



**TECHNICAL UNIVERSITY OF LIBEREC**  
**Faculty of Mechatronics, Informatics**  
**and Interdisciplinary Studies** ■

## **Manuál k softwaru GPU-DIC**

Ing. Petr Ječmen

Fakulta mechatroniky, informatiky a mezioborových studií  
Technická Univerzita v Liberci  
Studentská 2  
461 17 Liberec



# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Vybrané části algoritmu DIC</b>	<b>4</b>
2.1	Dělba obrazu na facetu . . . . .	4
2.2	Odhad deformace facetu . . . . .	4
2.3	Odhad pole přetvoření . . . . .	4
<b>3</b>	<b>Kroky výpočtu</b>	<b>6</b>
3.1	Vyznačení oblasti zájmu (ROI) . . . . .	6
3.2	Vyznačení reálné velikosti . . . . .	6
3.3	Velikost facetu . . . . .	6
3.4	Expertní nastavení . . . . .	6
<b>4</b>	<b>Ovládání aplikace</b>	<b>7</b>
4.1	Úvodní okno . . . . .	7
4.2	Výběr vstupu . . . . .	8
4.3	Zadávaní ROI . . . . .	9
4.4	Vyznačení reálné velikosti . . . . .	10
4.5	Expertní nastavení . . . . .	11
4.6	Spuštění výpočtu . . . . .	12
4.7	Uložení . . . . .	12
4.8	Výsledky . . . . .	12
<b>5</b>	<b>Doporučená nastavení</b>	<b>14</b>
<b>6</b>	<b>Parametry enginu</b>	<b>15</b>
<b>7</b>	<b>Konfigurační soubor úlohy</b>	<b>18</b>
<b>8</b>	<b>Ukázky konfiguračních souborů</b>	<b>19</b>



# 1 Úvod

Algoritmus digitální obrazové korelace (dále jen DIC) se historicky ukázal jako algoritmus vhodný pro analýzu záznamu mechanických dějů pro potřeby analýzy chování materiálu s co největší přesností. Nevýhodou algoritmu je citlivost na vstupní parametry a dlouhá výpočetní doba. Cílem aplikace GPU-DIC je hlavně zkrátit dobu výpočtu a alespoň částečný odhad vhodných vstupních parametrů pro potřeby analýzy konkrétního typu záznamů.

Popis jak funguje algoritmus DIC lze nalézt v "Two-dimensional digital image correlation for in-plane displacement and strain measurement: a review, Bing Pan et al; 2009 Meas. Sci. Technol. 23, <http://iopscience.iop.org/0957-0233/20/6/062001>." V tomto textu budou zmíněny pouze vybrané části, které byly vylepšeny nebo jejichž pochopením se uživateli ulehčí volba vstupních parametrů. V následující kapitole budou zmíněny pouze teoretické předpoklady pro úspěšnou volbu parametrů na základě teorie, detailnější doporučení týkající se volby parametrů budou pak uvedeny v kapitole 5.



## 2 Vybrané části algoritmu DIC

V této části textu bude odkazováno na text uvedený v minulé kapitole, text je přiložen k tomuto textu nebo je možné ho stáhnout z uvedené webové adresy. Odkazy budou ve formě "(DIC-5)", kde číslovka označuje číslo strany.

### 2.1 Dělbá obrazu na facetu

Prvním krokem výpočtu je rozdělení obrazu na oblasti, v případě algoritmu jsou nazývány "facetu" (ukázku lze nalézt v (DIC-5)). Výpočet probíhá nad jednotlivými facetu a lze tušit, že volba velikosti a umístění facetu je klíčová pro úspěch algoritmu. Menší velikosti umožní zachycení drobnějších detailů, zatímco větší pak nabízejí větší odolnost proti šumu. V praxi se běžně používají velikosti okolo 15 pixelů, pro zašuměná videa pak není problém používat velikost 20 pixelů a více. Pro další eliminaci vlivu šumu na výsledek používá implementovaný algoritmus většího počtu facetů, z kterých se pak na základě kvality výsledků vyberou ty nejlepší pro stanovení finálních výsledků.

### 2.2 Odhad deformace facetu

Výpočet pole posunů tkví v odhadu, jak se facet deformoval. Facet se deformuje dle tzv. tvarové funkce, která popisuje možné varianty posunu jednotlivých pixelů ve facetu. Detailní popis funkce lze nalézt v (DIC-5). Hlavním parametrem tvarové funkce je řád funkce - nultý řád popisuje pouze posuny, první řád pak je schopen popsat protažení nebo zkosení, druhý řád už pak dokáže popsat i nelineární deformace. Nesmíme zapomenout, že daná funkce popisuje facet jako celek, nikoliv pixel po pixelu. To je výhodou a zároveň nevýhodou celého algoritmu. Výhodou je podobnost s realitou, kde se testované vzorky deformují opravdu jako celky, ne jako malinkaté pod-části. Nevýhodou je, že algoritmus je schopen otestovat pouze omezený počet variant deformací facetu a pokud řešení není v množině testovaných řešení, algoritmus nám předloží nějaké jiné řešení a není lehké stanovit, jestli je vybrané řešení opravdu to pravé nebo jen nejlepší ze skupiny zadaných. Řešením je předat algoritmu dostatečně velkou množinu možných deformací, to ale velmi prodlužuje dobu výpočtu, takže je nutné najít nějaký kompromis na základě předpokládaných deformací.

### 2.3 Odhad pole přetvoření

Výstupem algoritmu DIC je pole posunutí jednotlivých bodů (typicky bod = pixel). Pro potřeby mechaniky by bylo dobré kromě samotným posunů znát i pole přetvoření (hodnoty epsilon). Hodnota přetvoření je dána poměrem koncové délky ku počáteční, v diskrétní oblasti lze tedy definovat jako rozdíl hodnot posunů v daném směru (prostá diferenciacie hodnot). Bohužel vlivem šumu nelze provést výpočet takto jednoduše. Existují dva přístupy, které dokáží potlačit vliv šumu na výsledné pole. Prvním přístupem je vyhlazení pole posunutí před vlastní diferenciací. Problém tohoto přístupu je volba vhodného vyhlazovacího algoritmu, který by nepotlačil užitečná data. Druhým přístupem a zároveň přístupem, který je implementován v aplikaci, je využití metody nejmenších čtverců. Podrobný popis lze nalézt v "Digital image correlation using iterative least squares and

pointwise least squares for displacement field and strain field measurements. Bing Pan, Anand Asundi, Huimin Xie, Jianxin Gao. 2008.” Principem metody je aproximace pole posunutí za pomoci lineární plochy (její předpis lze nalézt ve zmíněném článku) a hledání jejích koeficientů. Vzhledem k přímé závislosti koeficientů rovnice, hodnot posunutí a přetvoření, je možno koeficienty hledat za pomoci lineární algebry. Problém této metody je opět nutná volba parametru. Tím je velikost okolí, ve kterém se bude pole deformací aproximovat plochou. Pokud předpokládáme homogenní deformace, je možno volit velikost okna větší, protože se lépe potlačí vliv šumu na přesnost výsledku. V případě velkých variací v poli posunutí je nutné volit menší okno, aby byly zachyceny detaily, tím pádem ale může docházet ke zkreslení výsledků vlivem šumu. Typicky se velikost okna volí v rozsahu 10 - 30 pixelů.

## 3 Kroky výpočtu

V této kapitole budou zmíněny kroky, které by měl uživatel vykonat před spuštěním vlastního výpočtu. Nastavení lze provést buď za pomoci grafického rozhraní (viz kapitola 4) nebo lze parametry také nastavit za pomoci skriptu úlohy (viz kapitola 7).

### 3.1 Vyznačení oblasti zájmu (ROI)

Vzhledem k zaměření této aplikace na analýzu trhací zkoušky je i výpočet tomu uzpůsoben. Typický záznam obsahuje region, kde je zkoušený předmět uchycen do čelistí a zbytek obrazu je pro analýzu nezajímavý. Proto uživatel může buď vyznačit obdélníkovou oblast(-i), na jejichž ploše bude probíhat výpočet, nebo využije možnosti dynamické tvorby ROI na základě polohy čelistí. V případě dynamické tvorby je nutno vyznačit úvodní polohu čelistí za pomoci kružnic, program poté sleduje polohu čelistí a upravuje oblast zájmu tak, aby byla mezi nimi.

### 3.2 Vyznačení reálné velikosti

Tento krok je volitelný a slouží k možné unifikaci velikosti plochy pro odhad pole přetvoření. Aplikace používá poměr  $px / mm$ , který lze opět zadat dvěma způsoby. Buď ve formě čísla do skriptu úlohy nebo za pomoci GUI, kde se vyznačí známý rozměr, zadá se jeho reálná velikost a poměr se automaticky dopočítá.

### 3.3 Velikost facetu

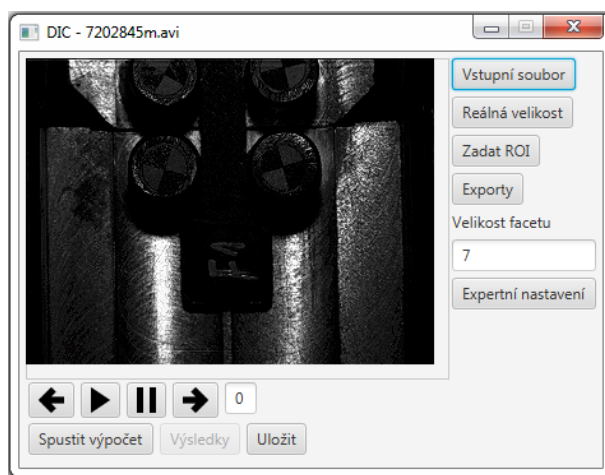
Velikost facetu je hlavním parametrem algoritmu a kvalita výsledky je na něm dost závislá. Doporučené hodnoty lze nalézt v kapitole 5. Číslo musí být celé a opět ho lze zadat přes GUI či skript.

### 3.4 Expertní nastavení

V některých speciálních případech může být žádoucí změnit i další parametry výpočetního enginu (jako limity deformací, limity kol výpočtu apod.). Co lze a nelze nastavit lze nalézt v kapitole 6, některá nastavení lze měnit v GUI, jiná se musí specifikovat za pomoci skriptu úlohy.

## 4 Ovládání aplikace

### 4.1 Úvodní okno



Hlavní okno programu

**Vstupní soubor** načte vstupní soubor(y), podporovány jsou videa (\*.avi), obrázky (\*.bmp, \*.jpg), konfigurace úlohy (\*.config), soubor projektu (\*.task) a skript pro více úloh (\*.scr)

**Reálná velikost** otevře okno pro vyznačení známého rozměru v obraze

**Zadat ROI** otevře okno pro vyznačení ROI

**Exporty** otevře okno pro specifikaci exportů provedených po dokončení výpočtu

**Velikost facetu** slouží pro zadání velikosti facetu použitou během výpočtu

**Expertní nastavení** slouží pro nastavení specifických parametrů enginu

**Šipky** slouží pro prohlížení obrázků

**Spustit výpočet** spustí výpočet

**Výsledky** otevře okno pro prezentaci výsledků (dostupné po dokončení výpočtu)

**Uložit** otevře dialog pro uložení skriptu úlohy dle konkrétního nastavení a pro uložení celého projektu (včetně výsledků)

## 4.2 Načtení vstupu

**Video** soubor musí mít koncovku avi, video se rozseká na obrázky za pomoci programu VirtualDub

**Obrázky** se řadí dle přirozeného třídění (tj. většinou dle abecedy)

**Config** soubor je textový soubor obsahující konfiguraci výpočtu, typicky po načtení může být rovnou spuštěn výpočet

**Task** soubor obsahuje kromě konfigurace úlohy také vypočtené výsledky, které lze okamžitě po načtení souboru prohlížet

**Scr** soubor by měl obsahovat seznam konfigurací, které se mají spočítat. Cesty ke konfiguracím musí být v absolutním tvaru (např. C:\temp\prvni.config), každá na samostatném řádku

Aplikace je schopna zobrazovat reálný čas děje, je ale nutné, aby ve složce, kde se nachází vstup, byl i "uda" soubor s informací o fps analyzovaného videa. Pokud nebyl automaticky vygenerován kamerou, je možno ho ručně vytvořit. Jedná se o prostý textový soubor, ve který by měl být řádek ve tvaru "Speed = 5000fps," kde číslo 5000 se nahradí počtem snímků za vteřinu daného vstupu.



## 4.3 Zadávání ROI

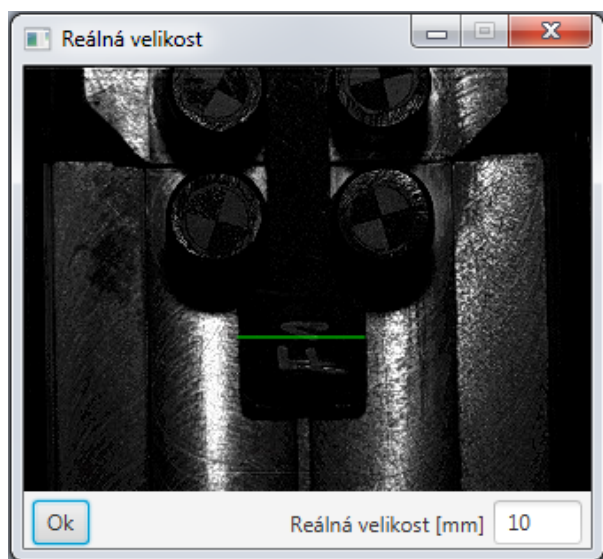


Okno pro vyznačení ROI

Zadáním ROI může uživatel enginu říci, které oblasti má počítat a tím pádem dost urychlit výpočet. Dostupné tvary ROI jsou obdélník a kruh. Na výběr jsou dvě možnosti vyznačení. První je statické umístění ROI (v téměř libovolném počtu). Tyto ROI se budou dodržovat po celou dobu výpočtu, případně je možno specifikovat na různá kola různá ROI s tím, že pokud nejsou uvedena jiná ROI, použijí se ROI z předchozího kola (takže pokud nás zajímá pouze jedna oblast během celého výpočtu, stačí ji označit pro první kolo).

Druhou možností je vyznačení polohy čelistí za pomoci 4 kruhových ROI (odtud pochází omezení na ne zcela libovolný počet statických ROI). V tomto případě se engine pokusí automaticky sledovat čelisti a podle toho upravuje sledovanou oblast. Sledovaná oblast může být v tomto případě definována dvěma způsoby - buď ji uživatel vyznačí sám za pomoci obdélníkové oblasti nebo vyznačení nechá na enginu. Engine vytvoří oblast dle jednoduchých pravidel - oblast je mezi čelistmi, začíná nad horními čelistmi a končí pod spodními čelistmi s tím, že se nechává mezera mezi čelisti a sledovanou oblastí pro alespoň částečné potlačení nepřesnosti vyznačení čelistí a tím pádem ovlivnění výpočtu.

## 4.4 Vyznačení reálné velikosti

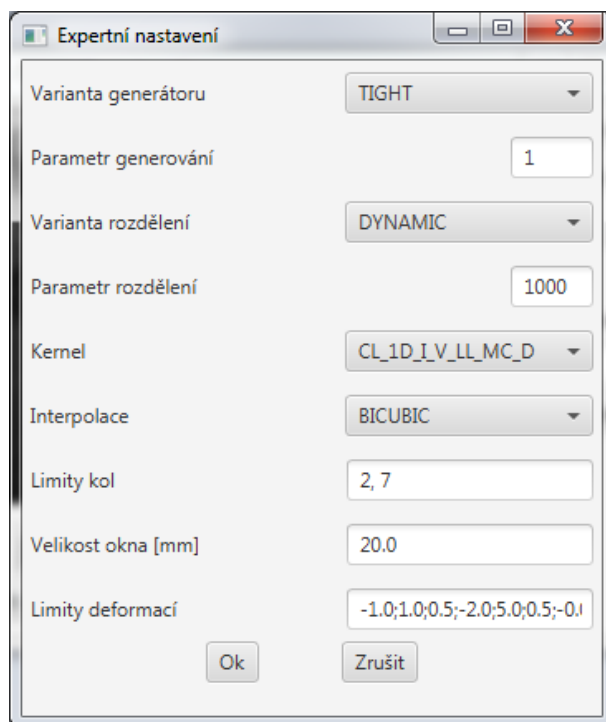


Okno pro vyznačení reálné velikosti

Konfigurace úlohy z hlediska fyzického uspořádání se může měnit, pro porovnávání výsledků by bylo dobré mít možnost zadávat parametry pro engine v reálných jednotkách. Pixely nejsou tolik vhodné, protože jeho rozměr se může vlivem změny vzdálenosti kamery a vzorku lišit. Proto je k dispozici okno pro vyznačení známého rozměru v obraze spolu se zadáním jeho reálné délky, což umožní spočítat převodní koeficient  $\text{px} / \text{mm}$ , který se využívá pro stanovení velikosti okna při výpočtu pole přetvoření. Tím lze zajistit stejné vlastnosti algoritmu (hlavně vyhlazení / vynechání detailů) pro různá uspořádání.

Proces vyznačení je prostý, uživatel vyznačí známý horizontální rozměr za pomoci kliknutí levého tlačítka myši a tažením a vyplní jeho délku v milimetrech. Program ulehčuje práci tím, že umožňuje vyznačit pouze horizontální čáru, takže uživatel nemusí řešit, jestli je čára skutečně rovná. Pokud uživatel chce nakreslit čáru znovu, stačí pouze znovu vyznačit za pomoci levého tlačítka, stará linie se automaticky smaže.

## 4.5 Expertní nastavení



Expertní nastavení

V tomto okně lze nastavit detailní parametry engine, většinu z nich není v běžném případě nutné nastavovat. V této sekci nebudou jednotlivé volby popisovány, pouze bude uveden jejich systémový název, který lze dohledat v kapitole 6, kde je uveden i jejich podrobný popis.

**Varianta generátoru** FACET\_GENERATOR\_METHOD

**Parametr generování** FACET\_GENERATOR\_PARAM

**Varianta rozdělení** TASK\_SPLIT\_METHOD

**Parametr rozdělení** TASK\_SPLIT\_PARAM

**Kernel** KERNEL

**Interpolace** INTERPOLATION

**Limity kol** ROUND\_LIMITS

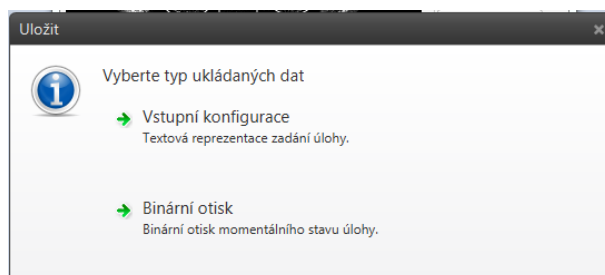
**Velikost okna** STRAIN\_ESTIMATION\_PARAM

**Limity deformací** DEFORMATION\_LIMITS

## 4.6 Spuštění výpočtu

Spuštění výpočtu neobnáší nic zvláštního, stačí pouze kliknout na tlačítko. Nenas-tavené parametry enginu (prakticky pouze hodnoty z expertního nastavení) se nastaví na výchozí hodnoty, které by měly být vhodné pro drtivou většinu záznamů trhacích zkoušek (konkrétní hodnoty lze nalézt v kapitole 6).

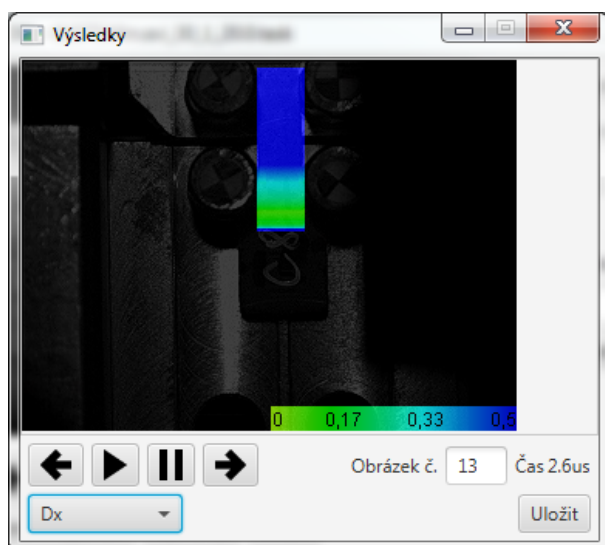
## 4.7 Uložení



Uložení projektu

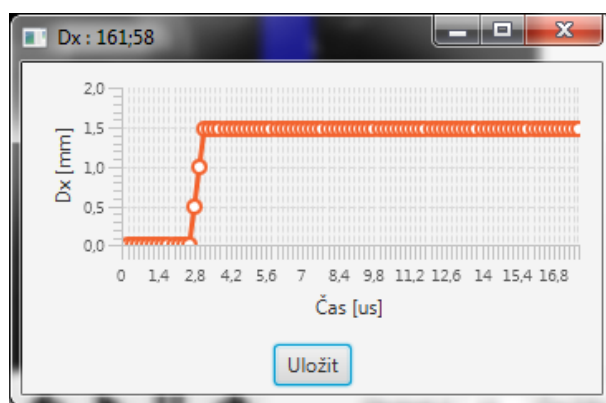
Úlohu lze uložit dvěma způsoby - jako textovou konfiguraci či binární otisk. Textová konfigurace se hodí v případě, že chcete opakovat výpočet nebo spustit více výpočtů po sobě za pomoci skriptu. Uložený soubor obsahuje všechny nastavené parametry výpočtu, takže po jeho načtení se může rovnou spouštět výpočet. Binární otisk úlohy také uloží nastavení úlohy, uloží také vypočtené výsledky, takže je možno je prohlížet i později (binární otisk se automaticky ukládá vždy po dokončení výpočtu).

## 4.8 Výsledky



Okno s výsledky

Okno výsledků prezentuje mapy s vypočtenými výsledky. Jsou dostupné jak kumulativní výsledky, tak diferenciální (ty mají před názvem písmeno "d"). Kromě prohlížení výsledků je také možno výsledky exportovat. K dispozici je buď export zobrazené mapy nebo sekvence map. Mapy lze exportovat buď jako obrázek (obdobu obrázku v okně) nebo jako seznam hodnot do souboru CSV. Kromě exportu po jednotlivých mapách lze také vytvořit animaci, která vytvoří video soubor prezentující zvolené výsledky jako video.



Výsledky pro bod

Kliknutím do obrázku je možno zobrazit vývoj vybrané hodnoty v daném bodě. Vývoj hodnoty je reprezentován grafem. Typ zobrazené hodnoty se automaticky aktualizuje vzhledem k vybrané hodnotě v hlavním okně s výsledky. Tlačítkem "Uložit" je možné uložit zobrazené hodnoty do CSV souboru (jméno souboru se generuje automaticky ze souřadnic bodu, typu dat a nastavení úlohy). Je možno mít otevřeno několik oken s různými body najednou.

## 5 Doporučená nastavení

**Velikost facetu** Vzhledem k šumu ve videu je vhodné volit velikost facetu minimálně 20, příliš vysoké hodnoty ale mohou způsobit ignorování detailů, takže není vhodné překročit hodnotu cca 35. Je také nutné dbát na to, aby generovaný facet nebyl větší než sledovaná oblast, v tom případě by skoro určitě docházelo ke zkreslení výsledků vlivem zahrnutí pozadí do sledování.

**Velikost okna** Tento parametr by měl nejlépe korespondovat se zvolenou velikostí facetu, případně může být větší. Hodnota menší než velikost facetu nemá příliš smysl, protože detaily jsou ignorovány díky zvolené velikosti facetu a promítnou se do pole posunutí, ze kterého se počítá pole přetvoření. Je nutné ale dát pozor na to, že tato velikost se zadává v milimetrech, zatímco velikost facetu je v pixelech.

**ROI** Nevhodnější je použít variantu dynamického výpočtu, tj. vyznačit pozice čelistí. Algoritmus pak na základě polohy čelistí vytvoří oblast materiálu a také se postará o posouvání oblastí dle detekovaného pohybu čelistí.

**Limity kol** Algoritmus vynechá výpočet deformací v případě, že detekuje nulový pohyb čelistí. Není tedy nezbytně nutné limity kol definovat, mohou se hodit v případě, že nás zajímá pouze konkrétní část videa, ne celý děj.

## 6 Parametry enginu

V této kapitole budou uvedeny parametry enginu, které lze měnit (ať už za pomoci GUI či konfiguračního souboru) spolu s popisem, jaké hodnoty lze použít. Za názvem parametru bude v hranaté závorce uvedena stručná charakteristika povolené hodnoty (int - celé číslo, double - reálné číslo, výčet povolených hodnot pro některé parametry, výchozí hodnota pak bude zvýrazněna **tučně**).

### **FACET\_GENERATOR\_METHOD** [CLASSIC — **TIGHT**]

Způsob generování facetu. CLASSIC používá klasický režim generování, kde se facetu pokládají vedle sebe s malým přesahem. Tento režim je vhodný pro deformace typu posunutí (tento režim se používá pro sledování posunu čelistí), v případě složitějších deformací nemusí být výsledné pole posunů homogenní. TIGHT klade facetu přes sebe s malou mezerou, což vylepší homogenitu výsledného pole a lépe odolá šumu, daní je ale mnohem vyšší počet facetů a tím pádem i mnohem delší výpočet. Ve výchozím stavu se používá TIGHT režim.

### **FACET\_GENERATOR\_PARAM** [int]

Parametr generování úzce souvisí se způsobem generování. Pro CLASSIC režim tento parametr určuje kolik pixelů z facetů se bude překrývat. V TIGHT režimu parametr určuje o kolik se posune okraj následujícího facetu oproti předchozímu. Výchozí hodnotou je 1. Je nutné dát pozor na jistá omezení hodnot, které vycházejí z principu generování facetů. V CLASSIC režimu má parametr smysl pro hodnoty maximálně do velikosti facetu, poté už se nové facetu umísťují špatně nebo výsledný počet facetu je nekonečný. V případě TIGHT režimu musí být hodnota kladné číslo, nulová hodnota by způsobila nekonečně mnoho facetů, záporné hodnoty by způsobily generování facetů na špatné pozici.

### **FACET\_SIZE** [int]

Tento parametr je velikost facetu. Doporučené hodnoty lze nalézt v kapitole 5.

### **KERNEL** [CL\_2D\_I — CL\_1D\_I\_V\_LL\_MC — **CL\_1D\_I\_V\_LL\_MC\_D**]

Typ OpenCL kernelu, který bude použit pro výpočet na GPU. Výchozí a doporučená hodnota je CL\_1D\_I\_V\_LL\_MC\_D, ostatní jsou vhodné pouze pro testovací režim nebo některé speciální případy. Pro kernely jiné než výchozí není garantována stabilita programu.

### **TASK\_SPLIT\_METHOD** [NONE — STATIC — **DYNAMIC**]

Vzhledem k omezenému množství paměti na GPU je někdy nutné rozdělit úlohu na menší části. Varianta NONE zakáže rozdělování úlohy. To lze úspěšně použít pouze pro malé úlohy, v ostatních případech výpočet spadne. STATIC umožní definovat, po kolika facetech se má úloha rozdělit. Tato varianta se může hodit v případě, kdy nechcete plně vytěžovat paměť GPU. Poslední varianta DYNAMIC se pokusí co možná nejvíce zaplnit paměť GPU daty, což je z hlediska délky výpočtu nejefektivnější, protože se minimalizují přenosy dat z / do GPU a díky tomu to je také výchozí hodnota.

## **TASK\_SPLIT\_PARAM** [int]

NONE a DYNAMIC rozdělování úlohy parametr nevyžadují, u varianty STATIC je nutné stanovit počet facetů tvořících jednu pod-úlohu. Výchozí hodnota je 1000, celková velikost úlohy ale také dost závisí na počtu testovaných deformací, takže nelze stanovit nějaké univerzální pravidlo, vesměs je nutno postupovat způsobem pokus-omyl, dokud nedosáhneme požadované paměťové náročnosti.

## **INTERPOLATION** [BILINEAR — BICUBIC]

Po deformaci facetu jsou souřadnice bodů facetu obecně reálná čísla, je tedy nutné nějakým způsobem zasadit bod zpět do rastru obrazu a získat jeho barvu. K tomu se používá proces interpolace barvy. Volba BILINEAR používá lineární interpolaci v ploše, zatímco BICUBIC používá kubickou funkci pro dopočet výsledné barvy. BICUBIC varianta by měla produkovat kvalitnější výsledky, je ale také náročnější na výpočet.

## **ROUND\_LIMITS** [int ; int]

Limity kol je možné stanovit k urychlení výpočtu (např. pro vynechání části záznamu, kde již došlo k přetržení vzorku). Hodnoty se zadávají ve formě indexů prvního a posledního snímku (číslováno od 0), čísla se oddělují středníkem.

## **RESULT\_QUALITY** [double]

Deformace se odhaduje na základě míry korelace barev pixelů původního a deformovaného facetu. Čím vyšší je tato hodnota, tím pravděpodobněji se jedná o správný odhad. Maximální hodnotu je pak 1.0, což znamená, že se vstup a výstup se naprosto shodují. Vlivem šumu a diskretizací prostoru možných deformací se ale maximální hodnoty prakticky nikdy nedosáhne, pouze se jí můžeme přiblížit. Často máme pro jeden pixel několik možných výsledků, což umožní částečně eliminovat vliv šumu na výsledek. Tento parametr říká, jak moc kvalitní musí být výsledek, aby byl přijat do skupiny výsledných. Výchozí hodnota je 0.5, což zaručuje vynechání nekvalitních výsledků při zachování dostatečného počtu výsledků pro úspěšný výsledek.

## **STRAIN\_ESTIMATION\_PARAM** [double]

Tento parametr určuje velikost okna při výpočtu pole přetvoření. Tento parametr se zadává v mm, engine si pak velikost v pixelech dopočítá sám za pomoci parametru MM\_TO\_PX\_RATIO.

## **MM\_TO\_PX\_RATIO** [double]

Koeficient pro přepočet rozměru okna přetvoření z milimetrů na pixely. Jedná se o hodnotu, která říká, kolik pixelů v obraze je jeden milimetr v reálném světě.

## **DEFORMATION\_LIMITS** [double ; double ; double...]

Limity deformací určují, jaké deformace se v obraze předpokládají a tím pádem jaké deformace se budou testovat. Hodnoty se pro každý koeficient zadávají ve tvaru MIN, MAX, KROK, koeficienty mohou být dva (testovat se budou pouze posuny), 6 (kromě posunutí také prodloužení a smyk) nebo 12 (přidají se nelineární deformace). Tím pádem se musí zadat 6, 18 nebo 36 hodnot oddělených středníkem. Matematické využití koeficientů lze nalézt v příložené literatuře. Jako výchozí hodnota se používají deformace prvního řádu s následujícím zadáním : -1; 1; 0.5; -2;



5; 0.5; -0.0; 1.0; 0.1; -0.5; 0.5; 0.1; -0.5; 0.5; 0.1; -0.0; 1.0; 0.1. V překladi jde o to, že posuny se testují po 0.5 pixelech, v ose X se testují pouze malé posuny. Deformace prvního řádu pak testují prodloužení od žádného (0) do prodloužení na dvojnásobek (1) jak v ose X, tak Y, a smyk pak záporný i kladný v obou osách (-0.5 až 0.5). Vliv koeficientů na deformaci lze vyzkoušet v přiloženém nástroji *DIC\_DeformationGenerator*.



## 7 Konfigurační soubor úlohy

Konfigurační soubor v běžném případě není nutné psát ručně, lze ho vygenerovat v aplikaci po nastavení všeho potřebného, ukázkový soubor lze nalézt v kapitole 7. Soubor je textový, každý parametr je uveden na samostatném řádku. Parametr se skládá z identifikátoru (název velkými písmeny), oddělovače (ten je tvořen dvojnásobnou dvojtečkou - "::") a hodnotou parametru.

Povinným parametrem je vstupní soubor. Tím může být buď cesta k video-souboru (cesta musí být absolutní) nebo seznam obrázků ke zpracování, názvy pak musí být odděleny středníkem. Název parametru pro vstupní soubor je pak "INPUT."

Dále je možno stanovit ROI pro jednotlivá kola. Identifikátor je "ROI\_" a číslo obrázku (např. ROI\_0 pro první obrázek). Hodnotou je pak výčet ROI pro dané kolo oddělených dvojitými středníky. ROI se definuje jako posloupnost hodnot (souřadnice ROI – limity deformací – velikost facetu) oddělených dvojitou pomlčkou ("–"). Souřadnice ROI mohou být buď souřadnice středu a poloměr (pro kruhový ROI) nebo souřadnice levého horního a pravého spodního bodu (obdélníkový ROI, v pořadí X1;Y1;X2;Y2), jednotlivé hodnoty jsou od sebe odděleny středníkem. Limity deformací se specifikují ve stejném tvaru jako globální limity (viz. kapitola 6), případně lze uvést slovo "NONE", které specifikuje, že se mají použít limity globální. Obdobně se chová velikost facetu, celočíselná hodnota určí velikost, NONE vynutí použití globálního nastavení.

Dále lze v souboru nastavit i parametry enginu. Identifikátor parametru se tvoří jako klíčové slovo "PARAM\_" následované názvem parametru (např. PARAM\_FACET\_SIZE), hodnota se pak musí řídit možnostmi hodnot daného parametru (opět viz. kapitola 6). Všechny neuvedené hodnoty budou nahrazeny výchozí hodnotou - nemusí se tedy nutně specifikovat každý parametr, pouze ty, které chceme změnit.

V ukázkách lze nalézt dva soubory. První je dalo by se říci minimalistický - obsahuje minimální počet parametrů pro úspěšný výpočet. Vstupem je video, ROI jsou čtyři kružnice označující pozici čelistí (což zajistí inteligentní posouvání jak ROI čelistí, tak automatické vytvoření ROI vzorku) a velikost facetu. Je nutné dát pozor na zápis ROI, který se bohužel nevejde na jeden řádek a je tudíž zalomen. V reálném souboru je ale nutné dodržet, aby ROI pro dané kolo byly všechny napsány na jednom řádku (čili v ukázce se jedná o jeden dlouhý řádek rozdělený na několik).

Druhá ukázka pak představuje plnou konfiguraci, ukazuje i definici ROI pro jednotlivá kola - druhé kolo má obdélníkový ROI používající specifikované limity deformací (pouze posuny) při velikosti facetu 25, zatímco ve třetím kole se použijí výchozí hodnoty jak pro limity, tak pro velikost facetu.

## 8 Ukázky konfiguračních souborů

Jednoduchý konfigurační soubor pro automatický výpočet

```
INPUT :: D:\temp\7202845m\7202845m.avi
ROI_0 :: 110.0;88.0;24.20--NONE--NONE;; 203.0;9.0;26.24--NONE--NONE;;
        203.0;87.0;28.28--NONE--NONE;; 109.0;12.0;26.24--NONE--NONE
PARAM_FACET_SIZE :: 20
```

Plný konfigurační soubor pro automatický výpočet

```
INPUT :: D:\temp\7202845m\7202845m.avi
ROI_2 :: 100.0;100.0;150.0;150.0--2.0,,2.0,,0.1,,5.0,,10.0,,0.1--25
ROI_3 :: 109.0;109.0;159.0;159.0--NONE--NONE
PARAM_FACET_GENERATOR_METHOD :: TIGHT
PARAM_FACET_GENERATOR_PARAM :: 1
PARAM_FACET_SIZE :: 20
PARAM_KERNEL :: CL_1D_I_V_LL_MC_D
PARAM_TASK_SPLIT_METHOD :: DYNAMIC
PARAM_INTERPOLATION :: BICUBIC
PARAM_ROUND_LIMITS :: 2,,4
PARAM_DEFORMATION_LIMITS :: -1.0,,1.0,,0.1,,2.0,,5.0,,0.1
```