

# 1. Detekce světelných stop v obraze

Pro analýzu vlastností broušeného kamene je důležité detekovat světelné stopy vzniklé dopadem laserových svazků na stínítko. Zároveň je třeba určit parametry stop, které se budou porovnávat s parametry svazků matematického modelu kamene.

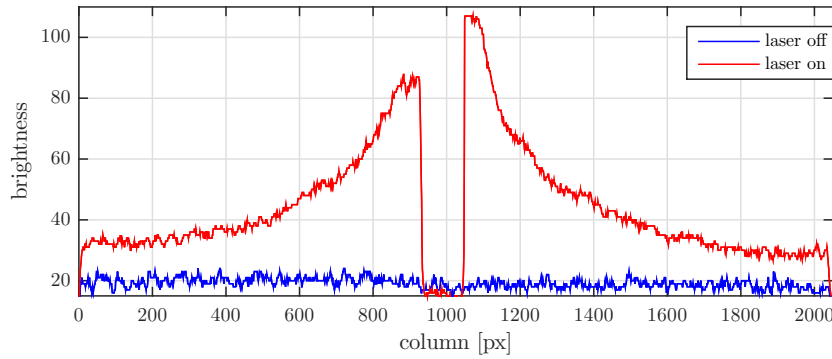
Intenzitu pixelu  $I$  můžeme vyjádřit jako

$$I = I_{beam} + I_{back} + I_{edge} + I_{\varepsilon}, \quad (1.1)$$

kde  $I_{beam}$  reprezentuje příspěvek světelného svazku,  $I_{back}$  intenzitu pozadí,  $I_{edge}$  intenzitu světelných ocásků a  $I_{\varepsilon}$  Poissonův šum (kapitola ??). Jedním z úkolů detektoru je oddělit pozadí od zbývajících složek intenzit. Po odečtení pozadí získáme obrazy laserových svazků.

Jednoduchým postupem pro určení intenzity pozadí by bylo prahování obrazu nad konstantní úrovní. V našem obraze však typicky konstantní není (kapitola ??).

Rozdílné pozadí se může také vytvořit odrazem zdrojového svazku od jiných předmětů, než broušeného kamene. Hlavním příspěvkem je v tomto případě odraz od podstavce, na který pokládáme broušený kámen (obr. 1.1).



Obrázek 1.1: Jasové úrovně ve vybraném řádku obrazu. Řádek protíná obraz podstavce. V případě červené charakteristiky dopadá na podstavec laserový svazek, rozptyluje se dopadá na stínítko. Modrou charakteristiku pozorujeme, pokud je laser vypnutý.

Hodnotu pozadí potřebujeme znát, abychom ze snímku mohli spočítat světelný tok pro jednotlivé dopadající laserové svazky. Určení intensity pozadí v každém pixelu komplikují obraz podstavce na kámen a okolí stínítka. Zde je intenzita světla podstatně nižší, než na povrchu stínítka. Vysoká změna jasu v obraze komplikuje určení pozadí.

V okolí stínítka můžeme detekovat falešné svazky, které je třeba odstranit.

Světelné stopy se mohou překrývat. Pro odlišení příspěvků jednotlivých svazků je třeba obraz prahovat v několika úrovních jasu.

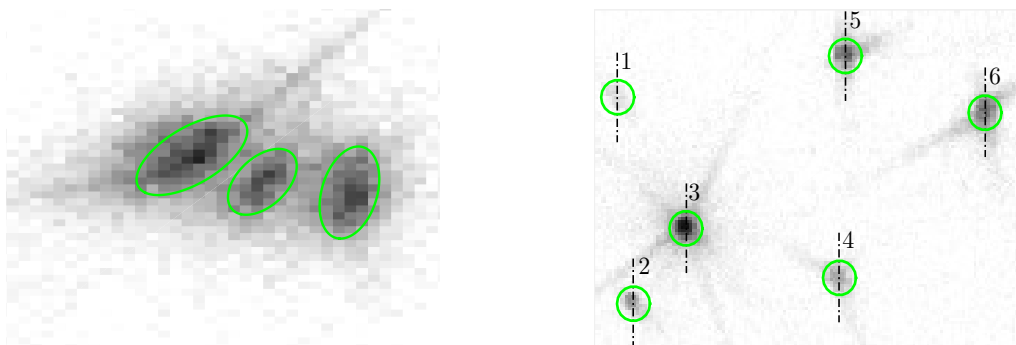
V místě, kde je vysoká koncentrace svazků mohou dokonce svazky dopadnout tak blízko sebe, že splynou v jednu stopu (obr. 1.2).

Obraz je třeba filtrovat. Filtrováním snížíme šum v odraze, ale zároveň zmenšíme kontrast mezi stopami.

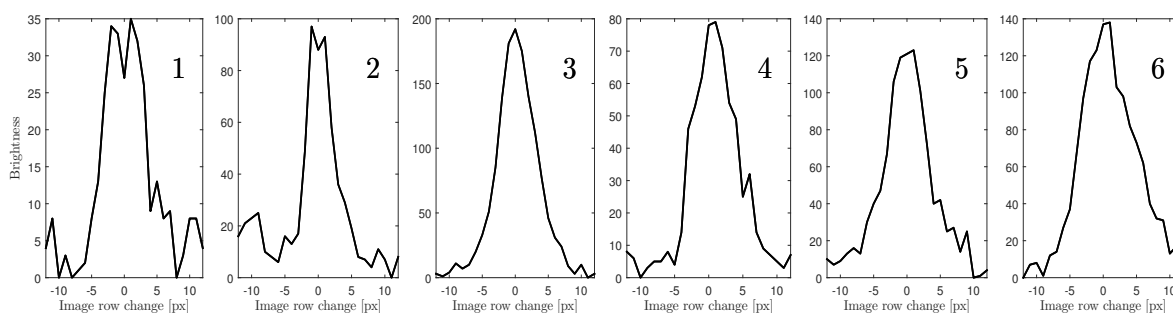
Ne všechny svazky vystupující z kamene je možné detekovat. Svazky s vícenásobným odrazem postupně ztrácí zářivý tok. Po dopadu na stínítko mohou být nerozlišitelné od šumu a jejich detekce je prakticky nemožná. Pro stopy s nízkým jasnem bude detekce často selhávat



Obrázek 1.2: Ilustrace slynutí dvou různých svazků. V pravém i levém snímku se nachází typově stejné laserové svazky. Na levém obrázku dopadly na stínítko příliš blízko sebe. V tomto případě nejsme schopni rozlišit příspěvek obou svazků a detekujeme pouze jednu stopu.



Obrázek 1.3: Problémové detekce. Nalevo jsou laserové stopy blízko u sebe. Stopy je nutné od sebe oddělit. Na levém snímku jsou znázorněny výrazné rozdíly mezi velikostí a intenzitou stop. Je nutné použít víceúrovňový detektor.



Obrázek 1.4: Jasové řezy ve totožném sloupci obrazu. Řez protíná pixel s maximální hodnotou jasu ve stopě. Číslování řezů odpovídá indexům stop na obr. 1.3.

## 1.1 Předchozí práce

V předchozí práci [5] jsme neměli možnost detekovat stopy s nízkým jasnem. Překrývající se svazky bylo nemožné oddělit.

Bohatší pojetí problému se objevilo v Bodlákově práci [3]. Snímek se prahoval více než jedním prahem, přičemž z oblastí nad prahem se sestavila stromová struktura a světelné stop se určily jako listy stromu s dostatečnou významností. Tento přístup je však pro svou výpočetní náročnost nepoužitelný pro snímky s rozlišením  $2050 \times 2050$ , které máme k dispozici.

Naše úloha detekce je velmi podobná detekci hvězd a galaxií v astronomických snímcích. V oblasti astronomie se hojně používá program s názvem Source Extractor [2]. Tento program má za sebou dlouholetý vývoj, je optimalizován z hlediska rychlosti a odzkoušený širokou veřejností. Tento software lze po naladění parametrů použít i pro náš případ. Nevýhodou však je, že nelze spustit v operačním systému Windows, který využíváme.

Po testu různých detektorů jsme se rozhodli pro detekci laserových stop v obraze využít relativně nový přístup uveden J. Matasem et al. [6] v roce 2002 - MSER detektor.

## 1.2 MSER (maximal stable extremal region) detektor

MSER detektor hledá v obraze maximálně stabilní extrémní oblasti. Původně byl využit pro robustní nalezení korespondencí mezi dvěma snímky stejného objektu pořízených z různého místa a v současné době se používá v mnoha oblastech počítačového vidění.

Princip spočívá v několikaúrovňovém prahování obrazu podle intenzity a nalezení spojitých oblastí, které jsou nad či pod prahovou hodnotou. Mezi úrovněmi jsou nalezeny korespondující oblasti a za MSER oblasti jsou označeny ty, jejichž velikost z předchozí úrovně se se zvyšující úrovní příliš nezměnila.

Výhodou MSER detektoru je invariance vůči afinní transformaci intenzity a vůči změně měřítka, což umožňuje současnou detekci malých a velkých oblastí s různou intenzitou. Podle studie [7], která porovnává MSER detektor s ostatními typy detektorů významných oblastí, dosáhl MSER detektor skvělých výsledků v detekci oblastí s vysokou hustotou a variabilní změnou velikosti. MSER detektor se tedy zdá být vhodným kandidátem pro detekci laserových stop v obraze.

## 1.3 Implementace

### 1.3.1 Filtrace

Nejprve se pokusíme minimalizovat Poissonův šum v obraze. Šum redukuje konvolucí s maskou, která se skládá z prvků odpovídajících Gaussově funkci. Parametry filtru: velikost masky - 3 px, směrodatná odchylka  $\sigma = 0.7$ .

### 1.3.2 Detekce

Dalším krokem je detekce MSER oblastí ve filtrovaném snímku. MSER detektor je již implementován v prostředí MATLAB ve funkci *detectMSERFeatures*. Pro aplikaci této funkce na snímek se světelnými stopami je třeba nastavit základní parametry detektoru. Mezi ně patří frekvence prahování snímku, maximální a minimální velikost MSER oblastí a dostatečná stabilita oblastí.

- **Frekvence prahování snímku.** - Určuje velikost kroku mezi prahovacími úrovněmi jasu (obr. 1.7). Prahování se používá pro nalezení extrémálních oblastí, na kterých se testuje stabilita.

- **Maximální a minimální velikost MSER oblasti.**
- **Dostatečná stabilita oblasti.** - Velikost stabilní oblasti se po změně úrovně prahu intenzity příliš nemění.

### 1.3.3 Okolí stínítka

Okraj obrazu stínítka má tvar kružnice. Kružnici popisuje funkce

$$(x - x_0)^2 + (y - y_0)^2 = r^2, \quad (1.2)$$

kde  $[x, y]$  je bod na kružnici,  $[x_0, y_0]$  střed kružnice a  $r$  poloměr. Je zřejmé, že k určení parametrů kružnice potřebujeme nalézt minimálně 3 body ležící na kružnici.

Úpravou rovnice 1.2 dostaneme

$$x^2 - 2x x_0 + x_0^2 + y^2 - 2y y_0 + y_0^2 = r^2, \quad (1.3)$$

$$\begin{bmatrix} -2x & -2y & 1 \end{bmatrix} \cdot \begin{bmatrix} x_0 & y_0 & -r^2 + x_0^2 + y_0^2 \end{bmatrix}^T = - (x^2 + y^2). \quad (1.4)$$

Polohu v obraze určujeme podle počtu pixelů v horizontálním (sloupec) a vertikálním (řádek) směru.

Body na kružnici nalezneme pomocí sečen. Sečny sestojíme ve dvou řádcích snímku a nalezneme sloupce, ve kterých protínáme kružnici.

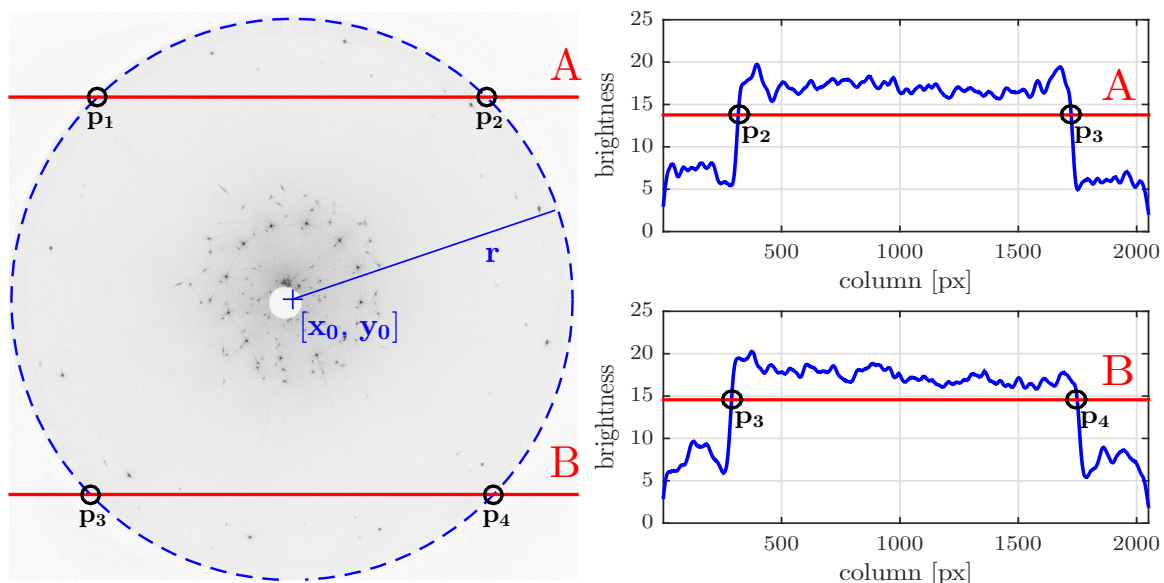
Sestrojením sečny získáme jasový řez v celé šířce snímku. Fotonový šum jasu vyfiltrujeme konvolucí s Gaussovým filtrem. Velikost filtru volíme 21 px a směrodatnou odchylku  $\sigma = 20$ .

Vyfiltrovaný jas oddělíme prahem. Práh určuje střední hodnota jasu v daném řezu. Nalezneme sloupce, kde je jas vyšší než prahové hodnoty. Sloupec s minimálním resp. maximálním počtem pixelů určuje bod na kružnici.

Každá sečna protíná kružnici ve dvou bodech, proto dostaneme celkem čtyři body na kružnici. Parametry kružnice určíme metodou nejmenších čtverců.

Okolí stínítka poté definuje funkce

$$(x - x_0)^2 + (y - y_0)^2 > r^2. \quad (1.5)$$



Obrázek 1.5: Jasové řezy A a B. Detekujeme body na kružnici  $p_1, p_2, p_3$  a  $p_4$ . Metodou nejmenších čtverců odhadneme parametry kružnice  $x_0, y_0$  a  $R$ .

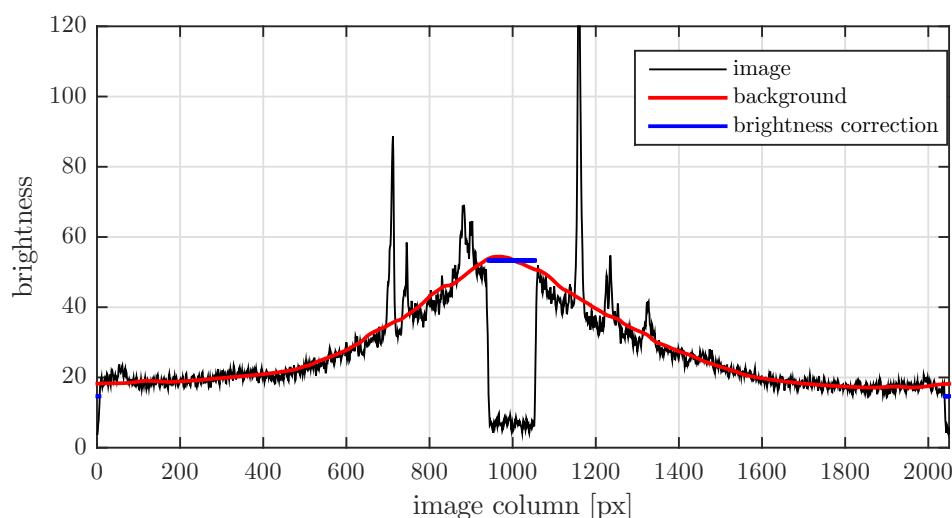
### 1.3.4 Pozadí snímku

V obraze nalezneme podstavec a okolí stínítka. Podstavec je specifický nízkou střední hodnotou jasu a jeho obraz je téměř ideální kruh. V seznamu MSER oblastí proto podstavec snadno nalezneme. Okolí stínítka již známe (kap. 1.3.3).

Velikost jasu v okolí stínítka nastavíme na hodnotu odvíjející se od střední hodnoty jasu snímku. Jas pixelů v oblasti podstavce nastavíme na střední hodnotu jasu pixelů v blízkém okolí podstavce.

Pozadí následně určíme konvolucí s Gaussovým filtrem. Tento filtr ignoruje vysoké změny jasu v obraze. Parametry filtru: velikost masky - 201 px, směrodatná odchylka  $\sigma = 201$ .

Samotná konvoluce s tímto filtrem by s použitím standardní funkce *conv2* byla příliš časově náročná, proto konvoluci provádíme efektivnějším způsobem, který využívá rozkladu masky filtru na singulární čísla.



Obrázek 1.6: Filtrace pozadí v HDR snímku znázorněná v řádku obrazu protínajícím obraz podstavce na kámen.

### 1.3.5 Odstranění nežádoucích detekcí

Výstupem detektoru je soubor MSER oblastí. U výrazné světelné stopy dostaneme data ve formě pyramidy MSER oblastí podle jednotlivých úrovní intenzity.

MSER detektor však najde nejen oblasti s výrazně vyšší intenzitou, ale i oblasti s nižší intenzitou než okolí. Ty je třeba vyřadit, protože nerepresentují světelnou stopu, kterou hledáme.

K odstranění nežádoucích detekcí použijeme následující postup.

1. Od filtrovaného snímku odečteme pozadí.
2. Ve vzniklém snímku vypočítáme střední hodnotu jasu MSER oblastí.
3. Pokud je střední hodnota jasu záporná, MSER oblast odstraníme.

## 1.4 Určení parametrů svazku

Základním parametrem svazku je směr šíření popsáný azimutem a elevací. Směr šíření svazku snadno dopočítáme, pokud nalezneme jeho obraz. Pozici světelné stopy v obraze lze určit jako polohu pixelu s maximálním jasnem v detekované oblasti. Šum v obraze ale situaci komplikuje.

Z obr. 1.4 ale vidíme, že pixel s maximálním jasem nemusí vždy určovat pozici dopadu a navíc nemusí být unikátním maximem.

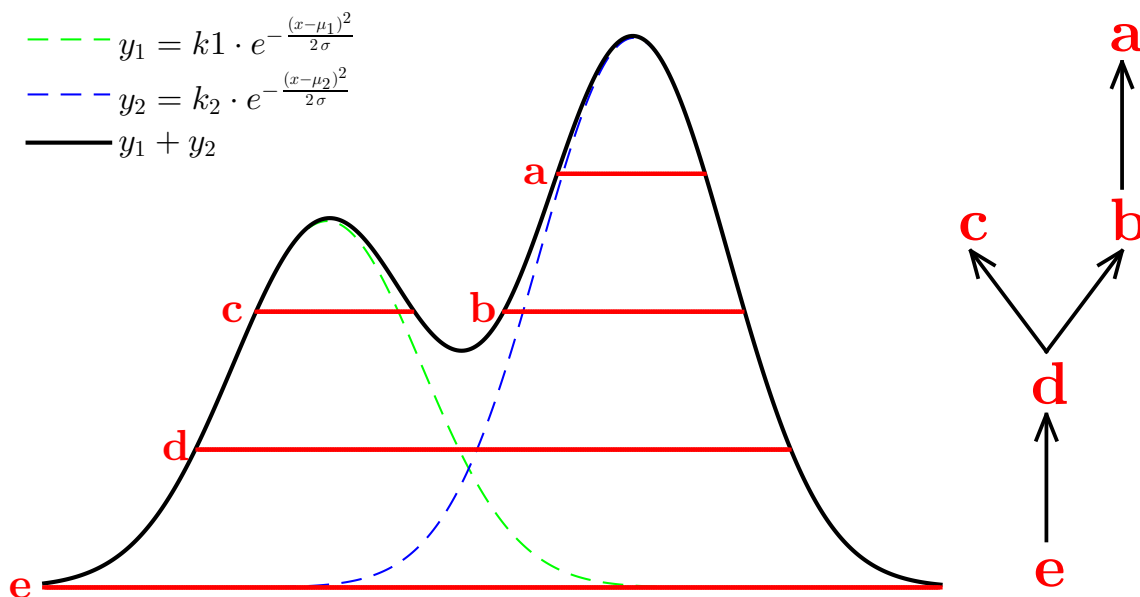
Laserové svazky se v reálném případě rozptylují. Kvůli různé rozbíhavosti svazků nejsme schopni určit údaje o ploše a intenzitě jejich obrazu. Určujeme pouze velikost zářivého toku svazku, kterou jsme schopni při nízkém rozptylu spolehlivě vypočítat.

V okolí obrazu svazků jsou patrné ocásky. Detekce ocásků a jejich klasifikace je popsána v samostatné kapitole 1.4.2.

Rozbíhavost svazků nemusí být ve všech směrech stejná. Na stínítku tak svazky tvoří stopy různých tvarů. Tvar stopy definujeme pomocí 3 parametrů.

### 1.4.1 Základní parametry

Máme detekované MSER oblasti. Nalezneme průniky oblastí a sestavíme stromovou strukturu. Kořenem stromu bude oblast s největší plochou a postupně se budou přidávat oblasti menší. Princip je patrný z 2D pohledu na prahovací úrovně MSER detektoru v obr. 1.7, kde vidíme i princip tvorby stromu. Výsledkem bude řada stromů s různým počtem listů. Počet všech listů určuje počet detekovaných stop v obraze.



Obrázek 1.7: Ilustrace překrytí stop v 2D řezu. Výsledná charakteristika je součtem dvou gaussových funkcí. Červeně jsou zakresleny prahovací úrovně MSER detektoru. Vlevo vidíme stromovou strukturu MSER oblastí a-f. Kořenem stromu je vrstva e. Vrstva d je jediný vnitřní uzel stromu. Listy představují vrstvy c a a. Důležité jsou podstromy e→d, c, b→a.

Cíleně prohledáváme jednotlivé stromy a nalézáme uzly, ze kterých počítáme parametry svazků.

- **Azimut a elevace** - Pozici dopadu světelné stopy určíme jako střed eliptické aproximace oblasti odpovídající listu stromu. Transformací z [5] získáme azimut a elevaci.
- **Zářivý tok** - Od filtrovaného snímku odečteme pozadí (obr. 1.6) a získáme snímek, ze kterého budeme odečítat intenzitu pixelů. Algoritmus výpočtu zářivého toku je následující:
  1.  $t_0$  = původní strom;  $i = 0$ ;  $q = 0$ ;  $n_0$  = počet listů v  $t_0$ ;
  2. Ve stromu  $t_i$  nalezneme podstromy  $\tau_1, \dots, \tau_n$  maximální velikosti bez vnitřních uzlů stromu  $t_i$  a obsahující jeden list stromu  $t_i$ .

3. Nalezneme křeny  $\xi_1, \dots, \xi_n$  podstromů  $\tau_1, \dots, \tau_n$ . Kořeny odpovídají oblastem s množinou pixelů  $\mathbb{M}_{q+1}, \dots, \mathbb{M}_{q+n}$ .
4. Pokud  $i = 0$  vypočítáme zářivý tok

$$\phi_{e_j} = \frac{\sum_{k \in \mathbb{M}_j} I_k}{N_j}, \quad j \in \{1, \dots, n_0\} \quad (1.6)$$

kde  $I_k$  je jas pixelu  $k$  ve snímku a  $N_j$  je počet pixelů v množině  $\mathbb{M}_j$ . Index  $j$  odpovídá indexu stopy ve stromu  $t_0$ .

Pokud  $i > 0$  nalezneme množiny  $\mathbb{P}_1, \dots, \mathbb{P}_n$ .  $\mathbb{P}_l$  je množina indexů listů, které jsou v  $t_0$  potomkem uzlu  $\xi_l$ , kde  $l \in 1, \dots, n$ . Zářivý tok stop upravíme.

$$\phi_{e_j} = \frac{\phi_{e_j}}{\sum_{q \in \mathbb{P}_l} \phi_{e_q}} \frac{\sum_{\{k \in \mathbb{M}_{q+l} \cap \{\mathbb{M}_1^c \cup \mathbb{M}_2^c \cup \dots \cup \mathbb{M}_q^c\} \mid \{1, 2, \dots, q\} = \mathbb{P}_l\}} I_k}{N_{q+l}} + \phi_{e_j}. \quad j \in \mathbb{P}_l \quad (1.7)$$

5.  $i = i + 1$ ;  $q = q + n$ ;
  6. Pokud  $n \neq 1$  odstraníme podstromy  $\tau_1, \dots, \tau_n$  z grafu  $t_{i-1}$ , získáme strom  $t_i$  a opakujeme od kroku 2.
- **Tvar** - Pro každou MSER oblast je určena elipsa, která uzavírá danou oblast. U této elipsy lze určit orientaci a velikost hlavních poloos.

Každé stopě odpovídá jeden list stromu. Nalezneme cestu  $\mathcal{C}$ , která je cestou od kořene k listu.

Orientace je určena jako medián orientací elips všech MSER oblastí v cestě  $\mathcal{C}$ . Velikosti hlavních poloos jsou určeny podle MSER oblasti, která je uprostřed cesty  $\mathcal{C}$ .

### 1.4.2 Detektor ocásků

Se znalostí směru a velikosti ocásků detekovaných svazků dostáváme nové informace, které mohou přispět k jejich správnému párování se svazky z matematického modelu kamene.

Ve snímaném obraze nelze rozpoznat všechny vznikající ocásky, ale pouze ty s dostatečně velkou intenzitou.

Princip detektoru ocásků zjednodušeně spočívá v převodu okolí stopy do polárních souřadnic (vzdálenost  $\rho$  a úhel  $\phi$ ) a nalezení oblastí úhlů, ve kterých je patrný výrazný vzestup intenzity oproti okolí. Takovéto oblasti jsou typicky důsledkem přítomnosti ocásků o obraze a lze tímto způsobem ocásky detekovat.

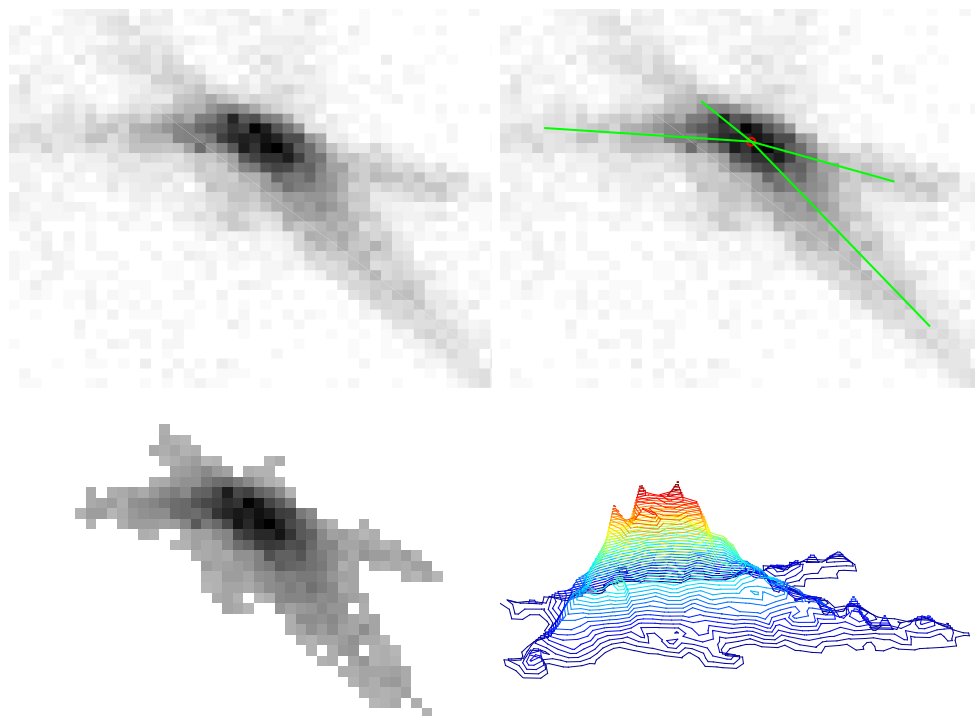
U rozvinutí do polárního grafu si však musíme všimnout překážek, které komplikují detekci ocásků.

- V blízkém okolí jedné stopy se může nacházet další stopa. V polárním grafu se tato blízká stopa jeví jako ocásek a dochází k falešné detekci.
- Stopy a ocásky jsou v obraze různě velké. Je třeba efektivně určovat vzdálenost  $\rho$  do které budeme převádět okolí stopy do polárního grafu. Pokud zvolíme malé  $\rho$  nepokryjeme oblast, kde se vyskytují ocásky a velké  $\rho$  zvýší časovou náročnost výpočtu.

Elegantní řešení přináší použití MSER detektoru, pomocí něhož získáme vymezení oblastí a tím i vzdálenosti  $\rho$ , kde se stopa i s ocásky nachází. Se znalostí oblastí náležící jednotlivým stopám jsme schopni od sebe stopy částečně oddělit a redukovat množství falešných detekcí. Na druhou stranu sousední stopa může ležet na pozici ocásku a odstraněním sousední stopy odstraníme současně i ocásek, který prozatím nejsme schopni v případě překrytí oddělit. Vzhledem k rozmanitosti stop, co do velikosti, intenzity, množství a tvaru ocásků apod. není jednoduché

stopu matematicky modelovat. Pokud by se podařilo vytvořit dostatečně přesný kompaktní model stopy, je možné uvažovat o situaci, kdy budeme schopni od sebe separovat překrývající se stopy a ocásky.

Pro znázornění postupu a mezivýsledků jsme si vybrali laserovou stopu (obr.1.8), která v obraze nekoliduje s další výraznou stopou. Zvolená stopa vznikla dopadem laserového svazku typu ....!!doplnit(až budou popsány všechny třídy svazků)!!.... V obraze jsou patrné čtyři ocásky různé intenzity.



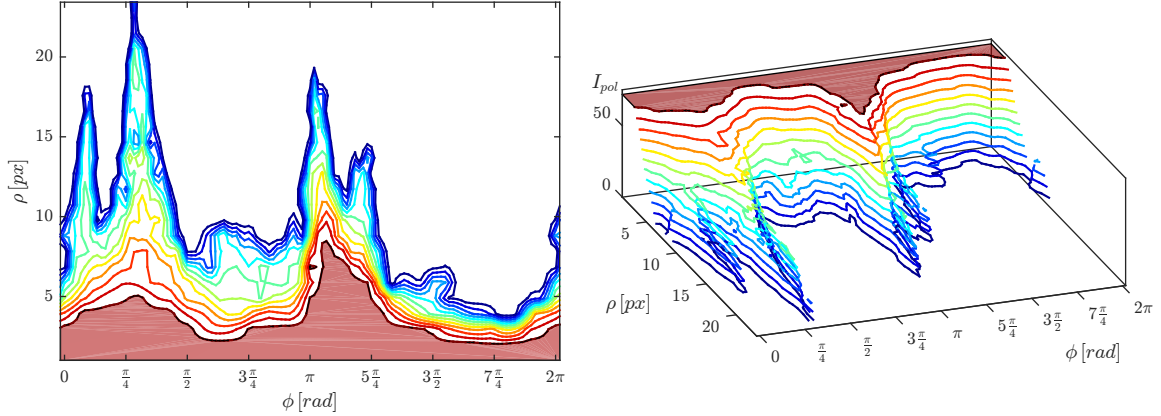
Obrázek 1.8: Zvolená laserová stopa k ilustraci algoritmu detektoru ocásků. Stopa vznikla dopadem svazku typu ..... na stínítko. V měřicí soustavě byl přitom umístěn kámen typu VIVA 12 s odstínem *Crystal* a průměrem 4.8 mm. Snímek vpravo nahoře znázorňuje výřez stopy ze snímaného obrazu. Můžeme zde pozorovat čtyři ocásky s různými vlastnostmi. Vlevo nahoře je výsledek detekce popsané níže. Obrázek vpravo dole ukazuje MSER oblast a poslední obrázek znázorňuje 3D pohled na zkoumanou stopu.

Jednotlivé kroky algoritmu:

- Vybereme stopu, u které chceme identifikovat ocásky a ze snímku vybereme oblast (obr.1.8), která náleží zkoumané stopě.
- Po odečtení okolního šumu  $z_{noise}$  prahujeme intenzitu pixelů stopy 2krát střední hodnota intenzity  $z_{mean}$ . Pixely s intenzitou vyšší než zvolený práh nastavíme intenzitu odpovídající prahové hodnotě a posuneme  $z_{noise}$  na hodnotu  $z_{mean}$  tak, že ke všem pixelům přičteme intenzitu  $z_{mean}$ . Důvodem tohoto kroku je snaha odstranit nežádoucí vlastnosti velkého šumu v hodnotách intenzity v blízkém okolí "těžiště" stopy a také to, že se chceme zvětšit relativní příspěvek pixelů s nižší intenzitou do součtového kritéria 1.8.
- Stopu s velikostí okolního šumu  $z_{mean}$  převedeme do polárních souřadnic  $(\rho, \phi)$ . Intenzitu  $I_{pol}$  v polárním grafu  $I_{pol} = f(\phi, \rho)$  určujeme pomocí bipolární interpolace, která pro větší efektivitu vynechává oblasti mimo oblast stopy, kde  $I_{pol} = 0$ . Důležitým parametrem při interpolaci je velikost vzorkování  $f_\phi$  úhlu  $\phi$  resp. vzorkování  $f_\rho$  vzdálenosti  $\rho$ . Experimen-



tálně jsme zvolili  $f_\phi = 3^\circ$  a  $f_\rho = 1$  px. Interpolaci počítáme v intervalech  $\phi \in \langle 0, 2\pi \rangle$  a  $\rho \in \langle 1, \rho_{max} \rangle$ , kde  $\phi_{max}$  je maximální vzdálenost všech pixelů v oblasti stopy od její pozice.



Obrázek 1.9: Dva pohledy na intenzitu okolí stopy převedené do polárního grafu  $I_{pol}$  zobrazené pomocí vrstevnic.

- Provedeme součet intenzit  $I_{pol}$  pro jednotlivé úhly  $\phi$  od minimální do maximální vzdálenosti  $\rho$  a získáme závislost  $I_\phi = f(\phi)$ , kde

$$I_{\phi_i} = \sum_{j=1}^{\rho_{max}} I_{pol}(i, j) . \quad i \in \left\{ 0, \frac{3}{180}\pi, \dots, 2\pi \right\} \quad (1.8)$$

Následně na  $I_\phi$  aplikujeme kubickou interpolaci sousedních hodnot s 5krát citlivějším vzorkováním  $f_{\phi_2} = \frac{f_\phi}{5}$  a rozšíříme rozsah  $\phi$  na  $\phi \in \langle -\frac{\pi}{2}, \frac{5}{2}\pi \rangle$ .

- Graf závislosti  $I_\phi = f(\phi)$  filtrujeme konvolucí s gausiánem  $g(x)$  se směrodatnou odchylkou  $\sigma = 3$  a získáme referenční závislost  $I_{filt}$ .

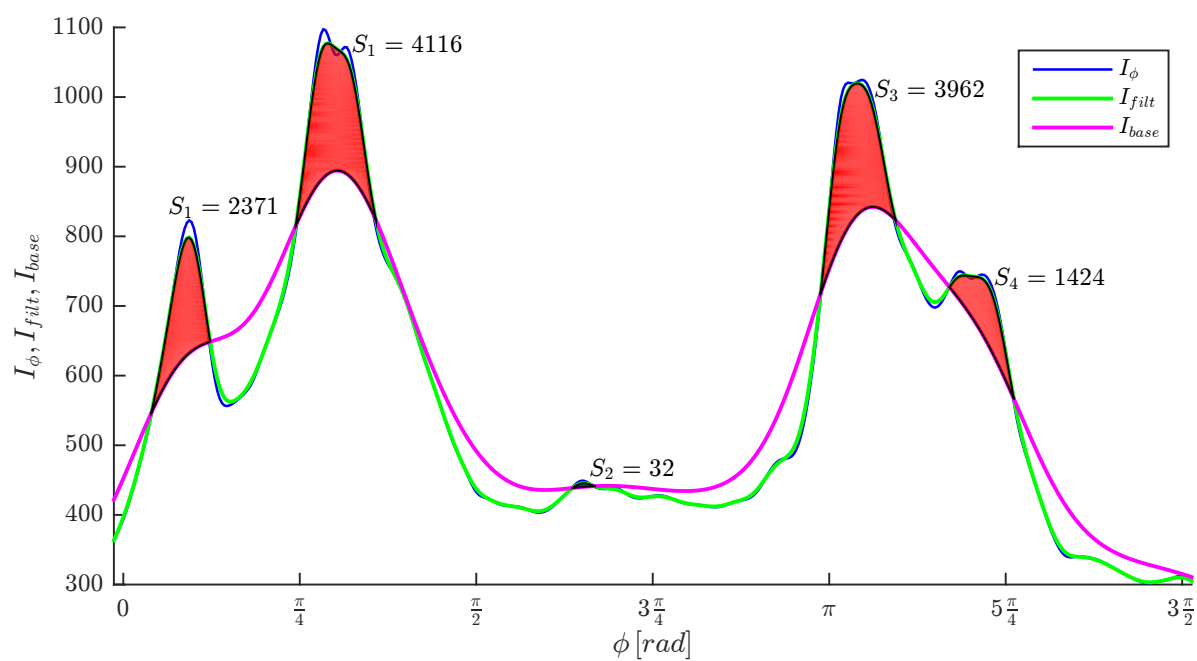
$$g(x) = \frac{1}{\sqrt{2\pi} \cdot \sigma} e^{-\frac{x^2}{2\sigma^2}} . \quad (1.9)$$

- Na graf  $I_{filt}$  následně opakovaně aplikujeme konvoluci, tentokrát s gausiánem  $g(x)$  s vyšší směrodatnou odchylkou  $\sigma = 8$ , abychom získali základnu  $I_{base}$ , kterou budeme porovnávat se signálem  $I_{filt}$ .
- Nalezneme souvislé oblasti  $\mathcal{R}_1, \dots, \mathcal{R}_n$ , kde graf  $I_{filt}$  má větší hodnotu než  $I_{base}$  a sečteme rozdíly  $I_{filt}$  a  $I_{base}$  v jednotlivých vzorcích. Velikost součtu  $S_1, \dots, S_n$  závisí na vzorkovací frekvenci  $f_{\phi_2}$ .

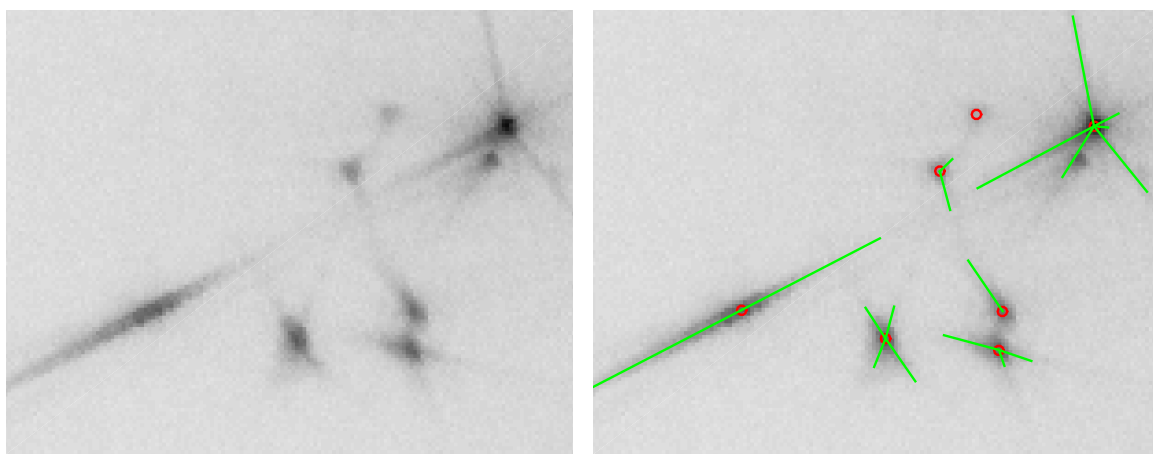
$$S_i = \sum_{\phi_j: \phi_j \in \mathcal{R}_i} I_{filt}(\phi_j) - I_{base}(\phi_j) . \quad i \in \{1, 2, \dots, n\} \quad (1.10)$$

- Za ocásek uvažujeme oblast  $\mathcal{R}_i$ , kde je součet  $S_i$  větší než prahovací úroveň  $s_{th}$  (pro  $f_{\phi_2}$  je  $s_{th} = 500$ ). Směr ocásku  $\varphi$  je určen jako úhel, ve kterém je graf  $I_{filt}$  v dané oblasti maximální a velikost ocásku  $\varrho_i$  určuje  $\rho_{max}$  a poměr součtu  $S_i$  k maximálnímu v pro danou stopu.

$$\varphi_i = \arg \max_{\phi_j: \phi_j \in \mathcal{R}_i} I_{filt}(\phi_j) , \quad \varrho_i = \frac{S_i}{\max_{j \in \{1, \dots, n\}} S_j} \rho_{max} . \quad (1.11)$$



Obrázek 1.10: Grafické vysvětlení funkce algoritmu pro detekci ocásků.



Obrázek 1.11: Ukázka funkce detektoru ocásků na vybraném vzorku z obrazu.

## 1.5 MSER detector configuration parameter list

Here is a list of the configuration parameters used for beam detection setting. All of them can be used with their default values. This description of the parameters can be useful for understanding the algorithm or debugging.

Name	Default	Description
<i>BackgroundFilt</i>	201	Size of Gaussian filter mask used for background estimation. <i>Sigma</i> of mask is equal to <i>BackgroundFilt</i>
<i>BackgroundKMean</i>	0.15	<i>BackgroundKMean</i> $\times$ <i>mean(image)</i> is added to background estimation.
<i>ComputeAllLayers</i>	1	1 - Divide all layers in marks variable calculation. 0 - Calculate only with upper layers.
<i>ComputeTails</i>	1	1 - Tails of marks are computed. It takes some time. 0 - No tails. Faster.
<i>CleanerAxisRatio</i>	10	MSER regions with upper axis ratio are deleted.
<i>CleanerFilterSigma</i>	5	<i>Sigma</i> of filter mask in image blurring.
<i>CleanerFilterSize</i>	31	Size of filter mask in image blurring.
<i>CleanerMean</i>	0.035	Regions with peak where difference of mean of surrounding intensity is lower than <i>CleanerMean</i> $\times$ <i>mean(image)</i> are deleted.
<i>CleanerSumWide</i>	5	Size of surrounding in cleaning process.
<i>FilterSigma</i>	0.7	<i>Sigma</i> of noise reducing filter mask.
<i>FilterSize</i>	3	Size of noise reducing filter mask.
<i>MSERMaxAreaVar</i>	0.9	This value specifies the step size between intensity threshold levels used in selecting extremal regions while testing for their stability. Decrease this value to return more regions.
<i>MSERRegion</i>	[2, 14000]	Two-element vector, [minArea maxArea], which specifies the size of the regions in pixels. This value allows the selection of regions containing pixels between minArea and maxArea, inclusive.
<i>MSERThresh</i>	1.2	Increase this value to return a greater number of regions at the cost of their stability. Stable regions are very similar in size over varying intensity thresholds.
<i>SurroundThres</i>	0.8	Multiple of <i>mean(image)</i> where surrounding intensity is moved.

## 2. Optimalizace orientace faset

### 2.1 Postup optimalizace

Před tím, než přistoupíme k problému optimalizace, je třeba získat potřebná data. Důležité jsou informace o parametrech svazků, které lze rozdělit na *simulované* a *reálné*.

#### Simulované svazky

1. Získáme potřebné parametry kamene, který zkoumáme. Zdrojem může být technický výkres, nebo předchozí měření.
2. Sestavíme model, který bude přibližně určovat tvar kamene. Tento model budeme považovat za *referenční*.
3. V programu LADOK simulujeme průlet svazku *referenčním* modelem. Pro simulaci je důležité znát elektromagnetické vlastnosti laserového svazku a index lomu kamene.
4. Výsledkem simulace jsou parametry *simulovaných* svazků.

#### Reálné svazky

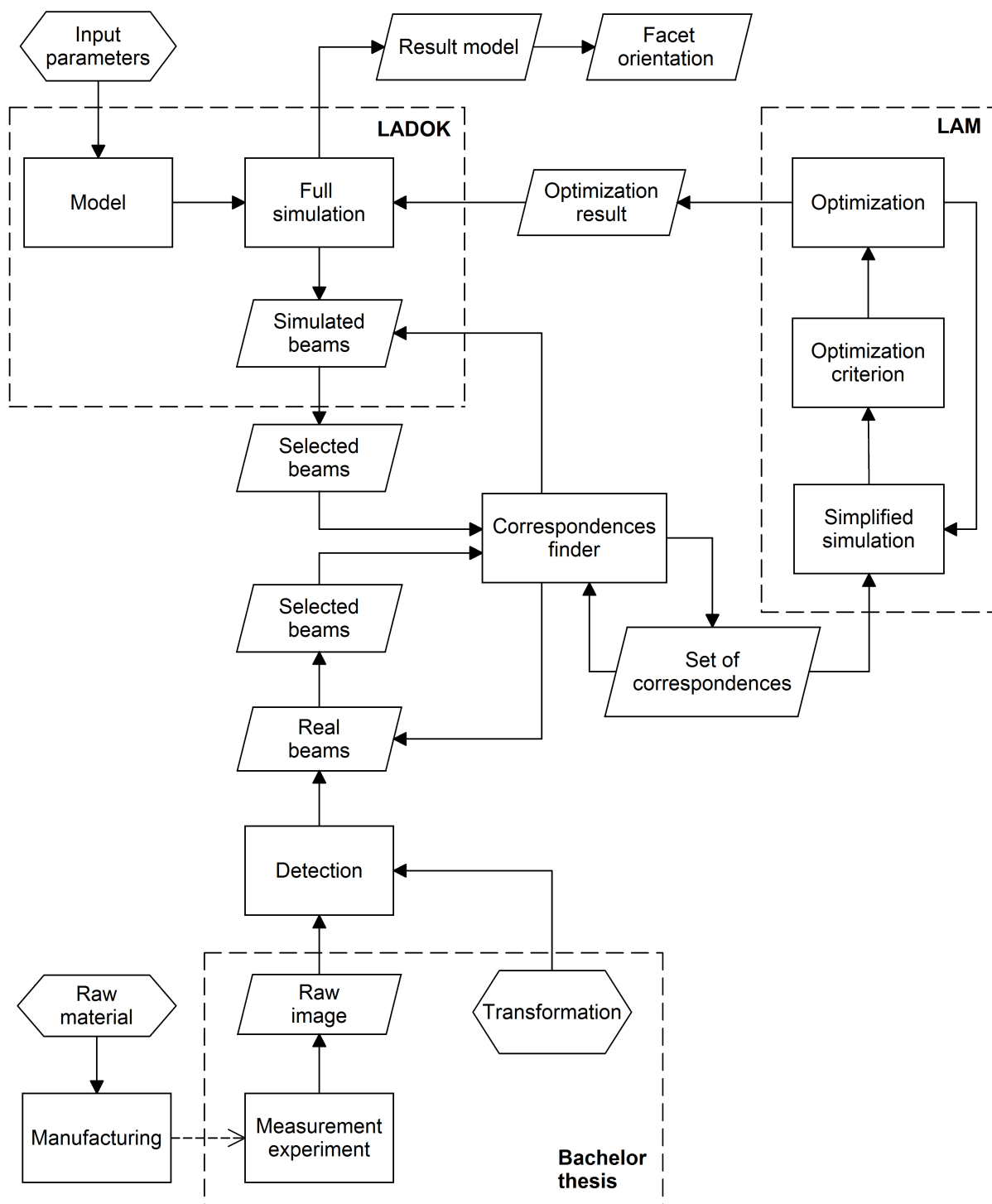
Předpokladem pro získání parametrů *reálných* svazků je sestavení a kalibrace měřicí soustavy podle [5].

1. Opracovaný kámen umístíme do měřicí soustavy.
2. Provedeme experiment průchodu svazku kamenem podobný situaci v simulačním programu LADOK.
3. Získáme obraz dopadu svazků na stínítko.
4. V obraze detekujeme světelné stopy (kapitola 1).
5. Z detekovaných stop vypočítáme parametry *reálných* svazků (kapitola 1.4).

Přecházíme k situaci, kdy máme dostupné informace o *simulovaných* i *reálných* svazcích. Mezi těmito dvěma množinami je třeba nalézt korespondence. Korespondující svazky si odpovídají seznamem faset kamene, na které při své cestě dopadají.

Pro korespondující páry určíme chybovou funkci a parametry budeme optimalizovat. Optimalizační algoritmus odhadne takové nastavení parametrů, aby bylo optimalizované kritérium co nejmenší.

Optimalizované parametry použijeme k výpočtu *optimalizovaného* modelu kamene. Orientace faset broušeného kamene odečteme z *optimalizovaného* modelu.



Obrázek 2.1: Diagram with princip of cut stone facet orientation estimator.

## 2.2 Korespondence

Zdrojový svazek odpadá na maximální možnou plochu kamene. Vlivem toho vystupuje z kamene velké množství svazků. Po dopadu svazků na stínítko je složité zpětně určit, které stopy náležejí kterým svazkům.

### 2.2.1 Podmíněnost

Základní otázkou optimalizačního problému je, zda je systém dostatečně podmíněný. První podmínku, kterou musíme splnit je získat minimálně stejný počet nezávislých rovnic, jako je počet optimalizovaných parametrů.

V optimalizačním kritériu máme celkem  $2r + q$  nezávislých rovnic. Základní podmínkou je, že počet korespondencí musí být minimálně  $s + q$ .

To, jak dobře korespondující svazky podmiňují daný model do jisté míry závisí na počtu dopadových faset. Svazky s velkým počtem dopadových faset logicky

Od tříd **1A** a **1B** lze obecně čekat velmi dobrou podmíněnost. Jednotlivý svazek z těchto tříd dopadne pouze na jednu fasetu a směr odraženého svazku jednoznačně určuje parametry fasety, od které se svazek odrazil. Pokud by náš matematický model přesně odpovídal reálnému experimentu, potom postačí nalézt korespondence třídy **1A 1B**, abychom určili parametry všech faset kromě spodku. Tyto třídy nepodmiňují index lomu kamene, pokud bychom chtěli zjistit index

Teoreticky korespondencí svazků třídy **3A** (např. UF1-TOP-BOT) získáme 2 rovnice  $\Delta e$  a  $\Delta \alpha$ , které jsou závislé na parametrech dopadových faset UF1, TOP a BOT. Problém spočívá v tom, že tato korespondence určuje pouze vzájemnou polohu dopadových faset a nelze u této třídy očekávat dobrou podmíněnost.

### 2.2.2 Výpočet odchylek parametrů

1. možnost - kámen během optimalizace rotujeme. Výsledky rotace si zapamatujeme a odchylky faset budeme počítat vzhledem k parametrům faset rotovaného kamene.

2.možnost - Kámen sice rotujeme a optimalizujeme, ale to nic nemění na tom, jak budeme počítat odchylky. Vezmeme si normály faset optimalizovaného a referenčního kamene. Nalezneme rotační matici mezi normálami spodku optimalizovaného a referenčního kamene. Rotační matici aplikujeme i na všechny ostatní fasety. Výsledkem bude to, že kamen srovnáme podle spodku. Chybové parametry pro spodek tak budou nulové.

Dále se postupuje tak, že se nalezne rotace mezi

### 2.2.3 Pozorování

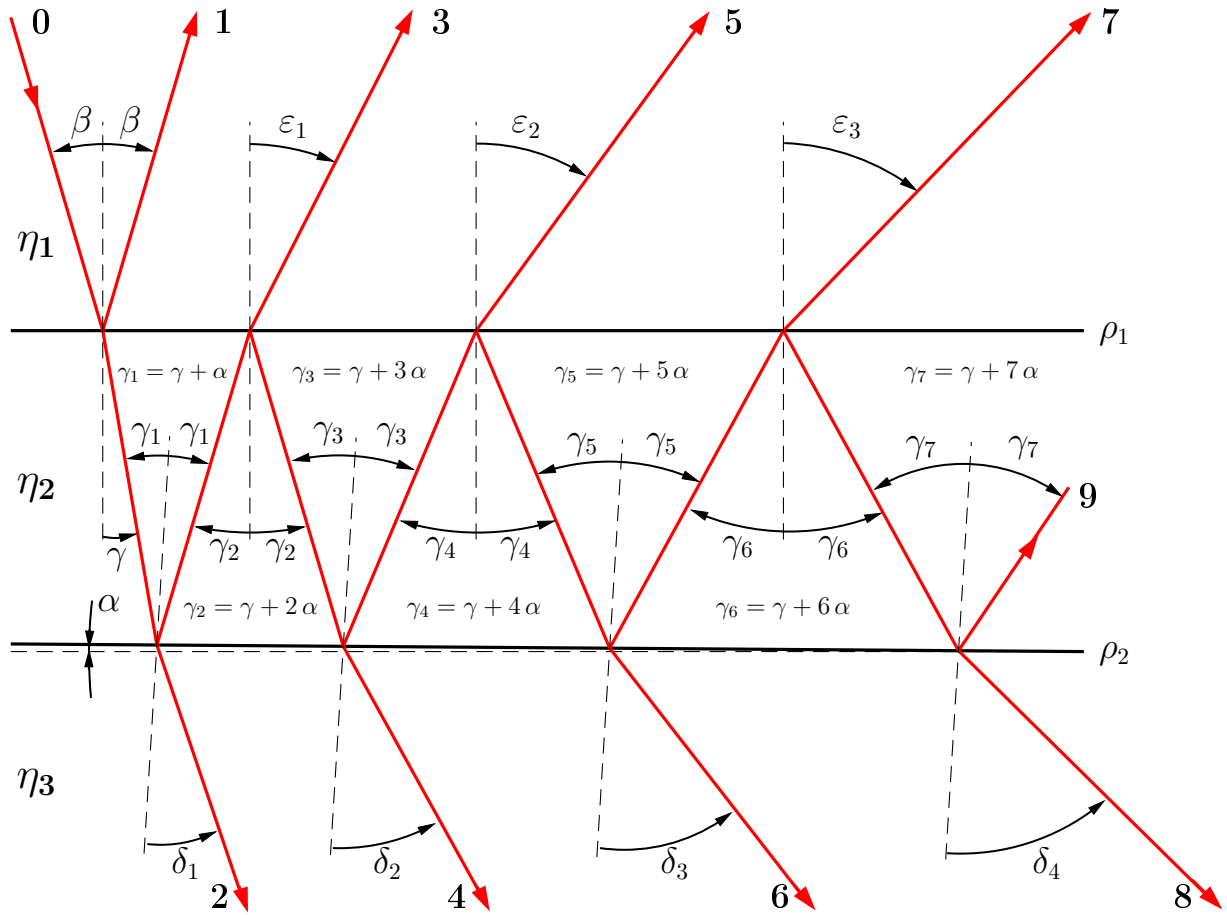
Simulujeme průlet světelného svazku optickým klínem (obr. 2.2). Prostředí s indexy lomu  $\eta_1$ ,  $\eta_2$  a  $\eta_3$  oddělují rozhraní  $\rho_1$  a  $\rho_2$ . Tyto rozhraní mezi sebou svírají úhel  $\alpha$ .

$$\gamma = \arcsin \left( \frac{\eta_1}{\eta_2} \sin \beta \right). \quad (2.1)$$

$$\varepsilon_1 = \arcsin \left( \frac{\eta_2}{\eta_1} \sin (\gamma + 2\alpha) \right) \rightarrow \begin{cases} \alpha > 0, & \varepsilon_3 > \varepsilon_2 > \varepsilon_1 > \beta \\ \alpha = 0, & \varepsilon_3 = \varepsilon_2 = \varepsilon_1 = \beta \\ \alpha < 0, & \varepsilon_3 < \varepsilon_2 < \varepsilon_1 < \beta \end{cases} \quad (2.2)$$

Svazek dopadá na rozhraní  $\rho_1$  pod úhlem  $\beta$ . Na obrázku 2.2 je tento svazek reprezentován paprsek č.0.

Svazek světla můžeme charakterizovat velikostí zářivého toku  $\phi_e$ . Z Fresnelových rovnic víme, že pokud na rozhraní  $\rho_1$  nedochází k totálnímu odrazu, tak po dopadu svazku na rozhraní vznikne



Obrázek 2.2: Lom a odraz paprsku v optickém klínu.

odražený a lomený svazek. Jaká bude velikost zářivého toku odraženého svazku  $\phi_{e_{reflect}}$  závisí na polarizaci světla, dopadajícím úhlu a poměrem mezi indexy lomu prostředí, které odděluje rozhraní  $\rho_1$ .

$$\phi_e = \phi_{e_{reflect}} + \phi_{e_{refract}}. \quad (2.3)$$

Při dopadajícím úhlu  $\beta = 0^\circ$  a indexech lomu  $\eta_1 = 1$ ,  $\eta_2 = 1.5$  se 4 % dopadajícího zářivého toku odrazí.  $\frac{\phi_{e_{reflect}}}{\phi_e} = 0.04$ ,  $\frac{\phi_{e_{refract}}}{\phi_e} = 0.96$ .

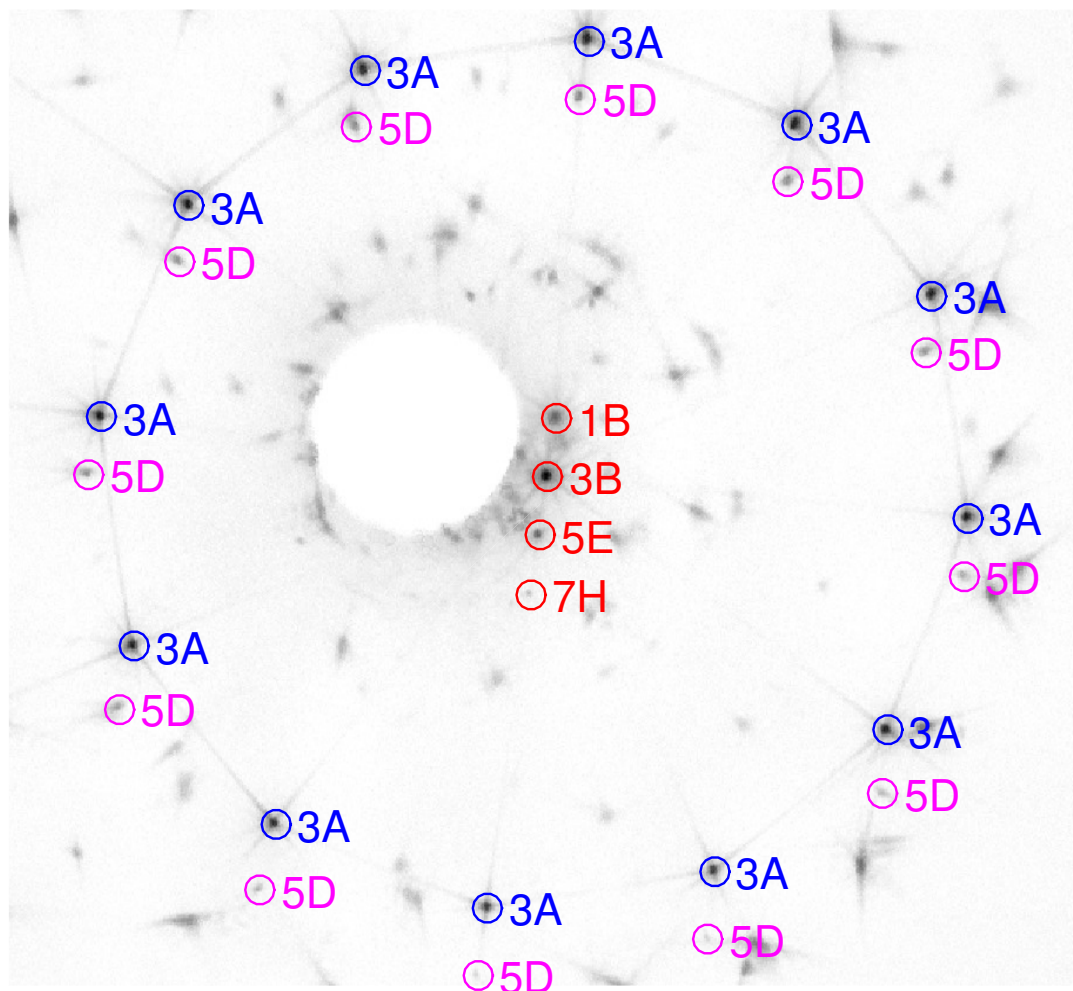
Z principu šíření světla optickým prostředím pozorujeme svazky 0 až 9, se specifickým směrem šíření.

V šatonové rúži nastává stejný optický jev mezi tabulkou (TOP) a spodkem (BOT), kde

$\eta_1$	index lomu vzduchu,
$\eta_2$	index lomu materiálu kamene,
$\eta_3$	index lomu odrazivé vrstvy,
$\rho_1$	tabulka
$\rho_2$	spodek
<b>0</b>	zdrojový svazek,
<b>1</b>	svazek třídy <b>1B</b> - TOP,
<b>3</b>	svazek třídy <b>3B</b> - TOP-BOT-TOP,
<b>5</b>	svazek třídy <b>5E</b> - TOP-BOT-TOP-BOT-TOP,
<b>7</b>	svazek třídy <b>7H</b> - TOP-BOT-TOP-BOT-TOP-BOT-TOP,
<b>2,4,6,8</b>	tyto svazky nevznikají - na rozhraní $\rho_2$ dochází pouze k odrazu.

V reálné situaci nejsou fasety TOP a BOT rovnoběžné. Důsledkem toho svazky **1B**, **3B**, **5E** a **7H** svírají s normálou tabulky různý úhlem. Svazek třídy **3B** bude mít vždy největší zářivý tok. Zajímavé je, že tyto svazky leží ve stejné rovině. Tato rovina je určena vzájemnou orientací mezi normálou tabulky a normálou spodku.

Po dopadu svazků **1B**, **3B**, **5E** a **7H** na stínítko můžeme v obraze pozorovat stopy ležící na jedné přímce obr. 2.3.



Obrázek 2.3: Zvýraznění obrazů svazků ve snímku. Svazky třídy **1B**, **3B**, **5E** a **7H** dopadají pouze na fasetu TOP a BOT. Svazky třídy **3A** a **5D** dopadnou nejprve na boční fasetu a poté následuje jeden resp. dva dopady na dvojici faset BOT-TOP.

Pokud bychom byli schopni přiřadit alespoň 2 tyto stopy ke svazkům můžeme určit orientaci tabulky a spodku. Situaci ovšem komplikuje fakt, že ne vždy můžeme nalézt obrazy těchto stop, protože jsou zastíněny podstavcem, na který pokládáme kámen.

U svazků třídy **3A** a **5D** dochází k podobnému optickému jevu pouze s tím rozdílem, že svazek do kamene nevstupuje tabulkou, ale boční fasetou. Můžeme si všimnout stejné orientace obrazů dvojice svazků se stejnou vstupující fasetou. Tato dvojice také určuje vzájemnou orientaci mezi normálou tabulky a normálou spodku. Pro jednoznačné této orientace je však nutné znát orientaci bočních faset.



## 2.3 Optimalizované kritérium

Optimalizační algoritmus je převzat z práce [3]. Některé části byly pozměněny. Ukážeme si stručný přehled metody optimalizace a zvýrazníme provedené úpravy. Optimalizační algoritmus používáme nejen k odhadu parametrů faset kamene, ale také k odhadu indexu lomu kamene a jeho orientace kamene v měřené soustavě

Definujeme kritériální funkci, kterou budeme optimalizovat. Funkci lze popsat vztahem

$$\vec{\varepsilon} = h(\vec{x}, \vec{v}, \vec{l}, \vec{p}), \quad (2.4)$$

- $\vec{p}$  Obsahuje směrové vektory *reálných* svazků. Směr popisujeme pomocí souřadnic azimutu a elevace. Pokud pro výpočet optimalizačního kritéria použijeme *s reálných* svazků, bude mít vektor  $\vec{p}$  délku  $2s$ .
- $\vec{l}$  Obsahuje seznam dopadových faset svazku. Tento seznam je využit pro výpočet směru výstupního svazku. Délka seznamu se musí rovnat  $s$ .
- $\vec{v}$  Popisuje směr zdrojového svazku světla
- $\vec{x}$  Vektor parametrů, které nastavuje optimalizační algoritmus.

**Parametry faset, index lomu** Jedná se převážně o parametry  $r$  uvolněných faset. Každou fasetu lze parametrizovat pomocí úhlové změny normálového vektoru  $\vec{n}$  (2 parametry) a změny vzdálenosti  $d$  fasety od souřadného systému (1 parametr).

Optimalizační metoda nemá dostatečnou citlivost na změny vzdálenosti  $d$  fasety [3]. Tento parametr proto považujeme za konstantu.

Nově lze mezi optimalizované parametry přidat index lomu kamene  $n_i$ . Celkově máme  $2r + q$  parametrů, kde  $q$  je 1 pokud je mezi  $\vec{x}$  parametr  $n_i$ , jinak je  $q$  rovno 0.

**Orientace** Orientaci kamene popisujeme rotací kolem vertikální osy  $R_z$  a dvou parametrů definujících náklon kamene  $R_x, R_y$ .

- $\vec{\varepsilon}$  Představuje vektor odchylek v elevaci a azimutu *simulovaných a reálných* svazků.

V práci [3] byla tato odchylka měřena jako chyba v pozici dopadu laserového svazku na stínítku. Důvodem proč bylo využíváno toto kritérium byla citlivější odezva při změně vektoru svazku. Hlavním důvodem volby azimutu a elevace pro výpočet optimalizačního kritéria je vyšší rychlost. Odpadá totiž potřebný výpočet, který transformuje směrový vektor do souřadnic na stínítku.

Vektor  $\vec{\varepsilon}$  má  $2s$  prvků.

## 2.4 Plná vs. zjednodušená simulace

V optimalizačním cyklu je použito dvou simulací, které simulují průlet světla broušeným kamenem

### 2.4.1 Plná simulace

Plnou simulací rozumíme klasický program LADOK, který modeluje odraz a lom svazků v konvexním tělese. Nevýhodou algoritmu je ale to, že simulace trvá příliš dlouho na to, aby mohla být úspěšně použita v optimalizačním procesu.

### 2.4.2 Zjednodušená simulace

Základní myšlenkou je, že se optimalizované parametry  $\vec{x}$  příliš nemění. Za tohoto předpokladu si podstatná část svazků zachová posloupnost dopadových faset. Vynecháme proto kontrolu vzniku či zániku svazků.

Zjednodušená simulace vyřadí z plné simulace zbytečné výpočty, které v procesu optimalizace nevyžíváme. Jediné, co potřebujeme znát je směr výstupních svazků. Pro výpočet směru nahradíme svazky nekonečně tenkými paprsky a můžeme s nimi pracovat jako s vektory. Fasety kamene reprezentujeme pomocí normálového vektoru.

Zjednodušená simulace přistupuje k paprsku jednotlivě. Funkci pro výpočet vektoru výstupního paprsku  $\vec{v}_o$  lze vyjádřit jako

$$\vec{v}_o = g\left(\vec{v}_i, \vec{N}\right), \quad (2.5)$$

kde  $\vec{v}_i$  je vektor vstupního paprsku. Vektor  $\vec{N}$  obsahuje normály  $\vec{n}_1, \dots, \vec{n}_m$  faset na které svazek dopadá. Normály jsou seřazené v pořadí odpovídající dopadovým fasetám při šíření paprsku od zdroje ke stínítku.

Při výpočtu musíme vědět, která faseta svazek odráží a která lomí. Situace je jednoduchá. Pokud  $m = 1$ , potom muselo dojít k odrazu. Pokud  $m > 1$  odpovídají normály  $\vec{n}_1$  a  $\vec{n}_m$  fasetám, přes které se svazek lomí. Na ostatních fasetách se svazek odrazí.

Zjednodušená simulace v LAMu pracuje s jinou funkcí. Základním rozdílem je to že navíc počítá polohu, kam paprsek na fasetu dopadl. Výpočet odrazu a jsou v LAMu pomalejší přibližně o 15 %.

## 2.5 Klasifikace příznaků

## 2.6 Implementace

Uvedeme algoritmy využitě pro hledání korespondencí měřených a referenčních svazků a popíšeme jejich aplikaci pro automatické určení náklonu faset broušených kamenů. Předtím, než se začneme zabývat navrženými přístupy, je třeba sjednotit značení jednotlivých parametrů svazků.

Referenční svazky dělíme do tří množin.

$\mathcal{R}_c$  - uspořádaná  $r_c$ -tice svazků, pro které byl nalezen korespondující měřený svazek,

$\mathcal{R}$  - uspořádaná  $r$ -tice svazků, ke kterým můžeme přiřadit korespondující měřený savek,

$\mathcal{R}_o$  - uspořádaná  $r_o$ -tice svazků, které není možné přidat do množiny korespondencí  $\mathcal{C}$ .

Pro tyto množiny platí  $\mathcal{R}_c \cap \mathcal{R} = \emptyset$ ,  $\mathcal{R} \cap \mathcal{R}_o = \emptyset$ ,  $\mathcal{R}_c \cap \mathcal{R}_o = \emptyset$ , kde  $\mathcal{R} = (\mathcal{R}_1, \dots, \mathcal{R}_r)$ . Dále definujeme uspořádanou  $r'$ -tici  $\mathcal{R}' = \mathcal{R} \cup \mathcal{R}_o$ .  $r' = r + r_o$ .

Podobně dělíme měřené svazky na  $\mathcal{M}_c$ ,  $\mathcal{M}$ ,  $\mathcal{M}_o$  a  $\mathcal{M}'$ , kde  $\mathcal{M} = (\mathcal{M}_1, \dots, \mathcal{M}_m)$ .

Algoritmy pracují s informacemi o směru, obrazu a zářivém toku  $\phi_e$  svazků. Směr vyjadřujeme pomocí azimutu  $\alpha$  a elevace  $\varepsilon$ . Polohu obrazu svazku určuje  $x$ -ová a  $y$ -ová souřadnice.

Parametry referenčních svazků označujeme  $\vec{\alpha}_r$ ,  $\vec{\varepsilon}_r$ ,  $\vec{\phi}_{e_r}$ ,  $\vec{x}_r$  a  $\vec{y}_r$ . Pro měřené svazky platí označení  $\vec{\alpha}_m$ ,  $\vec{\varepsilon}_m$ ,  $\vec{\phi}_{e_m}$ ,  $\vec{x}_m$  a  $\vec{y}_m$ . Platí, že  $\vec{\alpha}_r = \alpha_r(\mathcal{R})$ ,  $\vec{\alpha}_m = \alpha_m(\mathcal{M})$  apod.

Množinu korespondujících svazků značíme  $\mathcal{C}$ . Jednotlivá korespondence  $C_i$  je uspořádaná dvojice  $(\mathcal{R}_j, \mathcal{M}_k)$ ,  $C_i \subseteq \mathcal{C}$ .  $\alpha_r(C_i)$  označuje azimut referenčního svazku  $\mathcal{R}_j$ , kde  $\mathcal{R}_j \subset C_i$  a  $\alpha_r(\mathcal{C})$  vektor  $(\alpha_r(C_1), \dots, \alpha_r(C_n))$ , kde  $n$  je počet uspořádaných dvojic v množině  $\mathcal{C}$ .

### 2.6.1 Použité algoritmy

Korespondence hledáme buď pro specifickou třídu tříd svazků nebo pro svazky, které splní požadované parametry.

#### 2.6.1.1 Korespondence svazků třídy 1A

Víme, že svazky třídy **1A** se odráží od faset **UF1** - **UF2**. Úhlová odchylka normály referenční fasety skutečné normály se projeví dvojnásobnou úhlovou odchylkou referenčního svazku od měřeného. Očekáváme, že rozdíl mezi referenčními a reálnými parametry faset není příliš velký. Proto lze očekávat, že měřené svazky třídy **1A** budou mít velmi podobný směr jako referenční. Přibližně tedy známe směr těchto svazků.

V blízkém okolí stop třídy **1A** se mohou nacházet pouze stopy s řádově nižším zářivým tokem. Tato vlastnost je zajištěna geometrií kamene *viva12* a provedením experimentu. Pokud známe zářivý tok stop, můžeme v obraze snadno nalézt svazek třídy **1A**.

#### Parametry algoritmu

$d_{max}$  - kvadrát maximální vzdálenosti měřeného svazku od referenčního,

$n_{min}$  - minimální počet měřených svazků, v blízkém okolí referenčního svazku, s nižším zářivým tokem.

#### Popis algoritmu

1. Definujeme proměnné  $\alpha_c = 0$ ,  $\varepsilon_c = 0$ ,  $\mathcal{C} = \emptyset$ ,  $\mathcal{C}_l = \emptyset$ .
2.  $i = 0$ .
3. Vypočítáme kritérium hodnotící vzdálenost mezi svazky  

$$\vec{d} = (\alpha_r(R_i) \cdot \vec{1} - \vec{\alpha}_m - \alpha_c \cdot \vec{1})^2 + (\varepsilon_r(R_i) \cdot \vec{1} - \vec{\varepsilon}_m - \varepsilon_c \cdot \vec{1})^2.$$
4. Vybereme uspořádanou  $k$ -tici svazků  $\mathcal{N}$  z  $\mathcal{M}$ , která odpovídá rostoucí posloupnosti  $\vec{d}$  a omezení na  $\vec{d} < d_{max}$ .  $\mathcal{N}_1 \sim \arg \min(d)$  tak, že  $\mathcal{N} = (M_a, \dots, M_b)$ , kde  $a = \arg \min(\vec{d})$  a  $b = \arg \min\left(\frac{1}{\vec{d} - d_{max}}\right)$ . Dále bude platit, že  $N_1 = M_a$  a  $N_k = M_b$  apod.
5. Nalezneme uspořádanou  $k$ -tici  $\mathcal{O}$  takovou, že  $\mathcal{O} = (N_1, N_e, N_f, \dots, N_g)$ , kde  $e = \arg \min(\phi_{em}(N_1), \phi_{em}(N_2))$ ,  $f = \arg \min(\phi_{em}(N_1), \phi_{em}(N_2), \phi_{em}(N_3))$  a  $g = \arg \min(\phi_{em}(N_1), \dots, \phi_{em}(N_k))$ .
6. Určíme vektor  $\vec{n}_{\mathcal{O}}$  o velikosti  $k$  udávající četnost prvků z množiny  $\mathcal{N}$  v množině  $\mathcal{O}$ . Pokud  $O_q = \mathcal{O}_s$ , tak  $n_{\mathcal{O}}(q) = n_{\mathcal{O}}(s)$ , pro  $q = \{1, \dots, k\}$  a podobně  $s = \{1, \dots, k\}$ .
7. Určíme minimální  $l$ , které splňuje alespoň jednu z podmínek  $n_{\mathcal{O}}(l) > n_{min}$  a  $l = k$ . Do množiny  $\mathcal{C}$  přidáme korespondenci  $(R_i, N_l)$ .
8. Pokud  $i \neq r$ , tak  $i = i + 1$  a opakujeme body 3) až 7).
9. Pokud  $\mathcal{C}_l \neq \mathcal{C}$ , vypočítáme korekční parametry  $\alpha_c = \text{median}(\alpha_r(\mathcal{C}) - \alpha_m(\mathcal{C}))$ ,  $\varepsilon_c = \text{median}(\varepsilon_r(\mathcal{C}) - \varepsilon_m(\mathcal{C}))$ , položíme  $\mathcal{C}_l$  rovno  $\mathcal{C}$  a opakujeme body 2) až 8).
10. Z  $\mathcal{C}$  odstraníme prvky, ve kterých se obraz referenčního svazku nachází mimo obraz stínítka.
11. Optimalizujeme náklon kamene (kapitola 2.3).
12. Podle výsledku optimalizace upravíme model kamene a přepočítáme parametry referenčních svazků. Opakujeme bod 1) až 10). V příští iteraci tento bod vynecháme.
13. Z  $\mathcal{C}$  odstraníme prvky, ve kterých se obraz referenčního svazku nachází mimo obraz stínítka.

### 2.6.1.2 Korespondence svazků třídy 3A

Po optimalizaci podle třídy **1A** máme dobře odhadnuté parametry faset **UF1** - **UF12**. Pomocí rotace a náklonu kamene odhadneme přibližně parametry faset **TOP** a **BOT**. Podstatné je, že svazek třídy **3A** dopadá na tabulku **TOP** pod výrazně menším úhlem, než je kritický úhel. To zajišťuje přijatelnou citlivost směru svazků na změnu parametrů dopadových faset.

Referenční svazky třídy **3A** mají po třídě **3B** druhý nejvyšší zářivý tok. Ostatní třídy se vyznačují řádově nižším zářivým tokem. Podle zářivého toku  $\phi_{e_r}$  lze tedy snadno oddělit svazky se třemi dopadovými fasetami od ostatních svazků. Referenčních svazků třídy **3A** a **3B** je celkem 13. Často nastává situace, že svazek třídy **3B** nedetekujeme, protože je jeho obraz zakrytý podstavcem na kámen.

Separaci podle zářivého toku však nelze s jistotou použít u měřených svazků. Pokud dojde k překrytí obrazu svazků jiných tříd, může být zářivý tok tohoto shluku větší. Spolehlivý způsob, jak oddělit měřené svazky třídy **3A** a **3B** od ostatních, je porovnat maximální úrovně jasu v obrazech svazků.

Další vlastnost, která dobře charakterizuje svazky třídy **3A** jsou dlouhé a intenzivní ocásky. Při hledání korespondencí můžeme využít párování svazků podle charakteru ocásků (kapitola 2.6.1.5). Výhodou algoritmu je to, že dokážeme za vhodného nastavení nalézt korespondující svazky, a to i v případě vysokých směrových odchylek svazků. Nevýhodou je necitlivost na ostatní parametry svazků.

#### Popis algoritmu

1. Redukujeme počet měřených svazků a vytvoříme uspořádanou 13-tici  $\mathcal{M}$  obsahující prvních 13 měřených svazků s nejvyšší hodnotou jasu v obraze.
2. Podle podobnosti ocásků nalezneme počáteční odhad množiny korespondencí  $\mathcal{C}$  (kapitola 2.6.1.5)). Parametry algoritmu:  $\sigma = 0.2$ ,  $L_{min} = 2$ ,  $\Delta\alpha_{max} = 35^\circ$ ,  $\Delta\varepsilon_{max} = 15^\circ$ .
3. Uvolníme pouze parametry faset **TOP** a **BOT**. V tomto kroku prohlásíme parametry těchto dvou faset za totožné a optimalizujeme parametry uvolněných faset (kapitola 2.3). Pokud neznáme dostatečně přesně index lomu kamene optimalizujeme také index lomu.
4. Podle výsledku optimalizace upravíme model kamene a přepočítáme parametry referenčních svazků. Pro finální odhad množiny korespondencí  $\mathcal{C}$  použijeme algoritmus v kapitole 2.6.1.1 od bodu 1) do bodu 10).  $\mathcal{R}$  bude uspořádaná 24-tice referenčních svazků třídy **1A** a **3A**.  $\mathcal{M}$  bude obsahovat všechny měřené svazky.
5. K optimalizovaným parametrům přidáme parametry faset **UF1** až **UF12** a optimalizujeme.

### 2.6.1.3 Korespondence svazků třídy 5D

Svazky třídy **5D** můžeme v obraze pozorovat, pokud existuje úhlová odchylka mezi fasetou **TOP** a **BOT**. Předpokládáme, že pokud nějaká odchylka vznikne, bude max.  $1^\circ$ . Také předkládáme, že normály faset **UF1** až **UF12** nejsou vůči pravidelnému tvaru *vivy12* příliš vychýleny.

Podstatné je, že známe polohu svazků třídy **3A**. Za výše uvedených okolností lze pozorovat vzor určující vzájemnou polohu dvojice svazků třídy **3A** a **5D**, které mají v seznamu stejné dopadové fasety např. dvojice (UF1-TOP-BOT, UF1-TOP-BOT-TOP-BOT). Vzájemnou polohu mezi všemi těmito páry lze popsat pomocí polárních souřadnic vzdáleností  $\rho$  a úhlem  $\varphi$ .

Polohu  $j$ -tého měřeného svazku popisujeme souřadnicemi  $x_m(M_j)$  a  $y_m(M_j)$ . Měřené svazky rozdělíme na uspořádanou  $n$ -tici  $\mathcal{N}$  obsahující 12 svazků třídy 3A a uspořádanou  $o$ -tici  $\mathcal{O}$  obsahující zbylé svazky.

#### Parametry algoritmu

- $\rho_{max}$  - maximální vzdálenost obrazu svazků třídy **3A** a **5D** v pixelech,
- $\Delta\rho$  - maximální absolutní odchylka úhlu v obraze mezi párem svazků třídy **3A** a **5D**,
- $\Delta\varphi$  - maximální absolutní odchylka vzdálenosti v obraze mezi párem svazků třídy **3A** a **5D**,
- $p_{min}$  - minimální počet nalezených dvojic.

#### Algoritmus

1. Vypočítáme Euklidovu vzdálenost obrazů jednotlivých svazků

$$\rho_{j,k} = \sqrt{(x_m(N_j) - x_m(O_k))^2 + (y_m(N_j) - y_m(O_k))^2}$$

pro  $j = \{1, \dots, n\}$  a  $k = \{1, \dots, o\}$ .

Směrový úhel určíme podle vztahu  $\varphi_{j,k} = \arctan \frac{y_m(N_j) - y_m(O_k)}{x_m(N_j) - x_m(O_k)}$ .

2. Vybereme svazky vzdálené méně než  $\rho_{max}$  a dostaneme vektor vzdáleností  $\vec{\rho}$  a směrových úhlu  $\vec{\varphi}$ .

$\{\varphi_{j,k} \subseteq \vec{\varphi}, \rho_{j,k} \subseteq \vec{\rho} \mid \rho_{j,k} < \rho_{max}\}$  pro  $j = \{1, \dots, n\}$  a  $k = \{1, \dots, o\}$ ,

kde  $\vec{\varphi} = (\varphi_{j_1,k_1}, \dots, \varphi_{j_s,k_s}) = (\varphi_1, \dots, \varphi_s)$  a  $\vec{\rho} = (\rho_{j_1,k_1}, \dots, \rho_{j_s,k_s}) = (\rho_1, \dots, \rho_s)$ .

3. Definujeme funkce  $g(x) \rightarrow \begin{cases} 1, & |x| < \Delta\rho \\ 0, & jinak \end{cases}$ ,  $h(x) \rightarrow \begin{cases} 1, & |x| < \Delta\varphi \\ 0, & jinak \end{cases}$ .

Pro vektor  $\vec{x}$  délky  $n$  platí  $g(\vec{x}) = (g(x_1), \dots, g(x_n))$  a  $h(\vec{x}) = (h(x_1), \dots, h(x_n))$ .

Nechť  $a = \arg \max_{q=\{1, \dots, s\}} \sum_{i=1}^s g(\varphi_i - \varphi_q)$ ,  $b = \arg \max_{q=\{1, \dots, s\}} \sum_{i=1}^s h(\rho_i - \rho_q)$ , potom

$$\vec{v} = g(\vec{\varphi} - \varphi_a \cdot \vec{1}) + h(\vec{\rho} - \rho_b \cdot \vec{1}).$$

4. Naležeme množinu potenciálních korespondencí  $\mathcal{C}'$ .  $\{(R_t, O_{k_q}) \subseteq \mathcal{C}' \mid v_q > 1\}$ , pro  $q = \{1, \dots, s\}$ , kde  $R_t$  je referenční svazek třídy **5D** se stejným seznamem dopadajících faset jako měřený svazek  $N_{j_q}$ .
5. Pokud  $\mathcal{C}'$  obsahuje alespoň  $p_{min}$  prvků, přidáme  $\mathcal{C}'$  do množiny korespondencí  $\mathcal{C}$ .

### 2.6.1.4 Korespondence svazků podle polohy v obraze a zářivého toku

Tento algoritmus se snaží o to nalézt dvojici svazků, které se promítnou na podobnou pozici v obraze a mají vysoký zářivý tok. Snažili jsme se nalézt funkci, která by charakterizovala závislost mezi zářivým tokem  $\phi_{e_r}$  referenčních stop a zářivým tokem  $\phi_{e_m}$  měřených stop. Jednoduchou funkci jsme však nenašli. Ke korespondenci svazků budeme místo absolutního zářivého toku využívat jeho relativní velikosti vzhledem k ostatním svazkům v blízkém okolí.

#### Parametry algoritmu

- $\rho_{max}$  - maximální vzdálenost obrazu svazků třídy **3A** a **5D** v pixelech,
- $\Delta\rho$  - maximální absolutní odchylka úhlu v obraze mezi párem svazků třídy **3A** a **5D**,
- $\Delta\varphi$  - maximální absolutní odchylka vzdálenosti v obraze mezi párem svazků třídy **3A** a **5D**,
- $p_{min}$  - minimální počet nalezených dvojic.

#### Algoritmus

1.  $j = 1, \vec{w}_{m'} = \vec{1}, \vec{w}_{r'} = \vec{1}$ .
2. Nalezneme 3 svazky  $(M'_a, M'_b, M'_c) \subset \{\mathcal{M}' \setminus M'_j\}$  s nejmenší Euklidovou vzdáleností obrazu  $(d_a, d_b, d_c)$  od svazku  $M'_j$ .  

$$d_a = \sqrt{\left(x_m(M'_j) - x_m(M'_a)\right)^2 + \left(y_m(M'_j) - y_m(M'_a)\right)^2}.$$
3. Necht  $f(M'_x) \rightarrow \begin{cases} \phi_{e_m}(M'_x), & d_x < d_{max}, \phi_{e_m}(M'_j) > 1 \\ 1, & jinak \end{cases}$  potom  

$$\phi_{m'_{max}} = \max(\phi_{e_m}(M'_j), f(M'_a), f(M'_b), f(M'_c)).$$
4. Nastavíme váhy měřených svazků.  

$$w_{m'_q} = \frac{\phi_{max}}{f(M'_q)} \text{ pro } q = \{j, a, b, c\}.$$
5. Nalezneme uspořádanou  $s$ -tici  $\mathcal{S} = (S_1, \dots, S_s)$  referenčních svazků z  $\mathcal{R}'$ .  
 $\{R'_i \subseteq \mathcal{S} \mid d_{i,j} < d_{r_{max}}\}$  pro  $i = \{1, \dots, r'\}$ , kde  

$$d_{i,j} = \sqrt{\left(x_m(M'_j) - x_r(R'_i)\right)^2 + \left(y_m(M'_j) - y_r(R'_i)\right)^2}.$$
6.  $w_{r'_j} = \max(\phi_{e_r}(\mathcal{S}))$ .
7. Dokud  $j \neq n', j = j + 1$  a opakujeme body 2) až 6).
8. Necht  $g(x) \rightarrow \begin{cases} x, & x > 1 \\ 1, & jinak \end{cases}$ , potom můžeme určit kritériální funkci  

$$\mathbf{L}(i, j) = d_{i,j} \cdot w_{m'_j} \cdot g\left(\frac{\phi_{r_i}(\mathcal{R}'_i)}{w_{r'_j}}\right).$$
9. Potenciální korespondence  $\mathcal{C}'$  nalezneme podle vzájemně nejnížší velikosti kritéria v  $\mathbf{L}$ , které nepřekročí hodnotu  $L_{max}$ . Korespondující dvojice  $(R'_i, M'_j) \subseteq \mathcal{C}'$ , pokud  

$$i = \arg \min_{q=\{1, \dots, r'\}} \mathbf{L}(q, j), j = \arg \min_{q=\{1, \dots, m'\}} \mathbf{L}(i, q) \text{ a } \mathbf{L}(i, j) < L_{max}.$$
10. Vybereme korespondence z  $\mathcal{C}'$ , které můžeme přidat do množiny korespondencí  $\mathcal{C}$ .  

$$\{\mathcal{C}'_l \in \mathcal{C} \mid \mathcal{C}'_l \subseteq \{\mathcal{R}_f \cup \mathcal{M}_f\}\}.$$

### 2.6.1.5 Korespondence svazků podle ocásků

Ocásky svazků definujeme pomocí velikosti a směrového úhlu.

Počet ocásků referenčního svazku závisí na počtu stran polygonu, kterým popisujeme tvar svazku. Intenzitu ocásku v simulaci určuje zářivý tok svazku a na velikost hrany, na které ocásek vzniká.

Velikost  $\vec{\xi}_r(R_i)$  a směrový úhel  $\vec{\psi}_r(R_i)$  ocásků  $i$ -tého referenčního svazku  $R_i$  určuje vektor o délce  $n_i$ .

$\vec{\xi}_r(R_i) = (\xi_{r_1}(R_i), \dots, \xi_{r_{n_i}}(R_i))$ ,  $\vec{\psi}_r(R_i) = (\psi_{r_1}(R_i), \dots, \psi_{r_{n_i}}(R_i))$ , kde  $n_i$  je počet ocásků  $R_i$ .

Detekce ocásků měřeného svazku je popsána v kapitole 1.4.2. Obecně platí, že v obraze jsou detekovatelné ocásky, pro které byla v odpovídající simulaci vypočítána vysoká intenzita.

Velikost ocásků  $j$ -té měřené stopy značíme  $\vec{\rho}_m(M_j)$ , směrový úhel  $\vec{\varphi}_m(M_j)$ .

Podobnost ocásků hodnotíme podle kritéria, a to především na základě podobnosti směru ocásků. Velikost ocásků určuje váhu jednotlivých příspěvků.

#### Parametry algoritmu

- $\Delta\alpha_{max}$  - maximální absolutní odchylka azimutu měřeného svazku od referenčního,
- $\Delta\varepsilon_{max}$  - maximální absolutní odchylka elevace měřeného svazku od referenčního,
- $\sigma$  - citlivost kritériální funkce na úhlovou odchylku ocásků,
- $L_{min}$  - minimální velikost kritéria  $\mathbf{L} \in \mathbb{R}^{r \times m}$  pro přidání dvojice svazků do množiny korespondencí  $\mathcal{C}$ .

#### Algoritmus

1.  $i = 1$

2. Normujeme velikost referenčních ocásků  $\vec{\xi}_r(R_i) = \frac{\vec{\xi}_r(R_i)}{\max(\vec{\xi}_r(R_i))}$ .

3. Z referenčních ocásků vybereme pouze ty s dominantní intenzitou a získáme vektor velikost  $\vec{\rho}_r(R_i)$  a směr  $\vec{\varphi}_r(R_i)$  o délce  $n_k$ .  
 $\{\psi_{r_k}(R_i) \subseteq \vec{\varphi}_r(R_i), \xi_{r_k}(R_i) \subseteq \vec{\rho}_r(R_i) \mid \xi_{r_k}(R_i) > \xi_{min}\}$ , kde  $k = \{1, \dots, n_i\}$ .

4. Nalezneme uspořádanou  $n$ -tici  $\mathcal{N}$  z měřených svazků  $\mathcal{M}$ .  
 $\{M_j \subseteq \mathcal{N} \mid |\alpha_r(R_i) - \alpha_m(M_j)| < \Delta\alpha_{max}, |\varepsilon_r(R_i) - \varepsilon_m(M_j)| < \Delta\varepsilon_{max}\}$ , kde  $j = \{1, \dots, m\}$ .

5. Pokud  $M_j \subseteq \mathcal{N}$ ,

$$\mathbf{L}(i, j) = \sum_{k=1}^{n_k} \sum_{l=1}^{n_l} \frac{1}{\sqrt{2\pi} \sigma} \cdot e^{-\frac{(\varphi_{r_k}(R_i) - \varphi_{m_l}(M_j))^2}{2\sigma^2}} \cdot \sqrt{\rho_{r_k}(R_i) \rho_{m_l}(M_j)},$$

jinak  $\mathbf{L}(i, j) = 0$ , pro  $j = \{1, \dots, m\}$ , kde  $n_l$  je počet ocásků  $M_j$ .

6. Pokud  $i \neq r$ ,  $i = i + 1$  a opakujeme kroky 2) až 5).

7. Korespondující svazky nalezneme podle vzájemně nejvyššího kritéria v  $\mathbf{L}$  s minimální velikostí  $L_{min}$ . Korespondující dvojice  $(R_i, M_j) \subseteq \mathcal{C}$ , pokud  $i = \arg \max_{q=\{1, \dots, r\}} \mathbf{L}(q, j)$ ,

$$j = \arg \max_{q=\{1, \dots, m\}} \mathbf{L}(i, q) \text{ a } \mathbf{L}(i, j) > L_{min}.$$

### 2.6.2 Automatická optimalizace

Kámen je do měřicí soustavy umístěn ručně tak, aby se vertikální osa kamene nacházela v těžišti dopadajícího zdrojového svazku. Kámen je položen na spodku a laserový svazek dopadá na tabulku a boční fasety. Takto je zajištěna poloha kamene, ovšem rotace kamene okolo vertikální osy může být libovolná a musí být nalezena před optimalizací parametrů faset kamene.

Podle plochy, na kterou pokládáme kámen nemůžeme automaticky určit orientaci fasety BOT. Podstavec není pevně přichycen a mezi fasetou BOT a podstavcem je navíc reflexní vrstva. Pokud není reflexní vrstva nanесena rovnoměrně je kámen v měřicí soustavě nakloněn.

Náklon kamene určíme při hledání korespondencí svazků třídy **1A**.

## 2.7 Výsledky

### 2.7.1