PRIDOBIVANJE MERITEV IZ POSNETKA

POVZETEK

Obdelava posnetkov je potekala v več fazah. Najprej posnetke izvozimo v urejevalnik posnetkov, kjer jih obrežemo po času in obrežemo tudi odvečne robove.

Nato v Mathematici z uporabo 2 barvnih filtrov iz posnetkov poiščemo piksle na katerih se nahaja flumastrova oznaka.

S podobnimi trikotniki in upoštevanjem loma svetlobe v steklu in vodi lahko določimo, kje glede na kamero se v prostoru nahajajo najdeni piksli. Iz znane lege pikslov črte v prostoru lahko dobimo kako se premika središče črta, pa tudi, kako posodica niha v smeri, pravokotni na os snemanja. S krepko pisavo so označene meritve.

V UREJEVALNIKU POSNETKOV

Uporabljali smo urejevalnik Premiere Pro, bi pa lahko to naredili tudi v Davinci Resolvu <u>(alternativa za nepirate)</u>.

00:1	11:3	39:1	98	j ,				📇 c 🗸	[Windows10] 218.015.528 k of 498.655.228 k free			1	
🔆 n 🕸		2		્ય		.16 00:25:58:108 00:26:28:101 00:26:58:094 00:27:28:087 00:27:58:080 00.	👻 🗲 🔪 🗖	▼ ◀ \Documents\DRUGI PROGRAMI\Wolfram Mathematica\saška ura\posnetk ▶					
								Name		Ext	Size	↓ Date	Attr
								1 []			<dir></dir>	03.04.2022 14:21	
								📄 [posl	kusna]		<dir></dir>	03.04.2022 00:06	
	6		8	0				opor 📄	nbe	txt		02.04.2022 22:17	
								🔺 🕹 🛯 1					
			8					🔒 81 - 2					
	6		8	0				🔺 🕹 81 - 3					
	-							- 🛓 81 - 4					
						⊎		🔺 🕹 11 - 2					
	6							🔺 🕹 11 - 2					
								🔺 11 - 2					
								🔔 11 - 2					
								🔺 🕹 11 - 4					
								📥 3.3m					
			81			<u>v</u>		🛓 📥 4.9m					
_								🛓 📥 4.9 m					
								📥 4.9m					

V Premiere Pro najprej izbrane posnetke "drag&dropamo" na časovnico.

Tam jih označimo in z desnim klikom odpremo meni. Izberemo Unlink. Nato lahko označimo in odstranimo zvok.

Posnetke nato spet označimo in v istem meniju izberemo Speed/Duration. Odpre se naslednji meni:



Izberemo Ripple...Clips In v rubriki speed vpišemo 60/fps * 100. V našem primeru je fps = **240**. Na časovnici lahko s pritiskom tipk Ctrl+K poljubni posnetek pri nekem času ločimo na 2. (po časovnici se premikamo z miško, puščicami na tipkovnici in Alt+koleščkom). Na ta način odstranimo odvečne dele posnetkov pred in po spuščanju posodic. Ko to storimo, izberemo poljubni posnetek ter na začetku pritisnemo I, pred predzadnjo sliko pa O.



Nato pritisnemo Ctrl + M. V meniju, ki se odpre, gremo na zavihek Source, ter pritisnemo na ikono pod tem. Zdaj lahko obrežemo posnetek:



Na desni moramo pri Width vpisati: resolucija_x – Left – Right. Za Height naredimo podobno. V našem primeru je resolucija **(1080 × 1920)**, Left = **10** Top = **693**. Te podatke bomo še rabili pri nadaljnem računanju. Na desni še preverimo, da so izbrane nastavitve Format=H.264, Output Name= (želeno ime in pot izhodne datoteke), Preset=Custom, Frame Rate = 60, Aspect= Square Pixels (1.0), Use Maximum Render Quality. Pritisnemo Export in počakamo. Za ostale posnetke bodo nastavitve ostale enake kot smo jih izbrali, zato od vseh teh korakov za ostale posnetke samo spremenimo ime izhodne datoteke in izvozimo.

PREGLEDOVANO OBMOČJE

Če bi v vsakem koraku pregledovali celo sliko, bi program deloval zelo počasi, in našel veliko napačnih pikslov. To bi bilo zelo nestabilno. Program je zato narejen tako, da na vsaki sliki pregledujemo smo manjše pravokotno območje. Za 1. sliko to območje določimo ročno, tako, da v spremenljivko ops1 (območje pregledovanja slike 1) v tem vrstnem redu vnesemo podatke (vrs->vrs,sto->sto). To je med katerima vrsticama in stolpcema slike pregledujemo.

Za sledeče slike bodo pregledovana obomočja enake velikosti, bodo pa centrirana na težiša (povprečja) najdenihi pikslov iz prejšnjih dveh korakov (središče iz prejšnjega koraka premaknemo za razliko središč prejšnjega in predprejšnjega koraka). Če kak del območja pregledovanja gleda čez robove slike, ta del iz območja pregledovanja izločimo.

2 FILTRA

Testirali smo okoli 10 različnih enačb za izbiro, ali je piksel del oznake s flumastrom, ali ni. Na koncu se je za najboljšo (najzanesljivejšo) izkazala kombinacija 2 zelo enostavnih filtrov. S 1. filtrom odstranimo ozadje (komplement posodice). Za barvo ozadja vzamemo povprečje levega roba pregledovanega območja. Potem za vsak piksel iz pregledovanega območja pogledamo, ali je piksel dovolj podoben ozadju. To lahko učinkovto naredimo z Normalize[RGB_{ozadje}].Normalize[RGB_{piksla}]. Če je to dovolj blizu 1, vzamemo piksel za piksel iz ozadja. Prednost skalarnega produkta je, da v vektorizirani obliki s tem Mathematica precej hitro računa. Normirane vzamemo, saj se svetlost ozadja rahlo spreminja in s tem izločimo vpliv svetlosti. Ko imamo izločeno ozadje, smo na boljšem za naslednji filter, ker ostane posodica, ki je precej svetla. Pri naslednjem filtru vzamemo za piksle flumastra, tiste, ki so glede na ostale piksle dovolj temni (Norm@RGB). Program je narejen tako, da strogosti obeh filtrov na začetku vsakega posnetha določimo z 2D sliderjem <u>(slovenska ustreznica??)</u>. Ko pomikamo sliderja sproti vidimo, kaj je rezultat 1. in kaj 2. filtra. Filtra je ponavadi dobro dati na kar strogo (tako, da pobere kakih 80% pikslov na črti). Če sta preveč na ohlapno, lahko hitro začneta najdevati napačne piksle, kar narobe prestavi območje pregledovanja in vseskupaj podivja.



ZORNI KOT KAMERE

Zorni kot kamere lahko določimo tako, da iz zaporednih razdalj, razmaknjenih za ΔY=**0.8cm**, posnamemo meter.


Izmeriti moramo naslednje razdalje:

 $-Y_z$ (zrak) med goriščem kamere in akvarijem

- Y_s (steklo)

- Y_v (voda) med steklom in sprednjim (bližnjim) delom spuščane posodice

IZ PIKSLOV V KOORDINATE [m] NAJDENIH PIKSLOV

Zamislimo si da je kamera v izhodišču in snema v smeri y. Namišljena slika, ki jo snema kamera naj se nahaja na razdalji 1 od nje. Slika je tako visoka k=1.606 in široka v razmerju resolucije. Recimo, da smo našli piksel, ki je označen s flumastrom. Če vemo, kateri je ta piksel, njegovo lego na namišljeni sliki v prostoru dobimo z razmerjem resolucije in stranic namišljene slike. Naj bo ta lega **r**. potem se lahko vprašamo, od kod je ta žarek (tega piksla) prišel v kamero? To ugotovimo tako, da žarek namišljeno pošljemo nazaj. Steklo zadane v točki: $\mathbf{r_s} = Y_s \mathbf{r}$. Zadene ga pod vpadnim kotom α =ArcTan $\left[\sqrt{r_x^2 + r_z^2}\right]$. Lomni kot β določimo z lomnim zakonom. Smer nadaljnega potovanja žarka dobimo tako, da z ustrezno totacijsko matriko vektor **j** zavrtimo okoli $\mathbf{r} \times \mathbf{j}$ za kot - β . Za poiskati to matriko ima Mathematica ugrajeno funkcijo RotationMatrix. Za drugi lom je postopek skoraj enak. Na ta način naposled ugotovimo, kje v prostoru leži najdeni piksel.

GIBANJE SREDIN NAJDENIH PIKSLOV IN STRANSKO NIHANJE

Za precej zanesljivo se izkaže, če sledimo sredini najdenih pikslov. Za sredino vzamemo kar povprečje leg teh pikslov v prostoru. Dobljene rezultate shranimo v h5 datoteke. x komponenta gibanja je zelo majhna in zato precej zašumljen, z komponenta pa je najdena zelo zanesljivo. Nihanje vzrtan dobimo tako, da najdenim pikslom prilagodimo premico s koeficintom *k*. Nagibi posodic so potem ArcTan[k]. Tudi to shranimo v h5 datoteke.