

Aplicație pentru stabilizare video

Autor: Croitoru Robert, 331AB

An: III

1.Introducere

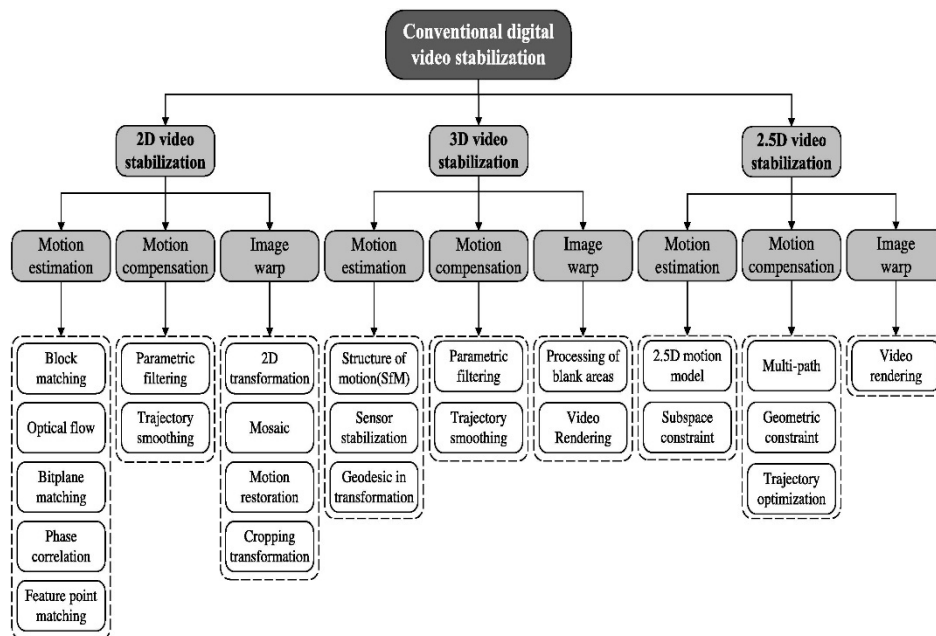
a. Context

Datorită creșterii accelerate a tehnologiei, tot mai multe dispozitive sunt echipate cu diferite metode de captură a imaginii, ducând astfel la o creștere a conținutului video digital, fapt ce implică o nevoie reală pentru dezvoltarea de tehnici video-audio de procesare din ce în ce mai sofisticate.

O astfel de creștere a numărului de dispozitive video duce în unele cazuri la videoclipuri instabile afectate de tremur și neclaritate, afectând experiența de vizionare[1].

Drept soluție la problema enunțată anterior se folosesc o serie de tehnici de stabilizare a cadrelor, menite să îmbunătățească calitatea vizuală a videoclipului prin reducerea tremurului, rezultat din corectarea mișcărilor camerei.

În funcție de modelul de mișcare utilizat, tehnicile de stabilizare video pot fi clasificate în metode 2D, 3D sau hibride, fiecare incluzând etape de estimare, compensare și transformare a mișcării[2].



b. Obiective

Scopurile principale ale acestei lucrări sunt reprezentate de crearea unui pipeline complet de stabilizare video bazat pe [3] :

- detecția trasăturilor imaginilor
- estimarea mișcării dintre cadre
- corectarea traiectoriei cadrelor

Totodată, aplicația urmărește implementarea unei interfețe grafice ce are rolul de a-i oferi utilizatorului un mod simplu și accesibil de:

- a încărca videoclipuri
- a vizualiza conținutul original
- a vizualiza conținutul transformat
- a stabili video clipul inițial încărcat
- a salva clipul anterior transformat

c. Motivația

Intenția pentru alegerea unei asemenea tematici provine dintr-o dorință de însușire a unor cunoștințe teoretice de prelucrare a imaginilor digitale, cât și a celor video. Mai mult, realizarea proiectului a oferit o oportunitate de extindere a cunoștințelor practice prin intermediul folosirii diferitelor biblioteci și limbaje de programare precum ar fi: Python, OpenCV, Numpy, PyQt5 și Scipy.

2. Prezentarea suportului tehnic

a. Detaliere

Proiectul se fundamentează pe diferite tehnici de procesare digitală a imaginilor. Principalul algoritm folosit este Lucas-Kanade ce are rolul de a estima mișcarea camerei prin intermediul fluxului optic, fiind folosit pentru urmărirea punctelor caracteristice. Fluxul optic este definit ca fiind o aproximare 2d a campului de mișcare obținut prin măsurarea deplasării luminozității de la un cadru la altul[4].

Pentru determinarea vectorilor de mișcare se rezolvă pentru fiecare punct de interes sistemul: [5]

$$\begin{bmatrix} \sum_w I_x(l,k)^2 & \sum_w I_x(l,k)I_y(l,k) \\ \sum_w I_x(l,k)I_y(l,k) & \sum_w I_y(l,k)^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -\sum_w I_x(l,k)I_t(l,k) \\ -\sum_w I_y(l,k)I_t(l,k) \end{bmatrix}$$

Unde I_x , I_y și I_t reprezintă derivata parțială în direcția Ox , I_y în direcția Oy și I_t derivata parțială în timp, de la un cadru la altul.

Detectarea punctelor caracteristice se realizează prin intermediul algoritmului Shi-Tomasi de detecție a colțurilor. Colțurile sunt definite ca acele zone din imagine unde o ușoară deplasare într-o direcție (fie pe axa verticală sau orizontală) produce variații mari de intensitate. Algoritmul de detecție poate fi disecat în 3 pași:

- determinarea ferestrelor care produc variații mari de luminozitate
- pentru fiecare fereastră se determină un scor :
 $R = \min(\lambda_1, \lambda_2)$, unde λ_i sunt valorile singulare ale matricei M [6],
 f_x , f_y reprezentând gradientii după direcția orizontală, respectiv după cea verticală
- dintre toate ferestrele se aleg cele care depășesc un prag specificat

$$M = \begin{bmatrix} f_x^2 & f_x f_y \\ f_x f_y & f_y^2 \end{bmatrix}$$

Pentru determinarea modelului de transformare între cadre se presupune că pentru a ajunge de la cadrul P la cadrul $P + 1$ s-a folosit o transformare afină. Transformarea afină reprezintă o înmulțire de matrice, urmată de o aditie vectorială[7]

Cu ajutorul acestei transformări se pot exprima:

- rotația
- translația
- scalarea
- forfecarea (shear distortion)

Pentru determinarea estimării transformării dintre 2 cadre succesive sunt necesare cel puțin 3 perechi de puncte.

Pentru selecția de puncte în cadrul estimării transformării dintre cadre se folosește RANSAC(Random sample consensus). Aceasta este o metodă iterativă de estimare a unui model matematic ce conține date aberante. Algoritmul RANSAC funcționează astfel:

- selectează un subgrup de date
- găsește un model pentru subgrup
- cuantifică numărul de date aberante
- se repetă procedura de mai multe ori[8]

După ce se obțin transformările se poate estima deplasarea camerei.

Pentru filtrarea deplasării s-a ales un filtru gaussian unidimensional.

Având forma(acesta este cazul bidimensional)[9] :

$$g_{\sigma}[m,n] = \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2}\left(\frac{m^2+n^2}{\sigma^2}\right)}$$

Iar rezultatul ieșirii se obține dintr-o convoluție cu vectorul respectiv.

Formă bidimensională a ieșirii[9]:

$$y[i,j] = \sum_{m=-M}^M \sum_{n=-M}^M g_{\sigma}[m,n] x[i-m,j-n],$$

b. Proiecte similare

Deshake Filter din Ffmpeg

Puncte tari:

- Gratuit
- Open-source
- Ușor de integrat în procese

Puncte slabe:

- Nu are interfață grafică

Adobe Premier Pro

Puncte tari:

- Algoritm avansat de detecție și analiză a mișcării
- Configurare avansată
- Rezultate cinematice

Puncte slabe:

- Cost ridicat(este necesară cumpărarea unui abonament de la ADOBE)
- Necesită hardware performant
- Este nevoie de specializare în folosirea acestei unelte

Davinci Resolve

Puncte tari:

- Oferă multiple moduri de stabilizare(Perspective, Similarity, Translation)
- Permite stabilizare pe bază de giroscop

Puncte slabe:

- Greu de învățat
- Consum intensiv de resurse

3. Prezentarea tehnică a etapei de implementare

a. Biblioteci si tehnologii

- Python este un limbaj de programare interpretat, orientat pe obiecte, care oferă suport pentru module, excepții și clase. Acesta permite dezvoltarea folosind mai multe paradigme de programare, printre care se regăsesc cea procedurală și cea orientată pe obiecte[10].
- NumPy este o bibliotecă open-source destinată calculului științific, care oferă suport pentru obiecte precum vectori și matrice multidimensionale, alături de o gamă variată de funcții eficiente pentru operarea pe aceste structuri[11].
- OpenCV este o bibliotecă open-source care include o gamă largă de algoritmi dedicați pentru procesarea imaginii și pentru computer vision[12].
- PyQt5 este o bibliotecă utilizată pentru dezvoltarea interfețelor grafice cu utilizatorul (GUI), bazată pe Qt Toolkit[13].
- SciPy este o bibliotecă utilizată în calcule științifice, care include algoritmi pentru optimizare, integrare, interpolare, precum și pentru rezolvarea ecuațiilor diferențiale și algebrice[14].

b. Algoritmi

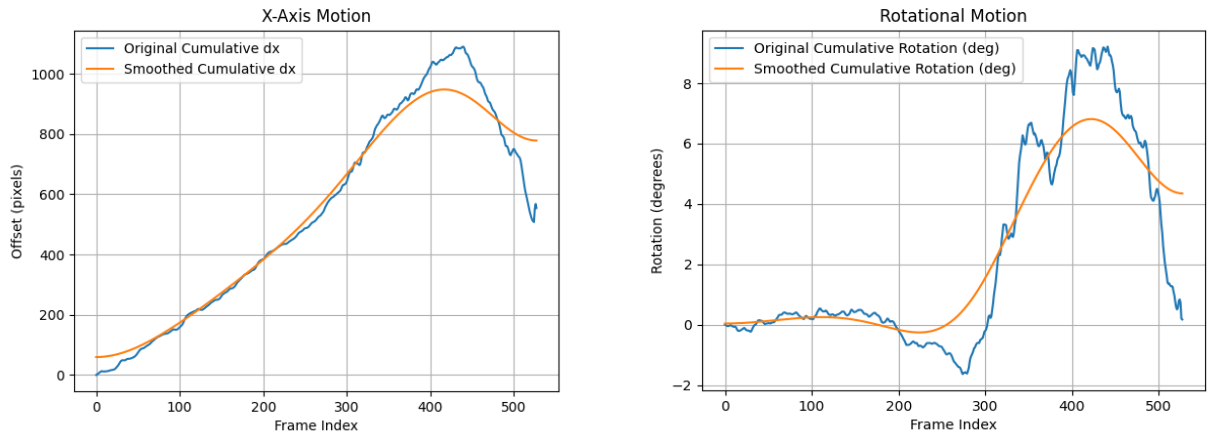
- Pentru detecția colțurilor s-a folosit funcția goodFeaturesToTrack din Open-cv cu următorii parametrii:
 - maxCorners=200; numărul maxim de colțuri ce vor fi detectate
 - qualityLevel=0.1; metrică ce determină calitatea unui colț
 - minDistance=30; distanța minimă între 2 puncte detectate
 - blockSize=3; dimensiunea ferestrei folosită în calculul derivatelor

- Pentru urmărirea colțurilor s-a utilizat funcția `calcOpticalFlowPyrLK` cu următorii parametrii:
 - `winSize=(20, 20)`; reprezintă dimensiunea zonei de căutare la fiecare nivel al piramidei
 - `maxLevel=3`; reprezintă numărul de elemente din piramidă; imaginea se scalează progresiv pentru detecția mișcărilor dintre cadre, diferitele rezoluții regăsindu-se în această piramidă
 - `criteria=(cv.TERM_CRITERIA_EPS|cv.TERM_CRITERIA_COUNT,10, 0.03)`; `TERM_CRITERIA_EPS` reprezintă condiția de terminare când poziția punctului nu variază mult între cadre
`TERM_CRITERIA_COUNT` indică numărul maxim de iterații
- Pentru determinarea transformării dintre 2 cadre successive s-a folosit `estimateAffinePartial2D`, având drept parametrii:
 - `method=cv.RANSAC`; ce are rolul de eliminare / ignorare a punctelor aberante
 - `ransacReprojThreshold=5.0`; indică eroarea maximă acceptată ca un punct să fie considerat valid
- Pentru transformarea unui cadru s-a folosit funcția `warpPerspective` cu următorii parametrii:
 - `flags=cv.INTER_LINEAR`; folosește interpolarea pentru determinarea valorii pixelului de la noua coordonată
 - `borderMode=cv.BORDER_CONSTANT`; în urma transformării există zone neacoperite și acest mod are rolul de umplere cu o culoare
- Pentru netezirea caracteristicilor astfel încât mișcarea nouă să elimine variațiile rapide de mișcare și rotație s-a folosit funcția `gaussian_filter1d`

În cadrul aplicației cadrele au fost stocate individual sub forma de matrice `numpy.ndarray`, având dimensiuni specifice (înălțime, lungime, 3), 3 reprezentând numărul de canale. Cele 3 canale sunt BGR, ordinea aceasta este o alegere de implementare a bibliotecii `open-cv`.

Cadrele în contextul unui videoclip au fost stocate într-o listă de `python` pentru a avea acces facil la cadre.

Metodologia de dezvoltare a aplicației a fost una iterativă. Un prim pas a fost încărcarea corectă a videoclipurilor și afișarea acestora. După aceea, a fost implementată opțiunea de urmărire și detecție a punctelor de interes. În continuare a urmat partea de aplicare a transformărilor inverse menite să stabilizeze clipul. Într-un final a fost realizată și partea de interacțiune cu utilizatorul, implementarea unei interfețe grafice.

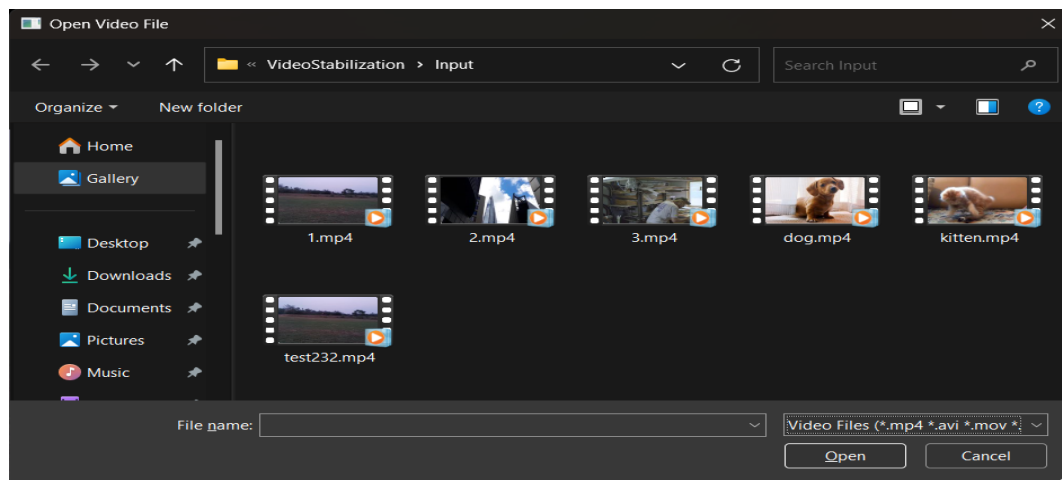


Aceste grafice reprezintă variația traiectoriei camerei estimată pe axa ox, rotația cumulată (cele cu albastru), respectiv valorile netezite folosite în cadrul operației de stabilizare(cele cu portocaliu).

4. Prezentarea modulului de utilizare

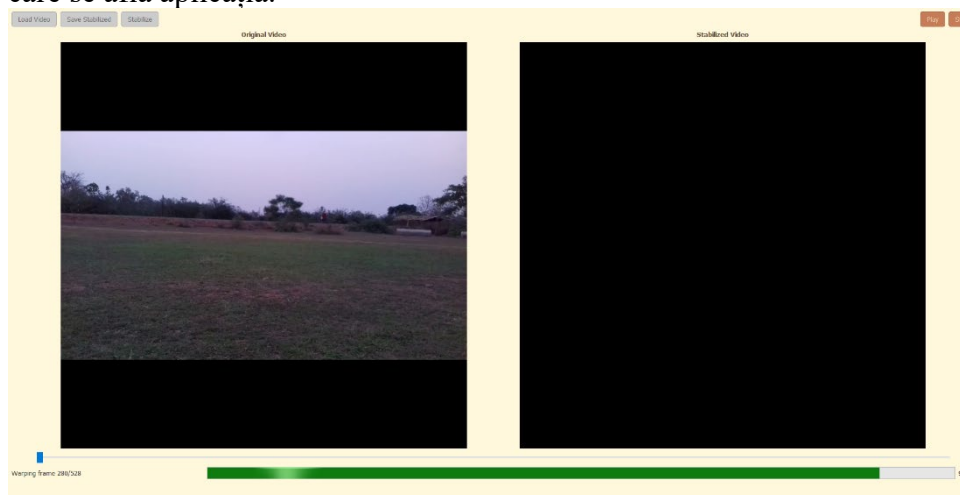
a. Meniul principal

În momentul deschiderii aplicației, utilizatorul îi sunt prezentate o serie de butoane: load, save stabilized, stabilize și 2 zone de reprezentare a videoclipurilor. Pentru o funcționare adecvată utilizatorul trebuie să apese pe butonul de load unde îi este afișat



o fereastră de dialog ce îi permite să aleagă un videoclip cu următoarele extensii:mp4, avi, mov, mkv, extensii compatibile cu open-cv.

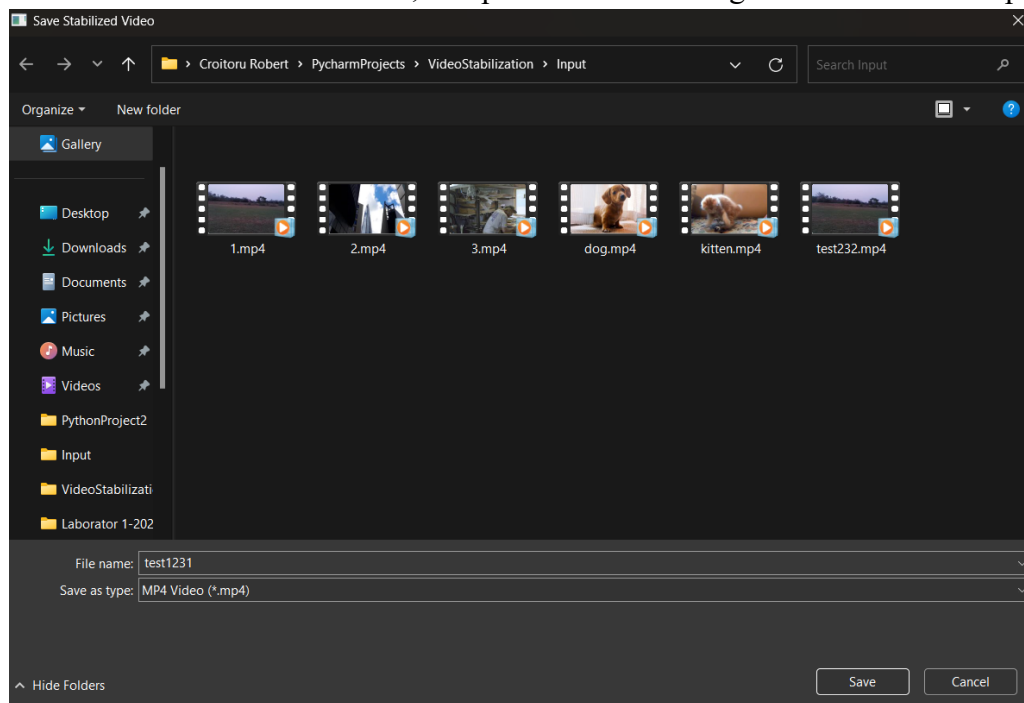
Odată încărcat videoclipul, utilizatorului i se deblochează posibilitatea de a porni sau de a opri rularea clipului, o bară de glisare ce-i permite să deruleze videoclipul inițial încărcat. Totodată, acesta are posibilitatea să înceapă procesul de stabilizare prin apăsarea butonului stabilize. În momentul procesului de stabilizare, utilizatorul poate să interacționeze în continuare cu interfața. În acest moment pe ecran este afișată o bară de progres ce afișează progresul, având drept rol informarea despre pasul de procesare în care se află aplicația.



După momentul stabilizării videoclipului, se pot vizualiza în paralel atât clipul original, cât și cel stabilizat. Atât butoanele de play și de stop, cât și bara sunt sincronizate cu ambele videoclipuri, astfel se pot derula ambele clipuri în paralel și se pot observa diferențele dintre cadre.



După acest moment de vizualizare se activează și ultimul buton de salvare a videoclipului, permițând salvarea videoclipului stabilizat într-un director ales, cu un nume introdus de la tastatură, cu posibilitatea de alegere a unui format presetat.



5.Concluzii

Obiectivele principale ale aplicației au fost realizate cu succes, dintre care amintim realizarea unui pipeline de stabilizare și oferirea unei interfețe intuitive pentru a fi folosită de către utilizator.

Utilitatea acestui proiect rezultă din faptul că se furnizează o soluție ușor de folosit pentru reducerea de zgomot a mișcării dintre cadre. Aplicația poate fi folosită atât de persoane non-tehnice ce își doresc o metodă simplă de stabilizare a videoclipurilor, cât și de persoanele ce doresc să învețe diferitele tehnici utilizate în cadrul dezvoltării aplicației.

Un avantaj al acestui proiect față de alte aplicații îl reprezintă faptul că este open-source și intuitiv de folosit.

Totodată, aplicația are un orizont infinit de dezvoltare și îmbunătățire, dintre care amintim:

- Folosirea unor biblioteci sau tehnici menite să delege efortul de calcul plăci video
- Integrarea unor metode de decupare manuală sau automată
- Integrarea unor funcționalități de prelucrare a mai multor videoclipuri în paralel
- Dezvoltarea unei metode de afișare a punctelor detectate de la cadru la cadru
- Implementarea unui grad mai mare de parametrizare; posibilitatea de a alege metoda de detecție, urmărire și netezire direct dintr-un meniu ușor accesibil.

6.Bibliografie

- [1]: https://www.researchgate.net/publication/346166370_Video_stabilization_Overview_challenges_and_perspectives
- [2]: <https://www.sciencedirect.com/science/article/pii/S092523122201270X>
- [3]: G. Balachandran and J. V. G. Krishnan, "A Review of Video Stabilization Algorithms," 2022 International Conference on Augmented Intelligence and Sustainable Systems (ICAISS), Trichy, India, 2022, pp. 643, figure (1), doi: 10.1109/ICAISS55157.2022.10011015.
- [4]: https://visionbook.mit.edu/optical_flow.html#d-motion-field-and-optical-flow
- [5]: <https://ieeexplore.ieee.org/document/8623648> figure(3)
- [6]: https://www.researchgate.net/publication/369670824_Lucas-Kanade_Optical_Flow_Machine_Learning_Implementations
- [7]: https://docs.opencv.org/4.x/d4/d61/tutorial_warp_affine.html
- [8] <https://www.mathworks.com/discovery/ransac.html>
- [9]: <https://www.mdpi.com/2076-3417/14/11/4664> figure(1,2)
- [10]: <https://docs.python.org/3/faq/general.html#what-is-python>
- [11]: https://numpy.org/doc/stable/user/absolute_beginners.html
- [12]: <https://docs.opencv.org/4.11.0/d1/dfb/intro.html>
- [13]: <https://www.pythonguis.com/pyqt5-tutorial/>
- [14]: <https://scipy.org/>