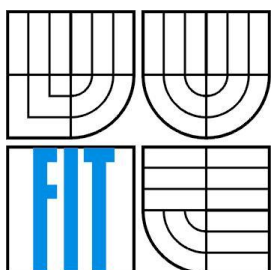


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY

DEPARTMENT OF COMPUTER SYSTEMS

DETEKCE OHNĚ A KOUŘE VE VIDEOSEKVENCI

DETECTION OF FIRE AND SMOKE IN VIDEOSIGNAL

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Peter Tomek

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Miroslav Švub

BRNO 2010

Zadání

1. Prostudujte základy zpracování obrazu. Zaměřte se na problematiku extrakce příznaků z obrazu a zpracování videosekvencí.
2. Zorientujte se v současných metodách vizuální detekce kouře a ohně v obraze.
3. Vyberte vhodnou metodu a navrhňte jednoduchý detektor kouře/ohně.
4. Experimentujte s vaší implementací a případně navrhňte vlastní modifikace metod.
5. Porovnejte dosažené výsledky a diskutujte možnosti budoucího vývoje. Zvažte další pokračování v rámci diplomové práce.
6. Vytvořte stručný plakát prezentující vaši bakalářskou práci, její cíle a výsledky.

Abstrakt

Cílem práce je analýza vstupní videosekvence a následné nalezení oblastí, které obsahujících oheň nebo kouř. Daný problém je rozdělen na dvě podúlohy—detekci ohně a detekci kouře. Základem a prvním krokem analýzy je detekce daných oblastí pomocí Gaussova zloženého modelu—Gaussian mixture model. Na jeho natrénování je použit algoritmus Expectation–Maximization, nebo skráceně EM. Pro detekci kouře se dále používá technika optického toku. Výsledné oblasti jsou pak upraveny pomocí aplikace různých morfologických metod a analýzou vzájemné polohy detekovaných oblastí. Výstupem algoritmu je opět videosekvence, ve které jsou zvýrazněny oblasti s pravděpodobnou přítomností ohně nebo kouře.

Abstract

This thesis aims to analyse a videosignal given in input and find segments that contains fire or smoke. The problem is divided into two cases—detection of fire and detection of smoke. The first and main step of analysis is detection of segments by Gaussian mixture model that is trained by Expectation–Maximization algorithm, or shortly EM algorithm. For smoke detection is than used method of optical flow. The final segments are than processed by some morphological methods and determination of their position is made. Finally, the output of algorithm is again a videosignal in which segments that probably contains fire or smoke are highlighted.

Klíčová slova

Oheň, kouř, Gaussov zložený model, GMM, Expectation–Maximization, EM, morfologické metody, eroze, dilatace, optický tok, barební model, RGB, HSV, digitální obraz, rastrová grafika, vektorová grafika.

Keywords

Fire, smoke, Gaussian mixture model, GMM, Expectation–Maximization, EM, morphological methods, erosion, dilatation, optical flow, color model, RGB, HSV, digital image, vector graphics, raster graphics.

Citace

Peter Tomek: Detekce ohně a kouře ve videosekvenci, bakalářská práce, Brno, FIT VUT v Brně, 2010

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Miroslava Švuba. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Peter Tomek
19.5.2010

Poděkování

Za aktivní pomoc a velké množství užitečných a konstruktivních připomínek děkuji panu Miroslavu Švubovi, který byl mým konzultantem.

© Peter Tomek, 2010

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod.....	7
2	Spracovanie obrazu.....	8
2.1	Digitálny obraz.....	8
2.1.1	Rastrový obrázok.....	8
2.1.2	Vektorový obrázok.....	8
2.1.3	Digitálne video.....	9
2.2	Farebné modely a priestory.....	9
2.2.1	RGB model.....	9
2.2.2	HSV model.....	9
2.2.3	Prevod RGB modelu na HSV.....	10
2.3	Optický tok.....	10
2.3.1	Matematický popis optického toku.....	11
2.3.2	Detekcia pohybu pomocou feature points.....	12
2.4	Gaussian mixture model.....	13
2.4.1	Matematický popis zloženého modelu.....	13
2.5	EM (Expectation Maximization).....	14
2.6	Morfologické operácie.....	14
2.6.1	Erózia.....	15
2.6.2	Dilatácia.....	16
2.6.3	Otvorenie.....	17
2.6.4	Zavretie.....	17
3	Návrh.....	19
3.1	Detekcia ohňa.....	19
3.1.1	Detekcia ohňa pomocou GMM.....	19
3.1.2	Určenie prahu.....	19
3.2	Výsledná úprava pomocou morfológických metód.....	20
3.2.1	Finálna detekcia ohňa.....	20
3.2.2	Detekcia dymu.....	20
3.2.3	Detekcia dymu pomocou GMM.....	20
3.2.4	Optický tok pri detekcii dymu.....	21
3.2.5	Výsledná úprava pomocou morfológických metód.....	21
3.2.6	Finálna detekcia dymu.....	21
3.3	Finálna realizácia—spojenie oboch metód.....	21

3.3.1	Segmentácia obrazu.....	21
3.3.2	Prienik segmentov	22
3.3.3	Vzájomná poloha segmentov.....	22
4	Implementácia.....	24
4.1	Gaussian mixture model — implementácia	24
4.2	Optický tok — implementácia	25
4.3	Morfologické metódy - implementácia.....	25
5	Experimentálne výsledky	26
5.1	Popis výstupu programu.....	26
5.2	Výber tréningových obrázkov a určenie prahu	27
5.3	Príklady chýb programu.....	28
6	Záver.....	30
6.1	Rozšírenia.....	30
7	Literatúra	31

1 Úvod

V posledných rokoch sa dramaticky zvýšil počet miest, ktoré sú monitorované rôznymi kamerovými systémami. Tieto miesta môžu byť verejné priestranstvá (ulice, parky, námestia,...), ale samozrejme aj súkromné priestory, do ktorých spadajú hlavne budovy a rôzne väčšie firemné alebo bytové komplexy. Hlavnou motiváciou pre inštaláciu kamerových systémov vo vyššie uvedených prípadoch je zvýšenie našej bezpečnosti. Správa a obsluha týchto zariadení vyžaduje však často nepretržitú prítomnosť človeka, čo zvyšuje finančné výdavky na fungovanie celého systému. Preto nastáva prirodzená potreba spracovávať informácie získané z kamerového systému strojovo (počítačom). Takýto prístup síce vyžaduje nemalú počiatočnú investíciu, napriek tomu nám umožní znížiť dlhodobé náklady a častokrát aj zvýšiť presnosť a rýchlosť detekcie vzniknutej situácie.

Jednou z mnohých nástrah, ktoré na nás číhajú v podstate na každom kroku je vznik požiaru. Na jeho včasnú detekciu sa vo veľkej miere používajú termodynamické snímače, ktoré ale majú oproti technike detekcie ohňa pomocou spracovania videa jednu vážnu nevýhodu. Tou je pochopiteľne dosah takéhoto snímača a tým súvisiaca vzdialenosť na akú je snímač schopný prípadný požiar detekovať. V tomto smere je výhoda detekcie ohňa z obrazu zrejmá. Presnosť detekcie je obmedzená hlavne kvalitou kamery a prípadným priblížením (zoom), čo však v dnešnej dobe dosahuje veľmi uspokojivých hodnôt.

V tejto práci sa teda zamerám špeciálne na detekciu ohňa a dymu vo videu.

2 Spracovanie obrazu

V tejto kapitole načrtnem, hlavne z teoretického hľadiska, základné techniky, ktoré sa používajú pri spracovaní obrazu.

Pod pojmom spracovanie obrazu sa chápe spracovanie dvojrozmerného obrazového signálu, ktorý môže mať charakter buď obrázku, videa, alebo rônych iných dát zozbieraných napríklad medicínskymi zariadeniami a pod. V tomto odvetví vzniklo množstvo rôznych techník a algoritmov, z ktorých popíšem tie najdôležitejšie, ktoré je možné aplikovať pri detekcii ohňa a dymu vo videu.

2.1 Digitálny obraz

V počítačovej technike sa pod pojmom *obraz* vždy myslí *digitálny obraz*, ktorý je reprezentovaný binárnou formou¹. V závislosti natom, či je rozlíšenie obrazu fixné, alebo sa môže meniť, sa digitálny obraz delí na *rastrový* a *vektorový*. Napriek tomu, ak nieje definované inak, sa pojem digitálny a rastrový obrázok v bežnej praxi zamieňajú².

2.1.1 Rastrový obrázok

Rastrový obrázok, je popísaný pomocou konečného počtu farebných bodov, ktoré sa nazývajú *pixeli*. Tieto body sú usporiadané do dvojrozmernej mriežky—matice, pričom každý má presne definovanú polohu. To znamená, že rozmery obrázku sú predom dané a nieje ich možné bez straty informácie zmeniť. Pixel je najmenšou zložkou obrázku a jeho hodnota definuje farbu daného bodu. Povolené rozsahy hodnôt, ktoré môže pixel nadobúdať definujú použité farebné modely, ktoré sú bližšie popísané v kapitole 2.2. Čím väčší počet hodnôt môže pixel mať, tým väčšie množstvo farieb je možné daným farebným modelom vyjadriť. Tento počet sa nazýva *farebná hĺbka*—*color depth* obrázku.

Rastrové obrázky je možné získať napríklad pomocou digitálneho fotoaparátu, kamery, scanneru, a pod.

2.1.2 Vektorový obrázok

Vektorový popis obrázku je druhý základný typ reprezentácie obrazu v počítačovej grafike. Na rozdiel od rastrových obrázkov, ktoré sú definované pravouhlou maticou, vektorový obrázok je zložený zo základných geometrických tvarov ako sú body, priamky, krivky a mnohouholníky.

Hlavnou výhodou vektorových obrázkov je, že nemajú fixne dané rozmery. To znamená, že je ich možné ľubovoľne zväčšovať alebo znižovať bez straty kvality. Ich reprezentácia v počítači je obvykle omnoho menšia ako v prípade rastrových obrázkov. Nevýhodou však ostáva náročnosť vytvorenia takéhoto snímku. Zatiaľ, čo rastrový obrázok môžeme vytvoriť jednoducho pomocou fotoaparátu alebo skeneru, na tvorbu vektorovej grafiky sa musia použiť špeciálne, nato určené editory³.

¹ Binárna reprezentácia znamená postupnosť núl a jednotiek.

² V literatúre sa niekedy označuje rastrový obrázok pojmom bitmapový obrázok. Jedná sa však o jeden typ.

³ Vektorové editory sú napr.: Adobe Illustrator, CorelDraw, Inkscape,...

2.1.3 Digitálne video

Digitálne video sa skladá zo série *digitálnych obrázkov*, ktoré sú zobrazované v rýchlej postupnosti konštantnou rýchlosťou. Tieto obrázky, ak sú chápané v kontexte videa, sa nazývajú snímky—frames. Parametrom videa je počet týchto snímok zobrazených za jednotku času. Väčšinou sa tento počet udáva v jednotkách *frames per second*—počet snímok za sekundu (FPS).

2.2 Farebné modely a priestory

Farebný model, je matematický model popisujúci spôsob, akým sa dá určitá farba vyjadriť pomocou postupnosti čísel. Táto postupnosť je väčšinou 3–4 prvková v závislosti od zložitosti modelu. Zoznam všetkých farieb, ktoré sa dajú pomocou farebného modelu vyjadriť sa nazýva *farebný priestor*.

2.2.1 RGB model

Farebný model RGB alebo *red–green–blue* je jedným z najrozšírenejších farebných modelov vďaka jeho využitiu v monitoroch a projektoroch. Využíva aditívne miešanie farieb⁴. Výsledná farba sa tvorí zmiešaním červeného, zeleného a modrého svetla v určitom pomere. Tento pomer je možné vyjadriť v percentách, alebo číselne v závislosti od počtu vyhradených bitov pre reprezentáciu jednej farebnej komponenty, pričom vyššia hodnota znamená väčšiu intenzitu farby. Typicky môže byť farebná intenzita jednej z troch základných komponent vyjadrená ôsmimi bitmi. Potom rozsah hodnôt pre danú komponentu je 0–255. Jednoduchým výpočtom sa získať množstvo všetkých farieb daného farebného priestoru: $255 \times 255 \times 255 \cong 16\,000\,000$ farieb.

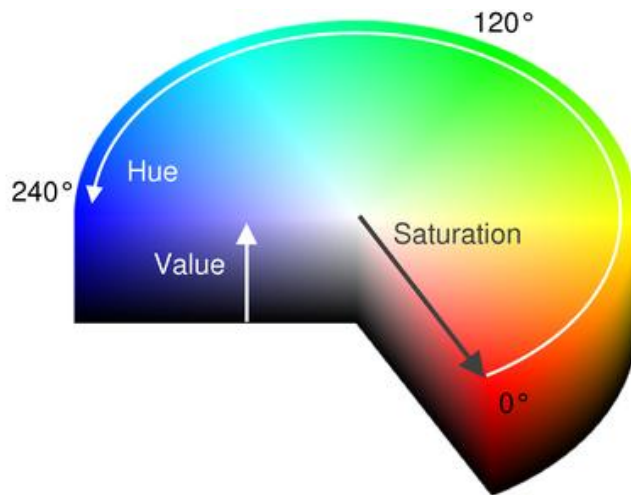
2.2.2 HSV model

HSV alebo *hue–saturation–value* model, na rozdiel od spomenutého modelu RGB, najviac odpovedá ľudskému vnímaniu farieb. Skladá sa z troch základných prvkov:

- *Hue* reprezentuje farebný tón alebo farebný odtieň. Vyjadruje sa v stupňoch v rozsahu hodnôt 0°–360°. Zvyčajne tento odtieň reprezentuje názov farby napr.: červená.
- *Saturation* vyjadruje sýtosť farby. Inak povedané to môže byť sila farby alebo jej čistota. Vyjadruje sa v percentách štandardne od 0–100%.
- *Value* reprezentuje jas alebo množstvo bieleho svetla obsiahnutého vo farbe. Táto hodnota vyjadruje koľko svetla farba odráža. Inými slovami, pre zníženie jas sa pridá do farby viac čiernej. Zadáva sa podobne ako saturation v percentách v rozsahu hodnôt 0–100%.

HSV model sa dá vyjadriť aj graficky pomocou farebného valca, kde jednotlivé zložky sú znázornené na obrázku nižšie.

⁴ Aditívne miešanie farieb (Additive color mixing) – spôsob miešania farieb, kedy sa jednotlivé zložky farby sčítavajú a vytvárajú svetlo väčšej intenzity.



Obrázok 1. Grafický popis HSV modelu

2.2.3 Prevod RGB modelu na HSV

Na transformáciu RGB modelu na HSV model neexistuje jednoduchá konverzná matica. Preto popíšem v jednotlivých krokoch princíp, akým je možné takúto transformáciu vykonať a spôsob, akým je možné takúto funkciu naimplementovať.

V nasledujúcich riadkoch bude funkcia $\max(a, b, \dots)$ určovať najvyššiu hodnotu z daných prvkov a $\min(a, b, \dots)$ hodnotu najnižšiu. Jednotlivé zložky farieb sú vyjadrené odpovedajúcimi písmenami (rgb, hsv) . Cieľom je teda získať pre dané hodnoty r, g, b odpovedajúce hodnoty h, s, v .

1. $v = \max(r, g, b)$
2. Ak $v = 0$ potom $s = 0$ a skonči (jedná sa o čiernu farbu $r = g = b = 0$)
3. Inak $\Delta = \max(r, g, b) - \min(r, g, b)$
4. $s = \Delta \div \max(r, g, b)$
5. Ak $r = \max(r, g, b)$ potom $h = (g - b) \div \Delta$
6. Inak ak $g = \max(r, g, b)$ potom $h = 2 + (b - r) \div \Delta$
7. Inak ak $b = \max(r, g, b)$ potom $h = 4 + (r - g) \div \Delta$

Ďalej nasleduje už len prípadná konverzia h, s, v hodnôt do daných rozsahov.

2.3 Optický tok

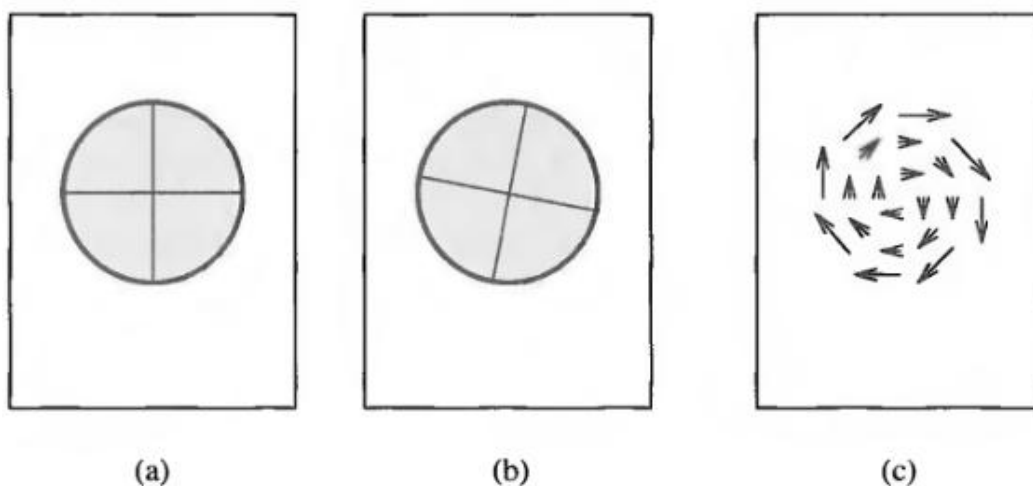
Optický tok určitým spôsobom reflektuje zmenu obrázku so zameraním sa na pohyb v priebehu daného časového intervalu. Jeho výstupom je smerové pole, ktoré určuje akým smerom sa daný objekt vo videu pohybuje. Do tohto procesu však zasahuje množstvo rušivých vplyvov, ktoré výsledok zkresľujú alebo v horšom prípade úplne znehodnocujú. Medzi hlavné patria zmeny osvetlenia, snimi súvisiace tieňe, potom súčasť pohybu kamery a snímaného objektu, atď. Cieľom je tieto nežiadúce vplyvy úplne eliminovať a získať smerové pole, ktoré sa čo najviac podobá skutočnému a teoretickému smerovému polu, pre daný obrázok.

Výpočet optického toku je preto možné s dobrou spoľahlivosťou aplikovať ak sú splnené tieto podmienky:

- Jas analyzovaného objektu je konštantný v priebehu času.
- Blízke body v obraze patria k rovnakému objektu—povrchu a pohybujú sa približne rovnakým smerom.

Výsledok algoritmu optického toku bude teda chybný, ak aspoň jedna z týchto podmienok bude porušená.

Ďalší problém nastáva v typickom prípade, kedy sa snímaná trojrozmerná scéna transformuje na dvojrozmerný obraz. V takejto situácii ostáva pohyb bodu v smere pozorovateľa skrytý alebo veľmi mierny. Vo výslednom vektorovom poli sa prejaví rozbiehajúcimi resp. zbiehajúcimi sa vektormi. Typicky sa optický tok dramaticky mení vo veľmi textúrovaných oblastiach, ktoré obsahujú množstvo pohybujúcich sa hrán.



Obrázok 2. Optický tok. (a) Čas t_1 (b) Čas t_2 (c) Optický tok—smerové pole

Optický tok teda udáva popis pohybu objektov v obraze a je preto veľmi dôležitou technikou, ktorá sa pri spracovaní obrazu využíva. Pohyb je možné zachytiť v podstate vo všetkých situáciách—pohybujúca sa kamera a statický objekt, statická kamera a pohybujúci sa objekt, alebo obe spolu.

2.3.1 Matematický popis optického toku

Cieľom rôznych metód na výpočet optického toku je detekovať pohyb medzi snímkami v časoch t a dt . Tieto metódy sa volajú diferenčné, pretože sú založené na lokálnej aproximácii obrazového signálu Taylorovým polynómom.

Majme jeden snímok vo videu. Potom $f(x, y, t)$ vyjadruje tento snímok, alebo jeho časť v čase t . Reprezentácia snímku ako funkcie miesta a času sa dá zapísať aj ako Taylorov rozvoj:

$$f(x + dx, y + dy, t + dt) = f(x, y, t) + f_x dx + f_y dy + f_t dt + O(\partial^2) \quad (1.0)$$

kde f_x, f_y, f_t značia parciálne derivácie f . Môžeme predpokladať, že za krátky časový okamih dt sa oblasť (x, y) posunie o malý úsek (dx, dy) . Chceme teda nájsť v dvoch rôznych snímkoch danej oblasti tak, aby platilo:

$$f(x + dx, y + dy, t + dt) = f(x, y, t) \quad (1.1)$$

Ak dx, dy, dt budú veľmi malé potom vyššie členy z rovnice (1.0) zanedbáme a:

$$-f_t = f_x \frac{dx}{dt} + f_y \frac{dy}{dt} \quad (1.2)$$

Cieľom je vypočítať rýchlosť

$$\mathbf{c} = \left(\frac{dx}{dt}, \frac{dy}{dt} \right) = (u, v) \quad (1.3)$$

f_x, f_y, f_t je možné vypočítať, alebo aspoň odhadnúť z $f(x, y, t)$. Rýchlosť pohybu môže byť potom vyjadrená ako

$$-f_t = f_x u + f_y v = \nabla f \mathbf{c} \quad (1.4)$$

Kde ∇f reprezentuje gradient dvojdimenzionálneho obrázku. Táto rovnica však obsahuje dve neznáme ∇f a \mathbf{c} a nedá sa priamo vyriešiť. Na výpočet je potrebné nájsť ďalšiu rovnicu, ktorá by bola definovaná nejakou špeciálnou podmienkou. Preto všetky metódy na výpočet vektorového poľa pomocou optického toku nejakú špecifickú podmienku predpokladajú. Napríklad to môže byť určité vymedzenie objektu, ktorý sa má sledovať a pod.

2.3.2 Detekcia pohybu pomocou feature points

Ako bolo v predošlom odseku spomenuté existuje na výpočet vektorového poľa pomocou optického toku niekoľko druhov metód. Väčšina z nich však funguje len ak je *snímková frekvencia*—*framerate*⁵ dostatočne vysoká. Detekcia pohybu založená na *záujmových bodoch*—*feature points* však funguje aj v prípadoch, keď je snímková frekvencia nízka.

Základom metódy výpočtu smerového poľa pomocou záujmových bodov je nájdenie navzájom si odpovedajúcich bodov v dvoch po sebe idúcich obrázkoch. Jednoduchšie povedané princípom tejto metódy je nájdenie bodu z predošlého obrázku v aktuálnom. Ak túto informáciu je možné získať odpovedajúce smerové pole je jednoduché zostrojiť.

Proces hľadania bodov je teda postupný—iteratívny a nie je potrebné mať informácie o budúcnosti. Takisto je možné predom určiť počet prepokladaných závislostí medzi jednotlivými bodmi, čo je vhodné hlavne vo veľkých obrázkoch. Každý dvojici nájdených bodov sa priradí pravdepodobnosť, ktorá vyjadruje mieru akou si nájdené body v dvojici obrázkov skutočne odpovedajú. Táto pravdepodobnosť sa následne prahuje aby sa získal žiadaný počet vektorov v smerovom poli.

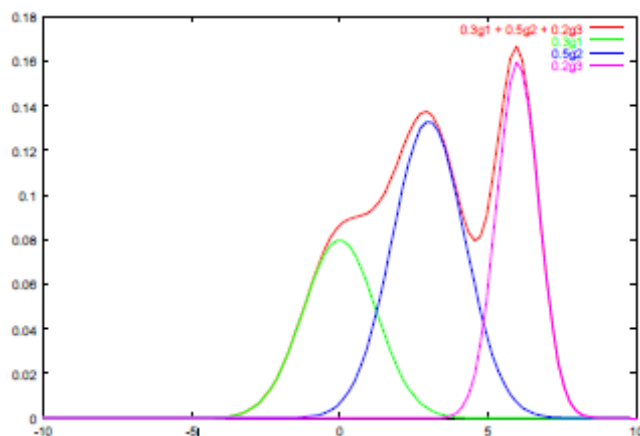
Prvým krokom tejto metódy teda je nájdenie výrazných bodov v dvojici obrázkou. Sú to body, ktoré sú napríklad podobné svojmu okoliu, body na hranách, okrajoch—inak povedané sú to body, ktoré nesú dôležitú obrazovú informáciu. Po tomto kroku nasleduje

⁵ Počet posebe idúcich obrázkov za jednotku času. Zvyčajne sa udáva v hertzoch alebo v fps—frames per second.

pokus o nájdenie väzieb a vzťahov medzi týmito bodmi v každom obrázku. Výsledkom celého procesu je hľadané smerové pole.

2.4 Gaussian mixture model

Gaussian mixture model, alebo v preklade Gaussovský zložený model, je pravdepodobnostný model, ktorý sa v štatistike používa na odhad funkcie hustoty pravdepodobnosti, určenej konečnou množinou experimentálne nameraných hodnôt. Skladá sa z kombinácie viacerých funkcií popisujúcich Gaussove—normálne rozloženie pravdepodobnosti. Tieto funkcie sa nazývajú aj komponenty zloženého modelu.



Obrázok 3. Gaussian mixture model (červenou) zložený z troch komponent.

2.4.1 Matematický popis zloženého modelu

Predpokladajme Gaussovu funkciu:

$$\phi(x; \theta) \quad (2.0)$$

kde $\theta = \mu\Sigma$ pričom μ označuje stred Gaussovej funkcie a Σ odpovedajúcu diagonálnu kovariančnú maticu. Potom Gaussov model zložený z k -komponent môžeme vyjadriť ako

$$f(x|\theta_{1..k}) = \sum_{j=1}^k \pi_j \phi(x; \theta_j) \quad (2.1)$$

Kde,

$$\sum_{j=1}^k \pi_j = 1 \text{ pričom } \pi_j \geq 0 : j \in 1, \dots, k \quad (2.3)$$

π_j označuje relatívnu váhu—dôležitosť danej komponenty a $\phi(x; \theta)$ je samotná komponenta. Váha π_j má charakter tzv. *a priori* pravdepodobnosti. To znamená, že jej hodnota je získaná experimentálne, v konkrétnom prípade zo vstupných hodnôt.

Na výpočet rovnice (2.1) je potrebné poznať neznáme π_j a θ , ktoré je možné aproximovať pomocou nižšie popísanej EM metódy.

2.5 EM (Expectation—Maximization)

Gaussov zložený model je teoretický model, na ktorého funkčné vyjadrenie je možné použiť rôzne metódy. EM je postupná optimalizačná metóda navrhnutá špeciálne pre pravdepodobnostné modely.

V matematickej štatistike a pravdepodobnosti je zvyčajne predom stanovená pravdepodobnosť, s akou daný jav môže nastať. EM sa však aplikuje na presne opačný prípad. Na vstup prijíma množinu už experimentálne nameraných hodnôt a na výstupe udáva pravdepodobnosť s akou daný jav v skúmanom modeli nastal. Výsledkom EM algoritmu sú teda koeficienty, ktoré popisujú práve danú pravdepodobnostnú funkciu.

Celý proces pozostáva z dvoch krokov, ktoré sa postupne striedajú. E-krok, často nazývaný aj *odhad* alebo *predpoklad* je definovaný ako

$$P(j, x_i) = \frac{\pi_j \phi(x_i; \theta_j)}{f(x_i)} \quad (3.0)$$

Odhad je následne maximalizovaný (M-krok) a tým sa získajú hľadané koeficienty.

$$\pi_j = \sum_{i=1}^n \frac{P(j|x_i)}{n} \quad (3.1)$$

$$\mu_j = \sum_{i=1}^n \frac{P(j|x_i)x_i}{n\pi_j} \quad (3.2)$$

$$\Sigma_j = \sum_{i=1}^n \frac{P(j|x_i)(x_i - \mu_j)(x_i - \mu_j)^T}{n\pi_j} \quad (3.3)$$

Tieto dva kroky sa postupne opakujú, až kým nieje dosiahnutá požadovaná presnosť konvergenencie.

2.6 Morfológické operácie

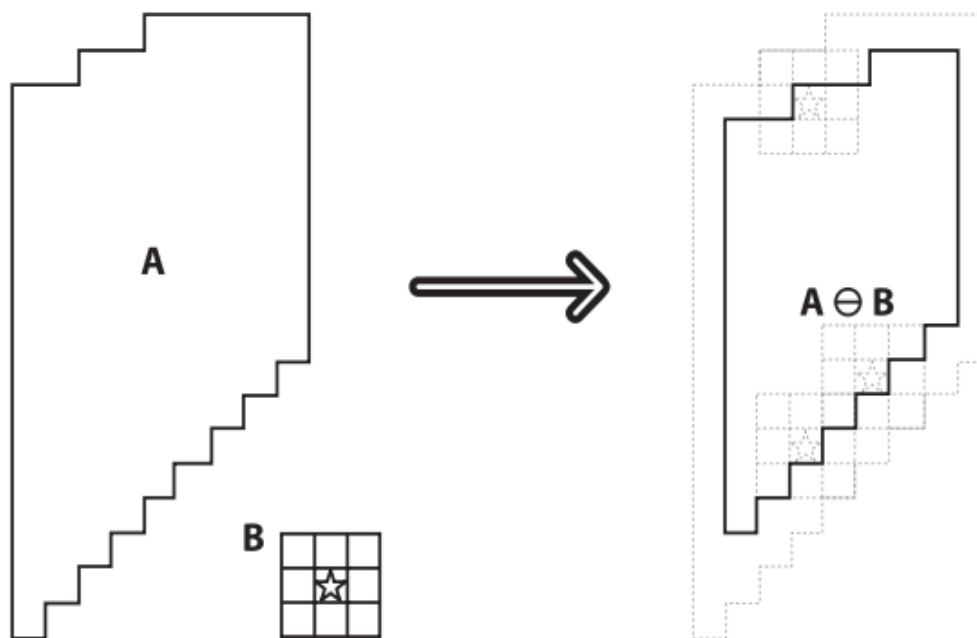
Morfológické operácie sa využívajú na spracovanie a analýzu geometrických štruktúr. V počítačovej technike sa najčastejšie uplatňujú v súvislosti s obrázkami, ale takžiet' ich je možné použiť na grafy, priestorové útvary alebo na povrchové siete.

Matematická morfológia vznikla okolo roku 1964. Rôzne techniky a funkcie boli preto navrhnuté hlavne na binárne obrázky⁶. Neskôr boli tieto metódy transformované aj na obrázky v stupňoch šedej.

Morfológické operácie sú odvodené zo základných pravidiel teórie množín. Rôzne postupy sa vo veľkej miere používajú na spracovanie binárnych obrázkov a bežné využitie zahŕňa detekciu hrán, potlačenie šumu, segmentáciu obrazu a iné.

⁶ Binárny obrázok—binary image je obrázok, ktorého body môžu nadobúdať len dve povolené hodnoty. Zvyčajne sa zobrazujú čiernobiele. Číselne je 0 pre čiernu farbu a 1 alebo 255 pre bielu. Binárny obrázok je napríklad výsledkom procesu prahovania.

Dve základné morfológické operácie sú erózia a dilatácia. Obe prijímajú na vstup dva parametre: obrázok, na ktorý sa má operácia aplikovať a jadro. Jadro je množina bodov definovaná relatívne od počiatku (napr. stredu). Zvyčajne sa jedná o mriežku tvaru obdĺžnika, často s rozmermi 3×3 . Od zdrojového obrázku je podstatne menšie a jeho počiatok nieje väčšinou v rohu, takže súradnice bodov majú aj záporné hodnoty.



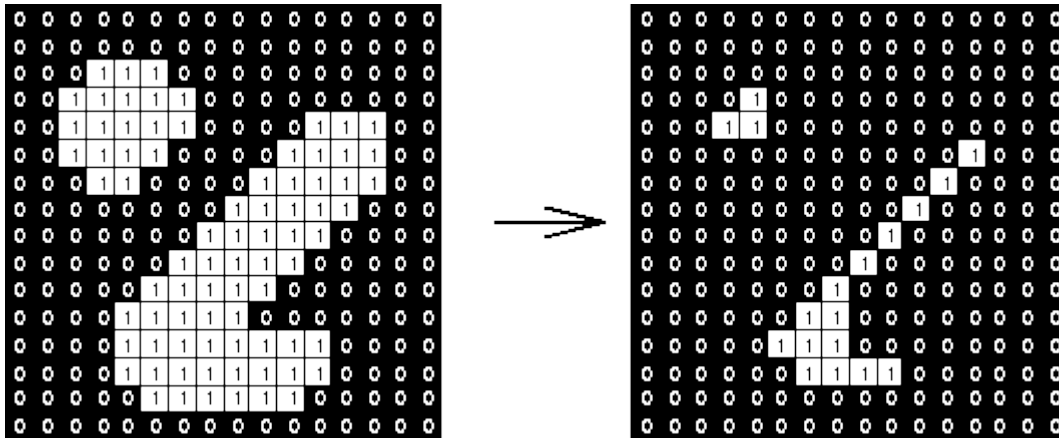
Obrázok 4. Príklad erózie objektu A. Použité jadro má štandardné rozmery 3×3 a počiatok v prostriedku.

2.6.1 Erózia

Hlavným účinkom erózie na binárnom obrázku je zmenšenie okraju okolo popredia obrázku, ktoré je zvyčajne definované bielou farbou. Oblasť popredia sa teda zmenší a diery vo vnútri tejto oblasti sa zväčšia. Erózia, tak ako bolo vyššie spomenuté, prijíma dva argumenty. Erodovaný obrázok a jadro, ktoré ovplyvňuje rozsah a tvar erózie.

Algoritmus výpočtu erózie pre vstupný binárny obrázok sa dá zhrnúť do týchto krokov:

1. Vezmi bod *popredia* (ďalej označovaný ako *vstupný bod*) a prelož ho z vrchu jadrom tak, aby sa počiatok jadra prekrýval so súradnicami *vstupného bodu*.
2. Ak pre každý bod jadra odpovedajúci bod v obrázku *popredím* potom *vstupný bod* ostáva nezmenený.
3. Inak, ak niektorí z bodov v obrázku odpovedajúcich jadru je bodom *pozadia*, potom aj *vstupný bod* sa zmení na *pozadie*.
4. Vezmi ďalší bod *popredia* a pokračuj na (1).



Obrázok 5. Výsledok erózie po aplikácii jadra s rozmermy 3×3 .

Výsledkom tejto operácie, ako je zremné z Obrázok 5, je zmena každého bodu popredia, ktorý nieje úplne obklopený⁷ bielymi pixelmi, na bod pozadia. Takýto bod musí ležať na hranici medzi popredím a pozadím a preto výsledkom je zmenšenie oblasti popredia.

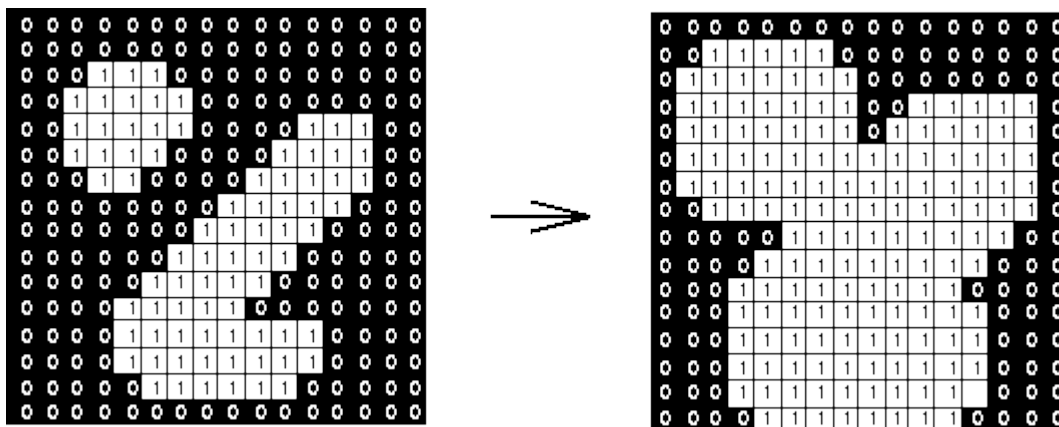
Erózia je duálna operácia k dilatácii—erózia bodov popredia je ekvivalentná k dilatácii bodov pozadia.

2.6.2 Dilatácia

Dilatácia je po erózii druhou základnou morfológickou operáciou. Jej vlastnosťou je zväčšenie okrajov popredia, ktoré býva typicky bielej farby. Tým pádom sa body popredia rozširujú a diery v tejto oblasti sa zmenšujú.

Algoritmus výpočtu *dilatácie* vstupný binárny obrázok sa dá zhrnúť do týchto krokov:

1. Vezmi bod *pozadia* (ďalej označovaný ako *vstupný bod*) a prelož ho z vrchu jadrom tak, aby sa počiatok jadra prekrýval so súradnicami *vstupného bodu*.
2. Ak pre každý bod jadra je odpovedajúci bod v obrázku *pozadím* potom *vstupný bod* ostáva nezmenený.
3. Inak, ak niektorí z bodov v obrázku odpovedajúcich jadru je bodom *popredie*, potom aj *vstupný bod* sa zmení na *popredie*.
4. Vezmi ďalší bod *pozadia* a pokračuj na (1).



Obrázok 6. Výsledok dilatácie po aplikácii jadra s rozmermy 3×3 .

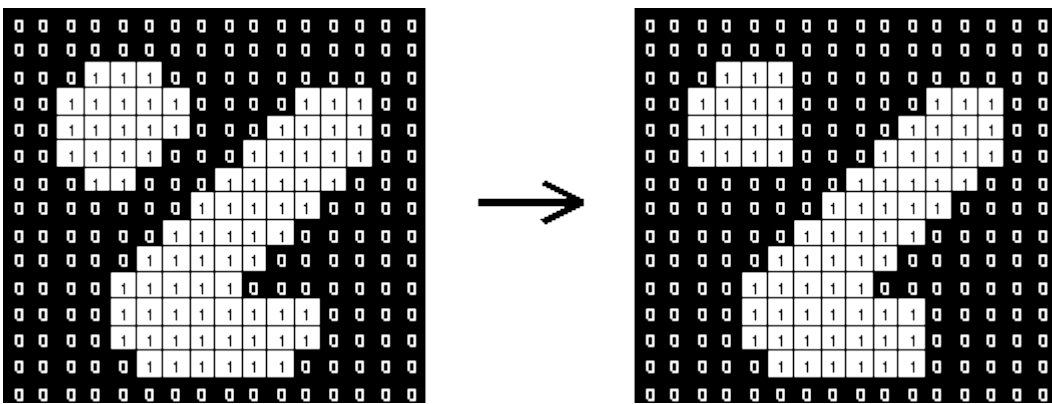
⁷ Úplne obklopeným sa rozumie vo všetkých ôsmich smeroch.

2.6.3 Otvorenie

Otvorenie je ďalšou morfológickou metódou, ktorá sa často využíva pri spracovaní obrazu. Jedná sa v podstate o kombináciu dvoch základných morfológických operácií erózie a dilatácie. Otvorenie sa tradične aplikuje na binárne obrázky, ale existujú jeho modifikácie aj na obrázky v stupňoch šedej.

Základnou vlastnosťou otvorenia je, podobne ako pri erózii, odstránenie bielych pixelov a teda zmenšenie oblasti popredia. Avšak otvorenie je menej deštruktívne ako erózia. Ako vstup prijíma binárny obrázok, na ktorý sa má otvorenie aplikovať a jadro, ktoré ovplyvňuje konečný efekt operácie.

Princíp je veľmi jednoduchý. Otvorenie sa definuje ako erózia bezprostredne nasledovaná operáciou dilatácie, pričom obe na vstup prijímajú *to isté jadro*. Výsledkom je odstránenie malých oblastí popredia—eróziou, a následné rozšírenie popredia dilatáciou.



Obrázok 7. Výsledok operácie otvorenia pri použití jadra s rozmermi 3×3 .

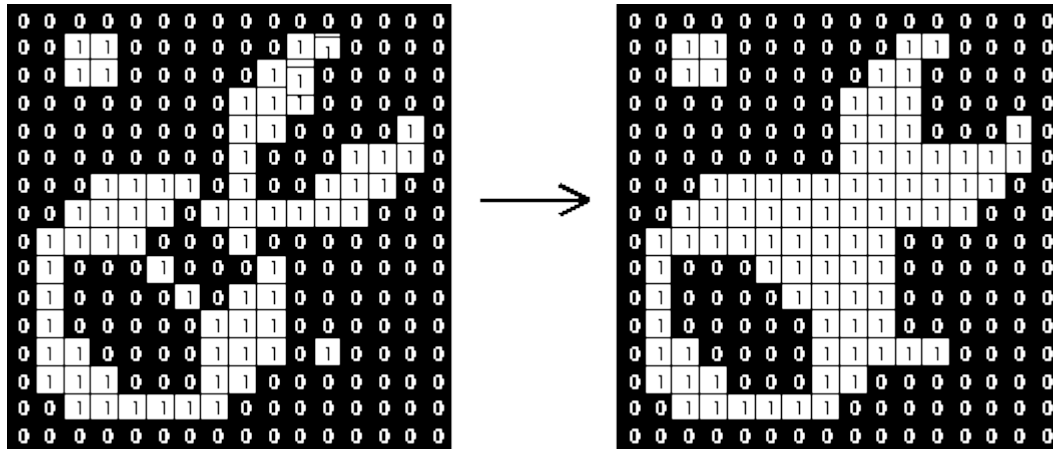
Z Obrázok 7 je možné vidieť, že ďalšia aplikácia operácie otvorenia by na výsledný obrázok už nemala žiadny efekt. Táto situácia nastala kvôli tomu, že všetky body popredia je možné pokryť jadrom tak, aby žiadny bod jadra nevyčnieval do oblasti pozadia. Preto zmenšenie popredia, ktoré spôsobí erózia, bude v plnej miere kompenzované následnou aplikáciou dilatácie.

2.6.4 Zavretie

Zavretie je *duálnou* operáciou k operácii otvorenia. To znamená, že aplikácia zavretia na body popredia má rovnaký efekt ako aplikácia otvorenia na body pozadia.

Definovaná je teda ako dilatácia bezprostredne nasledovaná operáciou erózie, pričom pre obe operácie je na vstupe použité *to isté jadro*.

Výsledkom zavretia je zmena oblastí pozadia, ktoré nie je možné úplne prekryť bodmi jadra, na oblasti popredia. Preto sa táto operácia často využíva na odstránenie dier v oblasti popredia. Všetko jasnejšie ilustruje Obrázok 8.



Obrázok 8. Aplikácia operácie zatvorenia na binárny obrázok pri použití jadra s rozmermi 3×3 .

3 Návrh

Ako aj názov práce naznačuje, riešenie úlohy je možné rozdeliť na dve podúlohy, ktoré sa budú riešiť samostatne a nakonci sa vhodným spôsobom prepoja. Jedná sa o *detekciu ohňa* a *detekciu dymu*.

V teoretickej časti som spomenul niektoré základné vlastnosti a metódy, ktoré sa používajú pri spracovaní obrazu. Tie teraz vhodným spôsobom využijem pri samotnom návrhu riešenia.

3.1 Detekcia ohňa

Vzhľadom na charakteristickú farbu plameňa, sa núka možnosť detekcie ohňa podľa jeho farby. Tu je možné s veľkou výhodou využiť spomínaný Gaussian mixture model. Ten určí pre každú farbu pravdepodobnosť, ktorá udáva ako veľmi odpovedá testovacia farba farbe ohňa. Dôležitú úlohu v tomto prípade preto zohráva obrázok, ktorý je použitý na tréning zloženého modelu. Týchto obrázkov môže byť samozrejme niekoľko, čím by sa mala zvýšiť presnosť detekcie.

Jednoduchou analýzou sa dá zistiť, že farba ohňa má vo veľkej miere vysoký jas. Farebný model, ktorý pri popise farby využíva práve jas je HSV model. Preto je vhodné previesť tréningové obrázky do HSV modelu a nastrénovať Gaussov zložený model vstupmi tohto formátu.

3.1.1 Detekcia ohňa pomocou GMM

Pre každý bod vstupného obrázku sa teda vypočíta pravdepodobnosť, akou mierou sa testovaný bod približuje teoretickej farbe ohňa a pomocou vopred zvoleného *prahu* sa určí výsledný záver. Ten môže byť buď kladný—testovacia farba je farbou ohňa, alebo záporný—testovacia farba nieje farbou ohňa.

3.1.2 Určenie prahu

Prah, pomocou ktorého sa rozhoduje, či testovacia farba odpovedá farbe ohňa alebo naopak je dobré určiť nie náhodne spôsobom pokus—omyl, ale čo najpresnejšie analýzou natrénovaného modelu. Vizualizáciu dát pre každú komponentu farebného modelu, dostaneme tri nezávislé zložené Gaussove funkcie, ktoré v každom bode určujú podobnosť danej farby s natrénovanou farbou ohňa. Je zrejmé, že čím väčšiu hodnotu v danom bode táto funkcia dosahuje, tým vyššia je pravdepodobnosť, že farba bude farbou ohňa. Preto sa za prah určí vhodná hranica, ktorá potom v praxi odpovedá realite. Vo finálnom riešení sa každá komponenta použitého farebného modelu (HSV) porovná s prahom a pokiaľ hodnota testovaného bodu patrí do intervalu prahu, testovaný bod je bodom plameňa.

Aplikáciou zloženého farebného modelu a spomenutej techniky prahovania sa získa oblasť s pravdepodobnou prítomnosťou ohňa. Táto oblasť však bude obsahovať množstvo *dier*—nedetekovaných oblastí, ako aj malé detekované miesta mimo samotnej oblasti ohňa. To je spôsobené tým, že charakteristickú farbu, ktorú má oheň, môžu mať aj iné objekty. Takéto poruchy vznikajú pri odraze svetla od skla, zrkadla, prípadne obloha a oblaky pri západe slnka často majú tento typ farby. Tieto objekty, aj napriek tomu, že nemajú s ohňom veľa spoločného sú po detekcii pomocou farebného modelu zahrnuté ako oheň v detekovanom obrázku. Tento efekt je dôležitý čo najviac eliminovať. Našťastie sú vo väčšine prípadov tieto

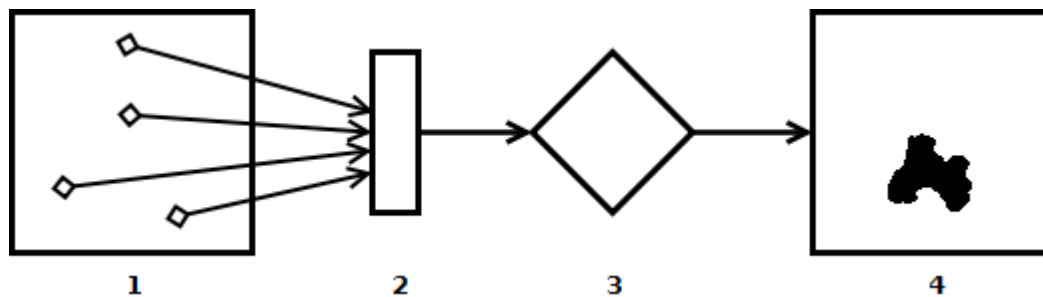
miesta podstatne menšie ako hlavná časť plameňa, preto je namieste použiť niektorú morfológickú metódu, prípadne ich kombináciu.

3.1.3 Výsledná úprava pomocou morfológických metód

Body, ktoré boli predošlou detekciou označené ako oheň budeme uvažovať ako body popredia. Všetky ostatné pixeli budú bodmi pozadia. Morfológická metóda, ktorá odstraňuje malé oblasti popredia a zároveň vyplní diery v oblasti popredia sa nazýva otvorenie. V závislosti od detekovaného poredia je možné použiť túto operáciu z rôznym typom jadra, avšak bežne by malo postačovať klasické jadro z rozmermi 3×3 s počiatkom v strede. Týmto spôsobom, by mala oblasť popredia označovať len miesta, v ktorých sa nachádza detekovaný oheň.

3.1.4 Finálna detekcia ohňa

Spojením týchto metód budú v analyzovanom obrázku označené oblasti, ktoré s vysokou pravdepodobnosťou obsahujú oheň. Celý postup ešte raz názorne ukazuje obrázok nižšie.



Obrázok 9. 1–vstupný obrázok 2–prah 3–morfológická metóda 4–výstupný obrázok

3.2 Detekcia dymu

Podobne ako pri detekcii ohňa, je pri návrhu spôsobu detekcie dymu potrebné určiť charakteristické vlastnosti, ktoré dym má. Za hlavnú sa dá označiť fakt, že dym stúpa smerom nahor. Druhou môže byť opäť jeho farba, aj keď tá už nieje taká jednoznačná ako pri ohni. To, že dym smeruje smerom nahor umožňuje použitie techniky optického toku. To však samo osebe nestačí, lebo v reálnom svete je veľké množstvo vecí, ktoré majú túto vlastnosť. Preto je potrebné využiť aj natrénovaný farebný model, tak ako pri ohni a ešte jeden zaujímavý fakt, že dym sa väčšinou nachádza v oblasti *nad plameňom* samotného ohňa.

3.2.1 Detekcia dymu pomocou GMM

Prvým krokom bude natrénovanie farebného modelu s charakteristickými obrázkami dymu. Situácia je však o niečo iná ako v prípade ohňa. Spektrum farieb, ktoré dym môže obsahovať je odosť rozsiahlejšie ako pri ohni. Farba dymu bude napríklad biela, pokiaľ sa v horiacom materiáli nachádza určité množstvo vody. Naopak môže byť aj sýto čierna ak horiaca látka obsahuje organické zlúčeniny alebo rôzne umelo vytvorené materiály. Dokonca v prípade ak farba dymu nieje dostatočne sýta, bude dym nadobúdať z časti aj farbu okolia—pozadia. Z toho vyplíva, že referenčný obrázok pre dym sa nedá zovšeobecniť, ale musí sa vybrať taký, ktorý odpovedá prostrediu, v ktorom sa systém na detekciu dymu nasadzuje. Tým sa podstatne zvýši presnosť detekcie.

3.2.2 Optický tok pri detekcii dymu

Výstupom optického toku pre aktuálne spracovávaný obrázok je smerové pole, ktoré určuje akým smerom a ako rýchlo sa jednotlivé body–objekty v obraze pohybujú. Rýchlosť pohybu určuje *veľkosť vektora* v smerovom poli. V riešenej situácii je vhodné, aby sa optický tok aplikoval len na oblasti, ktoré pravdepodobne obsahujú dym. Tieto oblasti určí predchodzia detekcia dymu pomocou farebného modelu.

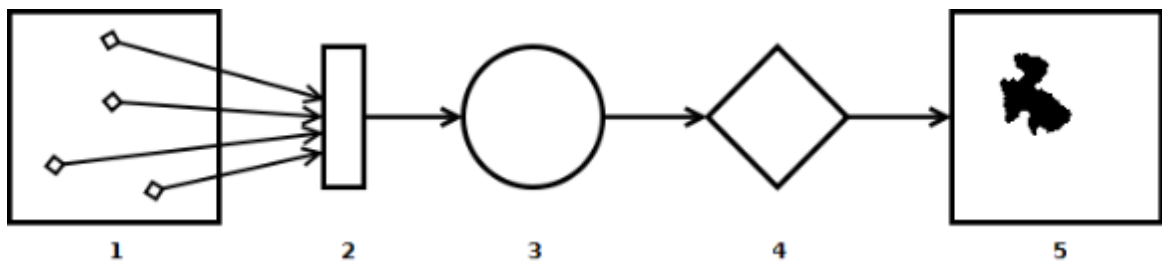
Potom ostáva len aplikovať fakt, že dym sa pohybuje smerom nahor, teda všetky doteraz detekované oblasti, ktoré nemajú jasný pohyb týmto smerom sa odstránia. To sa dá dosiahnuť podmienkou, ak *uhol vektoru v smerovom poli—uhol pohybu bodu v obrázku* nie je v intervale napríklad 30°– 150°, bod sa odstráni. Tým ostanú naozaj vyznačené len oblasti, ktoré spĺňajú dve základné podmienky a síce, že farba oblasti odpovedá farbe dymu a pohyb oblasti smeruje nahor.

3.2.3 Výsledná úprava pomocou morfológických metód

Rovnako ako pri detekcii ohňa, aj v tomto prípade nastávajú rôzne typy porúch v detekovaných oblastiach. Môžu to byť chybné detekované oblasti, ktoré neobsahujú dym, potom diery v oblastiach dymu, atď. Všetky tieto javy je s určitou úspešnosťou možné opraviť pomocou aplikácie morfológických operácií. V tomto prípade je postup úplne identický ako pri detekcii ohňa. Oblasti dymu sa považujú za body popredia a na vstupný obrázok sa aplikuje operácia otvorenia s jadrom 3 × 3. Tím sa odstránia malé oblasti mimo oblasť dymu a prípadné diery v oblasti dymu sa pridajú k oblasti poredia.

3.2.4 Finálna detekcia dymu

Celý postup detekcie dymu bližšie znázorňuje obrázok nižšie.



Obrázok 10. 1–vstupný obrázok 2–prah 3–optický tok 4–morfológická metóda 5–výstupný obrázok

3.3 Finálna realizácia—spojenie oboch metód

Po aplikácii vyššie spomenutých metód, sú v obraze detekované oblasti dymu a ohňa. Ostáva už len prepojenie týchto oblastí za predpokladu, že oblasť dymu sa nachádza nad oblasťou ohňa, a že môžu mať spoločný prienik.

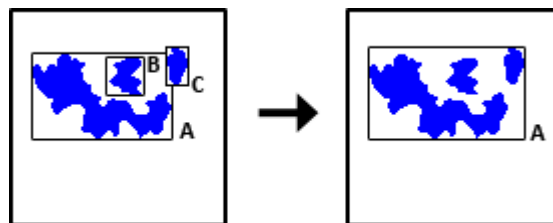
3.3.1 Segmentácia obrazu

Podľa doposiaľ detekovaných spojitých oblastí je možné rozdeliť obraz do segmentov—každá samostatná oblasť dymu alebo ohňa vytvorí jeden segment. Pre jednoduchosť bude mať každý segment tvar pravouhlého štvoruholníka, teda štvorca alebo obdĺžnika.

Predpokladajme teraz ľubovoľnú dvojicu segmentov s oblasťou dymu **A** a **B**. Môže nastať situácia kedy platí, že $A \subset B \Rightarrow A \cup B = A$. Inými slovami, že segment **A** úplne obsahuje segment **B**. Pokiaľ sa jedná o segment *rovnakého typu* (dymu), je možné predpokladať, že

tieto segmenty určujú rovnakú oblasť dymu ale z dôvodu šumu a podobných porúch v obraze boli vyhodnotené ako dve navzájom nezávislé oblasti.

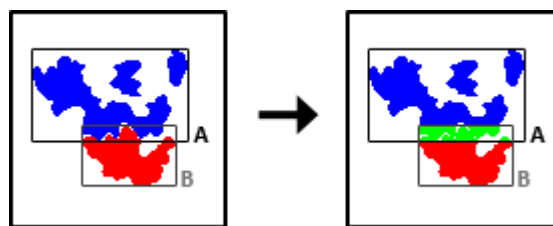
Teraz uvažujme situáciu, kedy ľubovoľné dve množiny **A** a **C** rovnakého typu majú spoločný prienik. Platí $A \cap C \neq \emptyset$. Opäť, ako v predošlom prípade je možné predpokladať, že segmenty určujú rovnakú oblasť dymu ale z dôvodu obrazových porúch boli vyhodnotené ako nezávislé oblasti. Z týchto dôvodov sa môžu segmenty, ktoré splňujú vyššie uvedené vlastnosti, zlúčiť a to štýlom ktorý popisuje obrázok nižšie.



Obrázok 11. Proces zlučovania segmentov

3.3.2 Prienik segmentov

Prienik segmentov sa počíta vždy medzi segmentom ohňa a segmentom dymu. Pokiaľ takýto prienik existuje, pravdepodobnosť, že sa jedná naozaj o plameň a dym je veľmi vysoká. Situáciu opäť ilustruje obrázok nižšie.

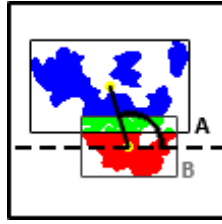


Obrázok 12. Prienik dvoch segmentov rôzneho typu.

3.3.3 Vzájomná poloha segmentov

Posledným krokom, ktorý je treba vykonať je určenie vzájomnej polohy segmentov, ktoré majú spoločný prienik. Jednoduchou a priamočiarou metódou, akou sa dá tento problém riešiť je nájdenie takého bodu v každom segmente, ktorý najlepšie charakterizuje jeho polohu. V riešenom prípade to je *aritmetický stred* segmentu. Potom sa úloha zjednoduší len na analýzu vzájomnej polohy dvoch bodov v rovine. Ak je teda stred z oblasti dymu nad⁸ bodom z oblasti ohňa, pravdepodobnosť, že sa jedná skutočne o horiaci plameň a s ním súvisiaci dym je veľmi vysoká.

⁸ Uhol, ktorý zvierajú priamka určená danými stredmi s vodorovnou osou je v intervale $0^\circ - 180^\circ$.



Obrázok 13. Poloha dvoch segmentov

Vzhľadom nato, že je to posledný krok analýzy obrazu, algoritmus vyhodnotí dané segmenty ako oblasti ohňa a dymu, čím sa analýza pre aktuálny obrázok videa ukončí a celý proces sa opakuje s nasledujúcim snímkom.

4 Implementácia

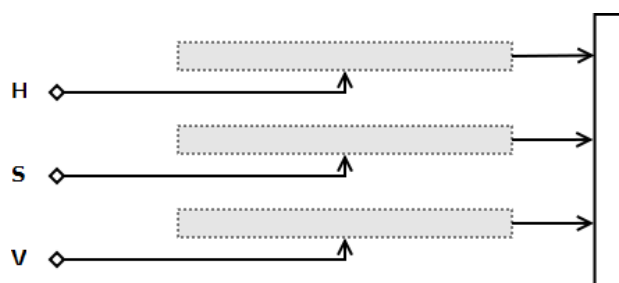
Na implementáciu výsledného algoritmu bol zvolený jazyk C++. Dôvodom je hlavne jeho prenositeľnosť medzi rôznymi platformami, ale taktiež množstvo knižníc, ktoré poskytujú základné prostriedky pre prácu s obrazom. Jedna z nich je známa a populárna knižnica OpenCv. Druhou je menejznáma, zato však veľmi kvalitná knižnica MDSTk.

4.1 Gaussian mixture model — implementácia

Prvým krokom pri realizácii výsledného algoritmu je implementácia funkcie na natrénovanie Gaussoveho zloženého modelu—Gaussian mixture model. Výstupom tejto funkcie bude pravdepodobnosť, akou sa testovací bod—testovacia farba približuje natrénovanej farbe ohňa. Táto myšlienka je bližšie popísaná v predchádzajúcich kapitolách.

Z trérovacieho obrázku, ktorý je prevedený do HSV farebného modelu, sa získa pre každú farebnú zložku pole, ktoré je následne použité ako vstup pre EM algoritmus, ktorý sa postará o samotné vyčíslenie hodnôt Gaussovej zloženej funkcie pre rôzne hodnoty. Dôležitým parametrom tohto algoritmu je počet jednotlivých Gaussových funkcií, ktorých kombináciou vznikne výsledný zložený model. Túto hodnotu som zvolil experimentálne na číslo 3, kedy Gaussov zložený model dosahoval presnosti postačujúcej pre daný účel.

Pre každý testovací bod analyzovaného obrázku potom natrénovaný Gaussov model vráti odpovedajúcu pravdepodobnosť. Tento proces však musí byť čo najrýchlejší vzhľadom na fakt, že analýza bude prebiehať v reálnom čase počas plynutia videa. Preto je vhodné si pre všetky možné farebné kombinácie HSV modelu vypočítať túto hodnotu vopred a uložiť do jednoduchej dátovej štruktúry. V imlementovanom algoritme som zvolil variantu, kedy pre každú zložku HSV modelu je vyčlenené pole, indexované hodnotu danej zložky farebného modelu a hodnota v tomto poli určuje pravdepodobnosť pre danú zložku. Vo výsledku sú potom tri polia—prvé pre zložku H, ktoré obsahuje 360 prvkov, a potom dve polia pre zložky S a V, pričom každé má po 100 prvkoch. Vyčíslenie pravdepodobnosti, pre každý analyzovaný bod sa teda zjednoduší len na výber hodnoty z poľa, pričom hodnota analyzovaného bodu je priamo index do tohto poľa. Takýmto spôsobom sa získajú tri pravdepodobnosti—pre každú zložku jedna, ktoré sú následne prahované spôsobom popísaným v predošlej kapitole o návrhu algoritmu. Celý tento postup bližšie ilustruje obrázok nižšie.



Obrázok 14. 1-zložka farby 2-pole pravdepodobností 3-prah

4.2 Optický tok — implementácia

Implementácia optického toku využíva vo veľkej miere knižnicu OpenCV, ktorá obsahuje potrebné funkcie—*cvGoodFeaturesToTrack*, *cvCalcOpticalFlowPyrLK*. Druhá menovaná funkcia okrem rôznych pomocných parametrov má dva dôležité. Tými sú aktuálny obrázok a obrázok predchádzajúci. V prvom obrázku sa definujú oblasti, ktoré sa majú sledovať a tie sa následne vyhľadajú v obrázku aktuálnom. Vypočtom veľkosti vektora, ktorý určuje smer posunu sa získa rýchlosť, akou sa objekt v obraze pohybuje. Je dôležité preto určiť minimálnu rýchlosť pohybu dymu, aby bolo možné vektory, ktoré sú príliš krátke odfiltrovať. Takéto vektory odpovedajú buď veľmi krátkym posunom, alebo sú to rôzne chyby, ktoré majú nežiadúci vplyv na výslednú analýzu. Minimálnu rýchlosť som určil experimentálne vzhľadom na daný prípad.

Ďalším spôsob, ktorým sa spresní analýza pomocou optického toku je fakt, že dym smeruje smerom nahor. Technika optického toku sa aplikuje vo výslednom algoritme len na detekciu dymu, preto je možné, všetky vektory smerujúce *nie nahor* takisto odfiltrovať. V praxi sa vypočíta uhol smeru vektoru. Ten sa testuje voči vopred zvolenému intervalu. Tento interval som opäť zvolil experimentálne na hodnotu (20°—160°).

4.3 Morfologické metódy - implementácia

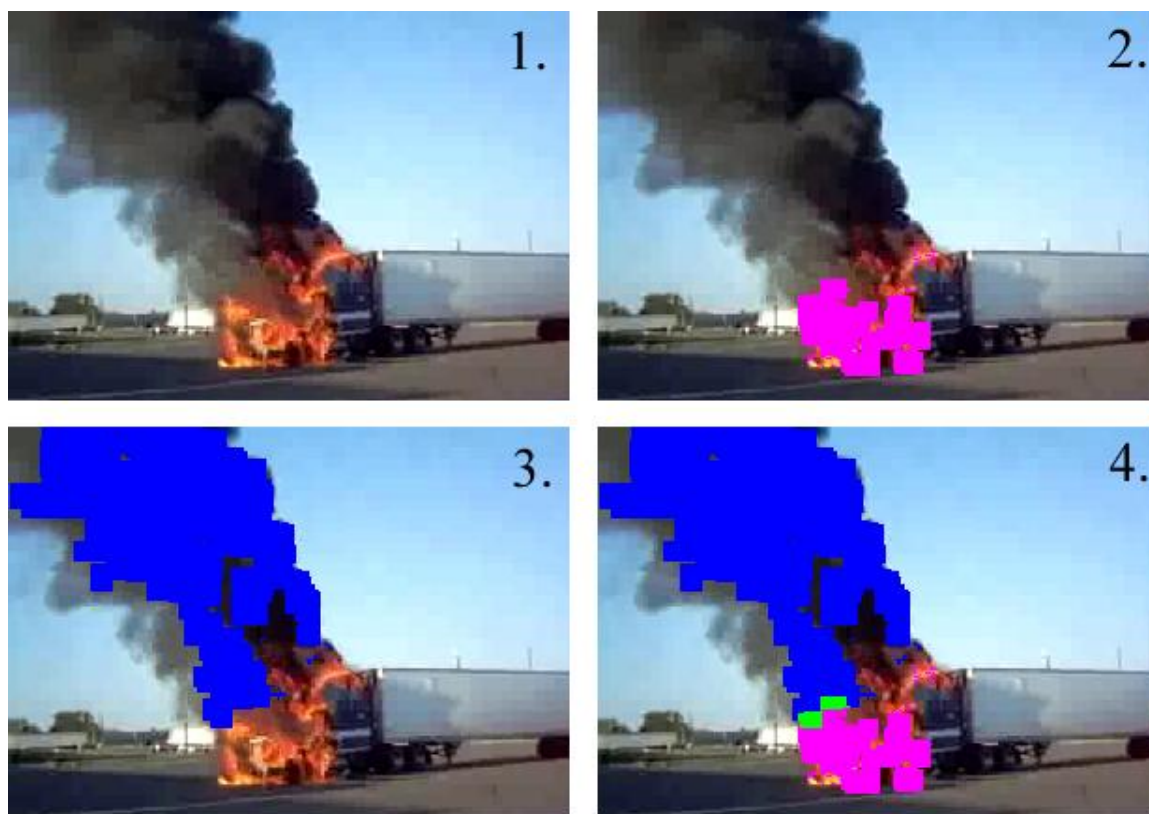
Implementáciu morfologických metód obsahuje knižnica OpenCV, preto sa na tento účel použijú konkrétne funkcie nato určené, ktorými sú *cvErode*, *cvDilate*.

5 Experimentálne výsledky

Implementáciou algoritmu proces tvorby systému na detekciu ohňa a dymu vo videu nekončí. Je ešte potrebné testovať rôzne situácie, ktoré môžu nastať, a snažiť sa nastaviť program tak aby, čo najlepšie plnil svoj účel. Možností je niekoľko—od výberu tréningových obrázkov, po experimentovanie s hodnotami použitými na prahovanie až po aplikáciu morfológických metód s rôznymi typmi jadier. Rozsah tejto práce však nedovoľuje sa venovať podrobne všetkým týmto oblastiam, preto sa zamerám hlavne na praktickú stránku veci, ktorou je výber tréningových obrázkov a ich vplyv na samotný výstup programu, ako aj ukážku chybné detekovaných objektov v testovacích videách.

5.1 Popis výstupu programu

Výstupom algoritmu je vizuálna interpretácia získaných hodnôt. Tieto hodnoty definujú oblasti, ktoré boli dôsledkom analýzy vyhodnotené ako oblasti ohňa a dymu. Celý tento proces bližšie popisujú obrázky nižšie.



Obrázok 15. 1–pôvodný obrázok, 2–detekcia ohňa(ružová), 3–detekcia dymu(modrá), 4–prienik oboch metód(zelená)

Prvý obrázok zobrazuje pôvodný snímok zo vstupného videa. V druhom je možné vidieť ružovou farbou zvýraznené oblasti, v ktorých bol detekovaný oheň. Hranatý okraj oblasti jasne ukazuje použitie morfológických metód popísaných v kapitole 2.6. Avšak nie celá oblasť bola vyhodnotená ako oblasť ohňa. Je to v plnej miere spôsobené tréningovým obrázkom, a zvoleným prahom, ktoré sú bližšie popísané v nasledujúcej podkapitole.

Na treťom obrázku je zvýraznená detekovaná oblasť dymu. Je dôležité poznamenať vysokú podobnosť medzi farbou dymu a farbou cesty—spodná časť obrázku. Ak by analýza

prebiehala len na základe Gaussovho zloženého modelu, bola by spodná časť obrázku takisto označená ako oblasť dymu. Preto tu dôležitú úlohu zohráva práve optický tok, ktorý vyčlení oblasť pohybujúcu sa smerom nahor a potom ešte aj analýza vzájomnej polohy oblasti ohňa a dymu. Vďaka týmto krokom je oblasť dymu určená správne.

Posledný obrázok 4. zobrazuje prienik detekovaných oblastí. Tento prienik však ani na takto jednoznačnej ukážke nieje veľký. Často ani nemusí existovať. Preto jeho prítomnosť slúži skôr ako dôkaz, že sa naozaj jedná o oheň a dym. Algoritmus ako finálnu hlavnú podmienku berie skôr vzájomnú polohu oblastí ohňa a dymu ako ich prienik.

5.2 Výber tréningových obrázkov a určenie prahu

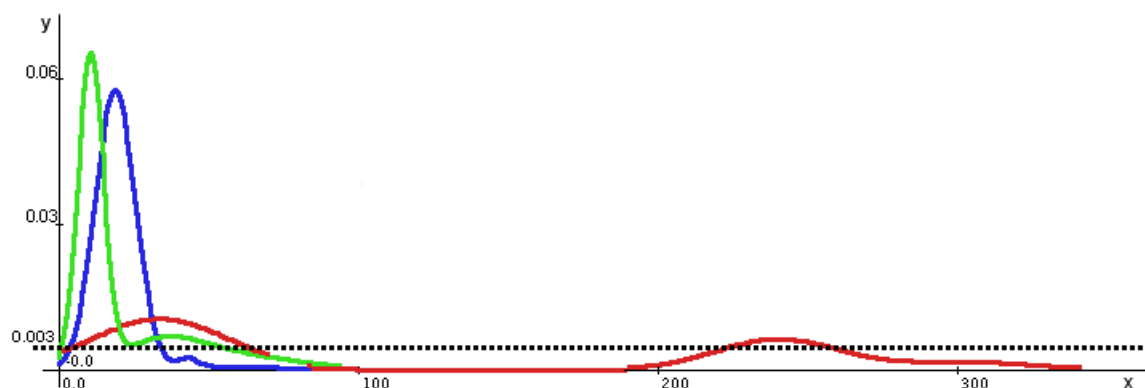
Ako som sa v prechádzajúcej kapitole zmienil, výber správnych obrázkov na natréningovanie zloženého modelu je kľúčová úloha hlavne pri detekcii ohňa. Ideálne by tento obrázok mal obsahovať pomer rôznych farieb tak, ako sa vyskytujú v horiacom plameni. Takisto by mal byť použitý obrázok, ktorý vznikol v analyzovanom prostredí, pretože je predpoklad, že prípadný oheň bude mať podobné vlastnosti, čo sa týka farby a jas.

Na natréningovanie zloženého modelu, ktorý bol použitý pri analýze snímku (15), som vybral obrázok priamo z analyzovaného videa. Takýto prípad v realite nenastane. Tréningový obrázok musí byť k dispozícii vopred, aby sa ním mohol natréningovať model, ktorý sa neskôr aplikuje na aktuálne snímané video. Avšak pre ukážku potenciálu použitej metódy som zvolil obrázok (16).



Obrázok 16. Tréningový obrázok.

Bližšou analýzou tohto snímku sa dajú určiť relatívne presne prahy, ktoré rozhodujú o príslušnosti testovacej farby k farbe ohňa alebo naopak. Prvým krokom takejto analýzy je vizualizácia zloženého modelu pre každú zložku farby H, S, V.



Obrázok 17. Gaussov zložený model pre každú zložku farby jeden. H-červená, S-modrá, V-zelená. Bodkovaná čiara určuje zvolený prah.

Zložky S a V majú rozsah hodnôt od 0–100, preto v bode 100 na x osi ich grafy končia. Prah, ktorý som zvolil experimentálne tak, aby výstup algoritmu dával čo najlepšie výsledky,

určuje minimálnu pravdepodobnosť akú musí daná zložka (x os) farby mať, aby bola vyhodnotená kladne. To platí pre všetky zložky a je medzi nimi vzťah AND. Znamená to teda, že pravdepodobnosť každej zložky testovacej farby musí byť vyššia ako prah. V ukázkovom prípade som zvolil prah 0,003 pre každú zložku rovnaký. Jedná sa však o špeciálny prípad, lebo prah môže byť, a často aj je, pre každú zložku iný. Môže nastať aj situácia, keď sa nedá prah určiť jedným číslom a je nutné definovať intervaly, do ktorých testovacia farba musí patriť.

5.3 Príklady chýb programu

Takto navrhnutý algoritmus však nie je bezchybný. Môžu nastať situácie, kedy určité objekty majú také vlastnosti, že ich program vyhodnotí chybné. To však môže platiť aj priamo pre oblasť ohňa alebo dymu, ktoré na výstupe niesú označené vôbec. Rozsah takýchto chýb určuje použitá analyzáčná technika. Práve princíp detekcie pomocou natrénovaného farebného modelu, môže byť často dosť nepresný. Dôvodom je množstvo rôznych objektov, ktoré môžu mať farbu podobnú plameňu. Oheň má vysokú farebnú intenzitu, tú majú aj lesklé objekty, ktoré dobre odrážajú svetlo. Môže to byť sklo, voda, oblaky prípadne samotné slnko. Pre bližšiu ilustráciu uvažujme nasledujúcu dvojicu obrázkov.



Obrázok 18. Ukážka chýb program.

Prvý obrázok je originál. Na druhom sú vyznačené oblasti z prítomnosťou ohňa, avšak niektoré sú chybné (oblaky vľavo hore). Dôvod je, ako som už vyššie spomínal, nasledujúci. Oblaky odrážajú svetlo, ktoré smeruje priamo k pozorovateľovi, preto má vysokú intenzitu. Navyše má bielu farbu, ktorá vo farbe plameňa prevláda. Preto natrénovaný model dáva pre takéto objekty vysoké hodnoty pravdepodobností a sú teda vyhodnotené chybné.

Detekcia dymu je v uvedenom prípade ešte zložitejšia. Z ohňa totiž takmer žiadny dym neuniká. Takáto situácia tradične vzniká pri dokonalom spaľovaní, čo však v reálnych prípadoch neplánovaného vziku požiaru nenastane. V každom prípade v takejto situácii nie je možné dym detekovať vôbec. Na natrénovanie farebného modelu je jeho farba príliš nejednoznačná a vzhľadom nato, že je veľmi slabý, sa rýchlo rozplynie a technikou optického toku ho nie je možné zachytiť. Preto v tejto ukážke bola detekcia dymu v algoritme vypnutá.

Uvedené príklady teda ukazujú situácie, kedy takto navrhnutý detektor zlyhá. Tieto informácie sú mimoriadne dôležité, lebo je možné podľa nich určiť, či je vhodné systém v danom prostredí nasadiť alebo nie. Taktiež je vidieť, že sa nastavenie algoritmu sa musí

prispôbiť analyzovanému prostrediu. To hlavne znamená, že tréningový obrázok a určenie prahov sa nedá zovšeobecniť, a musí sa pre každé prostredie vyhodnocovať zvlášť.

6 Záver

Na detekciu ohňa a dymu vo videu som použil techniku natrénovaného farebného modelu GMM a optického toku v kombinácii so základnými morfológickými metódami. Prítomnosť ohňa a dymu sa pomocou implementovaného algoritmu podarilo detekovať aj v rôznych špeciálnych situáciách. V každom prípade však celá metóda stojí na správne zvolených tréningových obrázkoch pre zložený model. Takýto obrázok je veľmi náročné nejakým spôsobom zovšeobecniť a takéto snahy, ako som sa experimentálne presvedčil, vedú k zníženiu kvality detekcie.

Rôzny materiál horí rôznym spôsobom. Niektorý produkuje hustý čierny dym, iný zas takmer žiadny. Osvetlenie objektu tiež prispieva k výslednej farbe ohňa a dymu. Preto pri voľbe tréningových vzoriek je dôležité brať tieto skutočnosti v úvahu a zanalyzovať prostredie, v ktorom bude systém nasadený. V praxi to môže, ale aj nemusí byť veľký problém. Servisné kamery v budovách a podobných objektoch, často snímajú len určitú časť priestoru, kde sa prípadný horlavý materiál dá vopred určiť a otestovať. Podobne osvetlenie môže byť stále a rovnomerné počas celého dňa, čo vedie k ideálnemu prípadu nasadenia systému.

Iná situácia nastáva vo verejných priestranstvách, kde sa vyskytuje veľké množstvo rôznych objektov často aj premenlivého charakteru—idúce auto. Svetelné podmienky sa tiež počas dňa pochopiteľne dramaticky menia, preto je nasadenie systému do takéhoto prostredia komplikovanejšie. V noci je prakticky nemožné detekovať dym zato sa tento nedostatok kompenzuje s veľmi veľkou presnosťou detekcie ohňa. Preto aj v takýchto situáciách dáva algoritmus postačujúce výsledky.

Ako bolo v úvode spomenuté celá technika predpokladá vstup snímaný statickou kamerou. Ak by táto podmienka splnená nebola, výstup optického toku by dával chybné výsledky a detekcia by prebiehala iba pomocou farebných modelov, čo by viedlo k zníženiu kvality detekcie, ale nie však k jej úplnému znehodnoteniu. To dokazuje, ako je možné vidieť na testovacích prípadoch, vysokú stabilitu použitej techniky.

6.1 Rozšírenia

Vo vývoji detekcie založeného na popísanom systéme sa dajú použiť ďalšie rozšírené techniky, ktoré by viedli, k čo najväčšej kvalite výstupu algoritmu. Jedným rozšírením by bolo rozšírenie implementácie aj o vstup, ktorý nie je snímaný len statickou kamerou. To sa týka hlavne implementácie optického toku a detekcie pohybu, ktorá by bola schopná sa s takou situáciou vyrovnáť. Problémom však je, že detekcia pohybu dymu týmto spôsobom vyžaduje veľmi veľkú citlivosť a presnosť, ktorá by sa takýmto spôsobom nedala možno dosiahnuť.

Dôležitou hodnotou, ktorá má veľký vplyv na celkový výstup je *prah*, podľa ktorého sa rozhoduje, či testovacia farba je farbou ohňa alebo nie. V implementácii som tento prah určil z časti experimentálne a z časti analýzou grafu⁹. Najvhodnejšie a najpresnejšie by sa však táto hodnota dala zistiť analyticky, pomocou rôznych matematických metód. Určenie presnej hodnoty prahu (pre každú zložku farby) by výrazne zvýšilo presnosť detekcie.

⁹ Bližšie v kapitole 3.1.2

7 Literatúra

Bradski, G., & Kaehler, A. (2008). *Learning OpenCV*. O'Reilly Media.

Dellaert, F. (2002, Február). *The Expectation Maximization Algorithm*. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.9.9735&rep=rep1&type=pdf>

Moore, A. W. (2004). *Clustering with Gaussian Mixtures*. Retrieved from <http://www.cs.cmu.edu/afs/cs/Web/People/awm/tutorials/gmm14.pdf>

Raut, C. K. (2004, 8 7). *Gaussian Mixture Model*. Retrieved from <http://mi.eng.cam.ac.uk/~ckr21/slides/gmm.pdf>

Sonka, M., Hlavac, V., & Boyle, R. (2008). *Image Processing, Analysis, and Machine Vision* (3rd Edition ed.). Thomson.