseda, lineární prediktor nebo jiné metody. Předpokládáme, že se vzhled objektu mění plynule v čase, proto navrhuje pro odhad vzhledových parametrů použít druhý PLLiP, Obr.3.8, který predikuje vzhledové parametry na základě vstupního vektoru intenzit a vektoru vzhledových parametrů naposledy pozorovaného obrázku. Princip predikce vzhledových parametrů je stejný jako u translace, rozdíl je v trénovací množině. Jedná se o uspořádanou trojici $(J, \Theta, \Delta_\Theta)$, kterou tvoří matice trénovacích intenzit J , která je zarovnaná s přesností úměrnou s přesností prvního PLLiP predikující translaci. Matice vzhledových parametrů Θ je stejná a místo matice translací T se jedná o matici vzhledových změn Δ_Θ .

Při zpětnovazebním zapojení, Obr.3.8, dvou PLLiPů za sebou získáme adaptivní prediktor, kde první PLLiP predikuje translaci pozorování t a druhý PLLiP po zarovnání predikuje aktuální vzhledové parametry Θ . Každý PLLiP může mít jinou velikost a uspořádání opěrné množiny. Tento systém zpětnovazebního zapojení dvou PLLiPů je testován v experimentech.



Obrázek 3.8: Zpětnovazební uspořádání dvou PLLiPů, první PLLiP predikuje translaci t a druhý po zarovnání aktuální vzhledové parametry Θ .

3.7 Výpočetní složitost PLLiP a LLiP

Porovnání výpočetní složitosti LLiP a PLLiP je rozdílné pro fázi učení a predikce. V průběhu sledování v obrázkové sekvenci (fáze predikce) v algoritmech LLiP a PLLiP je třeba nenáročných matematických výpočtů. Pro velikost opěrné množiny n je u LLiP $2n$ a u PLLiP $8n$ skalárních násobení, což jsou operace velmi rychlé, rozdíl se tak projeví až u sekvencí čítajících desetitisíce obrázků nebo sledovaných objektů. Ve fázi učení je rozdíl znatelnější. V případě PLLiP musíme počítat pseudoinverzní matici, která má $(q + 1)$ krát větší počet řádků než u LLiP, kde q je počet parametrů popisující vzhled objektu. Protože fáze učení je oddělená od sledování, tak se v experimentech nebudeme výpočetní náročností učení lineárních prediktorů podrobněji zabývat.

3.8 Sekvenční parametrický lineární prediktor (SPLLiP)

Se zvyšováním velikosti opěrné množiny se snižuje chyba predikce lineárního prediktoru. Čím větší je velikost množiny, tím menší je zlepšení. Pro příliš velkou opěrnou množinu dochází dále k přeučení prediktoru, které se projevuje stoupající chybou predikce na testovacích příkladech a klesající chybou na trénovacích příkladech. Navrhujeme proto použít sekvenci PLLiPů (dále označovanou jako SPLLiP), která dosahuje nižší chyby při konstantní velikosti opěrné množiny, Obr.3.9. Postupně učíme PLLiPy na menším rozsahu trénovacích translací, čímž snižujeme i chybu predikce. Trénovací translace $i + 1$ -tého prediktoru volíme rovné chybě predikce předchozího i -tého prediktoru, Obr.3.8. Nevýhodou SPLLiPu je vyšší výpočetní náročnost učení, protože počítáme více PLLiPů. Porovnání chyby PLLiPu a SPLLiPu je provedeno v experimentech.

Sekvenčním parametrickým lineárním prediktorem (SPLLiP) si definujeme uspořádanou r -tici PLLiPů. Zavedeme si označení SPLLiPu $\Lambda, \Lambda = (\lambda_1, \dots, \lambda_r)$, kde λ_i je i -tý PLLiP.

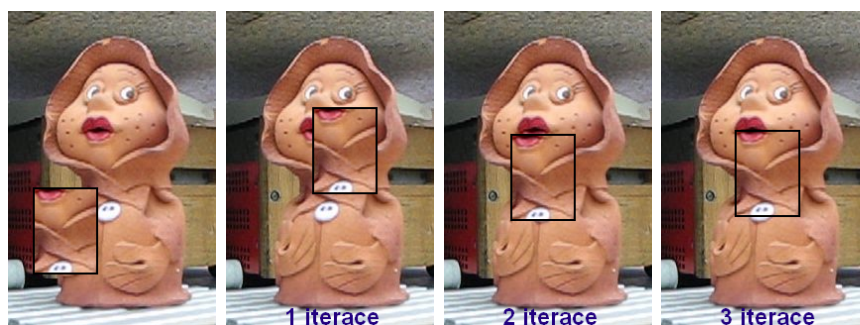
Algoritmus učení SPLLiPu:

1. definuj počet trénovacích translací p na jeden obrázek, opěrnou množinu \mathbf{X} , maximální rozsah translací t_{max}
2. vytvoř matici trénovacích translací \mathbf{T}_1 prediktoru λ_1 , matici vzhledových parametrů Θ
3. přes všechny prediktory λ_j , $j=1$ až r
 - vytvoř matici intenzit \mathbf{I}_j odpovídající \mathbf{T}_j
 - nauč λ_j pro trénovací množinu $(\mathbf{I}_j, \mathbf{T}_j, \Theta$
 - matice trénovacích translací \mathbf{T}_{j+1} dalšího prediktoru je rovna chybě λ_j
 - $j = j + 1$ nauč další prediktor, jdi na bod 3

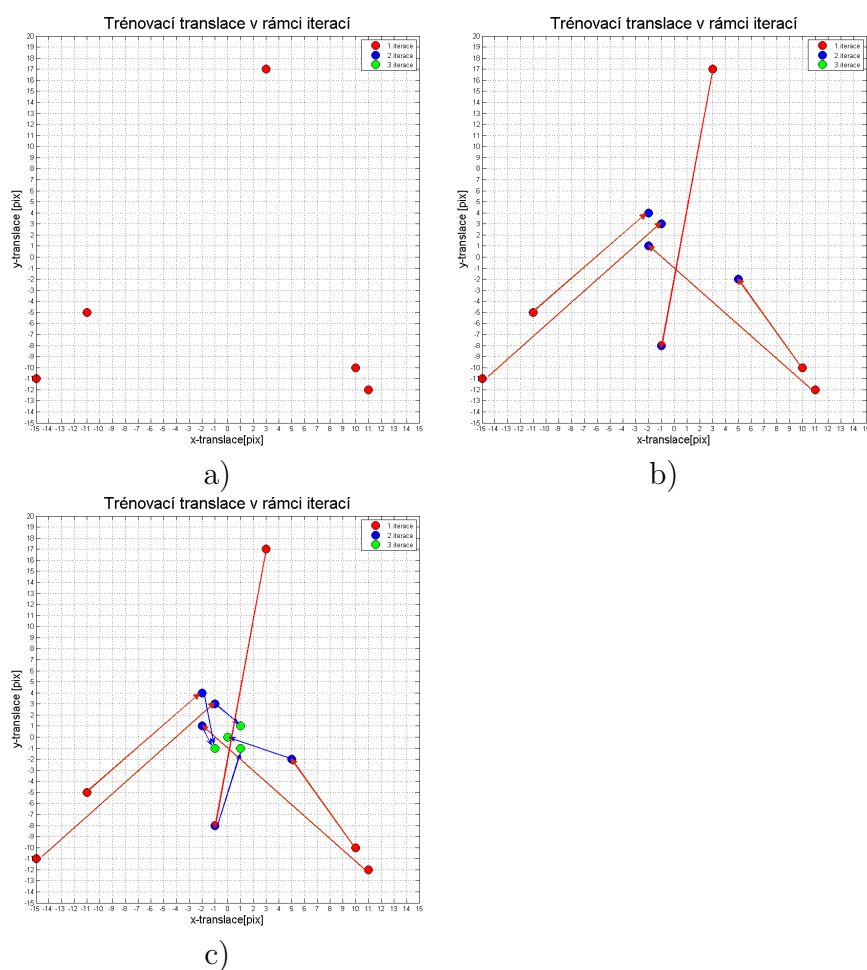
Algoritmus predikce SPLLiPu v obrázkové sekvenci:

1. načti první obrázek \mathcal{S}^1 , inicializuj pozici objektu
2. pro každý obrázek \mathcal{S}^i sekvence, $i = 1$ až konec sekvence
3. načti pozorování $\mathbf{I}_1^i(\mathbf{X})$ z \mathcal{S}^i a vektor vzhledových parametrů Θ^i
4. přes prediktory $\lambda_j \in \Lambda$, $j = 1$ až d
 - prediktorem λ_j predikuj translaci \mathbf{t}_j^i ze $\mathbf{I}_j^i(\mathbf{X})$ a Θ^i
 - aktualizuj opěrnou množinu $\mathbf{X} + \mathbf{t}_j^i \rightarrow \mathbf{X}$
 - načti nové pozorování $\mathbf{I}_{j+1}^i(\mathbf{X})$ z \mathcal{S}^i
 - použij další prediktor $j = j + 1$, jdi na bod 4
5. načti další obrázek v sekvenci $i = i + 1$, jdi na bod 3

3.8. SEKVENČNÍ PARAMETRICKÝ LINEÁRNÍ PREDIKTOR (SPLLIP)23



Obrázek 3.9: Postupá predikce translace vstupního pozorování SPLLIPem složeného ze tří PLLIPů



Obrázek 3.10: Při učení SPLLIP jsou trénovací translace T_j prediktoru λ_j rovny chybě prediktoru λ_{j-1} . Zobrazení trénovacích translací a) T_1 prediktoru λ_1 (červeně) b) T_2 prediktoru λ_2 (modře) c) T_3 prediktoru λ_3 (zeleně)

Kapitola 4

Experimenty

V experimentech ověřujeme teoretický návrh parametrického lineárního prediktoru a sekvenčního parametrického prediktoru, nejprve samostatně a pak ve zpětnovazebním zapojení. Chceme experimentálně ukázat, že PLLiP má menší chybu predikce než LLiP při stejné velikosti opěrné množiny. K tomuto účelu vypracujeme 3 experimenty:

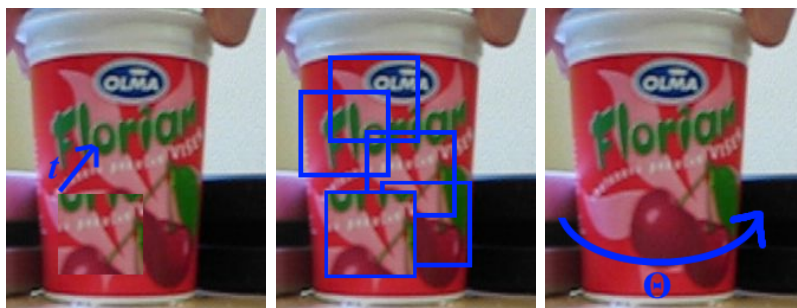
1. predikce translace objektu měnícího vzhled (známe parametry vzhledu) pomocí PLLiP a LLiP, SPLLiP-3 a SLLiP-3
2. predikce vzhledových parametrů (známe pozici) pomocí PLLiP a LLiP
3. predikce translace objektu měnícího vzhled (neznáme parametry vzhledu) pomocí LLiP a zpětnovazebního zapojení dvou PLLiPů pro predikci translace a následovanou predikci vzhledového parametru

4.0.1 Predikce translace

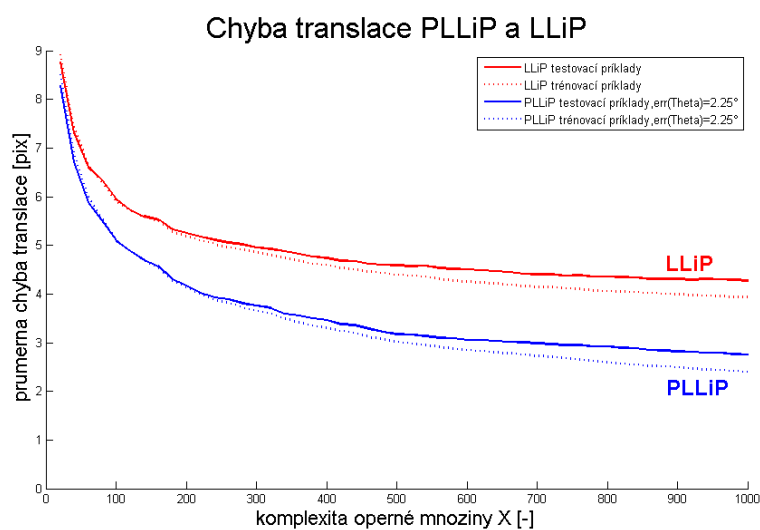
Cílem prvního experimentu je porovnání chyby predikce *translace* PLLiPem a LLiPem v závislosti na velikosti opěrné množiny i při chybě vstupního vzhledového parametru Θ . Dále je porovnána chyba predikce SPLLiP-3 a SLLiP-3.

Experiment provádíme na otáčejícím kelímku, jehož vzhled parametrizujeme skalárním parametrem $\Theta \in (0^\circ, 180^\circ)$ vyjadřujícím otočení, Obr.4.1. Rozsah učené translace volíme 20 [pix].

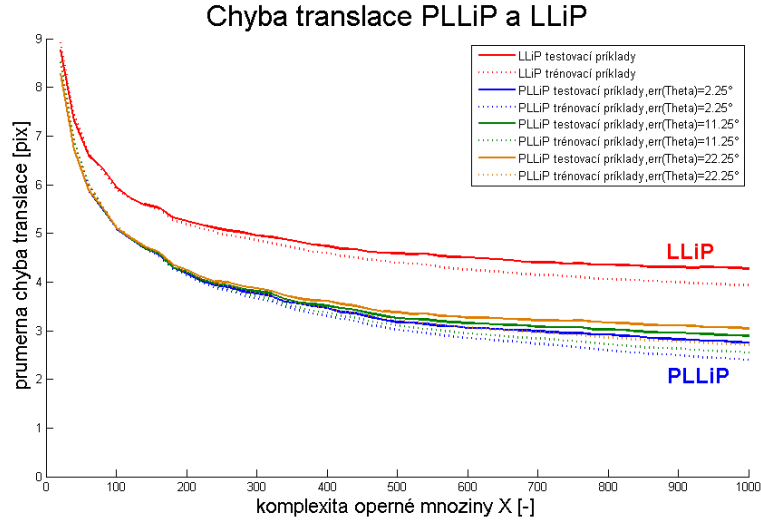
Z průběhů viz Obr. 4.2 je patrné, že PLLiP vykazuje menší chybu predikce translace než-li LLiP přibližně o 20%. S rostoucí velikostí opěrné množiny klesá velikost chyby obou lineárních prediktorů. PLLiP vykazuje menší chybu než LLiP i při chybě vstupního vzhledového parametru Θ , viz Obr. 4.3. S rostoucí chybou Θ mírně stoupá chyba predikce PLLiP.



Obrázek 4.1: Predikce translace pozorování pro vzhledovou parametrizaci $\Theta \in (0^\circ; 180^\circ)$



Obrázek 4.2: Chyba predikce LLiP a PLLiP v závislosti na velikosti opěrné množiny pro rozsah učení translace 20 [pix]



Obrázek 4.3: Chyba predikce translací LLiP a PLLiP v závislosti na velikosti opěrné množiny pro různou chybu vstupního vzhledového parametru err_{Θ} (při rozsahu učené translace 20 [pix])

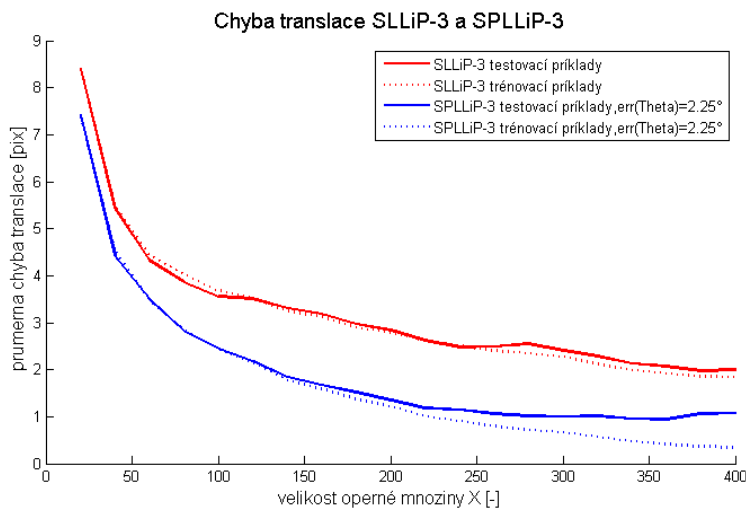
Z průběhu experimentu SPLLiP-3 a SLLiP-3, viz Obr.4.4 čteme menší chybu SPLLiP-3 než SLLiP-3 do velikosti opěrné množiny 450 bodů. Pro velikost opěrné množiny větší jak 300 bodů dochází u SPLLiP-3 k přeučení, kdy roste chyba na testovacích příkladech při stále klesající chybě na trénovacích příkladech. Z průběhu viz Obr.4.5 je patrné, že vliv chyby vstupního vzhledového parametru na SPLLiP-3 je větší jak na PLLiP.

4.0.2 Predikce vzhledového parametru

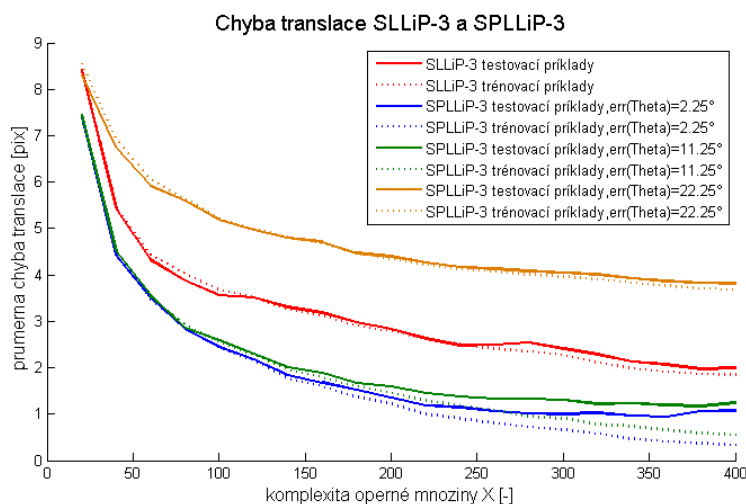
Cílem druhého experimentu je ukázat, že PLLiP má menší chybu predikce *vzhledového parametru* než LLiP a to i při chybě zarovnanosti vstupního obrázku. Predikujeme vzhledový parametr $\Theta \in (0^\circ, 180)$ otáčejícího kelímku (shodného s minulým experimentem, exp.4.0.1) v závislosti na velikosti opěrné množiny. Rozsah učené změny vzhledového parametru volíme $11,5^\circ$.

Z průběhů viz Obr. 4.6 je patrné, že PLLiP vykazuje menší chybu predikce vzhledového parametru než LLiP přibližně o 60%. Pro velikost opěrné množiny větší jak 65 bodů začíná růst chyba LLiP na testovacích příkladech, což je projevem přeučení LLiP

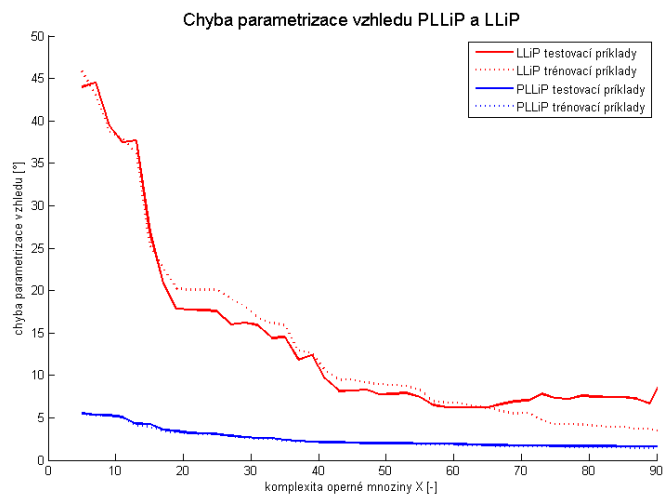
Ve třetím experimentu řadíme dva PLLiPy za sebou, proto teď ověřujeme schopnost predikce vzhledového parametru pro ne zcela zarovnaný vstupní



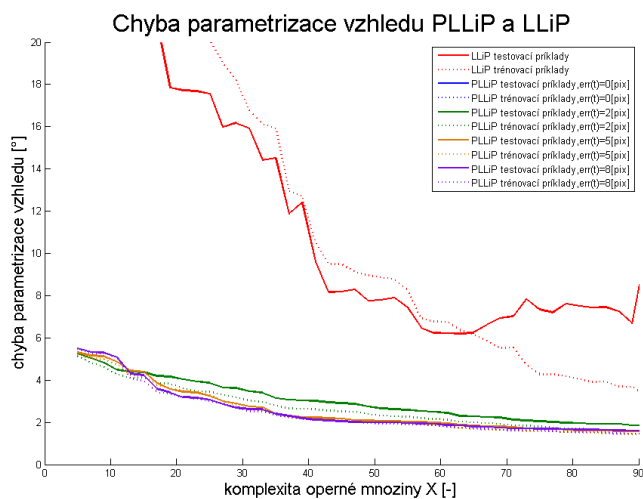
Obrázek 4.4: Chyba predikce SLLiP-3 a SPLLiP-3 v závislosti na velikosti opěrné množiny. Pro velikost opěrné množiny větší jak 300 bodů dochází u SPLLiP-3 k přeučení prediktoru, kdy roste chyba na testovacích příkladech při klesající chybě na trénovacích příkladech.



Obrázek 4.5: Chyba predikce SLLiP-3 a SPLLiP-3 v závislosti na velikosti opěrné množiny pro různou chybu vstupního vzhlédového parametru err_{Θ} .



Obrázek 4.6: Chyba predikce vzhledové parametrizace LLiP a PLLiP v závislosti na velikosti opěrné množiny. Pro velikost opěrné množiny větší jak 65 bodů dochází k přechodu LLiP, kdy roste chyba na testovacích příkladech při stále klesající chybě na trénovacích příkladech

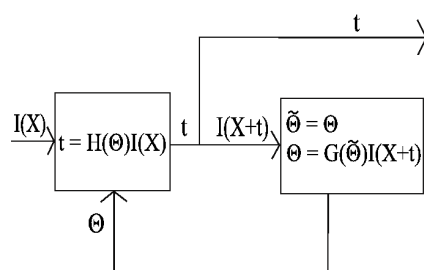


Obrázek 4.7: Chyba predikce vzhledového parametru Θ LLiP a PLLiP v závislosti na velikosti opěrné množiny pro různou chybu zarovnanosti vstupního obrázku err_t . Pro velikost opěrné množiny větší jak 65 bodů dochází k přechodu LLiP, kdy roste chyba na testovacích příkladech při stále klesající chybě na trénovacích příkladech

obrázek. Z Obr.4.7 čteme, že s rostoucí chybou zarovnanosti err_t roste i chyba PLLiP pro predikci vzhledového parametru. Velikost chyby PLLiPu je však stále menší než LLiPu.

4.0.3 Zpětnovazební zapojení PLLiPů

Cílem tohoto experimentu je porovnání chyby predikce translace LLiPem se zpětnovazebním zapojením dvou PLLiPů, Obr .4.8, kde první predikuje translaci t a druhý po zarovnání obrázku predikuje aktuální vzhledový parametr Θ . Sledovaný objekt je otáčející kelímek shodný s předchozími experimenty, Obr.4.1. Testování proběhlo na dvou obrázkových sekvencích pro různé pohyby a změny vzhledu způsobené otáčením kelímku. V obou případech se jednalo o pohyb v rozsahu maximální trénovací translace $t_{max} = 20[\text{pix}]$ a maximální změnu otočení do 3° mezi každými dvěma obrázky.

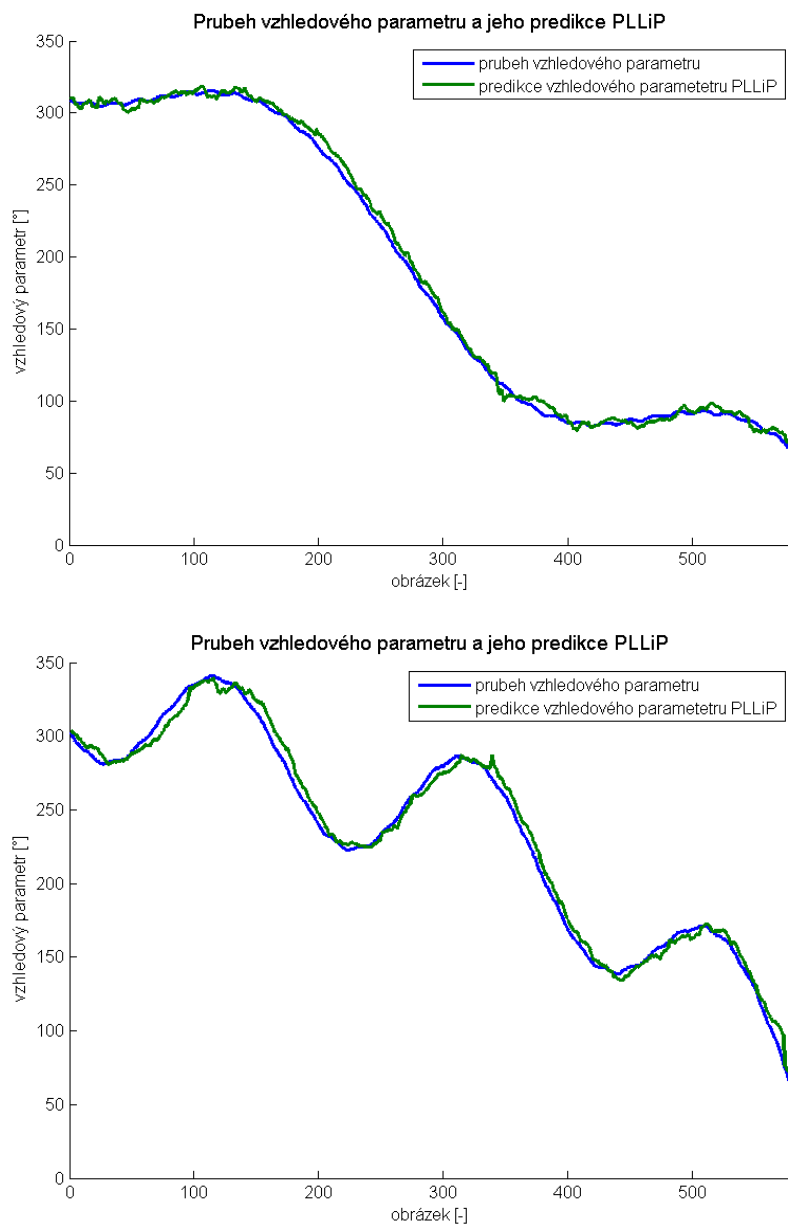


Obrázek 4.8: Zpětnovazební zapojení dvou PLLiP pro sledování objektu v obrázkové sekvenci

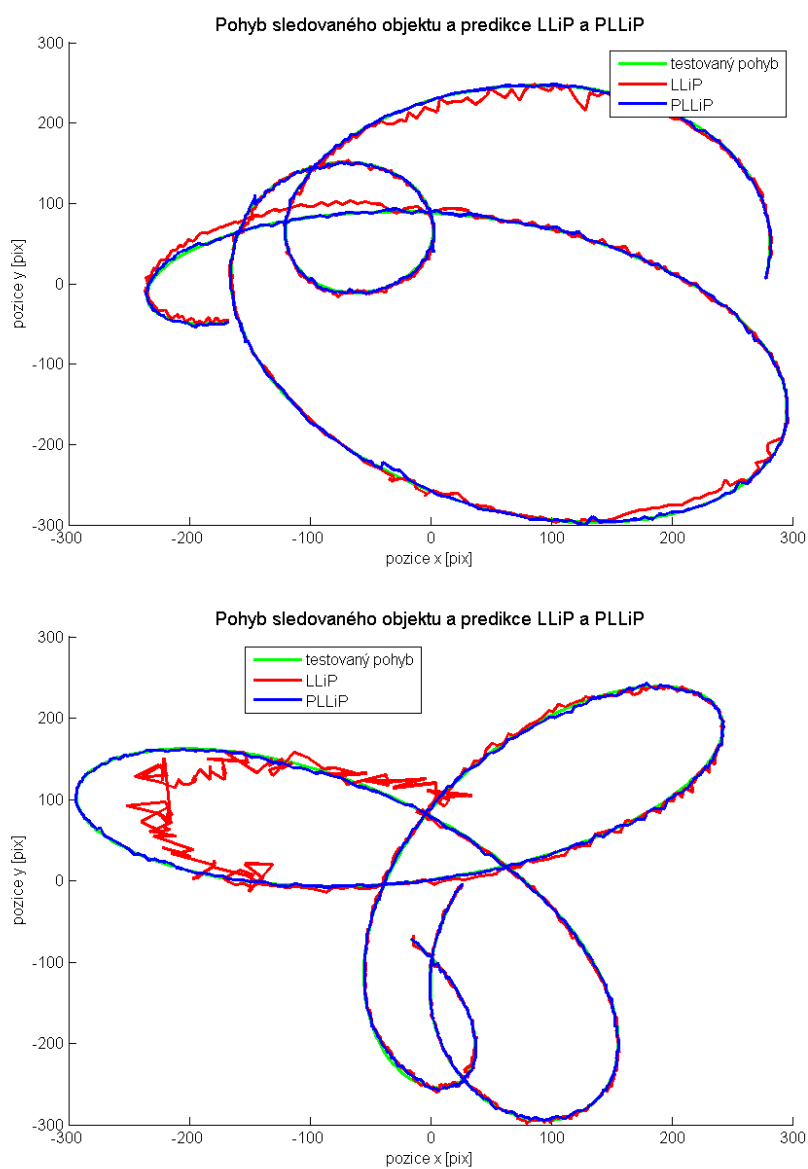
Testování první obrázkové sekvence proběhlo při velikosti opěrné množiny obou lineárních prediktorů 350 bodů. Průměrná chyba predikce pohybu byla u LLiPu 8.3 [pix] a u PLLiPů 5.2 [pix], Obr.4.10.

Z průběhu změny vzhledového parametru, Obr.4.9, je patrné, že PLLiP odhaduje s malou chybou aktuální vzhledový parametr a správně parametrizuje první PLLiP pro predikci translace dalšího vstupního obrázku. Můžeme říci, že zpětnovazební zapojení PLLiPů je schopné sledovat pohyb i změnu vzhledu sledovaného objektu a dochází ke zmenšení chyby predikce pohybu přibližně o 40% oproti LLiPu.

Při testování druhé sekvence byla velikost opěrné množiny obou prediktorů zmenšena na 300 bodů. Při odhadu translace LLiP již nedokázal udržet správnou pozici objektu a došlo k jeho ztrátě. V případě PLLiPu byla chyba predikce 5.8 [pix] a nedošlo ke ztrátě sledovaného objektu. Při porovnání s první sekvencí čteme zvětšení chyby predikce o 12% zapříčiněné menší velikostí opěrné množiny.



Obrázek 4.9: Sledovaný a predikovaný vzhlédový parametr PLLiP pro obě testované sekvence



Obrázek 4.10: Sledovaný a predikovaný pohyb LLiP a PLLiP pro obě testované sekvence. V druhé sekvenci při zmenšené velikosti opěrné množiny LLiP neudržel správnou pozici sledovaného objektu a došlo k jeho ztrátě.

4.0.4 Zhodnocení experimentů

V experimentu predikce translace vzhledově měnícího se objektu je ukázána menší chyba PLLiPu oproti LLiPu přibližně o 20%, a to i při chybě vzhledového parametru Θ . Sekvenční prediktor SPLLiP-3 vykazuje menší chybu predikce translace než PLLiP přibližně o 65%, zároveň je však více citlivý na chybu vstupního vzhledového parametru.

V experimentu predikce vzhledového parametru vykazuje PLLiP menší chybu než LLiP přibližně o 60%, a to i v případě chyby zarovnání obrázku do 5 [pix].

V experimentu pro odhad translace LLiPem a zpětnovazebním zapojení dvou PLLiPů, vykazují v první sekvenci PLLiPy menší chybu než LLiP přibližně o 40%. V druhé sekvenci při snížené velikosti opěrné množiny LLiP neudrží pozici sledovaného objektu a dochází k jeho ztrátě. Naproti tomu zpětnovazební zapojení PLLiPů je schopné objekt a jeho aktuální vzhled dále sledovat, při chybě predikce translace o 11% větší než v případě první sekvence.

Z experimentů lze říci, že parametrizací lineárního prediktoru, nebo-li využitím informace o aktuálním vzhledu sledovaného objektu, snížilo chybu predikce. Zpětnovazební zapojení PLLiPů umožňuje sledovat vzhledově proměnný objekt s menší chybou i za podmínek, kdy LLiP již selhává.

Kapitola 5

Závěr

Navrhli jsme novou metodu pro sledování objektů měnících svůj vzhled. Rozšířili jsme lineární prediktor (LLiP) na parametrický lineární prediktor (PLLiP), který je parametrizován aktuálním vzhledem sledovaného objektu. Vytvořili jsme teoretický koncept učení PLLiPu na uměle vytvořené trénovací množině. Pro sledování objektu v obrázkové sekvenci jsme navrhli zpětnovazební uspořádání dvou PLLiP, kde první PLLiP predikuje translaci a druhý po zarovnání aktuální vzhledové parametry objektu. Experimentálně jsme ukázali, že toto zapojení vykazuje menší chybu predikce pohybu než LLiP přibližně o 40% a umožňuje sledování objektu i za podmínek, kdy LLiP selhává. Další možností pro zmenšení chyby predikce PLLiPu je zapojení několika PLLiPů do sekvence (tzv. SPLLiP). Vytvořili jsme koncept SPLLiPu a popsali algoritmus učení a predikce. Experimentálně jsme ukázali menší chybu predikce SPLLiPu než PLLiPu přibližně o 60%, nevýhodou je vyšší citlivost na chybu vzhledového parametru. Metodu PLLiPu je možné dále rozvíjet nalezením optimálního počtu a rozložení opěrných bodů nebo hledáním optimální skladby trénovací množiny, kterou synteticky generujeme. Výhodou PLLiPu (SPLLiPu) je vysoká rychlost predikce a schopnost adaptace na změnu vzhledu objektu. Pro aplikace, kdy je dostatek času na trénování prediktoru a očekává se rychlost při sledování vzhledově proměnných objektů, se PLLiP (SPLLiP) nabízí jako vhodné a praktické řešení.

Kapitola 6

Přílohy

6.1 Přiložené CD

Přiložené CD obsahuje

- verzi diplomové práce ve formátu pdf
- zdrojové kódy matlabu
- materiály - sekvence obrázků

6.1.1 Digitální verze diplomové práce

Práce je uložena v adresáři: /DiplomováPracePDF/

Jméno souboru diplomové práce: *konecny_msc2007.pdf*

Prohlášení o samostatném vypracování diplomové práce: *prohlaseni.pdf*

Kompletní adresář T_EXsouborů diplomové práce: /DiplomováPracePDF/TeX/

6.1.2 Zdrojové kódy matlabu

Práce byla naprogramována v prostředí Matlab, verze 7.1.0.246(R14) Service Pack 3, MathWorks, Inc.

Adresářová struktura souborů je následující:

Soubory experimentu predikce translace (při známé vzhledové parametrizaci): `/Experimenty/Translace/`

Soubory experimentu predikce vzhledových parametrů (známe polohu): `/Experimenty/VzhledParam/`

Soubory experimentu predikce translace a vzhledových parametrů pomocí zpětnovazebního zapojení dvou PLLiPů: `/Experimenty/ZpetnaVazba/`

Experimenty je možné vždy spustit v souboru `a_set_test.m`, nastavení parametrů je uloženo v souboru `setParameters.m` v daném adresáři.

6.1.3 Materiály - sekvence obrázků

Na CD jsou k dispozici sekvence obrázků, které byly v průběhu vypracování diplomové práce natočeny. Některé jsou použity v experimentech. Mohou sloužit jako studijní materiály pro další práci v oblasti lineárních prediktorů. Materiály jsou v adresáři: `/VideoMateriály/`

Literatura

- [1] Baker, Simon, Matthews, and Iain. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004.
- [2] Michael J. Black and Allan D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. In Bernard F. Buxton and Roberto Cipolla, editors, *4th European Conference on Computer Vision*, volume 1064 of *Lecture Notes in Computer Science*, pages 329–342. Springer, 1996.
- [3] Comaniciu D. and Meer P. Mean shift: A robust approach toward feature space analysis. *PAMI*, 24(5):603–619, 2002.
- [4] Frédéric Jurie and Michel Dhome. Hyperplane approximation for template matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(07):996–1000, 2002.
- [5] Frederic Jurie and Michel Dhome. A simple and efficient template matching algorithm. *iccv*, 02:544, 2001.
- [6] Zimmermann K., Matas J., and Svoboda T. Tracking by an optimal sequence of linear predictors. *Transaction on Pattern Analysis Machine Intelligence*, 2008. To appear.
- [7] Bruce D. Lucas. *Generalized Image Matching by the Method of Differences*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, July 1984.
- [8] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision (darpa). In *Proceedings of the 1981 DARPA Image Understanding Workshop*, pages 121–130, April 1981.
- [9] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image Processing Analysis and Machine Vision*. Thomson, 3th edition, 2008.