

# Zaostrovanie tekutej šošovky v reálnom čase

Martin Zaťovič



## Abstrakt

Rýchlosť zaužívaného prístupu k zaostrovaniu kamerových systémov naráža na úzke hrdlo - mechanický pohyb motora, ktorý mení vzdialenosť šošovky od snímaného objektu. Tento pohyb spomaľuje zaostrovanie, ktoré je v mnohých prípadoch kritická operácia zariadení, ktoré využívajú kamerový systém, pretože vzniká potreba vyčkávať na správne zaostrenie. Riešenie predstavuje inovatívna technológia tekutej šošovky. Tekutá šošovka eliminuje potrebu mechanického pohybu a vďaka tomu je schopná takéto systémy urýchliť. Budeme sa venovať overeniu rýchlosti tejto technológie, jej výhodám a nevýhodám a ukážeme si jej využitie na príkladoch z reálneho sveta. Ideálnym príkladom pre využitie tekutej šošovky sú zariadenia, ktoré často preostrujú - hĺbkový skener, alebo linka, ktorá sníma štítky s textom na balíkoch rôznej veľkosti.

**Kľúčové slová:** Zaostrovanie — Tekutá šošovka — Kamera

**Priložené materiály:** [Demonstračné video](#) — [Kód\(github\)](#)

\*[xzatov00@fit.vutbr.cz](mailto:xzatov00@fit.vutbr.cz), *Fakulta informačných technológií, Vysoké učení technické v Brne*

## 1. Úvod

Tento projekt overuje a demonštruje rýchlosť preostrovania kamery za použitia tekutej šošovky. Absencia potreby mechanického pohybu motora pre preostrenie (ako tomu je pri klasických šošovkách) znamená podstatné zrýchlenie tejto operácie a väčšiu životnosť šošovky. Použitie tekutej šošovky dokáže zrychliť akýkoľvek systém, ktorý používa kameru a počíta s časom na preostrenie a zároveň predĺži jeho životnosť [1]. Práve týmito dvoma vlastnosťami tekutej šošovky vďaka za svoju stále rastúcu popularitu. Existuje množstvo rôznorodých projektov založených na tejto technológii - od biometrických zariadení a mikroskopov až po teleskopy a hĺbkové skenery.

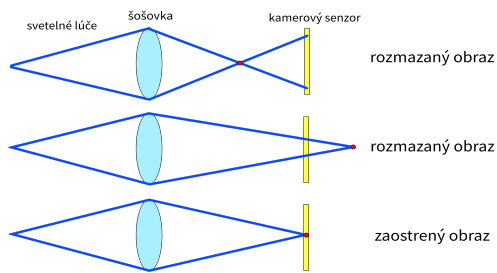
## 2. Šošovky

Šošovka je optické zariadenie, ktoré mení smer svetelných lúčov. V kamerových systémoch sa využíva na sústredenie obrazu, ktorý dokáže vonkajšia strana šošovky poňať na malý kamerový čip, ktorý lúče sníma. Fokálna vzdialenosť šošovky hovorí o tom ako veľmi šošovka sústredí, alebo rozptyľuje svetlo.

### 2.1 Princíp ostrenia kamerových systémov

Svetelným zdrojom nazývame objekt, ktorý vyžaruje svetelné lúče, napríklad slnko, alebo žiarovka. Objekt snímaný kamerou odráža svetlo zo svetelného zdroja pod takmer nekonečným množstvom uhlov. Úlohou šošovky, je sústrediť tieto lúče do jediného bodu - na

senzor kamery. Ak je svetlo sústredené na bod pred, alebo za sensorom, výsledný obrázok bude rozostrený.



**Obrázok 1.** Pozorujeme, že ak šošovka sústredí svetlo do bodu pred šošovkou, lúče pokračujú ďalej za tento bod a senzor pretnú na rôznych miestach. Naopak, ak je tento bod za šošovkou, lúče sa navzájom nepretnú, ale opäť dopadnú na rozličné body senzoru. Posledný príklad demonštruje prípad, kedy sú svetelné lúče sústredené do bodu, ktorý patrí senzoru - v takom prípade bude obraz ostrý.

Aby kamerový systém snímal ostrý obraz, je potrebné docieľiť sústredenie zachytávaných svetelných lúčov na senzor. Kamerové systémy s mechanickou šošovkou na to využívajú pohyb šošovky smerom k, alebo od senzoru. Pokiaľ sa bod v ktorom sa svetelné lúče pretínajú nachádza pred sensorom, posunú ju bližšie. Naopak, pokiaľ sa tento bod nachádza za šošovkou, je potrebné šošovku posunúť smerom od neho. Vzhľadom na čas potrebný na vykonanie takéhoto pohybu sa práve ostrenie stáva úzkym hrdlom kamerových systémov.

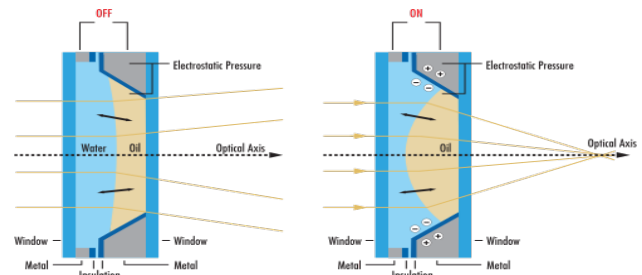
## 2.2 Tekutá šošovka

Riešenie spomenutého problému prináša futuristicky pôsobiaca technológia tekutej šošovky. Samotnú šošovku vytvára rozhranie dvoch nezmiešateľných tekutín. Tekutiny sú umiestnené medzi dve sklenené okná zo strán a dve železné časti zhora a zdola. Aplikovanie napätia na tieto železné časti spôsobí, že začnú byť menej hydrofóbne, následkom čoho sa k nim presunie viac vody [2]. Takýmto spôsobom tekutá šošovka mení svoje zakrivenie a tak aj fokálnu vzdialenosť (2). Šošovku je možné zakrivovať konvexne aj konkávne, čo zaručuje veľký interval možných hodnôt zaostrenia.

Proces zmeny ostroty sa skladá zo zmeny napätia na rozhraní šošovky a samotnej zmeny zakrivenia šošovky. Čas potrebný na zmenu zakrivenia šošovky sa pohybuje v jednotkách milisekúnd, a preto je zanedbateľný. Rýchlosť ostrenia obmedzuje preto predovšetkým rýchlosť zmeny napätia na výstupe ovládača.

Pre veľké tekuté šošovky zmena napätia na železných častiach nespôsobí dosť podstatnú zmenu v ich hydrofóbií na to, aby sa tekutina výrazne premiestnila. Pre väčšiu tekutú šošovku, taktiež platí, že čas potrebný na premiestnenie tekutiny a teda na zmenu

jej tvaru rastie spolu s jej veľkosťou a stáva sa tak menej efektívnou. Ďalšou nevýhodou je, obmedzenie rozlíšenia, ktoré taktiež plynie z veľkosti tekutej šošovky. Tekuté šošovky s veľkým priemerom nie sú schopné produkovať obraz o vysokom rozlíšení. Z týchto dôvodov nie je vhodné konštruovať tekuté šošovky s veľkým priemerom.



**Obrázok 2.** Princíp zmeny zakrivenia tekutej šošovky [1]

## 2.3 Tekutá vs. mechanická šošovka

Najvýraznejšou výhodou tekutej šošovky oproti mechanickej je jej rýchlosť, ale nie je to jediná výhoda, ktorú táto technológia ponúka. Absencia posuvného krokového motoru znamená, že samotná šošovka môže byť menšia ako jej mechanická alternatíva.

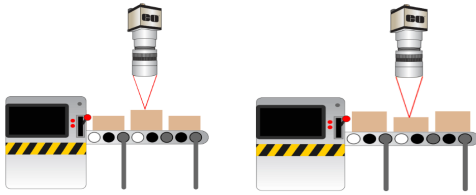
Pri mechanickej šošovke dochádza k opotrebovaniu mechanickej časti, ktoré môžu po dlhšom používaní zlyhať. Na tekutú šošovku nemá dlhšie používanie takmer žiaden vplyv, a preto vydrží až približne 50 000 000 cyklov preostrovania, čo je približne 500 krát viac ako tomu je pri mechanickej šošovke [1].

## 3. Využitia tekutej šošovky

### 3.1 Linka na snímanie balíkov na pošte

Vďaka jej rýchlosti sa tekutá šošovka dá využiť napríklad na linke na pošte, ktorá má za úlohu odfotografovať štítky s adresou na balíkoch. Štítok môže obsahovať text s adresou prijímateľa balíku, QR kód, čiarový kód, alebo iný rozpoznávací prvok, na ktorý je potrebné zaostriť. Kamera je umiestnená nad linkou. Keďže každý balík je inej veľkosti, linka musí pred zhotovením snímku preosťriť na každý balík osobitne. Modelová situácia je znázornená na obr. 3. Použitie tekutej šošovky zrýchli čas potrebný na zaostrenie na jednotlivé balíky, a dokáže tak výrazne zrýchliť takúto linku. Pre porovnanie budú použité dva kamerové systémy - jeden s tekutou šošovkou a druhý s mechanickej šošovkou.

Standardne sa tento problém rieši čakaním linky na preostrenie kamery - celá linka je zastavená a čaká na zhotovenie zaostreného snímku balíku. Takéto vyčkávanie je potrebné, aby nedošlo k situácií, kedy



**Obrázok 3.** Kamera s tekutou šošovkou, snímajúca balíky na linke. Každý balík môže byť rôzne vysoký a vzniká tak potreba správne zaostriť na každý balík zvlášť [1]

by sa balík posunul príliš ďaleko v rámci linky a nebol by tak v zornom poli kamery ešte počas zaostrovania. Pre optimalizovanie dĺžky čakania na preostrenie linka používa senzor vzdialenosti, vďaka ktorému určí vzdialenosť, na ktorú bude šošovka preostrovať. Pre systém to znamená, že čas na obsluhu jedného produktu bude približne rovný času potrebnému pre získanie vzdialenosti senzorom a preostrenie kamery na vypočítanú hodnotu.

Alternatívnym riešením je automatické zaoštrovanie na balík pomocou algoritmu, ktorý skokovo prechádza všetky možné úrovne zaoštrovania šošovky, pre každú zhotoví snímok a vypočíta jej skóre ostrosti. Ak je toto skóre vyššie ako doteraz maximálne nájdené skóre, uloží si ho spolu s hodnotou zaoštrovania. Takýmto spôsobom nájde najostrejší obrázok pomocou spracovania obrazu [3]. Nevýhodou tohto prístupu je niekoľkonásobné spomalenie, spôsobené potrebou snímať a analyzovať väčšie množstvo snímokov.

Naše riešenie vychádza z už existujúcich riešení a aplikuje ich za použitia tekutej šošovky. Keďže pri použití senzoru vzdialenosti stačí preostríť šošovku jedinýkrát - budeme testovať rýchlosť jediného preostrenia tekutej a mechanickej šošovky z bodu A na bod B. Pre druhý prístup, ktorý využíva vyhodnotenie ostrosti obrazu implementujeme aplikáciu, ktorá bude získavať a spracovávať obraz a následne zaostrí na objekt v strede zorného poľa kamery. Tento program budeme testovať nad oboma použitými šošovkami, výsledky následne porovnáme a vyhodnotíme [4].

### 3.2 Hĺbkový skener

Je zrejmé, že tekuté šošovky sú obzvlášť výhodné pre systémy, ktoré preostrujú často počas krátkeho časového intervalu. Príkladom takéhoto zariadenia je hĺbkový skener, ktorý je možné implementovať na základe obrázkov získaných preostrovaním na rôzne vzdialenosti. Zariadenie zaostrí najskôr na najmenšiu možnú vzdialenosť, získa snímok a vyberie z neho časti, ktoré sú v rámci obrázku ostré. Keďže zariadenie uchováva informáciu o tom na akú vzdialenosť za-

ostruje, vie ju v ďalšom kroku priradiť ostrým častiam obrázka, a získať tak 3D mapu fotografovaného priestoru. Takto zosníma celú vzdialenosť na ktorú dokáže zaostríť. Aby skener pracoval s rozumným rozlíšením, potrebuje získať veľké množstvo snímokov, pričom pred každým musí preostríť na správnu vzdialenosť. Hĺbkový skener je samozrejme možné zostrojiť aj pomocou mechanickej šošovky, ale takéto zariadenie bude výrazne pomalšie.

### 3.3 Smartfóny

Zakrivujúca sa šošovka neunikla ani pozornosti veľkých firiem. Firma Xiaomi v roku 2021 vydala prvý smartfón na svete obsahujúci kameru s tekutou šošovkou - Xiaomi Mi Mix Fold. Rozlíšenie kamery je 8MPx, pričom ponúka dva módy - mód teleobjektívu a makro mód, ktorý umožňuje zaoštrovanie už na vzdialenosť 3cm. Kamera podporuje maximálne 30x hybridné - optické a digitálne priblíženie [5].

## 4. Automatické zaoštrovanie

Automatické zaoštrovanie je obsiahla téma s viacerými možnými prístupmi. V tejto práci sa budeme venovať dvom z nich - prvý využíva senzor vzdialenosti na získanie optimálnej hodnoty zaoštrovania, zatiaľčo druhý nepoužíva žiaden ďalší senzor a optimálnu hodnotu zaoštrovania šošovky hľadá pomocou vyhodnocovania ostrosti obrazu.

Prístup založený na použití senzoru vzdialenosti je založený na tabuľke, ktorá uchováva informácie o tom, akú hodnotu je potrebné poslať ovládaču šošovky, aby ju zaostril na určitý interval vzdialeností. Najskôr senzor zmeria vzdialenosť od objektu, následne vyhledáme interval, do ktorého vzdialenosť spadá a príslušnú hodnotu pošle ovládaču, ktorý šošovku preostrí. Výhodou tohto prístupu je, že na správne zaoštrovanie stačí jediný preostrenie. Limituje ho však rýchlosť a presnosť senzoru vzdialenosti.

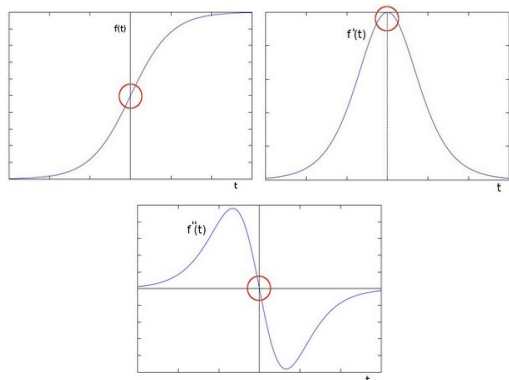
Existuje mnoho prístupov k automatickému zaoštrovaniu na základe vyhodnotenia ostrosti obrázku. V tomto článku sa budeme venovať len jedinému z nich - tzv. sweep algoritmu. Algoritmus snímá obrázky pri rôznych hodnotách zaoštrovania šošovky a počíta skóre ich ostrosti. Po preskúmaní celého intervalu hodnôt zaoštrovania vyberie obrázok s najvyšším skóre ostrosti a ovládaču šošovky zašle hodnotu pri ktorej bol zosnímaný [3].

V rámci projektu sme implementovali aplikáciu, ktorá využíva algoritmus sweep (14). Táto aplikácia prechádza celý interval možných hodnôt ostrosti a snaží sa nájsť hodnotu, pri ktorej je obraz najostrejší. Zvyčajne neskúmame každú možnú hodnotu, na ktorú

vieme zaostriť, ale v každej iterácii pričítame k hodnote posielanej ovládaču inkrement. V opačnom prípade by zaostrovanie trvalo veľmi dlho.

Na vyhodnotenie skóre ostrotí budeme používať tieto nástroje - Sobel operátor, Laplaceov operátor, Cannyho hranový detektor a Rýchlu Fourierovu transformáciu(FFT). Všetky sú implementované v rámci knižnice OpenCV [6]. Výstupom Sobelovho operátora, Laplaceovho operátora a Cannyho hranového detektoru je obrázok pozostávajúci z identifikovaných hrán.

Hrana je definovaná ako prudká zmena intenzity susediacich pixelov. Detektory hrán zvyčajne počítajú prvé, alebo druhé derivácie funkcie intenzity pixelov obrázku. Hrana je potom identifikovaná ako lokálne maximum funkcie, alebo hodnota 0 (4). čím viac hrán sa na obrázku nachádza, tým je vyššia pravdepodobnosť, že tento obrázok je ostrý. Túto vlastnosť využijeme pri odhade skóre ostroti obrázku - vypočítame priemernú hodnotu intenzity pixelov z výstupu hranového detektoru [7] [8].



**Obrázok 4.** Graf intenzity jedného riadku obrázku, ktorý obsahuje jedinú hranu a jeho prvá a druhá deriváciu. červený krúžok reprezentuje bod v ktorom je hrana identifikovaná bežnými hranovými detektormi.

#### 4.1 Grafy skóre ostrotí

Pred samotnou implementáciou sme nechali vykresliť grafy obsahujúce hodnoty, ktoré zasielame ovládaču na ose x a skóre ostrotí obrázku pri danej hodnote na ose y. Grafy sme vykreslili pre všetky metódy výpočtu skóre ostroti, ktorými sa budeme zaoberať.

Pre vykreslenie grafov používame 16-bitový ovládač, avšak pre zjednodušenie sme analyzovali len snímok pri každej 256. hodnote. Hodnoty na ose x teda reprezentujú hodnotu  $x*256$ . Tento graf je dôležitý pre určenie inkrementu. Ak určíme inkrement, ktorý bude príliš veľký, mohlo by sa stať, že by sme preskočili celú časť grafu, ktorá reprezentuje ostrý obrázok a nikdy by sme tak nezaostri. Naopak, ak by bol inkrement príliš malý, program bude snímať a analyzovať zbytočne

veľké množstvo snímok, a preto bude pomalý.

Pomocou grafov skóre ostrotí sme určili optimálnu hodnotu inkrementu pre jednotlivé metódy vyhodnotenia ostroti obrazu, ktorými sa budeme zaoberať. Tieto hodnoty boli otestované na algoritme automatického zaostrenia pomocou algoritmu sweep (4.7) a upravené, tak aby algoritmus s istotou našiel maximum v grafe. Pri zvolených inkrementoch potrebuje algoritmus na preskúmanie intervalu hodnôt, ktoré je možné ovládaču poslať, v priemere 25 snímok.

Grafy skóre ostrotí sú priložené k popisom jednotlivých metód výpočtu skóre ostroti. Scéna pre ktorú boli grafy ostrotí vykreslené obsahuje balík so štítkom vo vzdialenosti 15cm. Na štítku je vytlačená adresa príjemateľa balíku. Pre algoritmy automatického zaostrovania je dôležité, aby s istotou našli maximum v grafe, preto je dôležité zvoliť správny inkrement pre danú scénu.

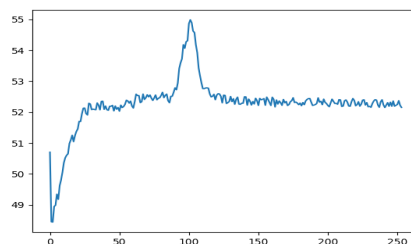
#### 4.2 Sobelov operátor

Sobelov operátor detekuje hrany na základe prvej derivácie funkcie intenzity pixelov obrázku. Tieto derivácie počíta pomocou konvolúcie obrázku s tzv. Sobelovou maticou. Ide o filter, ktorý identifikuje a dáva na výstup miesta s vysokým rozdielom v intenzite pixelov. Sobelov operátor používa rôzne Sobelové matice pre detekciu hrán v horizontálnom a vertikálnom smere [9].

Nižšie (1) sú zobrazené Sobelové 3x3 matice pre detekciu hrán v oboch smeroch. Matice tohto formátu detekujú hrany na základe veľmi blízkeho okolia práve spracovávaného pixelu. Existujú aj väčšie formáty matíc, ktoré počítajú s väčším okolím pixelu [7].

$$S_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & -0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (1)$$

Pre detekciu hrán v oboch smeroch je vykonaná konvolúcia s oboma filtermi a výstupy sú následne sčítané.



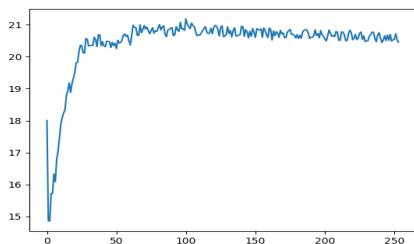
**Obrázok 5.** Graf ostrotí pri použití funkcie Sobel z knižnice OpenCV, osa x obsahuje hodnoty posielané ovládaču šošovky a osa y skóre ostroti obrazu.



### 4.3 Laplaceov operátor

Laplaceov operátor je založený na počítaní druhej derivácie funkcie intenzity pixelov obrázku. Hrany sú teda identifikované hodnotou 0 (4). Počítanie derivácií je opäť zjednodušené konvolúciou so Sobelovými maticami [8] :

$$LP_x = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} LP_y = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (2)$$

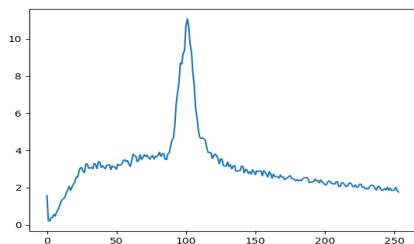


Obrázok 6. Graf ostrostí pri použití funkcie Laplacian z knižnice OpenCV.

### 4.4 Cannyho hranový detektor

Canny si uložil 5 základných podmienok na hranový detektor, ktorý následne vyvinul. Ide o komplexný detektor hrán, ktorý sa snaží o detekciu čo najväčšieho počtu hrán a následnej filtrácií nevýrazných, alebo neprepojených hrán [10].

V prvých krokoch vykonáva konvolúciu so štyrma Sobelovými maticami pre identifikáciu hrán aj v diagonálnych smeroch. Následne filtruje výrazné hrany [9].

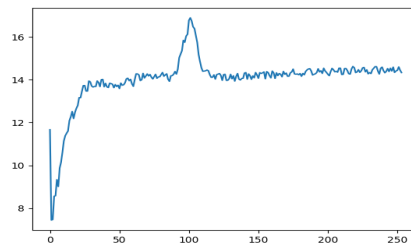


Obrázok 7. Graf ostrostí pri použití funkcie Canny z knižnice OpenCV.

### 4.5 Rýchla Fourierova transformácia

Táto metóda vyhľadáva hrany na základe rýchlej Fourierovej transformácie. Po jej výpočte aplikujeme vysoko-frekvenčný filter, keď že hrany sú zvyčajne reprezentované vysokými frekvenciami. Následne vypočítame inverznú Fourierovu transformáciu a výstupom je obrázok obsahujúci miesta, kde boli identifikované vysoké frekvencie - hrany. Na záver vypočítame priemernú in-

tenzitu pixelov výstupu inverznej transformácie, ktorá nesie informáciu o ostrosti obrázu [11].



Obrázok 8. Graf ostrostí pri použití funkcie fft a idft z knižnice OpenCV.

### 4.6 Výber metódy evaluácie ostrosti obrázu pre algoritmus automatického zaostrenia

Pri výbere vhodnej metódy výpočtu skóre ostrosti budeme sledovať dve vlastnosti - ako ľahké je nájsť maximum v grafe skóre ostrosti danej metódy a jej rýchlosť. Rýchlosť sledujeme pretože práve výpočet skóre ostrosti sa za použitia tekutej šošovky môže stať úzkym hrdlom systému. Snažíme sa teda docieľiť, aby vyhodnotenie obrázu prebiehalo čo najrýchlejšie.

Z grafov skóre ostrosti jednotlivých metód usudzujeme, že pri použití Laplaceovho operátora bude obťažnejšie nájsť maximum ako pri ostatných metódach. Usudzujeme tak na základe nevýrazného maxima v grafe. Čím plytšie graf stúpa do a klesá z maxima, tým jednoduchšie bude pre nás ho nájsť.

Keďže grafy skóre ostrosti Sobelovho operátora, Cannyho hranového detektoru a hranovej detekcie pomocou rýchlej Fourierovej transformácie vyzerajú veľmi podobne, budeme sledovať ich rýchlosti. Pre porovnanie necháme vypočítať skóre ostrosti pre 255 obrázkov, ktoré sme analyzovali pre vykreslenie grafu skóre ostrosti každou metódou a zistíme čas potrebný na všetky tieto výpočty.

- Sobelov operátor - 2.48232s
- Laplaceov operátor - 2.72255s
- Cannyho hranový detektor - 2.13336s
- Detekcia hrán pomocou FFT - 32.7225s

Cannyho hranový detektor napriek jeho komplexnosti nestráca na rýchlosti. Je spomedzi spomenutých metód najrýchlejší a najkomplexnejší, keď že zároveň filtruje hrany, ktoré nie sú výrazné. Z týchto dôvodov sme pre algoritmus automatického ostrenia vybrali túto metódu.

## 4.7 Algoritmus

### Algorithm 1 Algoritmus sweep

```
skore, max_skore, hodnota, max_hodnota = 0
increment = 2620
for i = 0; i <= 25; i++ do
    obrazok = get_image()
    hrany = Canny(image)
    skore = mean(hrany)
    if skore > max_skore then
        max_skore = skore
        max_hodnota = hodnota
    end if
    hodnota += increment
    ovladac_write(hodnota)
end for
ovladac_write(max_hodnota)
```

## 4.8 Nasadenie

Po krátkom testovaní kódu implementovaného v jazyku C++ bolo očividné, že šošovka za použitia takto rýchlej slučky nestíha preostrovať. Všimli sme si, že algoritmus často nezaostroval správne a tak sme analyzovali dáta, ktoré prijíma od kamery. Zistili sme, že napätie na šošovke nie je ustálené, keď kamera sníma obrázok, ktorý považuje za skutočne zaostrý na poslanú hodnotu.

Toto správanie sme spočiatku riešili funkciou `sleep` z knižnice `thread`. Neskôr sme presunuli príkaz na preostrenie na začiatok slučky `for`. Takýmto spôsobom systém zašle ovládaču príkaz na preostrenie pre ďalší snímok ešte predtým, ako začne počítať skóre ostrosti aktuálne spracovávaného snímku. Skóre ostrosti je teda vyhodnocované v rovnakom čase ako šošovka preostruje. Algoritmus počas počítania ostrosti nevykonáva ďalší kód a vďaka tomu môžeme odstrániť funkciu na uspanie `sleep`.

Algoritmus bol teda optimalizovaný tak, aby preostroval hneď na začiatku slučky `for`. Šošovka teda preostruje pre snímanie nového snímku v rovnakom čase ako sa počíta skóre ostrosti iného snímku. Pre snímanie obrazu z kamery a jeho živé zobrazenie sa starajú odlišné vlákna.

## 5. Testovanie šošoviek a použitý hardvér

Aby sme overili rýchlosť tekutej šošovky oproti zaužívanému mechanickému riešeniu, podrobili sme obe šošovky testovaniu. Pre testovanie tekutej šošovky sme použili nasledovné komponenty:

- Tekutá šošovka - [TECHSPEC® Liquid Lens Cx Series Fixed Focal Length Lens, 35mm](#)

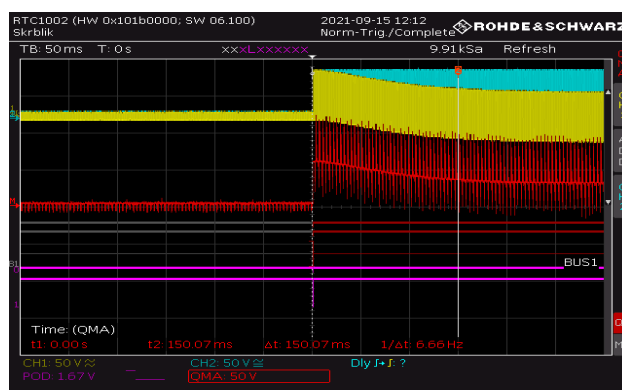
- Kamera - [IDS UI-3860CP Rev. 2](#)
- Ovládač šošovky - [Corning® Varioptic® Flexi-board with Maxim MAX14574 Driver](#)
- Výpočetná jednotka - [Raspberry Pi 4 model B](#)

Pre porovnanie sme otestovali aj mechanickú šošovku:

- Kamerový systém - [Arducam 5MP PTZ Camera](#)
- Výpočetná jednotka - [Raspberry Pi 4 model B](#)

## 5.1 Test rýchlosti tekutej šošovky

V kapitole 2.2 sme spomenuli, že pri preostrovaní tekutej šošovky je čas potrebný na samotnú zmenu zakrivenia šošovky zanedbateľný. Podstatná je rýchlosť zmeny napätia na rozhraní šošovky. Pre testovacie účely meriame zmenu z minimálneho možného napätia, ktoré dokáže ovládač na výstupe produkovať (najvzdialenejší bod na ktorý možno zaostriť) na maximálnu (najbližší bod na ktorý možno zaostriť). Túto zmenu sme zaznamenali osciloskopom a získali sme nasledovný výstup:



**Obrázok 9.** Zmena hodnoty diferenčného napätia na výstupe ovládača tekutej šošovky z minimálneho na maximálne vizualizovaná osciloskopom. Os x znázorňuje čas, pričom jeden diel reprezentuje 50ms. Os y znázorňuje merané diferenčné napätie v čase x. Pozorujeme, že ovládač pri prijatí signálu nastaví napätie na približne dvojnásobok požadovanej hodnoty a následne ho zníži na požadovanú hodnotu.

Na základe obr. 9 konštatujeme, že ustálenie diferenčného napätia v najhoršom prípade nastane približne po 150ms. Šošovka teda dokončí preostrovanie z najbližšieho bodu na najvzdialenejší za 150ms. Rovnakým spôsobom sme otestovali rýchlosť preostrovania cez 1/25, 1/5, 2/5, 1/2, 3/5 a 4/5 vzdialenosti, na ktorú dokáže šošovka preosťriť.

Druhý test bol vykonaný pomocou implementovaného algoritmu na zaostrenie pomocou spracovania obrazu. Bol sledovaný čas potrebný na kompletne vykonanie tohto algoritmu.

## 5.2 Test rýchlosti mechanickej šošovky

Mechanickú šošovku sme testovali rovnakým spôsobom. Najskôr sme zistili časy potrebné na preostrenie cez úseky vzdialenosti, na ktorú je možné zaostriť a potom sme spustili algoritmus na automatické zaostrenie pomocou spracovania obrazu.

## 5.3 Výsledky testov

Výsledok prvého testu sme zhrnuli do tabuľky (1). Pri jedinom preostrení z bodu A na bod B je tekutá šošovka v priemere približne 17 krát rýchlejšia ako mechanická.

-	1/25	1/5	2/5	1/2	3/5	4/5	1
TŠ	20	75	90	100	105	130	150
MŠ	200	740	1440	1780	2115	2820	3485

**Tabuľka 1.** Tabuľka zobrazuje časy potrebné na preostrenie tekutej a mechanickej šošovky cez rôzne dlhé úseky. Záhlavie obsahuje dĺžky týchto úsekov, ktoré sú udávané voči maximálnej dĺžke, cez ktorú dokážu šošovky preostríť. Časy sú udávané v milisekundách.

Z výsledkov (1) usudzujeme, že čas potrebný na preostrenie tekutej šošovky sa zvyšuje so zväčšujúcou sa vzdialenosťou, cez ktorú preostruje - šošovky preostrujú rýchlejšie cez krátku vzdialenosť. Usudzujeme tak, pretože čas potrebný na zmenu napätia na výstupe ovládača je kratší pre menší inkrement zasielanej hodnoty. Index 1/25 na osi x znamená pre 16-bitový ovládač zmenu hodnoty preostrenia o 2621:

$$2^{16} * (1/25) = 2621.44 \quad (3)$$

Obom šošovkám trvá preostrenie cez ľubovoľnú vzdialenosť v jednom kroku kratšie, ako preostrovanie cez rovnakú vzdialenosť vo viacerých krokoch.

Automatické zaostrenie pomocou spracovania obrazu za použitia tekutej šošovky trvalo zariadeniu v priemere 3,6 sekundy, čo je približne 4,48-krát rýchlejšie ako za použitia mechanickej šošovky. Mechanickej šošovke trval tento úkon v priemere 16,6 sekundy. Rozdiel nie je taký výrazný ako v predchádzajúcom teste pretože zariadenie spomaľuje rýchlosť čítania obrázkov z kamery.

Zaujímavou štatistikou je, že za čas potrebný na preostrenie mechanickej šošovky z najbližšieho možného bodu na najvzdialenejší, zvládne zariadenie s tekutou šošovkou vykonať automatické zaostrenie pomocou spracovania obrazu, ktoré obnáša 36 preostrení a zhotovení snímkov.

## 6. Záver

Táto práca potvrdila, že tekuté šošovky pracujú podstatne rýchlejšie ako mechanické. Rozdiel ich rýchlostí závisí na konkrétnej aplikácii, avšak v žiadnom prípade nie je zanedbateľný.

Použitie tejto technológie dokáže zrýchliť akékoľvek zariadenie, ktoré využíva kameru a počíta s časom potrebným na preostrenie. V prípade linky na pošte, ktorá sníma štítky na balíkoch, závisí na tom aký zvolíme prístup k automatickému zaostrovaniu. Za použitia senzoru snímajúceho vzdialenosť dokáže tekutá šošovka zrýchliť takýto systém približne 17 krát. V prípade zaostrovania pomocou spracovania obrazu sme pozorovali 4,48-násobné zrýchlenie.

Nevýhodou tekutých šošoviek je obmedzenie maximálneho dosiahnuteľného rozlíšenia. V prípade potreby použitia veľkej šošovky preto tekutá šošovka zrejme nie je vhodným kandidátom.

Automatické zaostrovanie na základe spracovania obrazu sa dá docieľiť pomocou detekcie hrán. Na túto detekciu slúžia operátory ako Sobel, Laplacian, alebo Canny. Úzkym hrdlom tohoto prístupu je zväčša rýchlosť prenosu dát z kamery do výpočtovej jednotky, alebo samotný výpočet detekcie hrán.

Čas potrebný na preostrenie tekutej šošovky sa pohybuje v radoch desiatok milisekúnd v závislosti na dĺžke úseku, cez ktorý preostruje. Takto nízky čas z nej robí čoraz žiadanejšiu technológiu, ktorú budeme pravdepodobne v blízkej budúcnosti vídiť čoraz častejšie.

Tekutá šošovka je lacná a inovatívna technológia, ktorá poskytuje veľa výhod. Vďaka jej rýchlosti a flexibilitě je možné, že v blízkej budúcnosti nahradí multikamerové systémy v dnešných smartfónoch. Jej použiteľnosť bude však vo veľkej miere závisieť na jej popularizácii a podporných softvéroch. Vďaka jej zloženiu je jej produkčná cena nízka a určite preto nájde využitie ako lacnejšia alternatíva mechanických šošoviek v mnohých odvetviach [12]. Je pravdepodobné, že nahradí veľkú časť mechanických šošoviek, avšak limitované rozlíšenie pri väčšom priemere šošovky naznačuje, že určite nenahradí všetky.

V budúcnosti by sme chceli vyvinúť kamerový systém s tekutou šošovkou a senzorom vzdialenosti a pridať osvetlenie časované uzávierkou kamery. Do systému založenom na výpočte ostrosti obrazu chceme pridať detekciu textu v obraze. Na základe takejto detekcie by algoritmus automatického zaostrovania počítal len ostrosť časti obrázka, ktorá obsahuje text a optimalizovali by sme tak čas potrebný na výpočet skóre ostrosti. Ďalej optimalizujeme pracovnú vzdialenosť systému, čo znamená, že obmedzíme interval

hodnôt zasielaných ovládaču na interval zodpovedajúci pracovnej vzdialenosti linky. Kamerový systém bude teda preostrovať cez menší interval hodnôt a bude tak na preostrovanie potrebovať menej času.

## PodĎakovanie

Za pomoc, odborný dohľad a poskytnutie hardvéru ďakujem pánovi inžinierovi Štefánovi Rydlovi, ktorý túto prácu vedie.

## Literatúra

- [1] EdmundOptics. Liquid lens features, applications, and technology. blogpost (english). <https://www.edmundoptics.com/knowledge-center/application-notes/imaging/liquid-lenses-in-imaging/>.
- [2] Second Wiki. Liquid lens - flüssiglinse. online, 11 2021. <https://second.wiki/wiki/flc3bcssiglinse>.
- [3] Chiou-Shann Fuh Wei-Sheng Liao. Autofocus. <https://www.csie.ntu.edu.tw/~fuh/personal/ImagesRecognition.Vol.9,No.4.Autofocus.pdf>.
- [4] Edmund Optics. Liquid lenses in machine vision. [https://www.edmundoptics.eu/ViewDocument/EO\\_Liquid\\_Lenses\\_in\\_Machine\\_vision\\_0419\\_EN.pdf](https://www.edmundoptics.eu/ViewDocument/EO_Liquid_Lenses_in_Machine_vision_0419_EN.pdf).
- [5] Jaron Schneider. The first smartphone to use a liquid lens is the xiaomi mi mix fold. <https://petapixel.com/2021/03/30/the-first-smartphone-to-use-a-liquid-lens-is-the-xiaomi-mi-mix-fold/>.
- [6] OpenCV. Opencv documentation. online, 01 2022. <https://docs.opencv.org/4.x/index.html>.
- [7] OpenCV. Opencv: Sobel derivatives. online, 01 2022.
- [8] OpenCV. Laplace operator. online, 01 2022. [https://docs.opencv.org/4.x/d5/db5/tutorial\\_laplace\\_operator.html](https://docs.opencv.org/4.x/d5/db5/tutorial_laplace_operator.html).
- [9] Albertus Joko Santoso Ninyikiriza Deborah Lynn, Arefin Islam Sourav. Implementation of real-time edge detection using canny and sobel algorithms, Mar 2021. [https://www.researchgate.net/publication/350151154\\_Implementation\\_of\\_Real-Time\\_Edge\\_Detection\\_Using\\_Canny\\_and\\_Sobel\\_Algorithms](https://www.researchgate.net/publication/350151154_Implementation_of_Real-Time_Edge_Detection_Using_Canny_and_Sobel_Algorithms).
- [10] OpenCV. Canny edge detection. online, 01 2022. [https://docs.opencv.org/4.x/da/d22/tutorial\\_py\\_canny.html](https://docs.opencv.org/4.x/da/d22/tutorial_py_canny.html).
- [11] Adrian Rosebrock. Opencv fast fourier transform (fft) for blur detection in images and video streams. online, 06 2020. <https://www.pyimagesearch.com/2020/06/15/opencv-fast-fourier-transform-fft-for-blur-detection-in-images-and-video-streams/>.
- [12] Brad Wendes. Are liquid lenses the future of photography? online, April 2021. <https://fstoppers.com/gear/are-liquid-lenses-future-photography-557255>.