Universitatea POLITEHNICA din București
Facultatea de Electronică, Telecomunicații și Tehnologia Informației

Analiza criminalistică a fluxurilor video folosind tiparele de zgomot recurent

Proiect de diplomă

Prezentată ca cerință parțială pentru obținerea
titlului de *Inginer*
în domeniul *Calculatoare şi Tehnologia Informaţiei*
programul de studii *Ingineria Informaţiei*

Conducător științific
Horia Cucu
Francesco De Natale

Absolvent
Aron Latiș

Anul 2021

Universitatea "Politehnica" din Bucureşti
Facultatea de Electronică, Telecomunicaţii şi Tehnologia Informaţiei
Program de studiu **INF**

Anexa 1

TEMA PROIECTULUI DE DIPLOMĂ

a studentului **LATIŞ V. Aron , 444A**

1. Titlul temei: Analiza criminalistică a fluxurilor video folosind tiparele de zgomot recurent

2. Descrierea temei și a contribuției personale a studentului (în afara părții de documentare):
Lucrarea va avea ca scop identificarea aparatului sursă pentru secvențe video și trebuie menționat că este o lucrare de cercetare. Pentru acest lucru vom exploata urma de PRNU (Photo-Responsive Non-Uniformity) prezentă în orice imagine și care până acum a oferit rezultate coerente în aplicații de acest gen.

Lucrarea va fi structurată în mai multe etape pentru clarificarea diverselor obiective pe care ni le-am propus. Prima dintre acestea va fi utilizarea urmei de PRNU pe imagini în scopul de a înțelege mai bine modul de funcționare și conceptele care stau la baza acestui instrument, cum ar fi: maximum likelihood estimate (MLE), normalized cross-correlation (NCC), peak energy correlation (PCE) și testele statistice utilizate în diversele aplicații care includ PRNU (asociere de amprente PRNU ale camerelor, identificarea aparatelor, descoperirea istoriei de procesare, etc). Următorul pas ar fi să extindem aplicarea algoritmilor de PRNU în secvențe video, mai exact în frame-urile individuale din care se formează un video, folosind astfel ceea ce este cunoscut în literatura de domeniu drept procesare intra-frame. Scopul acestui pas este a analiza cât de mult poate fi crescută acuratețea și eficiența atât prin alegerea frame-urilor pe care aplicăm PRNU cât și prin încercarea găsirii unei metode de a aplica PRNU folosind procesare inter-frame, exploatând astfel și informația abundentă care se găsește în domeniul timp.

Metodele de care am vorbit ulterior vor fi aplicate pe date aparținând unor aparate din setul de date VISION. Prima oară vom lucra pe aparate 'nestabilizate' – care implică faptul că imaginile (frame-urile individuale) nu au fost pre-procesate în niciun fel și sunt prezentate în forma lor inițială – urmând că ulterior să lucrăm pe aparate 'stabilizate' – care implică faptul că aparatele au aplicat o pre-procesare asupra imaginii inițiale-, pentru a înțelege ce diferențe vom obține din punct de vedere al acurateții și rezultatelor între cele două tipuri. Ca o paranteză, când ne adresăm problemei de aparate 'stabilizate'/'nestabilizate' ne referim la posibilele pre-procesări 'in-camera' aplicate de aparat în momentul în care este filmat un video, pentru a compensa pentru diverse turbulențe și imperfecțiuni. Un al 3-lea rezultat care trebuie analizat este acela în care se folosesc doar imagini, doar videouri și videouri combinate cu imagini pentru identificarea aparatului sursă.

Ultima etapa reprezinta modificarea unui algoritm "state-of-the-art" deja existent de calcul a amprentei PRNU ale video-urilor, prin combinarea acestuia cu alte idei personale si extrase din alte lucrari stiintifice deja existente, pentru a putea observa daca acuratetea categorisirii creste in urma schimbarilor facute. Mai exact, algoritmul actual lucreaza exclusive pe I-frames, neglijand informatia oferita de P-frames si B-frames. Astfel, utilizand o metoda de ponderare a frame-urilor B si P putem extrage informatie suplimentara care sa ne ajute la un calcul mai eficient decat celui in care se calculeaza amprenta pe fiecare I/P/B frame individual. Acesta este scopul final. In mod optional, daca studentului ii mai ramane timp, se va incerca si aplicarea unei idei de genul Wiener filtering in timp.

Tot proiectul va fi realizat în python. Contribuția studentului va fi aceea de a scrie algoritmii, de a realiza o analiza comparativa intre rezultatele obtinute in urma diverselor implementari si a concluziona in final daca ideile noi aduc imbunatatiri sau raman doar la stadiul de incercari.

3. Discipline necesare pt. proiect:
prelucrarea imaginilor , recunoasterea formelor si inteligenta artificiala

4. Data înregistrării temei: 2020-11-27 12:57:37

Conducător(i) lucrare,
Conf. dr. ing. Horia CUCU

Student,
LATIŞ V. Aron

Prof. Francesco De Natale, Universta di Trento

Director departament,
Ș.L. dr. ing Bogdan FLOREA

Decan,
Prof. dr. ing. Mihnea UDREA

Cod Validare: **2104a68886**

Declaraţie de onestitate academică

Prin prezenta declar că lucrarea cu titlul *Analiza criminalistică a fluxurilor video folosind tiparele de zgomot recurent*, prezentată în cadrul Facultăţii de Electronică, Telecomunicaţii şi Tehnologia Informaţiei a Universităţii "Politehnica" din Bucureşti ca cerinţă parţială pentru obţinerea titlului de *Inginer* în domeniul Calculatoare şi Tehnologia Informaţiei, programul de studii *Ingineria Informaţiei* este scrisă de mine şi nu a mai fost prezentată niciodată la o facultate sau instituţie de învăţământ superior din ţară sau străinătate. Declar că toate sursele utilizate, inclusiv cele de pe Internet, sunt indicate în lucrare, ca referinţe bibliografice. Fragmentele de text din alte surse, reproduse exact, chiar şi în traducere proprie din altă limbă, sunt scrise între ghilimele şi fac referinţă la sursă. Reformularea în cuvinte proprii a textelor scrise de către alţi autori face referinţă la sursă. Înţeleg că plagiatul constituie infracţiune şi se sancţionează conform legilor în vigoare. Declar că toate rezultatele simulărilor, experimentelor şi măsurătorilor pe care le prezint ca fiind făcute de mine, precum şi metodele prin care au fost obţinute, sunt reale şi provin din respectivele simulări, experimente şi măsurători. Înţeleg că falsificarea datelor şi rezultatelor constituie fraudă şi se sancţionează conform regulamentelor în vigoare.

Bucureşti, Iulie 2021.
Absolvent: Aron Latiș

.....

Cuprins

Listă de figuri

Listă de tabele

Mențiuni

Trebuie să mulțumesc în mod explicit lui Andrea Montibeller, doctorand în cadrul facultății din Trento - UniTN -, pentru ajutorul oferit de-a lungul realizării acestui proiect și pentru toate materialele și explicațiile primite de la dumnealui în cursul acestui an universitar.

Lista acronimelor

CFA = Color Filter Array
CN = Corelație normalizată
CPS = Cadre Pe Secundă
DARPA = Defense Advanced Reserch Projects Agency
JPEG = Joint Photographic Experts Group
FPS = Frames Per Second
LP = Linear Pattern
MDS = Mediere în domeniul spațial
ML = Maximum-Likelihood
MMEP = Minimul Erorii Erorii Patratice
MMSE = Minimum Mean Squared Error
NCC = Normalized Cross-Correlation
PCE = Peak to Correlation Energy
PM = Posibilitate maximă
PRNU = Photo Responsive Non-Uniformity
RPNU = Răspuns fotografic non-uniform
SDA = Spatial Domain Average
VEC = Vârf față de Energia Corelației
VFC = Vector Filtru de Culoare

Introducere

Am avut ocazia completării unui curs de procesare fundamentală a imaginilor și a semnalelor video înaintea studiului acestora și în cadrul facultății din care fac parte. În combinație cu un curs de fundamente ale Machine Learning-ului, acestea mi-au format bazele unei pasiuni pentru o nișă specifică Data Science, mai exact analiza obiectelor de tip media. Prin urmare, aceasta a fost motivația care a dus la acceptarea acestei teme în momentul în care mi-a fost propusă.

Scopul acestei lucrări este de a îmbunătăți un algoritm de analiză a semnalelor video stabilizate și nestabilizate propus în 2019. Astfel, pot afirma că gradul de noutate este unul destul de ridicat întrucât pornim având la bază o lucrare scrisă cu numai doi ani în urmă.

Obiectivele lucrării sunt de a înțelege treptat bazele conceptului de răspuns foto neuniform (Photo Responsive Non-Uniformity - PRNU), de a le aplica în moduri noi și dezvolta în scopul îmbunătățirii algoritmului menționat mai sus. Astfel, pornim de la imagini statice simple și le analizăm amprenta de zgomot, observând performanța rezultatelor și înțelegând modul de funcționare. Urmează să analizăm reziduul PRNU și în cazul de transformări geometrice de genul translație, scalare și distorsiune, utilizând algoritmi specifici pentru fiecare dintre acestea. După ce putem spune că stăpânim aceste concepte, facem trecerea la semnale video testând în mod sumar algoritmul de bază pe care vrem să îl modificăm.

Contribuția studentului consta atât în înțelegerea conceptelor prezentate mai sus, cât și de propunerea unor noi posibilități de abordare a problemei prezentate. Au fost încercate în jur de 5 metode diferite pentru a eficientiza implementarea deja existentă și acceptată în lumea științifică a analizei criminalistice. Drept rezultate palpabile, din acele câteva încercări, numai două au fost într-adevăr promițătoare, oferind rezultate relevante într-un timp de execuție mai mic decât cel al algoritmului original. Cu toate acestea, o analiză riguorasă nu a putut fi făcută din cauza limitării de putere de procesare. Întrucât lucrăm cu videoclipuri de calitate mare, până și cea mai rapidă metodă dura în jur de 90 de minute, pentru un singur video. Dacă am fi testat corect pe toate exemplele prezente în setul de date VISION (în jur de 3000), am fi ajuns la o durata de 4500 de ore, pentru a testa o singură amprenta a unui dispozitiv. Din acest motiv, ne-am rezumat la a testa 36 de videoclipuri pe 10 amprente și am făcut o analiză amănunțită a factorilor prezenți în acestea.

De asemenea, vom menționa atât scopul fiecărui capitol, cât și dacă acesta conține contribuții ale studentului:

- Capitolul 1 - Analiză criminalistică: scopul său este de a îl introduce pe cititor în subiect, la un nivel foarte înalt, prezentând generalități despre nișa în care se situează lucrarea. Acesta este strict teoretic, contribuția studentului fiind aceea de a fi înțeles conținutul său.

- Capitolul 2 - Concepte de bază: între [2.1] și [2.5] scopul său este de a oferi treptat cititorului informațiile fundamentale urmăririi ideilor implementate de către student în [2.6], dar și a rezultatelor prezentate în [4]. Acesta conține atât o parte teoretică asupra căruia studentul nu își asumă vreo contribuție dincolo de înțelegerea acesteia, cât și o prezentare unor idei inovative utilizate în vederea obținerii unor algoritmi mai eficienți în [2.6], care au fost și implementate.

- Capitolul 3 - Modele matematice: între [3.1] și [3.4.1] scopul său este de a oferi informații

despre partea matematică a conceptelor prezentate în capitolul anterior. Acesta conține atât o parte teoretică asupra căruia studentul nu își asumă vreo contribuție dincolo de înțelegerea ei, cât și o propunere alternativă pentru luarea deciziei finale în privința identificării dispozitivului, în [3.4.2].

- Capitolul 4 - Rezultate experimentale: între [4.1] și [4.2] scopul său este acela de a prezenta seturile de date pe care s-au realizat teste. Restul de subcapitole prezintă analiza realizată de student asupra rezultatelor obținute în urmă execuției testelor pe codurile scrise de mâna (toate în afară de cel din [4.3]). Contribuția studentului constă atât în realizarea implementărilor în limbajul Python, cât și în analiza și interpretarea rezultatelor obținute.

Cu toate că va fi menționat și de-a lungul lucrării, vom preciza încă de la început faptul că punctul de plecare a fost un cod obținut printr-o colaborarea cu universitatea UniTN din Trento. Testele din [4.3] s-au realizat pe baza acelui program pe care l-am modificat pentru a schimba o funcție. Restul de cod este scris de către student, având drept surse de inspirație lucrările și unele dintre codurile asociate acestora.

Capitolul 1
Analiză criminalistică

1.1 Date generale

A devenit deja un truism a spune că tehnologia avansează repede. Drept consolidare a afirmației anterioare, legea lui Moore spune că odată la 18 luni, viteza de procesare a calculatoarelor se dublează. Cu toate că pe termen lung acest principiu nu va reuși să își mențină corectitudinea, în momentul actual se dovedește a fi adevărat. Această evoluție a vitezei de procesare aduce cu sine o multilateralitate în modurile de utilizare a acestei noi capacități. O ramură puternic afectată de aceste noi posibilități computationale este cea a imaginilor și videoclipurilor de rezoluție înalta, care necesită un hardware puternic pentru a fi procesate.

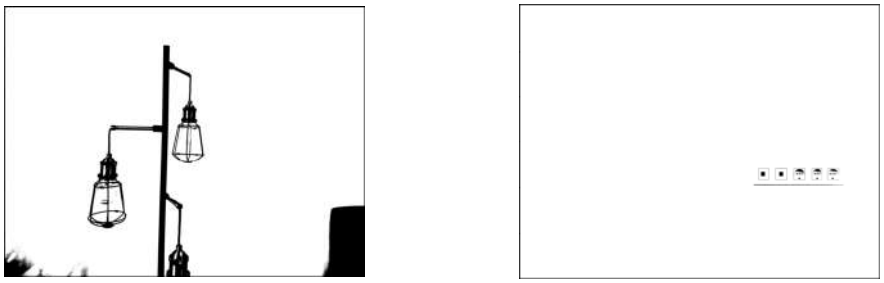
Această lucrare se va axa anume pe tehnicile care generează și manipulează tot ce ține de câmpul multimedia, sau mai exact, pe combaterea utilizări acestor tehnici în moduri ilegale. Numai pentru a oferi un pic de context al problemei, voi prezenta exemplifica unele moduri imorale în care pot fi utilizate aceste noi abilități ale calculatoarelor.

- modificarea pozelor și a videoclipurilor in defavoarea unei personalități publice
- postarea unor poze sau videoclipuri cu conținut ilegal pe diverse platforme sub aripa anonimității oferite de internet
- utilizarea tehnologiei "deepfake" pentru a uzurpa identitatea cuiva

Acest lucru nu este totuși o problema nouă. Manipularea imaginilor se realizează continuu de la începuturile fotografiei digitale, iar software-uri puternice cum ar fi Photoshop sau chiar Paint pentru modificari mai putin complexe, oferă moduri simple și convețională de a modifica imaginile, la îndemână tuturor. În următoarele figuri ofer niște exemple:

(a) Imagine originală (b) Imaginea modificată

Figura 1.1: Imagine făcută cu telefonul personal și modificată în Paint



(a) Obiecte prezente inițial (b) Obiecte adăugate

Figura 1.2: Conținutul imaginii anterioare, pe părți

O agenție a Departamentului Apărării al Statelor Unite, DARPA, a lansat în anul 2016 un program care se axează pe metode de analiză criminalistică a obiectelor media, cu scopul de a testa integritatea pozelor si videoclipurilor de pe internet. În urma acestei mișcări, au fost create diverse abordări, fiecare exploatând aspecte diferite, așa cum vom vedea în capitolele următoare.

În general, o imagine poate să fie caracterizată prin unele atribute definite arbitrar de către un analist criminalist (1.2.3.). În momentul în care orice fel de post-procesare este aplicată asupra imaginii, aceste trăsături nu mai rămân constante, urmând astfel să declanșeze un semnal de alarmă în privința obiectului multimedia respectiv.

În ciuda tuturor eforturilor de a crea un instrument de încredere, care lucreaza Împreună cu utilizatorul uman și care să ateste autenticitatea și integritatea unor imagini și videoclipuri, evoluția rețelelor neuronale adânci redefinesc standardele de eficiență setate precedent. Chiar și așa, acestea nu sunt incluse în zona noastră de interes și au fost menționate numai pentru a sublinia posibilitatea utilizării lor și în acest domeniu.

1.2 Metode ale analizei criminalistice

În această secțiune voi acorda atenție exclusiv metodelor dezvoltate înaintea renumitelor rețele neuronale utilizate actual, care încă mai pot fi îmbunătățite, oferind astfel rezultate satisfăcătoare. O proprietate definitorie a abordării problemei în aceste metode este existența cunoștinței a-priori pe care se bazează.

1.2.1 Metode "oarbe"

Primul tip de abordare este unul "orb", în care nu se folosește niciun fel de informație a-priori, ci doar se bazează pe conținutul din cadrul obiectului multimedia care este analizat. În mod grosier, se poate trage o paralelă între aceste metode și antrenarea nesupervizată. Mai exact, acestea caută artefacte generate de pre-procesare sau post-procesarea aparatului cu care se face captura conținutului multimedia. Spre exemplu, realizarea unei fotografii presupune niște operații hardware/software care își impun amprenta în mod sistematic asupra produsului final. Aceste metode exploatează aceste imperfecțiuni.

1.2.1.1 Distorsie de lentile

Fiecare dispozitiv fotografic presupune existența unui sistem complex de lentile. În mod ideal, acest sistem focalizează lumina într-un singur punct pe senzor. În realitate, sistemul nu poate focaliza perfect lumina la toate lungimile de undă. Aceste defecte de producție se numesc aberații cromatice și sunt de două tipuri: longitudinale și laterale.

- Aberațiile longitudinale se manifestă prin diferențe în planurile focale pentru valori diferite ale lungimii de undă. Ele pot fi modelate drept o convoluție a fiecărui plan de culoare cu un filtru trece jos specific.

- Aberațiile laterale se manifestă drept variații spațiale a zonelor în care ajunge în final lumina de diverse lungimi de undă. Acestea pot fi modelate drept o expansiune sau o contracție a unui plan de culoare față de celălalt.

În momentul în care o imagine este modificată, atunci valorile acestor aberații cromatice se schimbă și își pierd uniformitatea, lăsând astfel o amprentă care să poată fi analizată de către un expert criminalist.

Alte metode mai noi se folosesc și de distorsiunile asupra geometriei imaginii impuse de către sistemul de lentile, cum ar fi distorsiune "pincushion" sau distorsiune "barrel". Voi prezența aceste lucruri mai în detaliu în capitolele următoare.

1.2.1.2 Artefacte CFA

Mare parte din camerele digitale folosesc un filtru de culoare (CFA), care prezinta un model periodic, astfel încât fiecare pixel al senzorului fotoelectric înregistrează lumina numai într-un anumit interval de lungimi de undă (roșu, verde, albastru). Restul de informație este interpolată cu ajutorul pixelilor înconjurători, această operație fiind numită demozaificare. Procesul va impune un tipar periodic de corelație (deci un sablon de erori sistematice) în toate imaginile capturate. Prin urmare, în momentul în care o imagine este manipulată și modificată, acest model periodic intrinsec va fi distorsionat, oferind astfel expertului criminalist informații relevante.

Un factor important al acestei metode este că fiecare model de dispozitiv fotografic va avea aceeași configurație de CFA și același algoritm de interpolatie. Prin urmare, în momentul în care peste imaginea inițială se va plasa un obiect din altă imagine, capturată cu alt model de aparat foto, tiparul va fi evident anormal.

1.2.1.3 Nivel de zgomot

O abordare mai generală este aceea în care se evidențiază artefactele de zgomot introduse de către tot procesul de capturare a imaginii, indiferent de sursele lor de proveniență. Analiza zgomotului local (se poate trasa o paralelă cu filtrul Lee studiat în cadrul materiei Procesarea Imaginilor) poate oferi informații despre adăugarea unor obiecte străine în imagine. Acest gen de analiză locală a fost realizat atât folosind metode statistice în domeniul spațial, cât și în domeniul wavelet (care va fi prezentat în capitolele următoare).

Chiar și așa, intensitatea zgomotului nu este o sursă riguroasă de informație și trebuie consolidată. Astfel, imaginea convolutata cu o mască de filtru trece sus va fi folosită pentru a obține diverse caracteristici care să descrie mai bine imaginea decât doar valoarea intensității.

1.2.1.4 Artefacte de compresie

Diverse metode de compresie ale imaginilor și videoclipurilor oferă o cantitate vastă de informație brută care poate să fie ușor exploatată în domeniul criminalistic.

Un exemplu la îndemână este chiar compresia JPEG, în care imaginea este împărțită pe blocuri, este proiectată într-un spațiu virtual cu ajutorul transformatei cosinus discrete, urmând să fie cuantizata și să treacă prin ceilalți pași ai acestei metode. Studiind linia de asamblare a compresiei, experții în analiză imaginilor s-au gândit să se folosească de împărțirea în blocuri,

aceasta determinând un tipar periodic care se respectă în orice imagine. În momentul în care acest tipar periodic devine slab sau nu mai este găsit, înseamnă că imaginea a fost modificată.

Alte metode care se bazează tot pe compresia JPEG reușesc să extragă caracteristici specifice fiecărui algoritm de compresie, bazându-se pe modul în care se face conversia din real în întreg în etapa de cuantizare.

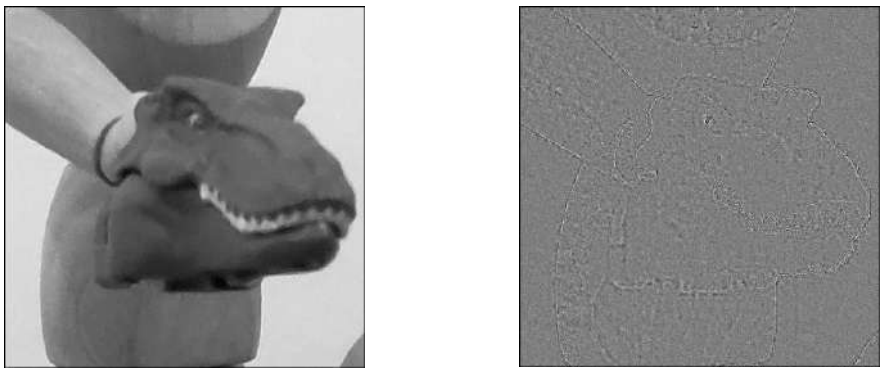
1.2.2 Metode cu informație a priori

Al doilea tip de abordare este unul care se bazează pe existența unei forme de informație anterioare, asa cum se mentioneaza si in titlu. O metodă importantă, care în mod întâmplător este și subiectul lucrării mele, este cea care se bazează pe senzorul camerei foto/video. Așa cum am menționat anterior, pentru a se realiza o poză, trebuie să existe o însumare a părților de hardware și software. Pe partea de hardware, printre primele acțiuni realizate de către o camera este aceea de a capta lumina prin intermediul unor senzori. În mod teoretic, acești senzori sunt identici din punct de vedere geometric și complet omogeni din punct de vedere al constituției chimice. În realitate, acest lucru este imposibil din cauza imperfecțiunilor evidente din partea producției umane. Prin urmare, fiecare camera va avea o amprentă proprie prin care poate să fie identificată. Această discuție se poate extinde și la modele de camere, nu neapărat la camere individuale, însă această discuție va fi continuată ulterior, în cadrul capitolului concepte de baza.

Metodele de detecție ale manipulării bazate pe PRNU constau în 2 pași fundamentali:

- Definirea unei amprente de bază a obiectului, care este utilizată drept punct de referință pentru celelalte imagini care trebuie analizate. Această amprenta se calculează utilizând mai multe imagini de integritate certă, care au fost captate de către aparatul în cauză.
- Calculul amprentei, sau a reziduului de zgomot al unei imagini pe care vrem să testăm dacă a fost sau nu manipulată.

Prin urmare, este evidentă dependența metodei de existența informației a priori, însă în ciuda acestui dezavantaj, aceasta oferă invarianță la tipurile de modificări, fiind capabilă să le descopere mai bine decât multe alte metode. Vom dezvolta mai mult acest subiect in urmatoarele capitole.



(a) Imagine originala (b) Reziduul zgomotului

Figura 1.3: Exemplu poza originala vs. poza PRNU

1.2.3 Metode ce implică "Machine Learning"

"Al treilea tip de abordare este unul care se bazează pe conceptul de machine learning. În aceste cazuri, se alege o lista de trăsături care oferă posibilitatea de deosebire între imagini

originale și imagini manipulate, acest pas fiind urmat de a antrena un clasificator pe un set de date mare. Trebuie menționat faptul că trăsăturile sunt alese în mod arbitrar de către utilizatorul uman, expertul în analiză criminalistică, lăsând astfel loc de imperfecțiuni."

De-a lungul timpului s-a încercat găsirea unor caracteristici specifice pentru fiecare tip de artefact; această abordare nu este una fezabilă, întrucât numărul de artefacte este unul ridicat iar testarea tuturor posibilităților nu este deloc eficientă.

Prin urmare, s-a încercat obținerea unor caracteristici universal valabile, care să își mențină verosimilitatea indiferent de context. Pentru a realiza acest lucru, experții in domeniu au concluzionat că primul pas trebuie să fie înlăturarea conținutului propriu-zis al imaginii (efectiv obiectele pozate ce pot fi văzute cu ochiul), rămânând astfel doar tiparele de zgomot impuse de proces. Astfel, folosindu-se de modele statistice, pot trage concluzii despre o imagine analizată, folosind fie domeniul spațial fie unul virtual, cum ar fi cel de tip Fourier sau Wavelet.

Acest capitol introductiv a fost realizat în mare parte utilizând cunoștințe din articolul[1] prezentat de Luisa Verdoliva, dar și din articolele publicate de către Farid & Hany[2] și Johnson & Micah & Farid & Hany[3].

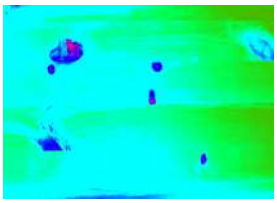
Capitolul 2
Concepte de bază

2.1 Teorie de bază in imagini si videoclipuri

2.1.1 Imagini - fundamente

O imagine este în general generată cu ajutorul unui dispozitiv de achiziție, care face translația dintr-un stimul fizic m dimensional într-un semnal bidimensional.

Acest stimul poate fi de mai multe forme, fie el optic, termic, raze X și altele, urmând să fie transformat într-un semnal digital analogic și apoi digitalizat printr-o serie de procese.



(a) Imagine termală

(b) Imagine cu raze X[4]

(c) Imagine optică

Figura 2.1: Tipuri de imagini

Pentru captarea unei imagini, aparatul realizează următorii pași:

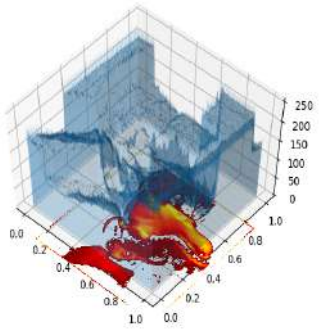
- Mini-panoul de senzori foto-sensibili combinat cu filtre de lungimi de undă este atins de rază incidentă de lumină, realizând transformarea energiei fotonilor în electroni.
- Cantitatea de sarcină generată de efectul fotoelectric este amplificată și trimisă către un convertor analog digital.
- Din cauza filtrului de lungimi de undă, este necesară o operație de interpolare prin care se deduc valorile celor trei planuri de culoare din așa numitul "mozaic" inițial. Această operație este numită demozaificare.
- Aici apar diverse pre-procesări ale imaginilor, cum ar fi reducerea zgomotului, manipularea culorilor, rectificarea distorsiunilor geometrice.
- Compresie într-un standard cunoscut, în general JPEG.

Până aici a fost făcută o descriere de nivel înalt al imaginilor, însă voi continua prin prezentarea dimensiunii matematice a acestora.

La fel cum sunetul este un semnal unidimensional, imaginile sunt semnale bidimensionale. Din punct de vedere conceptual, acestea pot fi considerate semnale continue, deoarece rezoluția unui aparat de captură media ideal poate tinde spre infinit, putând astfel observă orice fel de variații, chiar la nivel de quarki sau alte particule. În realitate, este evident că imaginile obținute prin niște dispozitive reale, manufacturate de către producători umani, vor fi de fapt niște semnale bidimensionale discrete.

Pentru o înțelegere mai intuitivă, imaginile pot fi prezentate drept suprafețe în spațiul tridimensional.

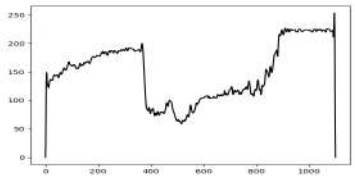
(a) Imagine originala



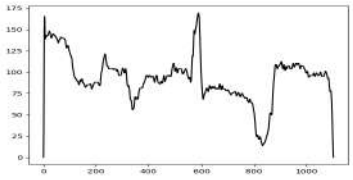
(b) Imagine tridimensionala

Figura 2.2: Translatia din spatiul bidimensional in spatiul tridimensional

O reprezentare mai simplă a imaginilor este printr-o multitudine de semnale unidimensionale, de-a lungul axelor Ox si Oy



(a) Secțiune pe 0x



(b) Secțiune pe Oy

Figura 2.3: Imaginea originală cu dinozaurul pe cele 2 direcții, in 2 valori alese arbitrar

2.1.2 Analiză Fourier

Continuând pe baza ideii anterioare, voi introduce și unele ecuații și concepte matematice necesare în următoarele capitole. Încep prin a face translația dintre transformata Fourier unidimensională și cea bidimensională.

Cazul unidimensional:

$$F(\omega) = \int_{-\infty}^{\infty} f(x) \mathrm{e}^{-j\omega x} dx \tag{2.1}$$

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) \mathrm{e}^{j\omega x} d\omega \tag{2.2}$$

Cazul bidimensional:

$$F(\omega_x, \omega_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \mathrm{e}^{-j(\omega_x x + \omega_y y)} dx dy \tag{2.3}$$

$$f(x, y) = \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(\omega_x, \omega_y) \mathrm{e}^{j(\omega_x x + \omega_y y)} d\omega_x d\omega_y \tag{2.4}$$

Pentru a explica formulele bidimensionale voi trasa o paralelă între ele și cele unidimensionale.

În spațiul 1-D, Fourier a demonstrat că orice semnal continuu poate fi reprezentat drept o însumare de sinusoidale de frecvențe diferite.

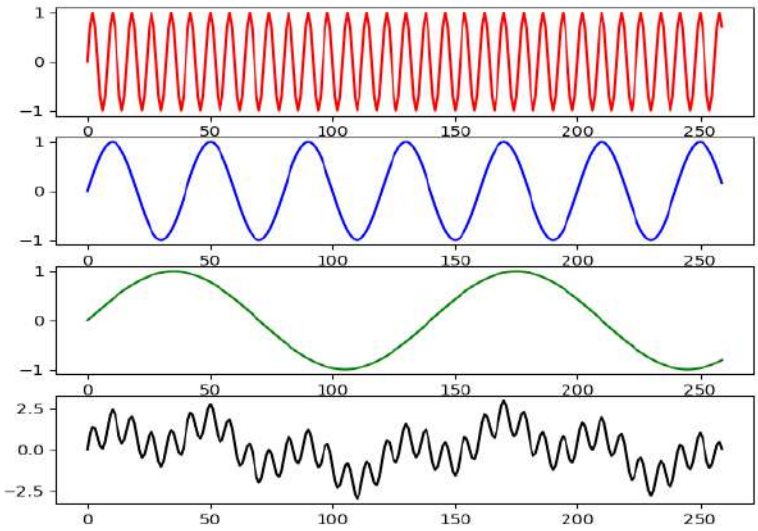


Figura 2.4: Explicație Fourier 1-D

Exact același lucru se întâmplă și în cazul 2-D, singura diferență fiind că sinusoidalele (respectiv cosinusoidalele) vor fi la rândul lor suprafețe bidimensionale care se adună între ele.

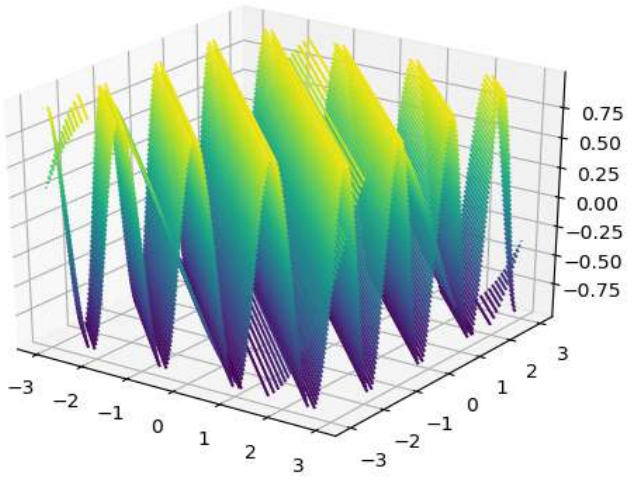
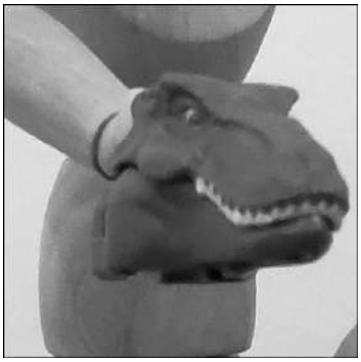


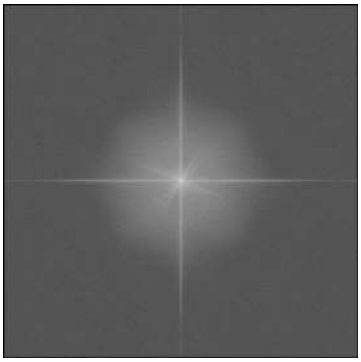
Figura 2.5: Explicație Fourier 2-D

Trebuie menționate unele lucruri de bază, tot în cadrul acestei paralele. La fel cum în 1-D frecvențele joase însemnau o variație mai lentă a semnalului, în 2-D frecvențele joase își păstrează semnificația, sugerând o oarecare uniformitate locală a imaginii. Similaritatea se poate întinde și la frecvențele înalte, care în cazul 2-D sunt cele responsabile pentru detaliile minore din poze, cum ar fi contururi sau alte obiecte de dimensiuni mici care sunt prezente. Din experiența mea, acesta este cel mai intuitiv mod de a înțelege conceptul de frecvență.

Evident, discuția despre frecvențe înalte și joase face parte din unitatea care conține și subiectul filtrării acestor frecvențe. În schimb, pentru a demonstra filtrarile de tip trece jos și trece sus mă voi folosi de graficele următoare:

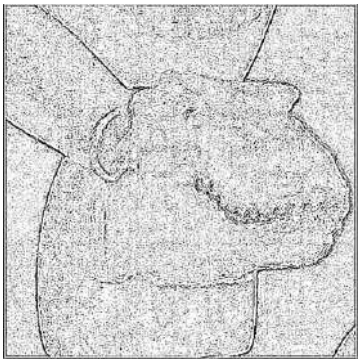


(a) Imagine originală in spatiu



(b) Imagine originală in frecvență

(c) Imagine cu filtru trece-jos cu masca 21x21



(d) Imagine cu filtru trece-sus cu masca 3x3

Figura 2.6: Imagini în domeniul Fourier și rezultate ale filtrărilor

2.1.3 Analiză Wavelet

Pentru a prezenta conceptele de teorie în domeniul Wavelet, am realizat anterior un subcapitol rezumativ pentru analiza Fourier. Pentru a continua în această direcție de eficientizare a fluxului de informații, voi face o scurtă prezentare a unui alt concept de bază, care leagă transformata Fourier de cea Wavelet și anume transformata Gabor.

Este un lucru cunoscut faptul că informația din spectrul Fourier, fie el de amplitudine sau de fază, nu oferă o asociere între frecvențele prezente și localizarea acestora în domeniul spațial sau temporal. Prin urmare, dacă se dorește realizarea unei analize mai complexe, domeniul Fourier devine limitat. Acest lucru a fost observat și de fizicianul Dennis Gabor, care a reușit să modifice formula transformatei Fourier, adăugându-i o restricție spațială sau temporală de tip Gaussian într-un anumit interval.

Explicat în cel mai simplu mod posibil, acesta a realizat ferestruirea funcției analizate inițial, într-un anumit interval și a aplicat transformata Fourier în acea delimitare. Prin această inovație, Dennis Gabor a oferit "o nouă axa de libertate", permițând analiză separată a diferitelor intervale pentru a putea observa ce frecvențe sunt prezente în ce porțiuni din semnal. Evident, această metodă are și ea limitările ei pe care le voi prezența în următoarele paragrafe.

$$\text{Formula pentru transformata Fourier: } F(\omega) = \int_{-\infty}^{\infty} f(x)\mathrm{e}^{-j\omega x}dx \tag{2.5}$$

$$\text{Formula unei functii fereastra: } g_\alpha(x) = \frac{1}{2\sqrt{\pi\alpha}}\mathrm{e}^{-t^2/(4\alpha)} \tag{2.6}$$

$$\text{Formula transformei Gabor: } (G_b^\alpha f)(\omega) = \int_{-\infty}^{\infty} f(x)\mathrm{e}^{-j\omega x}g_a(x-b)dx \tag{2.7}$$

În formula de mai sus, α este un parametru al funcției fereastră de tip Gaussian, echivalent al variației, în timp ce parametrul real b este cel responsabil de plasarea ferestrei în domeniul spațial sau temporal.

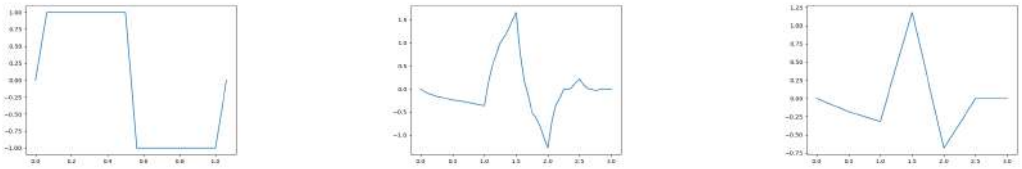
Drept o paranteză, o aplicație relevantă în domeniul facultății noastre o reprezintă chiar spectograma Gabor. La nivel matematic, transformarea în cauza este un caz particular al transformării Fourier de timp scurt, care este utilizată pentru realizarea spectogramelor clasice. În schimb, caracterul Gaussian al ferestrei oferă un avantaj din punct de vedere al rezoluției frecvență-timp, datorită formei ferestrei.

Așa cum am putut observa anterior, transformata Gabor are avantajele ei față de o transformare Fourier normală. În schimb, aceasta are și un dezavantaj evident și anume: fixarea ferestrei într-un punct individiual în domeniul temporal. Acest lucru este relevant prin prisma faptului că în mod ideal, este necesară aplicarea mai multor transformate Gabor, cu ferestre de amplitudini și localizări diferite pentru a putea capta toate frecvențele și a le identifica în domeniul inițial. Spre norocul nostru, aceste sarcini complexe sunt acoperite în totalitate de către transformata Wavelet.

Această transformată, așa cum am insinuat și mai sus, este similară cu cea Fourier, sau mai exact, cu transformata Gabor, inspirându-se din aceasta pentru a impune ferestre de parametri diferiți și forme diverse. Trebuie menționat faptul că în realizarea acestei transformate, primul pas și cel mai important este acela de a alege un wavelet mamă în conformitate cu sarcina pe care dorim să o realizăm. Pentru a clarifica, un wavelet mamă este funcția fereastră principală, a cărei parametrii îi variem pentru a obține o rezoluție timp-frecvență mult mai bună. În urma acestor două idei, se poate deduce ușor faptul că o transformare Wavelet este de fapt un set infinit de transformări, care depinde în totalitate de fereastra aleasă inițial.

$$F(a,b) = \int_{-\infty}^{\infty} f(x)\psi_(^*a,b)(x)dx \tag{2.8}$$

În formula de mai sus, putem observa funcția wavelet $\psi_(a,b)$, care depinde de cei doi parametri reali pentru a prezența o variație față de funcția fereastră mama. Mai exact, "a" este cel care determină forma funcției, adică lungimea și înălțimea, în timp ce "b" este parametrul care determina localizarea acesteia în domeniul temporal.

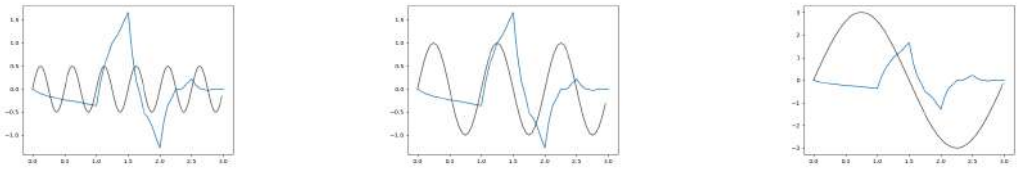


(a) Wavelet Haar (b) Wavelet Daubechie 2 (c) Wavelet Symlet

Figura 2.7: Tipuri de waveleturi

O abordare mai intuitivă, cel puțin în cazul meu, pentru a înțelege cum funcționează parametrul "a" este în felul următor: convenim de la început că păstrăm forma waveletului mamă intactă și alegem să variem scalarea funcției pe care dorim să o analizăm. În mod logic, dacă funcția va fi scalata supraunitar, aceasta va crește raportat la fereastra de analiză și prin urmare vom începe să detectăm frecvențe cât mai mari. Același raționament poate fi utilizat și în cazul în care funcția este scalata subunitar, ea urmând să scadă in raport cu waveletul mamă și prin urmare vom începe să detectăm frecvențe cât mai mici.

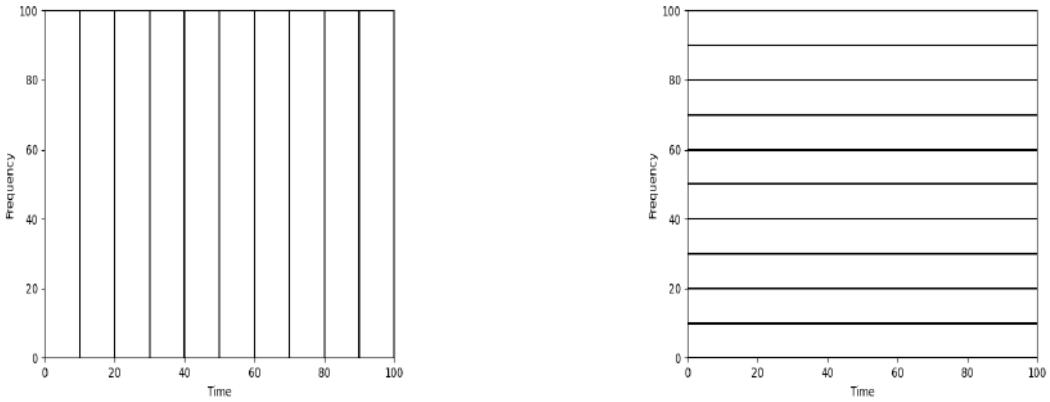
Ca să ilustrez explicația anterioară cu scalarea, voi folosi următoarele grafice:



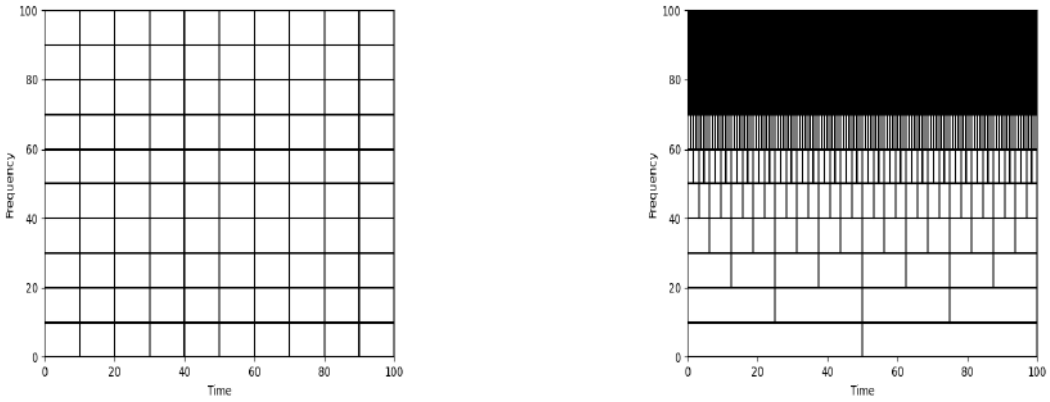
(a) Frecvențe mici (b) Frecvențe intermediare (c) Frecvențe mari

Figura 2.8: Variație a semnalului pentru a "modifica" parametrii waveletului mamă

Având în vedere explicațiile și graficele anterioare, putem deduce care este beneficiul major adus de transformarea Wavelet: oferă o rezoluție frecvență-timp foarte bună, în comparație cu precursorii săi. Pentru a evidenția acest lucru, voi urmări aceste desene:



(a) Rezoluție timp-frecvență semnal original (b) Rezoluție timp-frecvență transformata Fourier



(c) Rezoluție timp-frecvență Fourier scurt (d) Rezoluție timp-frecvență transformata Wavelet

Figura 2.9: Rezoluții timp-frecvență in diverse domenii

Pentru a înțelege figurile de mai sus, primul lucru care trebuie clarificat este faptul că punctele de intersecție dintre dreptele paralele cu Ox și Oy sunt cele care explică ideea de rezoluție timp-frecvența. Un sumar la nivel de suprafață ar fi că anumite frecvente pot fi localizate în domeniul temporal cu o anumită acuratețe, aceasta fiind direct proporțională cu mărimea ferestrei. Deci, cu cât fereastra este mai mare, cu atât vom avea un interval temporal mai mare, iar cu cât intervalul temporal este mai mare cu atât scade abilitatea noastră de a

oferi o locație concretă. Simultan, trebuie menționat că o fereastră foarte mare va cuprinde atât frecvente înalte cât și frecvente joase, în timp ce o fereastră mică va cuprinde exclusiv frecvente înalte.

Vom prezenta acum fiecare grafic în parte, pentru a adăuga explicații în fiecare caz:

(a) Rezoluție timp-frecvență semnal original: Așa cum se poate observa, nu există niciun punct de intersecție întrucât domeniul temporal nu oferă nicio informație despre localizarea în frecvență, deci nu se poate face nicio legătură între cele două.

(b) Rezoluție timp-frecvență transformata Fourier: Așa cum se poate observa, nu există niciun punct de intersecție întrucât domeniul Fourier nu oferă nicio informație despre localizarea în timp, deci nu se poate face nicio legătură între cele două.

(c) Rezoluție timp-frecvență Fourier scurt: Așa cum se poate observa, aici apar intersecții între paralelele cu cele două axe la distanțe constante. Conform paragrafului de mai sus, acest lucru denotă faptul că putem localiza unele frecvențe în timp, având totuși o restricție impusă de dimensiunea ferestrei utilizate. Astfel, fie nu vom putea găsi exact poziția unei frecvente foarte înalte, fie nu vom observa deloc existența unei frecvente joase.

(d) Rezoluție timp-frecvență transformata Wavelet: Așa cum se poate observa, aici apar intersecții între paralelele cu cele două axe la distanțe variabile. Acest lucru este marele avantaj al transformei Wavelet. Aceasta acoperă lipsurile transformatei Fourier scurte sau a transformatei Gabor, având implementată intrinsec o fereastră variabilă. În acest fel, conform desenului, frecvențele joase vor avea o fereastră mai mare (plecând de la prezumpția că aceastea sunt prezente permanent în semnal), care să le permită să fie descoperite în timp. În același timp, cu cât frecvența crește, cu atât intervalul temporal în care această poate fi găsit se rafinează, ajungând la o acuratețe de localizare foarte mare.

Pentru a putea continua cu explicatiile transformatei Wavelet, voi utiliza o poză cu rezultatul acesteia când este aplicată pe o imagine:

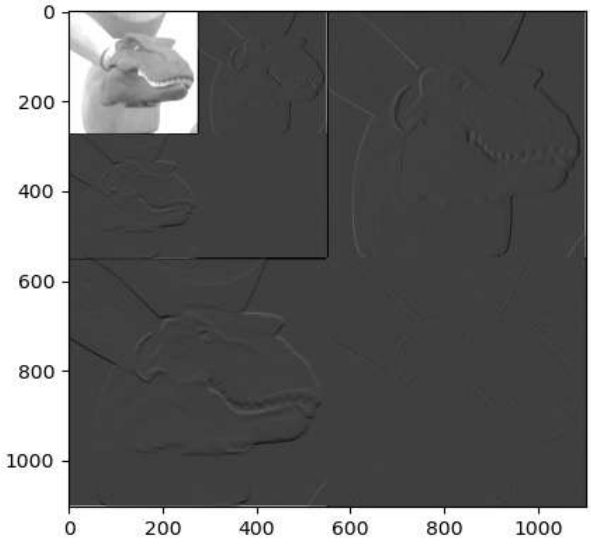


Figura 2.10: Exemplu transformata Wavelet

Se poate observa împărțirea imaginii în mai multe pătrate(nivele). Fiecare dintre acestea este împărțit la rândul său în alte 4 pătrate mai mici, cu o însemnătate specifică: stânga sus se

referă la imaginea cu rezoluție mai mică, deja filtrată orizontal și vertical, stânga jos se referă la o filtrare orizontală, dreapta sus la o filtrare pe diagonală la rezoluția inițială.

Strict drept exemplu practic, această transformată Wavelet se folosește în ciuda complexității sale, în domeniul compresiei de imagini. Motivul pentru care se acceptă acest sacrificiu al ușurinței de implementare este însușirea acestui gen de compresie care permite transmiterea în partiții de date a informației, în funcție de nivelul de complexitate al acesteia. Pe scurt, în momentul în care se deschide o poză mare într-un tab, mai întâi o să se încarce fundalul întreg, următorul pas va fi apariția unor trăsături vagi, următorul pas este apariția altor trăsături și tot așa până se încarcă toată imaginea. De asemenea, această transformare permite alegerea cantității de "detaliu" trimisă în diferite locuri ale imaginii, putând crea astfel zone de interes special în care să existe o rezoluție bună, restul pozei rămânând blurată sau pixelată.

2.1.4 Video - fundamente

Semnalele video sunt semnale multi-dimensionale caracterizate de existența unei axe temporale. În alte cuvinte, videourile sunt imagini care variază încet în timp. Această variație se datorează faptului că de fapt un video este doar o secvență de imagini statice înregistrate la intervale specifice de timp astfel încât să ofere impresia de continuitate.

Așa cum am văzut și în cazul imaginilor bidimensionale, semnalele video pot fi atât analogice cât și digitale. În schimb, până și în cazul analogic, nu poate exista un domeniu continuu de imagini, fiind necesară o discretizare în domeniul temporal. Astfel, in urma acestui proces de eșantionare apar așa numitele cadre. Acest lucru nu este o limitare a hardware-ului sau a software-ului, cât până și în cazul ochiului uman, ceea ce ni se pare un flux continuu de informație vizuală este de fapt doar o continuă eșantionare a impulsurilor fizice optice urmată de post-procesări în creier.

Mă voi axa mai mult pe partea de videoclipuri digitale. Acestea pot fi obținute în 2 moduri: fie printr-o conversie dintr-un video analog, fie direct cu o camera digitală care esantioneaza în timp real semnalul. Dacă în cazul imaginilor am explicat conceptul de eșantionare spațială, în cazul semnalelor video, pe lângă acesta se mai adaugă și eșantionarea temporală. În mod ideal, între frecvența temporală și cea spațială există o relație complexă, întrucât prima dintre ele depinde de conținutul cadrului, fiindu-ne astfel greu să aplicăm formula Nyquist. Prin urmare, pentru a evita o prolixitate, se alege o rată de eșantionare îndeajuns de mare încât ochii umani să nu mai poată perceapă cadrele drept imagini separate, obținând astfel iluzia de continuitate. Cea mai mică valoare posibilă pentru care se îndeplinește asta este 12 FPS, însă până și aici percepția noastră depinde de luminozitate.

Din punct de vedere al procesării, videourile pot fi tratate pe fiecare cadru individual, la fel ca niște operații pe o imagine statică. Acest lucru funcționează corect, însă nu este deloc eficient din punct de vedere al informației pe care o poartă semnalul video. Neglijând axa temporală, se pierde o oarecare continuitate și o corelație între imagini.

Din acest motiv, experții în domeniu au găsit un mod de a exploata această informație temporală pentru a realiza o compresie foarte bună a semnalelor video. Numai drept punct de referință, un video de dimensiuni 1920x1080 pixeli/cadru, cu 24 de biți per pixel și 30 FPS ar necesita o rată de transmisie de 1.5 giga-biți pe secundă. Evident, acest lucru nu este în niciun fel fezabil, motiv pentru care a apărut o soluție care să exploateze în totalitate redundanța dimensiunii temporale - 2 cadre apropiate sunt în mare foarte mare măsură similare -, utilizând tehnici predictive.

Evident, în cazul semnalelor video există atât o compresie intra-cadru, adică o compresie spațială, cât și o compresie inter-cadru, adică o compresie temporală. O vom explica pe cea de-a doua întrucât prima nu se află în zona de interes a acestei lucrări.

Pentru a putea explica mai clar, trebuie definite inainte:

- Toate cadrele fac parte din grupuri de dimensiuni prestabilite, în care există un singur cadru de tip I, mai multe de tip P și majoritatea de tip B.
- Cadre de tip I: sunt cadre captate în totalitate, care nu au fost influențate de către vreo predicție și sunt codate numai în mod intra.
- Cadre de tip P: sunt cadre care sunt calculate cu ajutorul metodelor predictive, din cel mai apropiat cadru de tip I sau P din grupul său.
- Cadre de tip B: sunt cadre în totalitate interpolate din conținutul cadrelor de tip P și I ce fac parte dintr-un anumit grup.

Având aceste cunoștințe, putem continua spunând că de fapt cadrele de tip I sunt utilizate mult mai rar decât celelalte 2, acestea fiind în general necesare când scena se schimbă brusc sau când se ajunge la finalul unui grup. Cadrele de tip P sunt utilizate pentru a obține o aproximare rezonabilă, rapidă și mai putin redundantă decât un alt cadru I ce l-ar putea înlocui. Cadrele de tip B sunt utilizate in general pentru a oferi precizie semnalului video, mai ales când mișcarea unui obiect permite observarea fundalului necunoscut.

Trebuie menționat că numai cadrele de tip I se pot decoda independent, celelalte 2 tipuri depinzând de decodarea cadrelor pe care au fost realizate predicțiile.

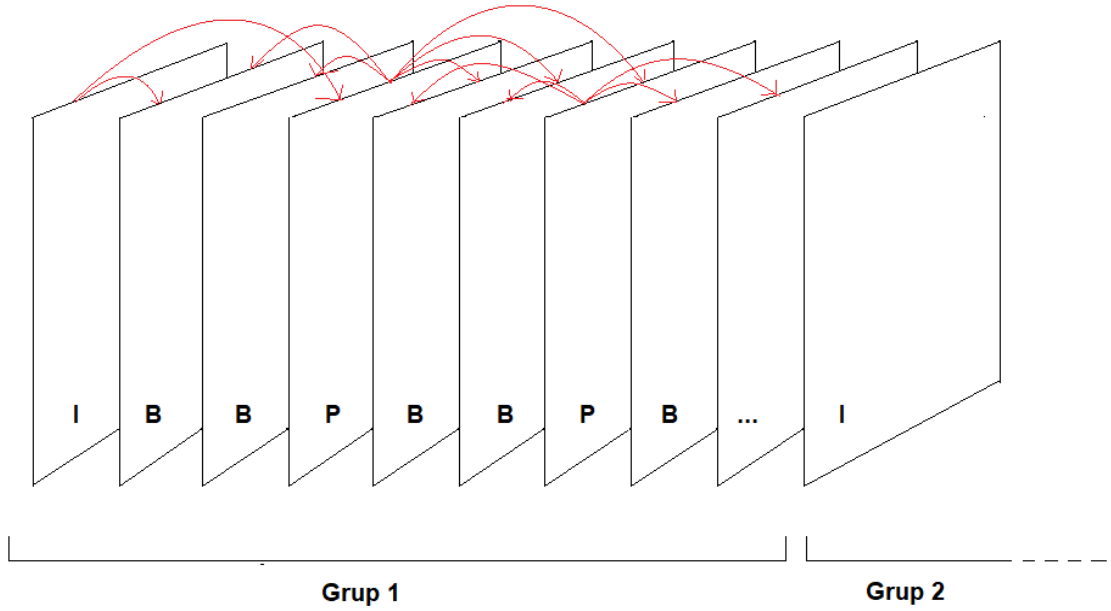


Figura 2.11: Explicație tipuri de cadre

Voi face de asemenea o scurtă prezentare a modului de filtrare în cazul semnalelor video, numai pentru a putea oferi niște posibile idei de dezvoltare în capitolele finale ale lucrării. În general, filtrarea se folosește pentru a îndepărta niște artefacte caracteristice videoclipurilor :

- Zgomot digital - cauzat în general de compresie și transmisie digitală, rezultând în artefacte de tip bloc, "fantome", etc.
- Artefacte de film - cauzate în general de praf, interferență, amprente, zgârieturi și alte influențe fizice asupra filmului.

- Zgomot analog - cauzele apariției acestuia sunt: interferență, zgomot de mediu, recepție imperfectă.

Evident, prima idee este aceea de a rămâne constanți în abordare, încercând să aplicăm metodele deja cunoscute de la imagini statice. Acest lucru funcționează și își îndeplinește scopul. În schimb, această operație are niște urmări neplăcute asupra calității obiective totale a semnalului video, deoarece poate induce artefacte temporale din cauza tratării independente ale cadrelor constituente și a lipsei de exploatare a corelației temporale între cadre separate. Ținând cont de aceste considerente teoretice, ne este ușor să tindem mai mult spre utilizarea unor filtre inter-cadre, care să folosească toată informația prezentă.

Pentru a explica mai ușor utilitatea acestor filtre temporale, vom folosi un exemplu la îndemână: presupunem că avem un semnal video în care am filmat un punct fix de pe un perete, fără a ne mișca. Acest lucru presupune că în video nu există mișcare, umbre, sau mai general variație de vreun fel a conținutului. Prin urmare, de la un cadru la altul, vom pleca de la prezumpția că valorile unui pixel de pe o poziție ar trebui să fie în mod ideal constante. În realitate, acea valoare a pixelului va fluctua datorită influenței zgomotului. Utilizând un filtru intra-cadru, putem să aplicăm o filtrare de tip trece-jos, având drept rezultat o blurare a detaliilor din imagine. Acest lucru nu ne convine și vom alege alternativa filtrării inter-cadru. Acest lucru presupune filtrarea printr-un "tub" temporal care folosește aceleași poziții pentru a realiza îndepărtarea zgomotului. În acest fel, reușim să facem uz atât de corelația în timp, cât ne și asigurăm că nu impunem discontinuități de niciun fel între cadre.

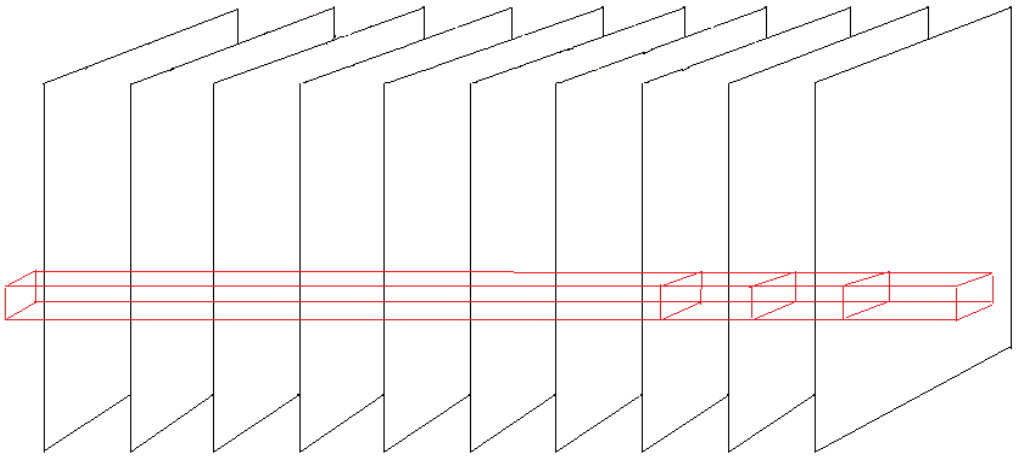


Figura 2.12: Explicație filtru temporal

Cu asta am concluzionat teoria de bază necesară înțelegerii a mare parte din conceptele utilizate în lucrare. De acum lucrarea se va axa mai mult pe subiectul ei principal, PRNU.

2.2 Răspuns fotografic non-uniform - PRNU

În general, există două tipuri de senzori utilizați în camere digitale, camere video și scannere: CCD și CMOS. Amândouă sunt formate dintr-o mulțime de mini foto-senzori numiți pixeli. Aceștia sunt confecționați din siliciu și captează lumina folosindu-se de efectul fotoelectric, realizând tranziția din fotoni în electroni. Astfel, sarcina obținută este amplificată și transmisă mai departe la un convertor analog digital, urmând să fie procesată în continuare pentru a obține produsul final, imaginea.

Acești pixeli sunt dreptunghiulari, de dimensiuni de câțiva microni. Mărimea și omogenitatea siliconului sunt cele care determina cantitatea de electroni generată de lumina incidența. Așa cum am menționat anterior, este evident că nu toți pixelii vor fi asemănători, rezultând astfel într-o amprenta de zgomot similară și recurentă în absolut toate pozele sau videoclipurile realizate cu acel senzor.

Prezumpția principală datorită căreia putem să aplicăm această metodă este că toți senzorii de imagini sunt construiți din semiconductori și modul lor de producție este asemănător.

Evident, fiecare pixel în parte va avea o fluctuație PRNU specifică indiferent de valoarea reală. Acest lucru este echivalent cu existența unei măști de diferențe de valori care se află peste imaginea inițială. Această mască este amprenta camerei. Ea poate fi estimată experimental prin captarea mai multor poze a unei suprafețe iluminate, medierea acesteia și scăderea mediei din poze pentru a scapă de conținutul real al pozei, rămânând doar cu componenta de zgomot. În schimb, informația deținută de această amprenta nu se rezumă doar la PRNU, cât include și alte părți importante, cum ar fi defectele sistematice de genul pixel morți, curentul negru (care este zgomotul ce apare în momentul în care dispozitivul ar face o poză cu capacul pus deasupra). Chiar și așa, PRNU rămâne cea mai importantă caracteristică a amprentei, fiind cea mai exploatabilă. Trebuie menționat că acesta nu este prezent nici în locurile foarte întunecate din imagine, nici în cele mult prea saturate.

Chiar dacă zgomotul non-uniform este o proprietate stochastică, acesta rămâne constant și stabil de-a lungul intervalului de viață al dispozitivului, acest lucru crescându-i fiabilitatea și răspândirea. Așa cum am menționat, această metodă este superioară celorlalte prezentate, urmând acum să dovedesc asta prin prezentarea caracteristicilor sale:

- Robustețe: Rămâne prezent în urma operațiilor de compresie, filtrare, corecții și alte procesări.
- Dimensionalitate: Este probabilistic din natură și are o cantitate mare de informație. Având în vedere mărimea senzorului complet (ceea ce presupune o cantitate foarte mare de pixeli individuali), există destul loc pentru caracterul stochastic de a își juca rolul.
- Stabilitate: Rămâne constant în apariție și valori indiferent de condițiile de utilizare ale aparatului.
- Generalitate: PRNU este prezent în imaginile captate de un dispozitiv foto indiferent de setările acestuia. Chiar și în cazul imaginilor extreme din punct de vedere al saturației, acesta este prezent în sens teoretic.
- Universalitate: Orice imagine digitală prezintă PRNU.

Amprenta PRNU are utilitate în mai multe aplicații, pe care le voi lista dedesubt. Înainte de asta, trebuie să precizez condițiile "experimentului". Presupunem existența unui eșantion de aparate foto. Fiecare aparat are la rândul sau un număr de poze garantat făcute de acesta. Utilizând imaginile caracteristice, putem determina o amprentă a dispozitivului pe care să i-o asociem și să o stocăm în memorie. Restul experimentului constă în concluzionarea dacă o imagine nouă analizată îndeplinește sau nu o condiție (ipoteza 1 sau ipoteza 0).

- Identificarea aparatului - Această utilizare este cea pe care o voi studia de-a lungul următorelor capitole, întrucât extinderea de la aceasta la celelalte implementări este ușor de realizat. Aici calculăm reziduul de zgomot al noii imagini de analizat, îl corelăm cu amprentele asociate fiecărui aparat, iar dacă valoarea calculată trece peste un anumit prag, atunci imaginea poate fi considerată ca fiind captată de aparatul asociat amprentei cu care s-a calculat cea mai mare corelație.

- Verificarea integrității - Observând lipsa formei de zgomot putem deduce ușor care sunt părțile din imagine care au fost modificate sau complet înlocuite. Acest lucru poate fi făcut printr-o analiză de tip bloc, mecanismul fiind același, însă aplicat de mai multe ori pe mai multe zone.

- Istoria de procesare - Putem folosi amprenta originală drept punct de referință și să încercăm diverse modificări inverse ale imaginii analizate. Prin modificări inverse mă refer la opusul operațiilor de scalare, tăiere sau rotație. Acest lucru va fi descris mai pe îndelete în următoarele capitole.

Întrucât nu ar avea sens să aglomerez acest capitol axat mai mult pe partea conceptuală de nivel mediu, voi prezenta sumar pașii necesari pentru calculul PRNU în diversele aplicații de mai sus, lăsând partea matematică pentru a fi aprofundată în 3.1, pe care vă invit să îl citiți după acest capitol, pentru a putea urmări restul lucrării cu ușurință.

Drept ipoteză, putem pleca de la o imagine nouă, despre care nu avem nicio informație și căreia vrem să îi calculăm amprenta PRNU. În stadiul ei inițial, aceasta conține subiectul pozei, adică diverse obiecte ce au reflectat lumina pentru a fi captate pe film/ în mod digital, conține reziduul PRNU, dar mai conține și zgomot provenind din alte surse, cum ar fi "curentul negru" sau compresia JPEG. Evident, scopul nostru este de a scăpa de toți ceilalți factori pertubatori și să rămânem doar cu partea de PRNU din imagine.

În această direcție, primul pas pe care îl parcurgem este acela de a înlătura de imagine partea propriu-zisă de conținut. Pentru a face acest lucru, aplicam o filtrare de tip Wiener adaptivă in domeniul Wavelet.

În primul rând, filtrarea de tip Wiener este o filtrare ce are drept scop restaurarea imaginii inițiale, înlăturând astfel artefactele de tip determinist. Aceste artefacte se pot simula în mod ideal printr-o convoluție a imaginii originale, perfecte, cu o mască ce caracterizează defectele modelate. Astfel, filtrarea Wiener are la bază un simplu filtru invers, care aplică o convoluție cu inversa măștii de care am discutat mai sus. Acest filtru brut, însă, este limitat din cauza faptului că nu ia în considerare prezența zgomotului pe care îl va amplifică. Prin urmare, experții în domeniu s-au gândit să impună o constrângere asupra energiei zgomotului, minimizând-o cu ajutorul informațiilor a priori despre această, dând astfel naștere filtrului Wiener. Acesta poate să fie privit drept un "pipeline", începând cu o filtrare inversă și urmată de o corectare a acestei filtrari. Partea matematică va fi acoperită în capitolul de modele matematice.

În al doilea rând, trebuie să răspund unei întrebări de genul "Ce sens are să aplici o filtrare de acel gen în domeniul wavelet?". Conform lui M. Kıvanc Mıhcak, lui Igor Kozintsev și a lui Kannan Ramchandran, există o logică solidă în spate. Totul pleacă de la algoritmii de compresie atât de necesari de care am vorbit fugitiv și mai sus. Teoretic, scopul acelor metode este de a capta cea mai mare cantitate posibilă de energie în cât mai puțin volum de date. Acest lucru se realizează prin neglijarea detaliilor minuscule care nu aduc niciun fel de beneficiu utilizatorului uman. Funcționează exact pe același principiu pe care, în ciuda faptului că un semnal unidimensional are o infinitate de componente spectrale infim de mici, le vom alege pe acelea care ne permit să obținem 99.99% din informația transmisă. Acele detalii minuscule, în cazul nostru, pot fi considerate zgomot, reușind astfel să fie trasată o paralelă între compresia imaginilor și filtrarea lor împotriva zgomotului. Pornind de la această idee și

cunoscând superioritatea algoritmilor de compresie ce folosesc transformata Wavelet, cei trei experți le-au împerecheat în următorul mod:

1. Se folosește un model de mixturi independente Gaussiene cu variații necunoscute, care se modifică raportat la locația lor în spațiu, reprezentând coeficientii Wavelet.

2. Se estimează variația fundamentală folosind o regulă ML, urmând să aplicăm o estimare de tip MMSE.

3. Se folosește o mască variabilă pentru a putea capta mai bine efectul contururilor.

Esența algoritmului stă în abilitatea sa de a putea afla parametrii Gaussienelor ce reprezintă coeficienții Wavelet. Pentru a putea îndeplini acest lucru, fiecare punct (fiecare coeficient Wavelet) va avea asociată o valoare a variației, bazată pe vecinătatea să locală. Din punct de vedere teoretic, se poate presupune că o vecinătate mai mare va oferi o aproximare mai corectă, însă cu cât este mai mare fereastră de analiză, cu atât riscul de a include coeficienți greșiți crește. Prin urmare, se folosește o fereastră adaptivă, încercând mai multe mărimi (3x3, 5x5, 7x7, 9x9), urmând să se aleagă cea mai mică. În final, cunoscând modelele caracteristice coeficientilor Wavelet, se aplică un filtru de tip Wiener pentru înlăturarea zgomotului.

Având astfel imaginea "perfectă", adică lipsită de zgomot, o putem scădea din imaginea de analizat, rămânând astfel doar cu suma componentelor de zgomot. Cu toate că PRNU este într-adevăr o caracteristică a dispozitivului, celelalte surse de zgomot sunt specifice modelului sau designului de senzor. Din fericire, acestea se manifestă drept semnale periodice în rânduri și coloane mediate ale amprentei PRNU și pot fi suprimate printr-o scădere a acestor medii de-a lungul celor două axe. De asemenea, reziduuri periodice pot rămâne în semnal, afectând rezultatele finale prin valori mari ale corelației cauzate de deplasare a unui șablon de artefacte, dacă nu sunt atenuate printr-o filtrare Wiener corectă.

Un pas opțional între medierea pe axe și ultimul filtru aplicat este trecerea în paleta de culori gri. Acest lucru poate micșora cantitatea necesară de calcul, fiind deci mai eficient, însă pașii de mai sus pot fi la fel de bine aplicați separat în fiecare plan de culoare RGB, urmând la final să fie ponderați conform unei transformări liniare dintr-un spațiu de culori în altul.

Până acum avem amprenta unui dispozitiv și amprenta unei imagini pe care dorim să o analizăm. De aici înainte, avem o multitudine de aplicații pe care putem să le testăm, însă numai una dintre ele a fost abordată în această lucrare: identificarea dispozitivului. În acest sens, este formulată o problemă de detecție într-o formă mult mai generală, adică o problema de testare a ipotezelor

$$H_0 : K_1 \neq K_2$$
$$H_1 : K_1 = K_2$$

În scopul rezolvării acestui test, se pot utiliza mai multe metode, însă voi alege să o prezint exclusiv pe cea mai simplă, care este și implementată în codul lucrării (Anexa 2). Ne interesează valoarea maximă a corelației bidimensionale a lui K_1 cu K_2, unde K_2 este amprenta imaginii analizate și K_1 este amprenta dispozitivului. Acest lucru se poate face atât în domeniul spațial, însă un avantaj mare îl reprezintă faptul că poate fi realizat și în domeniul frecvenței, întrucât corelația se rezumă la o convolutie inversă, care trecută în Fourier este o simplă înmulțire.

Întrucât decizia H_0 sau H_1 se face pe baza unei prăguiri a valorii maxime a corelației, dacă ne-am rezuma exclusiv la rezultatul brut, neprocesat al acesteia, există posibilitatea apariției unor concluzii fals pozitive, din cauza faptului că nu putem seta un prag global. Pentru a explica mai pe îndelete, vom presupune două imagini de analizat, una care a fost făcută cu dispozitivul în cauză și are o singură valoare maximă, peste prag, restul fiind vizibil mai mici,

în timp ce o altă imagine, care a fost făcută cu alt dispozitiv, are în mod întâmplător mai multe vârfuri de valori mari, peste prag, datorită caracterului stochastic al zgomotului. Amândouă sunt caracterizate că apartinând dispozitivului, când de fapt ce-a de-a două este doar rezultatul unei întâmplări nefericite.

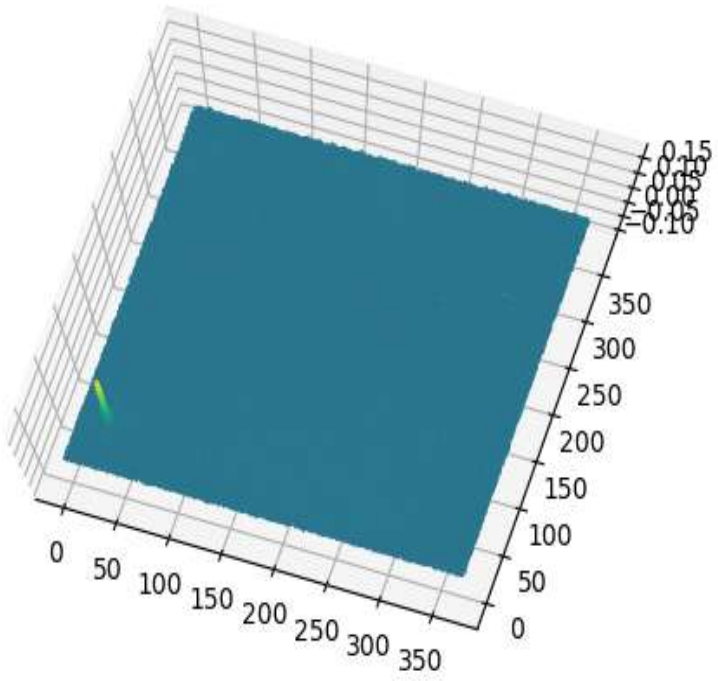


Figura 2.13: Exemplu rezultat corelație

Din acest motiv, se utilizează vârfulfață de energia corelației - PCE, care este doar o varianta puțin mai sofisticată a NCC-ului prezentat in frazele anterioare. Din punct de vedere teoretic, două amprente aparținând unui dispozitiv sunt corelate cu o valoare mare într-o singură poziție, celelalte locații prezentând rezultate mici, chiar negative. Astfel, dacă raportul între maximul corelației și energia matricei de corelație (care nu conține vecinătatea in care se află maximul) este destul de mare, înseamnă că într-adevăr imaginea analizată aparține dispozitivului. Pașii necesari calculului PCE sunt:

- Primește drept parametru de intrare matricea de corelație bidimensională scoasă din corelația normalizată - NCC.
- Identifică locația rezultatului maxim, o stochează și transformă vecinătatea de 3x3 sau 5x5 în care se află această în 0, urmând să calculeze energia noii matrice de corelație.
- Împarte valoarea stocată la rezultatul calculului energiei și de abia acum aplică un prag ales în mod empiric.

În cadrul lucrării, în momentul în care am lucrat cu imagini statice am folosit un prag al PCE-ului de valoare 70, conform rezultatelor experimentale extrase din lucrările de știință din bibliografie.

Acest capitol a fost realizat utilizând informații extrase din lucrarea[5] publicată de Jessica Fridrich, dar și din lucrarea[6] despre filtrul Wiener adaptiv.

2.3 Răspuns fotografic non-uniform - tăiere și modificarea mărimii

Așa cum am menționat anterior, un dezavantaj mare al acestei metode este ușurința cu care poate eșua din cauza sincronizării incorecte. Dacă imaginea în cauză a fost tăiată și scalată, atunci algoritmul inițial simplu nu va fi capabil să recunoască aparatul. Aceste operații constituie transformări geometrice, care cauzează desincronizare și introduc distorsiune adițională ca urmare a procesului de re-eșantionare. Există mai multe metode de abordare a problemei, însă pentru a evita prezentarea unor concepte prea stufoase, ne vom rezuma la cea pe care am ales să o analizăm și să o testăm cu exemple practice. Aceasta se bazează pe testarea de tip forță brută in domeniul spațial pentru a găsi valorile corecte ale parametrilor cu care au fost realizate transformările geometrice.

Acest capitol se rezumă la transformările geometrice de scalare și tăiere, ignorând rotația la un anumit unghi. Acest lucru se bazează pe o decizie similară luată de către Miroslav Goljan și Jessica Fridrich în [7], în urma unui studiu realizat pe studenții acestora. Rezultatele chestionarului au indicat că majoritatea persoanelor nu aplică alte procesări imaginilor după ce le salvează, iar cei care într-adevăr vor să le îmbunătățească, se mulțumesc la operațiile pe care le luăm în considerare. Singurele rotații pe care am putea să le avem in vedere cu ușurință, fără a adaugă un nivel de complexitate ar fi cele la 90 de grade. Chiar și așa, vom vedea în capitolul 2.5 importanța rotației în cadrul videoclipurilor și modul în care putem să o exploatăm pentru a obține rezultate mai bune.

Pentru început, trebuie clarificate procesele de scalare și tăiere. Vom presupune că imaginea originală este în paleta de culori gri, de dimensiune m x n $\mathbf{I}[i][j], i = 1..m, j = 1..n$. De asemenea, vom nota drept $\mathbf{u} = (u_r, u_l, u_t, u_b)$, vectorul care descrie procesul de tăiere, unde u_r,u_l,u_t și u_b reprezintă numărul de pixeli tăiați din imagine din dreapta, sus, stânga și jos. Astfel, dacă aliniăm imaginea originală $\mathbf{I}$ cu imaginea tăiată $\mathbf{Y}$ în colțul din stânga sus, procesul de tăiere va impune o translație spațială T_s determinată de vectorul $s = (u_l, u_t)$ raportat la imaginea originală. Dacă în urma operației de tăiere apare și o operație de scalare de factor r T_r, atunci vom obține o imagine $\mathbf{Z}[i][j], i = 1..M, j = 1..N$, unde $M = r(m - u_r - u_l)$ și $N = r(m - u_t - u_b)$. Mecanismul acestei operații de scalare depinde la nivel granular de algoritmul de eșantionare folosit în spate.

Astfel, problema ce trebuie rezolvată este aceea de a descoperi dacă imaginea Z aparține de un dispozitiv sau nu. La fel ca și în cazul PRNU simplu de mai sus, putem calcula reziduul de zgomot al imaginii, adică pe $\mathbf{K}$, urmând să formăm un test de ipoteze pentru a stabili care dintre $\mathbf{H_0}$ și $\mathbf{H_1}$ este adevărată, pe baza valorii maxime de PCE obținute. Trebuie menționat că am avut de ales între a sub-eșantiona amprenta dispozitivului și a supra-eșantiona reziduul imaginii analizate și am folosit cea de-a două varianta pentru a urmări algoritmul prezentat în lucrarea studiată.

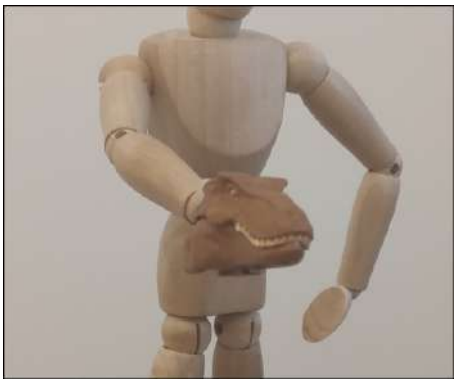
Am menționat în primul paragraf că vom folosi o abordare de genul forță brută în care trecem prin toți parametrii și calculăm PCE-ul în fiecare caz. Pentru a putea face acest lucru, trebuie mai întâi să definim intervalul de valori al parametrilor, utilizând dimensiunile imaginii analizate MxN. Astfel, vom căuta în parametrii de scalare cu valori discrete pornind de la $r_0 = 1$ până la $r_{min} = MAX(M/m, N/n)$. Pentru un parametru de scalare fix r_i nu trebuie calculat PCE-ul pentru toate translațiile posibile, ci numai pentru cele care mișcă imaginea scalată $T_{1/r_i}(\mathbf{Z})$ in interiorul imaginii mari $\mathbf{K}$. Astfel, presupunând că dimensiunile $T_{1/r_i}(\mathbf{Z})$ sunt M/r_i X N/r_i, putem avea următorul interval de translații spatiale $s = (s_1, s_2)$

$$0 \leq s_1 \leq m - M/r_i$$
$$0 \leq s_2 \leq n - N/r_i$$

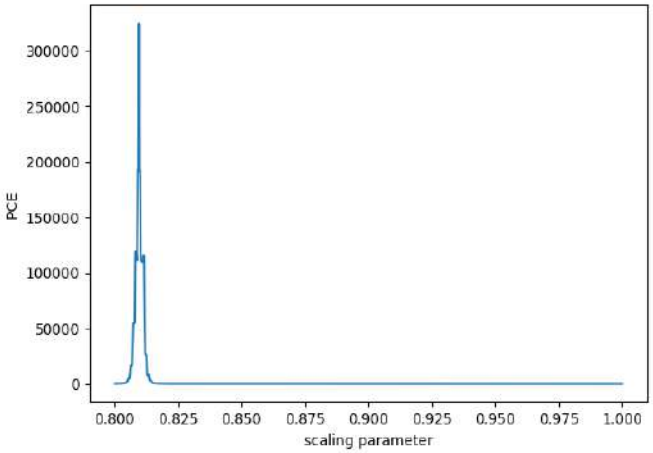
Revenind la parametrii de scalare, valorile discrete ale acestora sunt alese în două moduri,

fiecare dintre acestea apartinând unei faze din algoritm. Prima dintre acestea reprezintă partea mai grosieră, în care trecem mai rapid prin posibilele valori, decrementând cu 0.5% valoarea lui r_i cu fiecare incrementare a indexului, ajungând astfel mai rapid către r_{min}. În urma acestei prime părți, vom avea o valore maximă de PCE pentru un anumit coeficient de scalare. Aici vom întra în cea de-a două faza a algoritmului, în care valorile discrete se află intr-un interval mai mic raportat la cel inițial, dar sunt mult mai apropiate unele de celelalte, realizând ceea ce se numește o căutare "de aur". În oricare dintre faze, dacă valoarea PCE găsită este îndeajuns de mare, atunci algoritmul se oprește și trece la faza următoare sau chiar poate categorisi imaginea ca fiind sub ipoteza H_1. Parametrii de tăiere sunt impliciți calculați în momentul obținerii rezultatului PCE, deoarece acesta este localizat in spațiu.

(a) Imagine originala



(b) Imagine tăiată și scalată



(c) Grafic final

Figura 2.14: Demonstrație de funcționalitate pentru tăiere și scalare

Mai sus am plecat de la imaginea originală din stânga, am tăiat-o în jurul capului de dinozaur din mijloc, am sub-eșantionat-o cu un factor de scalare de 0.81 și am trecut-o prin algoritmul propus de Goljan și Fridrich. Dacă aș fi lăsat algoritmul să ruleze în mod convențional, atunci aș fi obținut rezultatul în aproximativ 30 de secunde, întrucât de îndată ce se apropie de 0.81, atunci valorea PCE-ului deja trece de pragurile impuse. În schimb, din motive experimentale, am continuat algoritmul, ajungând la valori foarte mari ale PCE-ului exact în jurul parametrului de scalare egal cu 0.81 în aproximativ 3 minute.

Acest capitol a fost realizat utilizând informații din lucrarea[7] publicată de Jessica Fridrich și Miroslav Goljan.

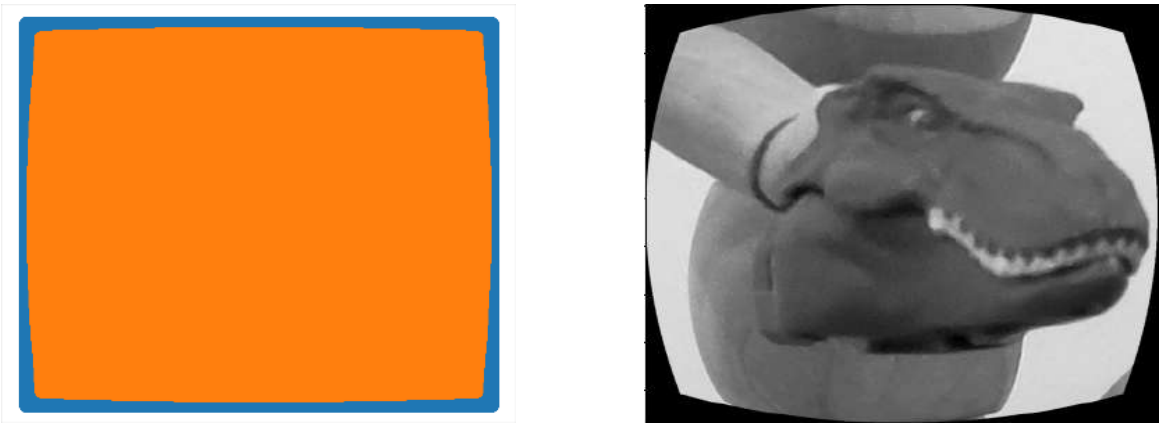
2.4 Răspuns fotografic non-uniform - imagini cu distorsiuni

În ultima perioadă, producătorii de camere foto au început să se bazeze din ce în ce mai mult pe partea de software în scopul de a compensa pentru distorsiunile imaginilor cauzate de către lentile la folosirea măririi sau micșorării (zoom). Acest lucru este înrădăcinat în niște fundamente economice, întrucât realizarea unui set de lentile care să nu impună distorsiuni geometrice mari este mult mai costisitoare decât utilizarea unui micro-procesor care să primească imaginea brută și să o perfecționeze. La fel ca și multe alte concepte din partea de inteligență artificială, această metodă are drept fundament corpul uman, în care partea optică este destul de slab dezvoltată, imaginile pe care le percepem fiind atât de clare datorită post-procesării realizate de către creier.

Acest lucru reprezintă însă un mare impas în cadrul algoritmului pe care l-am utilizat până acum, întrucât post-procesarea respectivă se bazează pe un model matematic care va fi descris în [3.3], combinat cu un proces de interpolare. Astfel, în urma acestor modificări, zgomotul din imagine va deveni desincronizat față de cum a fost obținut de fapt din camera, fiind realizată o oarecare mascare a reziduului PRNU. Din acest motiv, am ales să implementăm o idee propusă de Goljan și Fridrich în lucrarea[8], în care încercăm printr-o căutare de tip forță brută să aflăm istoria corectării de distorsiune, putând astfel să o inversăm și să obținem sincronizarea adevărată a PRNU-ului. Mai exact, ne vom axa pe două tipuri de distorsiuni în mod special, care sunt prezente mult mai des în imagini, distorsiunea de tip butoi și distorsiunea de tip pernă.

Din punct de vedere teoretic, aceste distorsiuni își pot face anunțată prezența prin intermediul valorii distanței focale, care ar trebui să varieze raportat la cât de mult mărim/micșorăm poza în momentul captării. Pe baza acestor considerente am lucrat în prima parte a lucrării, utilizând imagini de lungimi focale asemănătoare, pentru a evita problema distorsiunilor. În schimb, distanțele focale nu sunt întotdeauna prezente în meta-datele unei imagini, deci avem nevoie de modalitatea menționată mai sus pentru a putea obține mai multe informații necesare aplicării unei metode ce ține de PRNU.

Pentru a clarifica exact modurile în care apar distorsiunile și cum sunt ele modificate pentru a deveni poze coerente, ne vom folosi de următoarele figuri:



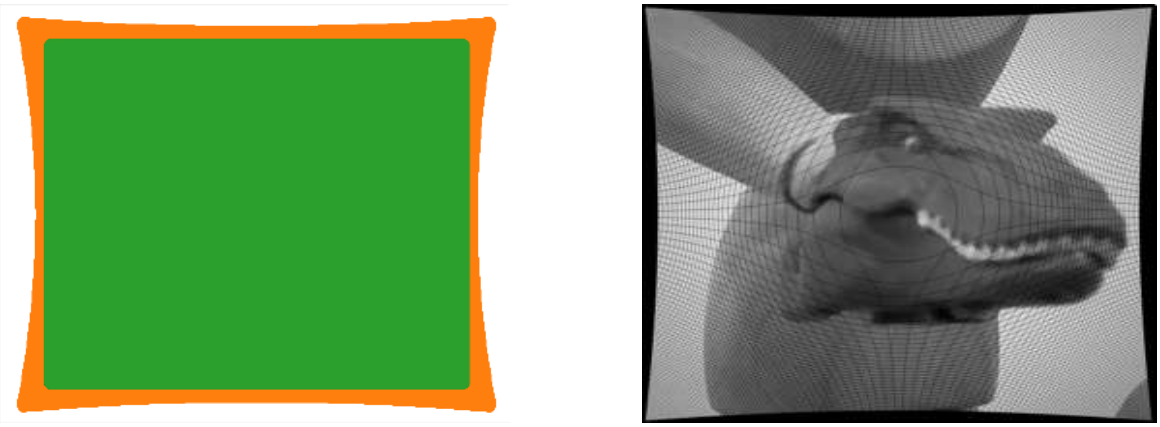
(a) Distorsie de tip butoi conceptual

(b) Distorsie de tip butoi practic

Figura 2.15: Distorsie de tip butoi

În pozele de mai sus, imaginea originală captată de către senzor este reprezentată prin culoarea portocalie în timp ce imaginea procesată, produs final, prezentată utilizatorului este reprezentată prin culoarea albastră. Acest gen de distorsiune apare în general într-un tip specific de lentile, adică cele de unghi-larg. Datorită curburii acestora, obiectivele care se află mai aproape de centrul lentilei vor fi mult mai exagerate, comparativ cu cele aflate mai

departe de centrul acesteia, dând astfel impresia arcuirii catre exterior. Exact acesta este și motivul pentru care în fotografie acest tip de lentile este folosit pentru a cuprinde mai mult din fundal. Astfel, lumina incidentă va trece prin lentilă și nu va ajunge să atingă întregul senzor fotoelectric, ci numai pixelii colorați cu portocaliu, cea ce înseamnă că unii pixeli nu își vor impune amprenta imperfecțiunii asupra reziduului de zgomot.



(a) Distorsie de tip pernă conceptual (b) Distorsie de tip pernă practic

Figura 2.16: Distorsie de tip pernă

În cazul imaginilor de mai sus, imaginea originală captată de către senzor este reprezentată de punctele portocalii, în timp ce imaginea finală, procesată și livrată utilizatorului este reprezentată prin verde. Acest gen de distorsiune este specific lentilelor tele-foto care permit captarea imaginilor la distanțe foarte mari. Datorită curburii acestora, tot ce este apropiat de mijlocul lentilei va fi mărit puțin, în timp ce obiectele mai depărtate de mijloc vor fi exagerate, dând astfel impresia arcuirii către interior. La fel ca și mai sus, lumina incidentă în dispozitiv va trece prin lentilă și va fi deviată într-un mod in care să nu atingă întregul senzor foto-electric, unii pixeli rămânând deci ne-excitați, adică nu își vor imprima amprenta imperfecțiunii asupra reziduului de zgomot.

De asemenea, drept o paranteză, un lucru interesant de observat în cadrul imaginilor pentru distorsiunea de tip pernă, în ce-a de-a două poză, este existența unor linii negre fine de-a lungul imaginii, care ne oferă informații cu privire la modul în care am aplicat distorsiunea asupra obiectului media original. Acestea rămân prezente în imagine deoarece am ales să nu aplicăm și o interpolare, pentru a înțelege rezultatul în cazul absenței acesteia.

Un alt lucru care trebuie clarificat este că în pozele de mai sus, am aplicat distorsiunile de tip butoi și pernă asupra unor imagini normale, strict pentru a obține niște exemple cu scop academic. În cadrul implementării noastre, nu vom face același lucru, pe baza următorului considerent: în momentul în care imaginea a fost captată în mod distorsionat și procesată cu ajutorul interpolării, aceasta a fost deja desincronizata; dacă am impune o altă distorsiune peste o imagine desincronizata, urmând să realizăm altă interpolatie a distorsiunii, noi nu vom obține cea mai apropiată imagine de cea originală, ci una care conține erori sistematice impuse de interpolări. Astfel, vom încerca să inversăm procesul de corectare, adică vom aplica un model matematic invers imaginii in cauză si vom încerca să interpolăm rezultatul în forma inițială a imaginii, obținând un rezultat mai apropiat de cel real.

Algoritmul este unul destul de simplu, în care trecem în mod iterativ prin mai multe valori ale intensității distorsiunilor in intervalul [-0.22, 0.22], calculăm imaginile rezultate și aflăm valorea PCE maximă asociată unei imagini. Partea costisitoare din punct de vedere al eficienței, este că raportat la valorile de PCE, putem mări sau micșora mărimea pasului cu care trecem prin interval. Pentru a clarifica, voi adăuga o poză care să explice modul în care înjumătățim continuu intervalul, cu pași cât mai mici, pentru a trece prin toate posibilitățile. În momentul

în care se gasete o valoare îndeajuns de mare, sau doar am ajuns la pasul minim acceptabil și am încercat toate posibilitățile, atunci se ține minte valoarea parametrului de distorsiune pentru care s-a obținut PCE maxim și se mai trece printr-o căutare "de aur", care are un interval discret mai fin de valori, pentru a găsi exact cea mai mare valoare posibilă.

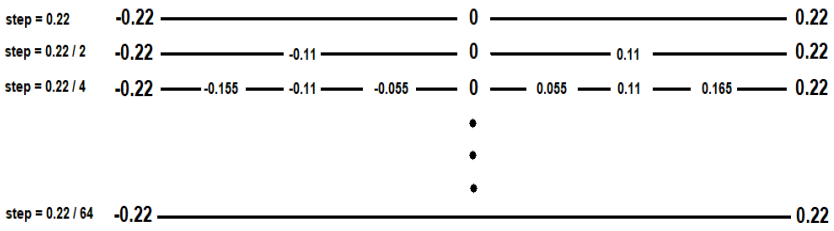


Figura 2.17: Exemplu modificare interval

Atât rezultatele cât și partea matematică vor fi prezentate în capitolele următoare.

Acest capitol a fost realizat utilizând informații din lucrarea[8] publicată de Jessica Fridrich și Miroslav Goljan.

2.5 Video - Răspuns fotografic non-uniform - concepte de baza

Așa cum s-a putut observa până acum, există diverse moduri de a realiza identificarea de camere foto prin intermediul imaginilor. În același timp, metodele utilizate în cadrul video-urilor sunt mult mai puține și mai slab dezvoltate prin comparația acurateții finale. Acest lucru se datorează faptului că numărul de operații de post-procesare în cazul videoclipurilor este considerabil mai mare în comparație cu cazul imaginilor. Drept explicație pentru afirmația anterioară, voi oferi exemplul funcționalității împotriva tremuratului mâinii; aici, software-ul combate în timp real imperfecțiunea umană, rezultatul final fiind un video stabilizat. În cadrul acestei lucrări, vom propune atât o metodă mai simplă de analiză și detectare a semnalelor video nestabilizate, cât și o încercare de eficientizare a analizei semnalelor video stabilizate.

Majoritatea metodelor oferite de analiză a videoclipurilor pleacă de la prezumpția existenței unui video de referință nestabilizat, sau sunt cazuri speciale care utilizează anumite caracteristici specifice numai unor camere, cum ar fi standardele de compresie și informația pe care o pot oferi acestea. În schimb, lucrarea de bază de la care am plecat [9] propune o metodă generală de abordare a problemei, având drept cunoștințe a priori doar niște imagini realizate de aceeași cameră folosită la analiză. Se poate face această presupunere în contextul în care dispozitivele fotografice ale telefoanelor mobile au devenit din ce în ce mai performanțe și o mare parte din persoane aleg să le folosească în defavoarea aparatelor foto.

Înainte de a detalia metoda care stă la baza implementării personale, trebuie explicate diferențele fundamentale între un video și o imagine statică, dincolo de evidenta axa temporală. În general, videoclipurile sunt captate cu o rezoluție mult mai mică decât imaginile normale. Cât timp se înregistrează un semnal video, aparatul va aplica o tăiere asupra matricei de informație primită de la senzorul fotoelectric, iar rezultatul va fi scalat negativ pentru a îl aduce la rezoluția impusă a video-ului. Voi prezenta acest lucru în figurile de mai jos:

(a) Imagine originală (b) Imagine tăiată (c) Imagine scalată

Figura 2.18: Demonstrație înregistrare video

În schimb, procesul de captare nu este atât de simplu, întrucât nu se rezumă doar la tăiere și scalare. Acesta presupune și alte operații, așa cum am menționat mai sus, cum ar fi stabilizarea videoclipului pentru a combate mișcarea mâinilor. Această prelucrare presupune ca partea digitală de camera să estimeze mișcările mâinii și să aleagă în timp real care parte din matricea obținută de la senzorul fotoelectric va fi utilizată pentru a înregistra semnalul. Este evident că va impune desincronizări majore, care ne vor afecta rezultatul final dacă am folosi metode simple de obținere a amprentei PRNU.

Din acest motiv, am ales să folosim drept punct de plecare metoda propusă de Massimo Iuliani, Marco Fontani, Dasara Shulani și Alessandro Piva în [9]. Aceasta se ocupă atât de videoclipurile nestabilizate cât și de cele stabilizate, plecând, așa cum am menționat și mai sus, de la o informație apriorică bine definită și corectă obținută din calcularea amprentei dispozitivului utilizând imagini statice. Pașii pe care ii urmăm în scopul implementării sunt:

1) Calculul amprentei dispozitivului folosind exemple de imagini statistice pre-existente înregistrate de acesta.

2) Determinare dacă semnalul video este stabilizat sau nu - acest lucru se realizează împărțind toate cadrele de tip I în două grupuri, calculând câte o amprentă de grup pentru fiecare și calculând un PCE care se prăguiește cu o valoare prestabilită; dacă rezultatul este destul de înalt, atunci videoclipul este considerat nestabilizat, în celălalt caz el este tratat drept fiind stabilizat.

3) Raportat la categorisirea videoclipului din stadiul anterior, se poate continua în două moduri:

   3.1) În cazul în care videoclipul este nestabilizat, atunci fiecare cadru de tip I este tratat în parte ca fiind o imagine statică, ceea ce înseamnă că nu există rotație de niciun fel. Pentru prima "imagine" se caută parametrii de translație și scalare pentru care valoarea PCE este maximă și se stochează. Pentru următoarele "imagini", întrucât parametrii sunt deja cunoscuți, nu se mai efectuează vreo căutare, ci doar scalarea urmată de translație pentru a obține masca PRNU individuală care va fi folosită pentru calculul măștii PRNU a dispozitivului, conform modului prezentat atât în capitolele anterioare, cât și în [3.1].

   3.2) În cazul în care videoclipul este stabilizat, atunci scalarea, translația și (aici apare) rotația pot fi variabile de la un cadru la celălalt, dar nu cu o variație prea mare. Prin urmare, fiecare cadru de tip I va fi verificat cu o metodă de trip forță brută pentru mai multe valori ale parametrilor de scalare și de rotație, iar cei care oferă o valore a PCE-ului îndeajuns de mare pot fi folosiți pentru a reduce intervalul de căutare pentru restul de cadre. Amprenta finală este calculată utilizând numai cadrele cu cele mai înalte valori de PCE.

4) Având astfel două amprente, una inițială obținută din imagini și una finală obținută din cadrele semnalului video, putem calcula PCE-ul și raportat la valoarea acestuia putem determina dacă videoclipul a fost înregistrat de către dispozitiv sau nu.

Încă un lucru ce trebuie luat în considerare în analiza semnalelor video este modul de stocare al acestora. Ideal ar fi ca ele să fie salvate la dimensiunile lor inițiale, deci fără pierderi de informație. În realitate, acest lucru nu se întâmplă pe sursele principale de semnale video, adică pe platforme de socializare. Atât Youtube cât și Facebook scalează și taie videoclipurile cu parametrii necunoscuți pentru utilizator. Astfel, metodele pe care le folosim trebuie să ia în considerare și probabilitatea utilizării atât unor imagini statice de calitate joasă (descărcate de pe internet) cât și a unor videoclipuri de altă rezoluție față de cea inițială.

Acest capitol are drept referință lucrarea științifică[9] și a fost prezentat pentru a pune bazele algoritmului care va fi dezvoltat în următoarele rubrici. Acolo va fi argumentat și de ce am preferat o altă implementare față de cea actuală.

2.6 Video - Răspuns fotografic non-uniform - implementare personala

Analizând algoritmul propus de Iuliani în [9], am observat câteva moduri în care putem să îl îmbunătățim din punct de vedere al eficienței de computatie și al rezultatului final. Voi prezenta în parte modificările aduse fiecărui pas al imeplementării prezentate în capitolul anterior. Însă cel mai important aspect inovativ pe care l-am utilizat în această lucrare este folosirea atât a cadrelor de tip I, cât și a cadrelor de tip P, lucru ce ne oferă informație adițională de exploatat, pentru un cost mic din punct de vedere temporal.

De-a lungul experimentelor am încercat diverse metode pentru eficientizarea calculului, pe care le voi prezenta în subcapitole diferite. Tot ce se află în acest început de capitol reprezintă "numitorul comun" între toate metodele încercate.

Calculul amprentelor dispozitivelor utilizând imagini statice rămâne asemănător, singura optimizare care poate fi făcută este paralelizarea procesului.

Categorisirea videoclipurilor drept stabilizate sau nestabilizate nu mai este realizată prin împărțirea cadrelor de tip I în două grupuri, calcularea amprentelor pe grup și aflarea valorii PCE dintre ele, chiar dacă acuratețea acestei metode era rezonabilă. Abordarea pe care o propunem se folosește, așa cum am spus mai sus, și de cadrele de tip P. Astfel, vom obține o singură amprenta globală pentru tot videoclipul în mod direct, pe baza unui procent mai mare de cadre, urmând să calculăm o valoare PCE între aceasta și amprenta obținută din imagini statice. Apare întrebarea de genul "de ce să repari ceva ce nu este stricat"? Și este o întrebare pe bună dreptate, întrucât la prima vedere implementarea noastră nu aduce nimic nou, ci doar crește timpul de rulare al primului pas. Acest lucru ar fi adevărat dacă nu am folosi o metodă mai rapidă de calcul al reziduului unui grup de imagini, numită medierea în domeniul spațial - SDA. Așa cum se va putea observa în cadrul rezultatelor experimentale, metoda funcționează și aduce beneficii din punct de vedere al informației disponibile utilizate.

2.6.1 Mediere în domeniul spațial

Motivul principal al existenței acestei metode este de a combate timpul îndelungat de analiză a semnalelor video. Așa cum am menționat și mai sus, majoritatea metodelor actuale fac un compromis și anume: preferă să utilizeze exclusiv cadrele de tip I pentru calculul amprentei dispozitivului întrucât este mai puțin costisitor din punct de vedere computațional. Compromisul apare în momentul in care se alege să nu se folosească informație relevantă pentru obiectivul experimentului.

Acest schimb între eficientă și acuratețe este de fapt problema esențială în cazul analizei videoclipurilor. Se poate afirma asta deoarece pe un calculator personal cu un număr mediu de nuclee și o cantitate decentă de memorie cu acces aleator - RAM, calculul reziduului de zgomot PRNU în mod individual pe fiecare cadru și apoi medierea acestora pentru a afla amprenta generală ar dura mai mult de 24 de ore pentru un videoclip de câteva zeci de secunde. Evident, acest lucru impune niște constrângeri asupra testelor ce pot fi realizate. Neglijând cadrele de tip P, putem risca să pierdem imagini mai puțin stabilizate care să indice o valoare mai mare a PCE-ului pentru anumiți parametri, care ar mări viteza totală de execuție a programului.

În cazul clasic de calcul al amprentei PRNU pe baza unui grup mai mare de imagini statice (cadrele individuale pot fi considerate imagini statice), am prezentat mai sus procesul simplu care este realizat. Se calculează reziduul de zgomot individual pentru fiecare imagine, iar apoi se mediază ponderat toate reziduurile, obținând amprenta dispozitivului. În cadrul metodei SDA însă, apare un pas adițional înainte de calculul PRNU. Aici mai întâi se alege din câte cadre să fie format un grup, raportându-ne la numărul total de imagini disponibile, pe care îl vom numi m. Odată ce s-a stabilit acest parametru, o să îl numim d, vom lua d cadre succesive, le vom aduna valorile pixerilor și le vom media, obținând astfel niște poze fără sens din punct de vedere optic, dar cu multă informație din punct de vedere al zgomotului. Astfel, la sfârșitul primului pas vom avea m/d imagini, care sunt semnificativ mai puține decât numărul inițial de cadre. Acestora le vom calcula reziduurile individual și le vom media la final, obținând astfel amprenta aparatului.

Un avantaj oferit de SDA este că acesta se asigură de imparțialitatea amprentei finale la conținutul propriu-zis al imaginii. Este evident că filtrele intermediare din cadrul calculului de reziduuri individuale PRNU sunt imperfecte, deci permit o eroare sistematică cauzată de obiectele din imagine. Medierea utilizată ar trebui să reducă din influențele acestora asupra rezultatului final.

(a) Imagine originală nr. 1

(b) Imagine originală nr. 2

(c) Imagine originală nr. 3

(d) Imagine SDA

Figura 2.19: Demonstrație SDA

O problemă cu care ne-am confruntat în momentul implementării acestui algoritm este cantitatea uriașă de memorie ocupată în momentul decompresiei videoclipurilor în imagini statice. Cum am menționat și în subcapitolul de teorie anterior, rata de compresie în cadrul semnalelor video este foarte mare, deci rezultatul respectiv era de așteptat. Acest lucru a fost un impas gigantic în contextul în care, pentru rapiditatea experimentului am distribuit datele și am rulat mai multe teste în paralel. Dacă am fi rămas la abordarea inițială, am fi

ocupat întreagă memorie cu numai 3 videoclipuri decompresate. Pentru a soluționa această provocare, am ales să încărcăm treptat cadrele constituente ale videoclipurilor, să le prelucrăm - adică le grupăm și aplicăm medierea specifică SDA - și să ținem minte exclusiv imaginile rezultate din algoritm, restul cadrelor fiind șterse din memorie. În acest fel, după ce trecem prin tot videoclipul rămânem exact cu aceleași imagini mediate cu care am fi rămas dacă încărcam întregul video. Evident, acest lucru presupune un cost temporal adițional, dar este un compromis bun raportat la privirea de ansamblu a experimentului.

Acest subcapitol a fost prezentat utilizând informații din lucrarea [10] publicată de Taspinar, Mohanty și Memon.

2.6.2 Abordarea cu energie liniară

Una dintre cele mai mari probleme ale algoritmului propus de Iuliani este timpul foarte mare de execuție al programului. Presupunând că un semnal video rulează cu 30 de FPS-uri timp de 30 de secunde, acest lucru înseamnă aproximativ 60 de cadre de tip I. Analiza completă, a tuturor combinațiilor de parametri de scalare și de rotație pentru un singur cadru este de 10 minute. Putem deci concluziona că pentru un singur videoclip, analizat raportat la un singur dispozitiv, timpul minim ar fi de 10 ore. Acest lucru este un mare semnal de alarmă.

Din acest motiv, am încercat să găsim alte abordări care să ofere o acuratețe asemănătoare și să fie mai puțin costisitoare computațional. Astfel am ajuns la lucrarea propusă de Miroslav Goljan și Jessica Fridrich, in care se încearcă descoperirea unor parametri asemănători făcând uz de o altă proprietate intrinsecă a dispozitivelor fotografice, textura liniară.

Această metodă a venit drept răspuns necesității de determinare a parametrilor distorsiunilor geometrice corectate, fără a avea vreo informație a priori, deci în lipsa unei amprente PRNU a dispozitivului. Așa cum indică și numele, textura liniară este un șablon de periodicitate liniară în cadrul imaginii. Acesta apare în momentul în care se face conversia analog digital a semnanelor electrice trimise de senzor, urmate de procesul de demozaificare și de compresie.

Astfel, indiferent de tipul de distorsiune, imaginea brută și necorectată va avea un șablon perfect liniar de zgomot periodic. În momentul în care aparatul realizează o pre-procesare a imaginii, corectând imperfecțiunile impuse de lentile, aceasta modifică acest șablon, impunându-i niște forme asemănătoare cu cele observate în figura (2.16.b). Ideea fundamentală a acestei implementări se bazează pe homeostaza fizică a lumii. Mai exact, a fost observat că energia șablonului linear este maximă în momentul în care acesta nu este alterat în niciun fel. Prin urmare, metoda lor se rezumă conceptual la ceva simplu, se inversează distorsiunea impusă pentru fiecare parametru și se calculează energia maximă a șablonului. Valoarea parametrului asociată celei mai mari energii reprezintă intensitatea și tipul distorsiunii. Partea importantă a acestei metode este că oferă rezultate la fel de bune într-un timp mult mai scurt întrucât nu calculează o corelație bidimensională între două imagini de rezoluție înaltă.

Trebuie reamintit că în cadrul semnalelor video, distorsiunile geometrice prezente nu se rezumă numai la cele ale lentilelor, cât și scalări și rotații ale imaginii. Cu toate că nu luăm în considerare distorsiunile lentilelor la videoclipuri în cadrul acestei lucrări, am plecat de la ideea prezentată în lucrarea[11] conform căreia rotația impune la rândul ei o scădere în energie. În acest mod am găsit o metodă mai rapidă de căutare a parametrilor ce ne interesează. Rămâne să fie prezentate rezultatele în [4.7]

Vom tiliza o imagine ce ține de partea de implementare a energiei modelului liniar. Întrucât acest șablon presupune o formă dreptunghică, in cazul nostru, unde lucrăm cu rotații de imagini, a trebuit să realizăm o tăiere a pozelor rotite, prelevând cel mai mare dreptunghi posibil care își păstrează centrul original.

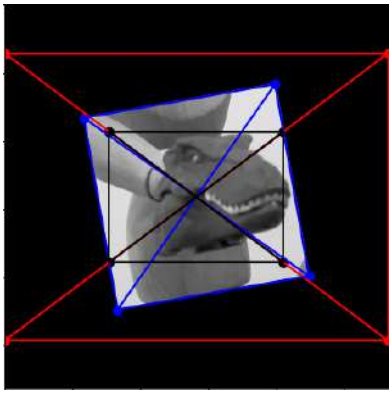


Figura 2.20: Exemplu extragere dreptunghi din imaginea rotită

unde dreptunghiul mare roșu este amprenta dispozitivului, dreptunghiul albastru este conturul imaginii iar dreptunghiul negru este cel extras și utilizat pentru calculul energiei.

Acest subcapitol a fost prezentat utilizând informații din lucrarile [11][12] publicate Miroslav Goljan si Jessica Fridrich.

2.6.3 Abordarea exclusiv cu mediere în domeniul spațial - SDA

Așa cum am precizat în [2.6.1], prima parte din toți algoritmii dezvoltați are la bază tehnica SDA de calcul a unei amprente a dispozitivului folosind cadrele acestuia. Așa cum se va observa în rezultatele experimentale, această idee funcționează cu o acuratețe ridicată la detectarea semnalelor video stabilizate și nestabilizate.

În timpul realizării unor teste cu metode care vor fi prezentate ulterior, am observat un tipar ocazional pentru videoclipurile stabilizate. Va fi subliniat faptul că pentru videoclipurile nestabilizate, valoarea PCE între amprenta SDA scalata maxim și amprenta originală este una foarte mare, deci identificarea este aproape imediată. Revenind la semnalele stabilizate, în momentul în care variam parametrul de scalare pentru amprenta SDA, am surprins un impuls crescător în valoarea PCE, lucrând cu videoclipuri aparținând de dispozitivul analizat. Astfel am obținut ideea de a încerca să identificăm și videoclipurile stabilizate utilizând tot metoda SDA, printr-o căutare de tip forță brută care trece cu o granularitate fină prin valorile posibile ale parametrilor. Comparativ cu celelalte abordări, aceasta durează mult mai puțin și utilizează toate informațiile prezente in semnalul video.

Preconizăm că această metodă funcționează datorită existenței unor cadre a căror procesare de stabilizare nu a fost atât de puternică în comparație cu restul. Astfel, în momentul în care acel gen de cadre sunt prezente într-un număr rezonabil (minimul pe care l-am observat a fost 4 cadre de tip I, deci 124 de cadre în totalitate), atunci corelația poate să detecteze influența acestora asupra amprentei calculate prin SDA și să ofere o valoare mare.

2.6.4 Abordarea cu scalarea cadrelor

Această secțiune va fi una mai scurtă întrucât nu diferă cu mult de algoritmul original, singura divergență fiind modul final de decizie H_1 sau H_0 pentru videoclipurile stabilizate (aceasta va fi prezentată in capitolul urmator). În schimb, restul de algoritm rămâne asemănător cu cel propus de Iuliani. Fiecare cadru de tip I este analizat individual, este scalat, rotit și se calculează valoarea PCE.

În cadrul experimentelor, am renunțat destul de rapid la această variantă întrucât este costisitoare computațional și deoarece mai impune o problema la nivelul acurateții. În momentul în care realizăm o supra-eșantionare, adică o scalare cu un parametru supraunitar asupra unei

imagini, vom folosi o interpolare. Acest proces va impune o eroare sistematică, deci o altă direcție de analiză care poate fi evitată prin utilizarea altor metode. De asemenea, cantitatea de informație conținută de o imagine supra-scalată rămâne constantă sau scade și va fi clar mai mică decât cantitatea de informație din amprenta originală de dimensiuni mai mari.

2.6.5 Abordarea cu sub-eșantionarea amprentei originale

Așa cum am precizat în secțiunea anterioară, metoda care utiliza scalarea cadrelor era nefezabilă din punct de vedere computațional. Acest lucru era cauzat de calculul unei corelații bidimensionale între matrice de dimensiuni de ordinul 3000x4000, care nici măcar nu oferea rezultate atât de concluzive. Din acest motiv, ne-am gândit că sub-eșantionarea amprentei dispozitivului ar putea fi o soluție mai eficientă conform următoarelor considerente:

În primul rând, timpul necesar calculul unei corelații între imagini de 1000x2000 este mult mai mic decât cel anterior. Chiar și dacă realizăm aceste calcule în domeniul Fourier, diferențele de dimensiuni tot joacă un rol important în timpul total de execuție.

În al doilea rând, cantitatea de informație stocată inițial în matricea mare se pierde întrucât imaginea se compactează, dar problema trebuie să fie privită din perspectiva relației între cele două imagini finale analizate. Pentru a clarifica, vom prezența cele două cazuri. În amândouă, inițial, avem o amprentă de dimensiuni mari și o matrice de reziduuri calculată folosind cadrele de dimensiuni mai mici.

- În prima metodă vom supra-eșantiona amprenta mai mică și vom calcula corelația asociată; acea supra-eșantionare însă nu oferă niciun avantaj amprentei determinate din cadre, ba chiar poate să aibă un efect dăunător. Toate acestea în timp ce amprenta dispozitivului, abundență în informație, rămâne intactă.
- În a doua metodă vom sub-eșantiona amprenta mai mare și vom calcula corelația asociată; acea sub-eșantionare oferă un avantaj amprentei determinate din cadre întrucât îi permite acesteia să își păstreze informația intactă și doar scade puțin din informația amprentei dispozitivului, care oricum era foarte mare prin comparație.

Astfel scopul nostru este acela de a oferi amprentei determinate din cadre cel mai propice punct de referință, fără a o modifica și a îi impune erori cauzate de interpolare.

Astfel am clarificat modul de impunere a parametrilor de scalare. Parametrii de rotație rămân impuși amprentei mici, întrucât ea este cea care este compensată rotativ.

Revenind la paralela cu algoritmul de bază, vom preciza faptul că în lucrarea [9] ar urma o căutare foarte riguroasă a parametrilor și mai exact a cadrelor care dau valorile PCE maxime. În unele cazuri, aici este permisă o căutare mai restrânsă a parametrilor pentru cadrele ulterioare găsirii unei valori îndeajuns de mari de PCE, fiind de obicei peste un prag predefinit. Urmăm aceeași logică, însă pasul următor este cel în care ne depărtăm complet de metoda clasică. În aceasta, concluzia se bazează exclusiv pe cadrele I cu cel mai mare PCE, întrucât se formează o nouă matrice de zgomot cu ajutorul lor și se calculează valorea PCE raportată la amprenta originală a dispozitivului.

În cazul nostru, alegem într-adevăr cadrele de tip I cu cel mai mare PCE, însă numai primele trei. Dincolo de acestea, ne interesează pozițiile lor în cadrul semnalului video, pentru a putea preleva cadrele de tip P care urmează după ele. Ideea a fost implementată plecând de la prezumpția că un cadru mai puțin stabilizat va fi urmat de alte cadre la fel de puțin stabilizate, oferind astfel o grupare îndeajuns de mare de reziduuri de zgomot procesate slab, care să consolideze decizia finală a algoritmului. Simultan, căutarea mai restrânsă a parametrilor pornește imediat după primul cadru I, indiferent de valorea PCE finală a acestuia. Inițializarea parametrilor se face cu valorile pentru care s-a găsit valoarea maximă anterioară și intervalul

de căutare este micșorat, eficientizând astfel căutarea. Chiar și în cazul nefericit al unui prim cadru foarte stabilizat, algoritmul se poate redresa întrucât intervalul de căutare este îndeajuns de mare cât să ajungă încet într-un maxim global.

După ce obținem cadrele de tip P, le trecem printr-o căutare foarte restrânsă în jurul valorilor parametrilor asociați cadrului de tip I după care se situează aceastea. Pentru fiecare set de numere calculăm o valoare PCE și o stocăm într-un vector pe care îl vom numi **A**. Aici intervine o altă diferență în comparație cu algoritmul original: decizia finală este luată pe baza unui model matematic probabilistic, în loc de un prag de valori. Acest lucru va fi prezentat mai riguros în [3.4]. Vectorul A conține toate valorile de PCE, fie că acestea țin de H_1, fie de H_0. Acestora le aplicăm funcțiile specifice distributiilor utilizate, obținând doi vectori $\mathbf{A_1}$ și $\mathbf{A_0}$, fiecare reprezentând probabilitățile valorilor de a aparține unei ipoteze adevărate sau unei ipoteze false. Însumând valorile pe fiecare vector, comparăm rezultatele finale, iar în cazul in care suma pentru $\mathbf{A_1}$ este mai mare decât suma pentru $\mathbf{A_0}$, atunci putem spune că videoclipul analizat aparține de dispozitivul a cărei amprentă am folosit-o. În caz contrar, spunem că videoclipul nu aparține de dispozitivul in cauză.

Capitolul 3
Modele matematice

"Încă de la început, trebuie menționat faptul că oriunde în acest capitol, notațiile boldate **X**, **Y** reprezintă vectori de lungime precizată, iar **X**[i] reprezintă al i-lea element din vectorul **X**. De asemenea, poate apărea o notație de genul **X**[i][j] care reprezintă un vector bidimensional, deci o matrice."

3.1 Răspuns fotografic non-uniform - inițial

"Vom pleca de la notația **I**[i] care este semnalul cuantizat la pixelul i, din cei m x n pixeli posibili, unde m și n sunt dimensiunile imaginii. **Y**[i] va fi intensitatea luminii incidente în pixelul i. Pentru ușurința cititorului, vom înlătura indicii pixelilor și vom lucra cu o formă mai generală.

$$\mathbf{I} = g^{\gamma}[(\mathbf{1} + \mathbf{K})\mathbf{Y} + \mathbf{\Omega}] + \mathbf{Q} \tag{3.1}$$

Aici g este factorul de câștig specific fiecărui plan de culoare, γ este factorul de corecție gama, K este amprenta ideală de zgomot PRNU, Ω este combinația altor surse de zgomot, cum ar fi "curentul negru" iar Q este distorsia impusă de compresia JPEG.

În părți ale imaginii care nu sunt total negre, termenul dominant din ecuația de mai sus este intensitatea luminii **Y**. Realizând un factor comun și o expansiune Taylor și păstrând exclusiv primii doi termeni, ajungem la următoarea formă:

$$\mathbf{I} = (g\mathbf{Y})^{\gamma}[\mathbf{1} + \mathbf{K} + \mathbf{\Omega}/\mathbf{Y}] + \mathbf{Q} = (g\mathbf{Y})^{\gamma}(\mathbf{1} + \gamma\mathbf{K} + \gamma\Omega/\mathbf{Y}) + \mathbf{Q} = \mathbf{I}^{(0)} + \mathbf{I}^{(0)}\mathbf{K} + \Theta \tag{3.2}$$

Unde $\mathbf{I}^{(0)} = (g\mathbf{Y})^{\gamma}$ este rezultatul ideal al senzorului fără existența zgomotului. De asemenea, $\mathbf{I}^{(0)}\mathbf{K}$ este termenul PRNU iar $\mathbf{\Theta} = \gamma\mathbf{I}^{(0)}\mathbf{\Omega}/\mathbf{Y} + \mathbf{Q}$ este restul de zgomot.

Pentru a putea estima o valoare a PRNU, trebuie să trecem prin pașii menționați în partea teoretică. Prin urmare, primul lucru pe care trebuie să îl facem este să filtrăm imaginea și să înlăturăm conținutul propriu-zis. Astfel, vom nota cu $\tilde{\mathbf{I}}^{(0)}$ imaginea filtrată.

$$\mathbf{W} = \mathbf{I} - \tilde{\mathbf{I}}^{(0)} = \mathbf{IK} + \mathbf{I}^{(0)} - \tilde{\mathbf{I}}^{(0)} + (\mathbf{I}^{(0)} - \mathbf{I})\mathbf{K} + \mathbf{\Theta} = \mathbf{IK} + \mathbf{\Psi} \tag{3.3}$$

Este mai ușor de estimat termenul PRNU din **W** decât din I deoarece nu mai există obiecte în imagine, numai zgomot. Prin $\mathbf{\Psi}$ am notat suma dintre $\mathbf{\Theta}$ și alți termeni introduși de către filtrarea Wiener.

Teoretic, dacă lucrăm pe o imagine individuală, ar trebui aplicată o mediere și o scădere pe ambele direcții, urmate de încă o filtrare Wiener. Acum voi descrie procesul prin care se ajunge la o amprentă pe care o asociem unui dispozitiv, pașii finali rămânând aceeași, schimbându-se numai partea intermediară.

Vom pleca de la presupunerea existenței unui set de imagini garantat înregistrare de aparatul analizat, $\mathbf{I}_1$, $\mathbf{I}_2$, $\mathbf{I}_3$, ... $\mathbf{I}_d$. Pentru fiecare dintre aceste imagini, zgomotul pixelului i $\mathbf{\Psi}[i]$ este modelat drept zgomot Gaussian cu variație σ^2. Acest termen nu este tehnic independent

de semnalul PRNU **IK**, însă valoarea energiei sale este atât de mică prin comparație încât prezumpția de independență este una rezonabilă.

Din 3.3 putem scrie pentru fiecare imagine k :

$$\frac{\mathbf{W}_k}{\mathbf{I}_k} = \mathbf{K} + \frac{\mathbf{\Psi}_k}{\mathbf{I}_k}, \mathbf{W}_k = \mathbf{I}_k - \tilde{\mathbf{I}}_k^{(0)} \tag{3.4}$$

Pornind de la presupunerea independenței discutată mai sus, atunci posibilitatea observării $\mathbf{W}_k/\mathbf{I}_k$ cunoscându-l pe $\mathbf{K}$ este:

$$L(\mathbf{K}) = -\frac{d}{2}\sum_{k=1}^{d} log(2\pi\sigma^2/(\mathbf{I}_k))^2 - \sum_{k=1}^{d}\frac{(\mathbf{W}_K/\mathbf{I}_k - \mathbf{K})^2}{2\sigma^2/(\mathbf{I}_k)^2} \tag{3.5}$$

Derivând parțial ecuația de mai sus cu elementele individuale ale lui K, putem să obținem cea mai posibilă estimare a PRNU:

$$\frac{\partial L(\mathbf{K})}{\mathbf{K}} = \sum_{k=1}^{d}\frac{(\mathbf{W}_K/\mathbf{I}_k - \mathbf{K})}{2\sigma^2/(\mathbf{I}_k)^2} = 0 => \tilde{\mathbf{K}} = \frac{\sum_{k=1}^{d}\mathbf{W}_k\mathbf{I}_k}{\sum_{k=1}^{d}(\mathbf{I}_k)^2} \tag{3.6}$$

Așa cum am menționat și mai sus, valoarea estimată $\mathbf{K}$ conține și alte artefacte de zgomot. Pentru a le înlătura, vom face o medie pe fiecare linie și coloană și le vom scădea din valoarea K.

$$\begin{aligned} r_i &= \frac{1}{n}\sum_{j=1}^{n}\tilde{\mathbf{K}}[i][j], i = 1..m => \tilde{\mathbf{K}'}[i][j] = \mathbf{K}[i][j] - r_i \\ c_j &= \frac{1}{m}\sum_{i=1}^{m}\tilde{\mathbf{K}}[i][j], j = 1..n => \tilde{\mathbf{K}'}[i][j] = \mathbf{K}[i][j] - c_j" \end{aligned} \tag{3.7}$$

Pentru a nu ne repeta, vom lăsa formulă NCC pentru următorul subcapitol.

Aceste formule au fost preluate din lucrarea[5] aparținând lui Jessica Fridrich.

3.2 Răspuns fotografic non-uniform - tăiere/scalare + rotatie

"Vom începe prin formula NCC ce se folosește la acest algoritm:

$$\begin{aligned} NCC[s_1, s_2; r] &= \frac{\sum_{k=1}^{m}\sum_{l=1}^{n}(\mathbf{X}[k][l] - \bar{\mathbf{X}})(\mathbf{Y}[k+s_1][l+s_2] - \bar{\mathbf{Y}})}{||\mathbf{X} - \bar{\mathbf{X}}|| \cdot ||\mathbf{Y} - \bar{\mathbf{Y}}||} \\ X &= \frac{\tilde{\mathbf{K}}}{\sqrt{\sigma_2^2 + \sigma_1^2(T_{1/r}(\mathbf{Z}))^2}} \end{aligned} \tag{3.8}$$

Următoarea este formula PCE utilizată:

$$PCE(r_i) = \frac{\mathbf{NCC}[s_{peak}; r_i]^2}{\frac{1}{mn-|N|}\sum_{s,s\notin N}\mathbf{NCC}[s; r_i]^2} \tag{3.9}$$

Trebuie precizat că în cazul acestui test de ipoteze, detectorul optimal a fost descris de către Holt, rezultând în niște formule matematice complexe și costisitoare din punct de vedere computațional. Merită menționată existența lor, însă nu intră în interesul acestei lucrări."

Aceste formule au fost preluate din lucrarea[7] aparținând lui Jessica Fridrich si lui Miroslav Goljan.

3.3 Răspuns fotografic non-uniform - distorsiuni

Pentru a putea fie aplica distorsiune fie inversa distorsiunii unei imagini, primul pas care trebuie făcut este de a trece din coordonate carteziene în coordonate polare. Acest lucru se face ușor cu ajutorul formulelor:

$$\begin{aligned} r &= \sqrt{x^2 + y^2} \\ \theta &= arctg(\frac{y}{x}) \end{aligned} \tag{3.10}$$

Având aceste coordonate, am implementat un model destul de simplu pentru distorsiuni de tip butoi și pernă:

$$\begin{aligned} x' &= x_c + (x - x_c)(1 + kr^2) \\ y' &= y_c + (y - y_c)(1 + kr^2) \end{aligned} \tag{3.11}$$

unde (x, y) sunt coordonatele punctului inițial, (x', y') sunt coordonatele punctului distorsionat, (x_c, y_c) sunt coordonatele centrului imaginii, r este distanță radială calculată în funcție de x și y, iar k este parametrul de intensitate al distorsionării.

Trecând totul în coordonate polare și presupunând că centrul optic este în $(0, 0)$ putem ajunge la o formă mai scurtă a unei transformări directe:

$$r' = r(1 + kr^2) \tag{3.12}$$

Cu toate că această formulă a fost utilizată inițial, rezultatele obținute cu ajutorul său nu erau satisfăcătoare, așa că am trecut la aplicarea formulei inverse:

$$r = r'(1 - kr'^2 + 3k^2r'^4 + O(r'^6)) \tag{3.13}$$

Aceste formule au fost preluate din lucrarea[8] aparținând lui Jessica Fridrich si lui Miroslav Goljan.

3.4 Răspuns fotografic non-uniform - video - final

3.4.1 Mediere în domeniul spațial

Prima formulă care necesită explicații este una simplă, cea pentru SDA:

$$\mathbf{I}_i = \frac{\sum_{j=(i-1)d+1}^{id} \mathbf{I}_j}{d} = \frac{\sum_{j=(i-1)d+1}^{id} \left(\mathbf{I}_j^{(0)} + \mathbf{I}_j^{(0)}\mathbf{K} + \mathbf{\Psi}_j \right)}{d} \tag{3.14}$$

urmând ca amprenta finală să fie calculată utilizând formula (3.6).

Aceste formule au fost preluate din lucrarea[10] aparținând lui Taspinar, lui Mohanty si lui Memon.

3.4.2 Distribuție exponențială

În continuare vom prezenta modelul matematic utilizat la decizia finală în privința ipotezelor H_0 sau H_1. Am folosit concluziile unui studiu realizat de către dr. Andrea Montibeller din universitatea UniTN asupra distribuției valorilor PCE în cadrele de tip I și P ale unor semnale video drept punct de plecare. Pentru clarificare, pot fi analizate pozele din anexa 1. Acolo se poate observa forma specifică unei distribuții exponențiale care reiese din rezultatele cuantizate

ale experimentului. Din acest motiv, am ales să le modelăm și în cazul nostru cu o distribuție exponențială.

$$F(x;\lambda) = \begin{cases} 1 - e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{3.15}$$

Am realizat un experiment asemanător celui realizat de dr. Montibeller, doar că la scală mult mai mică. Au fost testate videoclipuri ce aparțineau cert de dispozitivele față de care erau analizate - H_1. S-a trecut prin fiecare cadru de tip I sau P și s-a notat valoarea maximă PCE ce se putea obține pentru o variație a parametrilor precum cea pe care o utilizăm și noi. De asemenea s-au testat aceleași semnale video împreună cu dispozitive de care nu aparțineau - H_0 -, urmându-se același procedeu de notare a valorii PCE maxime. S-a calculat media fiecărei grupări de valori (I & P pentru H_1 și I & P pentru H_0).

Prin urmare, s-au luat următoarele decizii:

- Pentru cazul H_1, în ciuda faptului că rezultatele noastre indicau către valoarea de 35, am ales să utilizăm valorea 38 pentru cadrele de tip I și pentru cele de tip P, pentru a avea o prăguire mai riguroasă conform lucrării[9] prezentate de Iuliani et al..
- Pentru cazul H_0 valoarea parametrului λ s-a ales a fi 20.

În urma execuției unor teste pe acești parametrii s-a observat că valorile obținute erau extrase dintr-un eșantion de exemple mult prea mic (și destul de stabilizat), deci nu erau în conformitate cu valorile reale de PCE.

Astfel s-a ajuns din nou la parametrii obținuți cu ajutorul graficelor din anexa 1, și anume:

- Pentru cazul H_1 am ales să utilizăm valorea maximă dintre distribuțiile cu $\lambda = 491$ și $\lambda = 442$ pentru cadrele de tip I; pentru cadrele de tip P am ales valoarea maximă dintre distribuțiile $\lambda = 247$ și $\lambda = 185$.
- Pentru cazul H_0 valoarea parametrului λ s-a ales a fi 35 pentru toate tipurile de cadre.

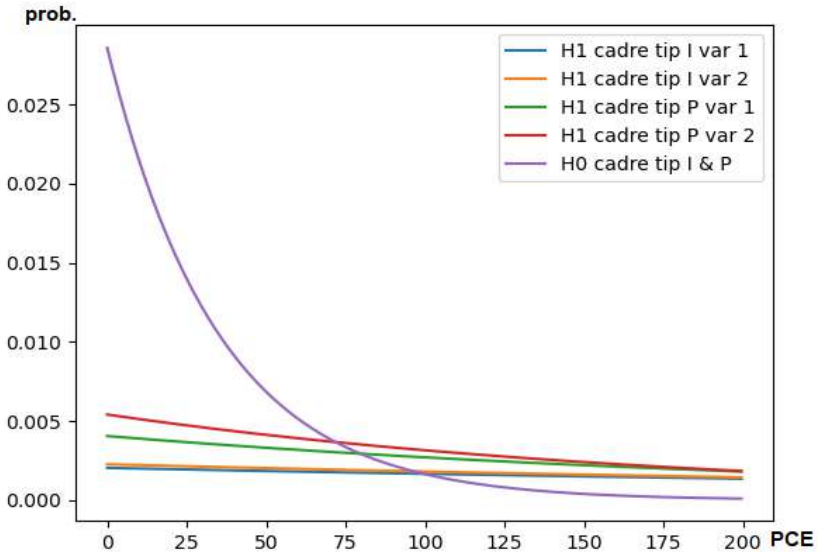


Figura 3.1: Distribuțiile PCE

Capitolul 4
Rezultate experimentale

4.1 Setul de date Dresden

Această baza de date este folosită pentru majoritatea experimentelor de tip PRNU încă din 2010, întrucât prezintă un volum mare de imagini la cea mai mare rezoluție posibilă oferită de dispozitivul digital și, în cazul în care era posibil, care erau stocate într-un mod lipsit de pierderi de informație.

Imaginile au fost captate de pe 73 de dispozitive, acestea fiind categorisite la rândul lor în 25 de modele aparținând unor producători de renume de camere digitale sau telefoane inteligente cum ar fi: Nikon, FujiFilm, Kodak, Canon Ixus, Samsung.

Am ales să folosim această baza de date atât datorită numărul mare de exemple disponibile, cât și grație variației conținutului, (birouri, universități, clădiri, câmpii, plante, figuri) și modului în care acestea au fost captate. Există grupuri de imagini care au fost realizate cu diverse distanțe focale în cadrul aceluiași dispozitiv, oferindu-ne astfel posibilitatea de a trata cazurile de distorsionare a lentilei și a modului în care scalarea și tăierea influențează amprenta PRNU.

Așa cum se va observa în următoarele capitole, am ales să rulăm codul pe două seturi de imagini, unul conținând figuri asemănătoare din punct de vedere al distanței focale, iar celălalt prezentând variații mari ale acesteia, fiind prezente chiar și imagini scalate digital, nu optic.

4.2 Setul de date Vision

Un mare dezavantaj în această nișă de analiză criminalistică a obiectelor media de tip video este reprezentat de numărul redus de exemple ce pot fi utilizate în vederea realizării unor experimente. Așa cum se prezintă și în lucrarea [13], sunt foarte puține baze de date disponibile pentru public, iar cele care deja există nu oferă și posibilitatea testării algoritmilor implementati asupra videoclipurilor stabilizate. Semnalele video disponibile sunt înregistrate dezactivând opțiunea de stabilizare. Acest lucru este cauzat de faptul că stabilizarea este un proces complex, care încă nu a fost rezolvat într-un mod eficient, fie din punct de vedere al acurateții, fie din punct de vedere al costului computațional.

O altă problemă care doar întărește primul paragraf al acestei lucrări este viteza avansului tehnologic la momentul actual. Dispozitivele fotografice propriu-zise devin din ce în ce mai rare, fiind utilizate exclusiv de pasionați sau de profesioniști în domeniu. În secolul 21, majoritatea oamenilor dețin un telefon "inteligent", care să le permită să facă atât poze cât și videoclipuri utilizând același senzor fotoelectric, în cazurile fericite pentru analiști. În cazurile nefericite, ultimele modele de telefoane oferă o gamă de sisteme de lentile (cum ar fi ultimul model Apple) pentru diverse situații și moduri de fotografiere, fapt care evident mărește complexitatea problemei. Prin urmare, experimentele în domeniul criminalistic ar trebui realizate pe exemple cât mai relevante raportat la tehnologia contemporană, adică pe imagini și videoclipuri înregistrate cu precădere de pe telefoane.

Din motivele de mai sus, am concluzionat că un set de date potrivit pentru problema noastră de identificare a despozitivelor folosind semnale video ar fi setul Vision. Acesta oferă în jur

de 30000 de imagini captate cu diferite nivele de calitate (compresie impusă de aplicații cum ar fi Facebook sau Whatsapp) și 2000 de videoclipuri (originale sau modificate de Youtube sau Whatsapp). Un lucru ce trebuie neapărat menționat este că toate obiectele media au fost captate exclusiv cu diferite modele de telefoane inteligente de la companii de renume cum ar fi Asus, Apple, Huawei, OnePlus, Samsung, Sony și altele.

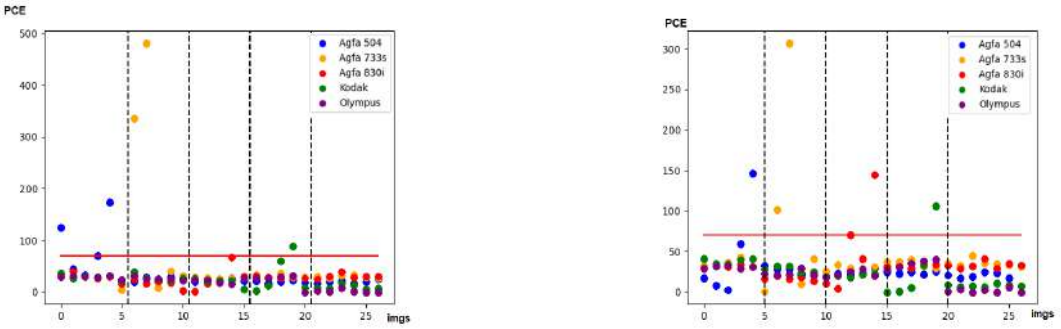
Pentru a finaliza descrierea setului de date, voi prezența și tipurile de video ce au fost înregistrate, din care am avut de ales pentru a ne realiza experimentul:

- Plate - conțin cadre ce au drept obiect de interes suprafețe plate, cum ar fi pereții sau cerul
- Înăuntru - conțin cadre ce au drept obiect de interes interiorul unor cladiri
- Afară - conțin cadre ce au drept obiect de interes zone în aer liber
- Mod static - utilizatorul stătea pe loc în momentul înregistrării videoclipului
- Mod mișcare - utilizatorul se mișca în momentul înregistrării videoclipului
- Mod panrot - mișcare de rotație combinată cu o mișcare dintr-o parte într-alta

Atât acest capitol cât și cel dinainte au fost construite pe baza informațiilor extrase din lucrarea[13] despre setul de date VISION.

4.3 Răspuns fotografic non-uniform - simplu

Această secțiune este destul de fundamentală, scopul ei fiind acela de a sublinia performanța slabă a algoritmului inițial, care nu ia în considerare toate procesările posibile aplicate unei imagini. Prima oară am lucrat pe un set de 5 dispozitive cu 5/7 poze fiecare, având distanțe focale asemănătoare.

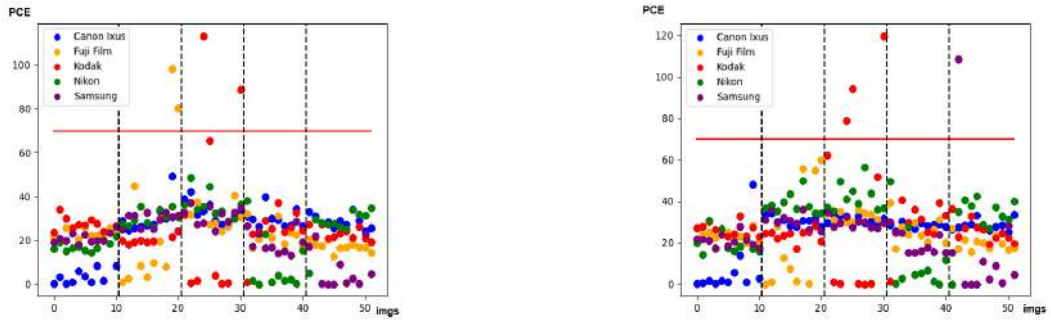


(a) Rezultate PCE cu zgomot de $\sigma^2 = 5$ (b) Rezultate PCE cu zgomot de $\sigma^2 = 2$

Figura 4.1: Rezultate PCE pe set de date cu aceeași distanță focală

Am menționat în cadrul secțiunii [2.6.1] faptul că până și în urma filtrarilor Wiener repetate și a medierii de-a lungul celor două axe, conținutul imaginii își exercită influența asupra amprentei PRNU. În momentul aplicării filtrărilor Wiener, fie ele normale sau adaptive, facem o presupunere în legătură cu valoarea zgomotului prezent în imagine. Cantitatea remanentă de conținut al imagini se raportează la cât de aproape (sau departe) este valoarea aleasă arbitrar de noi față de valoarea reală a variației zgomotului. Se poate observa acest lucru dacă analizăm cele două grafice și observăm că în momentul în care am ales variația egală cu 5, ceea ce înseamnă un zgomot foarte puternic, am reușit să categorisim corectă în jur de 8 imagini din

cele 27 intiale. În celălalt caz, când am ales variația că fiind egală cu 2, am categorisit alte 4 din cele 27 de imagini inițiale. Pe lângă asta, imaginile în cauză probabil au fost supuse unor procesări sau distorsionări pe care le vom arăta în capitolul [4.5]



(a) Rezultate PCE cu zgomot de $\sigma^2 = 5$ (b) Rezultate PCE cu zgomot de $\sigma^2 = 2$

Figura 4.2: Rezultate PCE pe set de date cu distanțe focale diferite

Aceleași lucruri care au fost menționate mai sus pentru cazul cu distanțe focale asemănătoare pot fi repetate și aici. O diferență fundamentală, însă, este procentul mult mai mic de categorisiri corecte ce se datorează distorsiunilor diferite, deci nesincronizării pixelilor de zgomot; anume 4/51 de imagini în cazul cu $\sigma^2 = 5$ și alte 4 în cazul cu $\sigma^2 = 2$

Lucrul fundamental care este observabil în absolut toate graficele este că rata de rezultate fals pozitive este inexistentă. Ceea ce înseamnă că probabilitatea de identificare a unei imagini că aparținând unui alt dispozitiv este infimă. Acest lucru oferă credibilitate in prezumpția de bază a conceptului, anume că fiecare dispozitiv are o amprentă proprie.

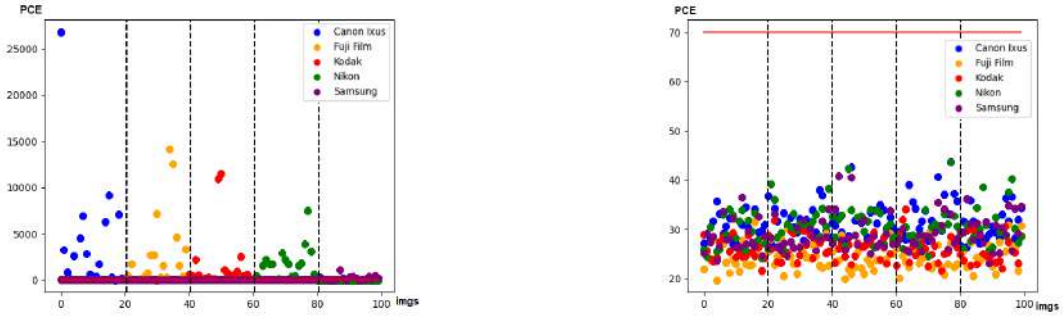
Timpul de calcul al amprentei PRNU individuale pentru o imagine de dimesiuni $(3000, 4000)$ este aproximativ 10 secunde, în timp ce calculul corelației și valorii PCE între aceasta și o amprenta asemănătoare în dimensiune este de 4 secunde. Astfel, pentru un grup de 5 dispozitive cu 5/7 imagini fiecare, avem un timp total de execuție de aproximativ 30 de minute. Aceste lucruri sunt menționate pentru a avea un punct de referință al eficienței viitoarelor metode.

Codul de la bază acestei implementări a fost preluat în proporție de 95% de la doi studenți din Universitatea din Milano. Restul codului implementat a fost scris de către autorul lucrării.

4.4 Răspuns fotografic non-uniform - tăiere și scalare

Vom începe acest subcapitol printr-o mică explicație atașată unui experiment opțional. În general, calculul amprentei unui dispozitiv se face cu un set diferit de imagini față de setul de antrenare, exact ca și în cadrul unui algoritm de machine learning. Acest lucru se datorează faptului că filtrările Wiener setate cu o valoare stabilă a variației zgomotului nu reușesc să înlăture complet conținutul imaginii din reziduul de zgomot PRNU.

Pentru a observa mai clar influența celor două operații menționate în titlu, vom încălca acest standard. Am făcut două teste, pornind de la exact același set de imagini pe care s-a calculat amprenta și folosindu-l și drept set de test. În primul caz, am lăsat imaginile intacte și neprelucrate, în timp ce în al doilea am tăiat o parte de dimensiune 1000x1000 pe care am scalat-o cu o valoare între 0.8 și 1. Rezultatul poate fi observat în figura 4.3, unde în poza din stânga, este evident cum rata de categorisire a imaginilor că aparținând dispozitivelor mamă este de aproape 100%, în timp ce în cazul din dreapta nu există nicio categorisire corectă. Astfel putem afirma cu încredere că cele două operații au un efect major asupra identificării bazate pe PRNU.

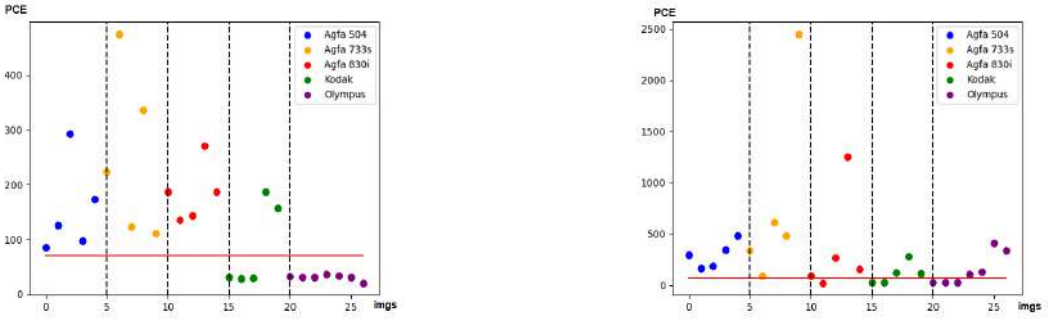


(a) Rezultate PCE normale

(b) Rezultate PCE tăiate si scalate

Figura 4.3: Rezultate PCE pe set de date cu distanțe focale diferite

Dincolo de experimentul de mai sus, am mai efectuat altele două pe un set de date test diferit de cel de "antrenare". Mai exact, am lucrat pe setul utilizat în capitolul anterior, care conținea imagini cu distanțe focale asemănătoare. Ne amintim faptul că acuratețea în acel caz era în jur de 20% cu indulgență. Pentru a vedea diferența, vom studia următoarele grafice:



(a) Rezultate PCE cu zgomot de $\sigma^2 = 5$

(b) Rezultate PCE cu zgomot de $\sigma^2 = 2$

Figura 4.4: Rezultate PCE pentru tăiere si scalare

Este evidentă îmbunătățirea adusă categorisirii, până și în cazul ultimului dispozitiv, reprezentat prin mov (Olympus), care nu a fost identificat deloc în cazurile anterioare. La fel ca și mai sus, imaginile de test au fost tăiate și scalate cu aceleași valori. Trebuie discutate două noțiuni, motivul îmbunătățirii și una deja prezentat mai sus, influența variației.

În primul rând, în momentul în care distanțele focale ale imaginilor sunt salvate în metadatele acestora, ele sunt stocate numai cu două zecimale după virgulă, lucru care la nivel de sincronizare de zgomot poate avea o influență foarte mare, așa cum se poate observa și în cazul de față. Scalând corect imaginea, chiar puțin peste scalarea impusă de noi, am reușit să creștem acuratețea de la 20% la 60%. Acest lucru este doar o reamintire a cât de esențială este sincronizarea poziției zgomotului pentru calculul corelației.

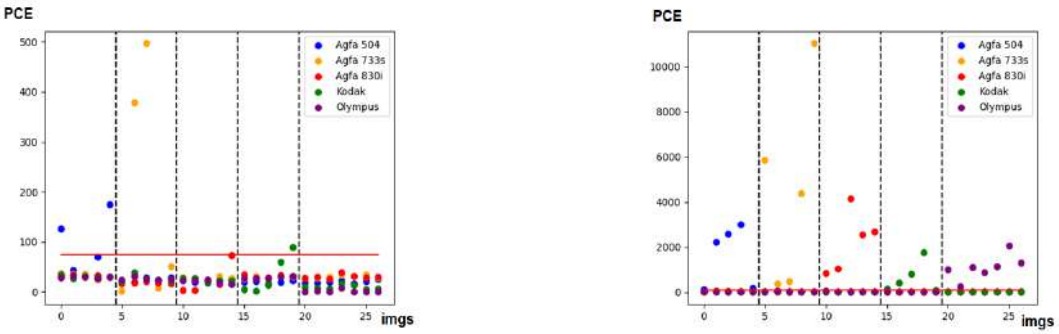
În al doilea rând, modificând valoarea variației zgomotului am reușit să identificăm corect alte 5 imagini, însă aceeași schimbare a scăzut valoarea PCE a tuturor celorlalte și a creat rezultate fals negative. Din nou, apare discuția influenței conținutului imaginii.

Timpul de execuție pentru o imagine care aparține într-adevăr de un dispozitiv analizat este între 1 și 3 minute. Simultan, timpul de execuție pe o imagine care nu aparține de dispozitivul analizat este în jur de 12 minute. Astfel, dacă vrem să rulăm teste pe mai multe imagini raportându-ne la câteva dispozitive, atunci vom ajunge cu ușurință la un timp de execuție de minim 24 de ore.

4.5 Răspuns fotografic non-uniform - imagini cu distorsiuni

Pentru a sublinia îmbunătățirea rezultatelor de către această metodă, le voi prezenta în paralel cu rezultatele originale obținute asupra două seturi de date. Am folosit un prag de valoare 80, care a fost desenat cu roșu în toate graficele. Culorile rămân constante din stânga în dreapta. Testul pe care l-am realizat a fost să luăm cinci dispozitive și amprentele lor asociate, pentru fiecare set de date și un număr de poze realizate garantat de aceste dispozitive. Apoi am analizat fiecare imagine individuală raportat la amprenta fiecărui dispozitiv, pentru a obține atât rezultatele adevărat-pozitive cât și cele fals-pozitive.

Primul set de date presupune 5 imagini pentru fiecare dispozitiv cu distanțe focale asemănătoare dar nu egale:

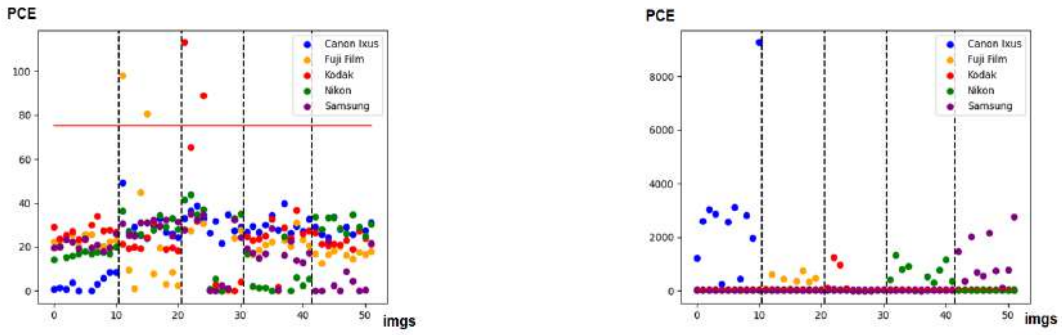


(a) Algoritm PRNU simplu (b) Algoritm PRNU cu distorsiuni

Figura 4.5: Rezultatele implementarii algoritmului pentru distorsiuni pe un set de date cu distante focale similare

În ambele cazuri, valorile PCE ale imaginilor raportate la amprenta cu care sunt analizate sunt reprezentate prin puncte individuale. Se poate observa că în cazul algoritmului PRNU simplu, imaginile distorsionate sunt ocazional detectate în mod corect, datorită suprapunerii valorii distanței lor focale cu distanța focală a imaginilor utilizate pentru calculul amprentelor. Chiar și așa, nu foarte multe dintre acestea trec peste pragul impus, deci nu sunt niște rezultate satisfăcătoare. În cadrul algoritmului care ia în considerare distorsiunile, se observă atât valorile foarte mari ale PCE-ului cât și faptul că toate imaginile sunt categorisite corect.

Al doilea set de date conține 10 imagini pentru fiecare dispozitiv cu distanțe focale care variază:

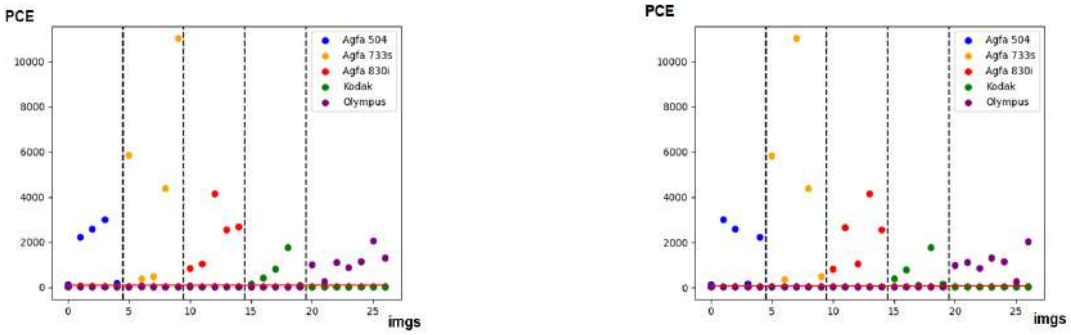


(a) Algoritm PRNU simplu (b) Algoritm PRNU cu distorsiuni

Figura 4.6: Rezultatele implementarii algoritmului pentru distorsiuni pe un set de date cu distante focale variate

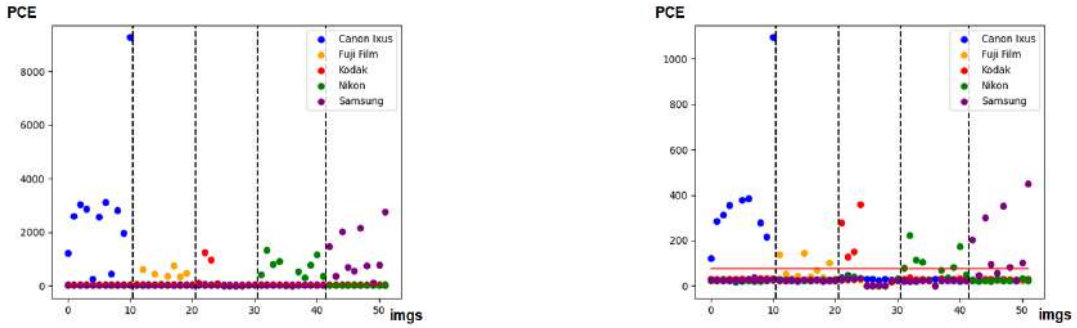
Aici este mai evident cât de inferior este algoritmul PRNU simplu față de cel cu distorsiune, întrucât având distanțe focale diferite, valorile PCE calculate în mod simplu sunt foarte mici, numai 4 dintre acestea fiind caracterizate corect. În același timp, algoritmul cu distorsiuni oferă rezultate net superioare, având în jur de 40 din cele 50 de poze identificate în mod corect. Marja de eroare de 10 imagini neidentificate este cauzată de apariția unor distorsiuni în imagini pe care nu le-am luat în considerare în modelul matematic.

O problemă mare a implementării propuse este durata mare necesară execuției acestuia, drept urmare a dimensiunii imaginilor care sunt 4K. Deci, conform sfaturilor autorilor, am încercat să sub-eșantionăm imaginile cu un coeficient de 3, începând din mijloc pentru a evita impunerea unei erori sistematice și am reușit să reducem timpul de execuție de la 18 ore la 2 ore, rămânând la o acuratețe comparabilă.



(a) Algoritm PRNU fără sub-eșantionare (b) Algoritm PRNU cu sub-eșantionare

Figura 4.7: Rezultatele implementării algoritmului pentru distorsiuni pe un set de date cu distanțe focale similare cu/fără sub-eșantionare



(a) Algoritm PRNU fără sub-eșantionare (b) Algoritm PRNU cu sub-eșantionare

Figura 4.8: Rezultatele implementării algoritmului pentru distorsiuni pe un set de date cu distanțe focale variate, cu/fără sub-eșantionare

Este evident că metoda cu sub-eșantionare oferă rezultate mai bune din punct de vedere al eficienței, însă există un schimb implicit pe care îl facem, oferind o scădere a acurateții pentru timp redus de execuție. Chiar și așa, valorile PCE nu scad foarte mult, rămânând o alternativă bună pentru teste rapide de volum mare.

Timpul de executie mediu pentru aflarea parametrilor pentru o poză este de aproximativ 40 de minute pentru varianta ne-eșantionată si 7 minute pentru varianta eșantionată.

4.6 Video cu implementare Iuliani (supra-eșantionare a amprentei cadrelor)

În cadrul acestei secțiuni nu vom prezenta rezultate cuantizabile ale acurateții întrucât nu am considerat relevant să rulăm o sumedenie de teste pentru un algoritm pe care vrem să îl îmbunătățim. Am rulat individual diverse părți din el și niște experimente izolate și am obținut următoarele concluzii:

- Execuția unei scalări, a unei rotații și a altor câteva operații mici asupra unei imagini statice fără calculul PCE: 15-20 de secunde
- Execuția unei scalări, a unei rotații, a altor câteva operații mici asupra unei imagini statice și inclusiv a calculului PCE: 75-90 de secunde
- Timpul necesar de execuție pentru un cadru de tip I fără valori de plecare pentru parametri: 52 de minute
- Timpul necesar de execuție pentru un cadru de tip I cu valori de plecare pentru parametri: 8 minute
- Valorile finale de PCE nu ofereau valori mai mari de 200, în testele realizate de noi.

Astfel am concluzionat că în ciuda faptului că oferă rezultate bune, acest algoritm este foarte încet și există posibilități de eficientizare dincolo de o simplă paralelizare.

4.7 Video cu energie liniara

Rezultatele din această secțiune vor fi sumare și nu vor prezenta valori ale acurateții. La prima implementare a acestui algoritm, se arată a fi foarte promițător din punct de vedere computațional:

- Execuția unei scalări, a unei rotații și a altor câteva operații mici asupra unei imagini statice fără calculul PCE: 15-20 de secunde
- Execuția unei scalări, a unei rotații, a altor câteva operații mici asupra unei imagini statice și inclusiv a calculului PCE: 25-30 de secunde
- Timpul necesar de execuție pentru un cadru de tip I fără valori de plecare pentru parametri: 15 de minute
- Timpul necesar de execuție pentru un cadru de tip I cu valori de plecare pentru parametri: 3 minute

Marea problema de care ne-am dat seama ulterior a fost că energia liniară nu este deloc concluzivă pentru parametrii pe care îi căutam. Am observat că indiferent de ce imagine ofeream la intrarea în algoritm, parametrul de scalare pentru energia maximă era fie mereu 1, adică cel mai mare parametru posibil. Am explicat acest lucru prin faptul că indiferent de schimbarea în șablonul linear impusă de rotație, energia este mai strâns legată de numărul de valori prezente în imagine. De asemenea, noi nu aplicăm nicio transformare inversă care să contracareze corectura unei distorsiuni geometrice impusă de lentile, deci nu modificăm în vreun fel această energie dincolo de rotații la unghiuri mici.

Prin urmare, am renunțat și la această metodă și ne-am axat pe ultimele două ce urmează a fi prezentate.

4.8 Video cu implementare Mediere în domeniul spațial

În cadrul acestei secțiuni vom privi rezultatele pentru semnalele video stabilizate și pentru cele nestabilizate simultan. De asemenea, trebuie menționat că am folosit mai multe seturi de date de test pentru a verifica fezabilitatea abordării. Argumentul pe care îl aducem pentru fraza anterioară este că stabilizarea impusă în videoclipuri este strâns legată de mișcarea camerei digitale utilizate.

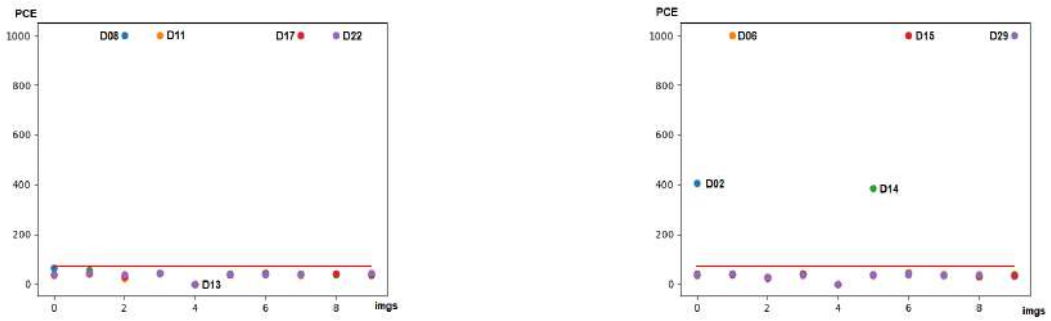
Primul lucru pe care trebuie să îl facem este să prezentăm un tabel în care menționăm numele dispozitivelor și dacă acestea sunt sau nu stabilizate:

	D02	STABILIZAT
	D06	STABILIZAT
	D08	NESTABILIZAT
	D11	NESTABILIZAT
	D13	NESTABILIZAT
	D14	STABILIZAT
	D15	STABILIZAT
	D17	NESTABILIZAT
	D22	NESTABILIZAT
	D29	STABILIZAT

Tabela 4.1: Tipurile dispozitivelor

Astfel am analizat mai multe tipuri de videoclipuri captate de către 10 camere digitale, atât cu amprentele originale (cele deduse din imagini statice) proprii, cât și a celorlalte videoclipuri pentru a ne asigura că nu există rezultate fals pozitive. O convenție pe care ne-am permis să o utilizăm în toate imaginile este aceea de a limita valorile PCE la 1000, pentru a obține grafice inteligibile. Linia roșie este pragul PCE pentru imagini deja cunoscut, de valoare 70.

Primul set de date pe care l-am folosit a fost unul cât se poate de avantajos pentru identificarea dispozitivelor. Spunem asta deoarece conținutul semnalelor video era format din cadre lungi și statice care prezentau suprafețe plate, cum ar fi cerul sau pereții dintr-o cameră, iar camera a rămas fixă; deci șanse mici de stabilizare puternică.



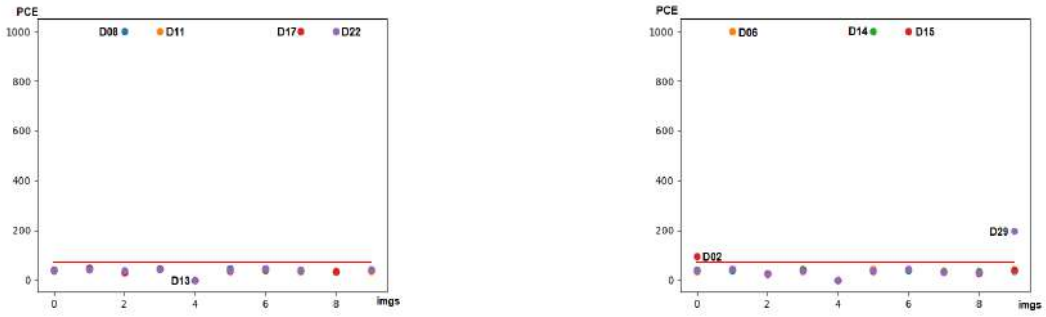
(a) Rezultate pentru dispozitive nestabilizate (b) Rezultate pentru dispozitive stabilizate

Figura 4.9: Rezultate pentru set de date plat si cameră fixă

Se poate observa cu ușurință faptul că rata de rezultate fals pozitive este de 0%, în timp ce rata de adevărat pozitive este de 90%. Toate cele 5 videoclipuri stabilizate au fost corect identificate, în timp ce numai 4 dintre cele nestabilizate au putut obține rezultatul ideal. Acest

lucru ridică niște semne de întrebare în legătură cu procesarea dispozitivului D13 nestabilizat care nu a putut fi identificat. Vom trage concluzii cu privire la acesta după ce trecem și prin celelalte rezultate.

Al doilea set de date utilizat presupune tot o camera digitală fixă, însă conținutul nu mai este la fel de favorabil, întrucât acesta prezintă interiorul unor case. Acest lucru ar trebui să testeze cât de influențat este algoritmul nostru de cantitatea de conținut prezentă în imagine.

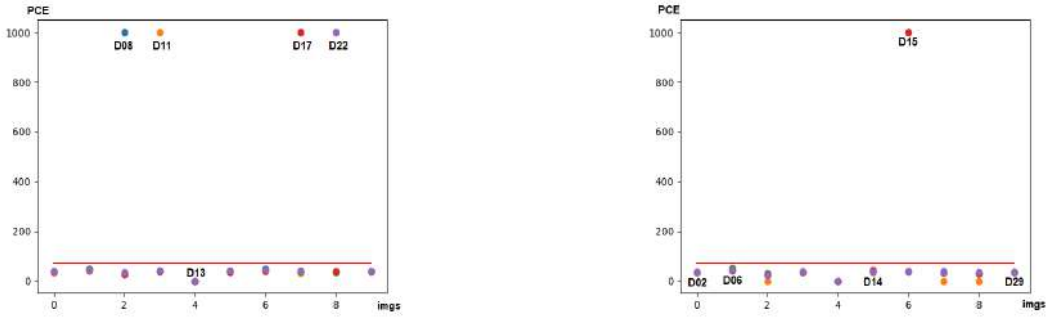


(a) Rezultate pentru dispozitive nestabilizate (b) Rezultate pentru dispozitive stabilizate

Figura 4.10: Rezultate pentru set de date de interior si cameră fixă

Avem din nou plăcuta surpriză de a observa că rata de rezultate fals pozitive este 0% în timp ce rata de rezultate adevărat pozitive rămâne la 90%. În ciuda conținutul lor, semnalele video stabilizate au fost identificate corect în totalitate, chiar dacă unul dintre cazuri (D02) era la limită. În același timp, semnalele nestabilizate rămân la aceleași valori înalte de PCE, mai puțin D13 care își păstrează valoarea sub pragul impus.

Ultimul set de date utilizat este cel mai defavorabil dintre cele alese. În ciuda faptului că obiectele din conținut sunt tot pereți și cerul, marea problema impusă de aceste semnale video este faptul că aparatul digital nu mai este fix, ci se mișcă, deci stabilizarea va fi mult mai puternică.



(a) Rezultate pentru dispozitive nestabilizate (b) Rezultate pentru dispozitive stabilizate

Figura 4.11: Rezultate pentru set de date plat si cameră dinamică

Rata de fals pozitive rămâne într-adevăr la 0%, însă rata de rezultate adevărat pozitive a scăzut la 50% și preconizăm că acest rezultat este încă unul favorabil întrucât am calculat acuratețea finală bazată pe întregul set de date. Așa cum se poate observa, identificarea dispozitivelor nestabilizate rămâne constantă, singurul impas fiind reprezentat de dispozitivul D13 care cel mai probabil prezintă și alte tipuri de prelucrări care au deteriorat amprenta de zgomot PRNU. Problema apare însă în momentul în care privim semnalele video stabilizate și concluzionăm că numai unul dintre acestea a fost corect identificat. Acest lucru s-a întâmplat

cel mai probabil deoarece a existat o porțiune mai puțin stabilizată pe care algoritmul a putut să o exploateze, însă aceasta este o situație ideală. În majoritatea cazurilor pentru videoclipuri in care există mișcare excesivă, ne așteptăm ca această metodă să nu ofere rezultate bune.

Timpul necesar execuției acestui program pentru o imagine analizată raportată la o amprentă a unui dispozitiv este de 90 de minute. Deci timpul total al execuției paralele a testelor noastre a fost de 13 ore. [cont]

Concluzia acestui capitol este că abordarea pe care am ales să o testăm oferă rezultate de încredere într-un timp mult mai scurt decât celelalte metode încercate. Trebuie menționat totuși că există un compromis de acuratețe pentru semnalele video complexe, care conțîn și mișcare excesivă a camerei foto. De asemenea, există și alte tipuri de prelucrări (cazul D13) pe care această metodă nu le ia in considerare.

4.9 Video cu implementare de sub-eșantionare a amprentei dispozitivului

Inițial atât metoda aceasta cât și cea anterioară au fost testate pe un set de date izolat care nu ținea de setul "VISION" folosit în restul experimentelor. Pe baza a două exemple din acesta, unul cu un videoclip stabilizat și altul cu un videoclip nestabilizat s-a realizat testarea incipientă a diverselor implementări propuse. În acele rezultate inițiale s-a observat faptul că abordarea cu sub-eșantionarea amprentei dispozitivului oferă valori PCE mai mari decât abordarea cu supra-eșantionarea amprentei obținute din cadre, într-un timp ce demonstra un slab câștig din punct de vedere al eficienței. Chiar și așa, metoda SDA oferea rezultate mult mai bune și mult mai rapide în comparație, deci s-a decis ca majoritatea experimentelor să fie săvârșite asupra ei.

Așa cum am menționat și mai sus, timpul alocat și cerințele computationale ale experimentelor nu ne-au permis să realizăm o cantitate foarte mare de experimente și asupra acestei metode.

Am precizat în partea de modele matematice [3.4.2] că inițial am început cu un anumit set de valori pentru distribuțiile exponențiale, însă am observat rapid că în ciuda unei ratei de rezultate adevărat pozitive de 60%, a apărut o problemă: rata de rezultate fals pozitive care a rămas 0 de-a lungul întregii lucrări a început să crească. Acest lucru a reprezentat un semnal de alarmă pentru schimbarea valorilor. Astfel, ne-am întors la valorile inițiale obținute din sursele externe menționate și am rulat setul de teste. Aici rezultatele de acuratețe au fost dezamăgitoare, fiind identificat corect un singur videoclip din cele 6 testate; aici totuși rata de fals pozitive a rămas 0. Timpul de execuție a fost în jur de 5 ore pe videoclip, care este mai scurt decât cele 6 ore asociate unei execuții normale de algoritm original.

Drept concluzie pentru acest subcapitol vom spune că metoda propusă se arată a fi promițătoare, însă cu alte valori pentru parametrii distribuției exponențiale decât cele folosite în experimentele noastre. Această metodă este mai slabă prin comparație cu cea ce utilizează SDA.

Concluzii și contribuții personale

Cel mai important lucru care trebuie observat drept numit comun al tuturor experimentelor și metodelor abordate este că rata de rezultate fals pozitive este 0%, acest lucru consolidând prezumpția fundamentală a conceptului de PRNU: fiecare camera digitală are propria să amprenta de zgomot.

În cazul testelor realizate asupra imaginilor, putem pleca de la observația necesității de sincronizare între amprenta dispozitivului și amprenta pozei analizate. În lipsa unei suprapuneri corecte de valori ale zgomotului, corelația nu va detecta similitudini între semnale și va oferi rezultate PCE slabe, așa cum se poate observa în cazul algoritmului PRNU brut. Corectarea acestei lipse de sincronizare duce la obținerea unei rate de identificare mare, peste 70%, în comparație cu 20% anterior. Acest lucru poate fi realizat fie prin inversarea operațiilor de tăiere și scalare, fie prin neutralizarea corecturii distorsiunilor impuse de lentile. Ambele cazuri reprezintă transformări geometrice ce pot fi tratate ușor cu modele matematice, așa cum am observat în cadrul rezultatelor experimentale.

În toate metodele de mai sus s-a mai observat și influența conținutului din imagini prin imperfecțiunea filtrelor Wiener utilizate. Chiar și în urma unor filtrări de acel, gen modificate pentru a fi adaptive, tot se pleacă de la o presupunere în legătură cu nivelul de zgomot prezent în imagine, care evident nu va fi egal pentru toate pozele. Prin urmare, în mare parte din analize, valoarea arbitrară aleasă pentru variația zgomotului joacă un rol important în rezultatele finale.

În cazul semnalelor video, s-a încercat îmbunătățirea unui algoritm publicat și acceptat de comunitatea științifică în anul 2019. Din toate metodele și variațiile testate, numai două dintre acestea au oferit rezultate decente într-un timp de execuție mai scurt. Una dintre ele, și anume metoda SDA, este destul de apropiată de abordarea propusă în lucrarea[10] de către Memon, însă acest lucru a fost descoperit numai după implementarea ei. Chiar și așa, există diferențe care să îi permită studentului să își însușească ideea. Din punct de vedere al costului computațional, s-a observat o scădere de timp de aproximativ 4 ore, ajungând de la 6 ore la 90 de minute. Întrucât nu am avut hardware-ul necesar realizării unor teste de dimensiuni mari, ne-am rezumat la cele trei seturi de date care au oferit rezultate promițătoare atât pentru videoclipurile nestabilizate cât și pentru cele stabilizate. Cealaltă metodă este mai apropiată de cea originală, atât din punct de vedere al logicii de implementare, cât și a costului computațional. Întrucât ne-am axat pe metoda prezentată anterior, nu am avut timp să realizăm o testare la fel de riguroasă și pe aceasta. Chiar și așa, rezultatele obținute par satisfăcătoare.

Drept posibile idei de dezvoltări următoare avem:

- Pentru algoritmul SDA: luarea în calcul și a posibilelor rotații la unghiuri mici și testarea pe un set mult mai voluminos de date.
- Pentru algoritmul cu sub-eșantionare: găsirea unei metode de a determina cadrele mai puțin stabilizate pentru a evita o căutare exhaustivă de parametrii pe toate cadrele de tip I și experimentarea pe mai multe tipuri de semnale video.
- În cazul în care nivelul de acuratețe pe un set de date mai mare este îndeajuns de bun, se poate publica o lucrare științifică în scopul îmbunătățirii metodelor actuale de identificare a videoclipurilor stabilizate.

Bibliografie

[1] Luisa Verdoliva. Media forensics and deepfakes: An overview. *IEEE Journal of Selected Topics in Signal Processing*, 14(5):910–932, 2020.

[2] Hany Farid. Exposing digital forgeries from jpeg ghosts. *IEEE Transactions on Information Forensics and Security*, 4(1):154–160, 2009.

[3] Micah K. Johnson and Hany Farid. Exposing digital forgeries through chromatic aberration. In *Proceedings of the 8th Workshop on Multimedia and Security*, MMamp;Sec '06, page 48–55, New York, NY, USA, 2006. Association for Computing Machinery.

[4] Free xray of hands stock photo. `https://www.freeimages.com/photo/xray-of-hands-1526780`. Accessed: 2021-06-13.

[5] Jessica Fridrich. Digital image forensics. *IEEE Signal Processing Magazine*, 26(2):26–37, 2009.

[6] M. Kivanc Mihcak, I. Kozintsev, and K. Ramchandran. Spatially adaptive statistical modeling of wavelet image coefficients and its application to denoising. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258)*, volume 6, pages 3253–3256 vol.6, 1999.

[7] Miroslav Goljan and Jessica Fridrich. Camera identification from cropped and scaled images - art. no. 68190e. *Proceedings of SPIE, Media Forensics and Security*, 03 2008.

[8] Miroslav Goljan and Jessica Fridrich. Sensor-fingerprint based identification of images corrected for lens distortion. *Proceedings of SPIE - The International Society for Optical Engineering*, 8303:12–, 02 2012.

[9] Massimo Iuliani, Marco Fontani, Dasara Shullani, and Alessandro Piva. Hybrid reference-based video source identification. *Sensors*, 19:649, 02 2019.

[10] Samet Taspinar, Manoranjan Mohanty, and Nasir Memon. Camera fingerprint extraction via spatial domain averaged frames. *IEEE Transactions on Information Forensics and Security*, 15:3270–3282, 2020.

[11] Miroslav Goljan and Jessica Fridrich. Estimation of lens distortion correction from single images. *Proceedings of SPIE - The International Society for Optical Engineering*, 9028, 01 2014.

[12] M. Goljan, J. Fridrich, and Matthias Kirchner. Image manipulation detection using sensor linear pattern. *electronic imaging*, 2018:119–1–119–10, 2018.

[13] Dasara Shullani, Marco Fontani, Massimo Iuliani, Omar Alshaya, and Alessandro Piva. Vision: a video and image dataset for source identification. *EURASIP Journal on Information Security*, 2017:15, 10 2017.

Anexa 1
Imagini din surse externe

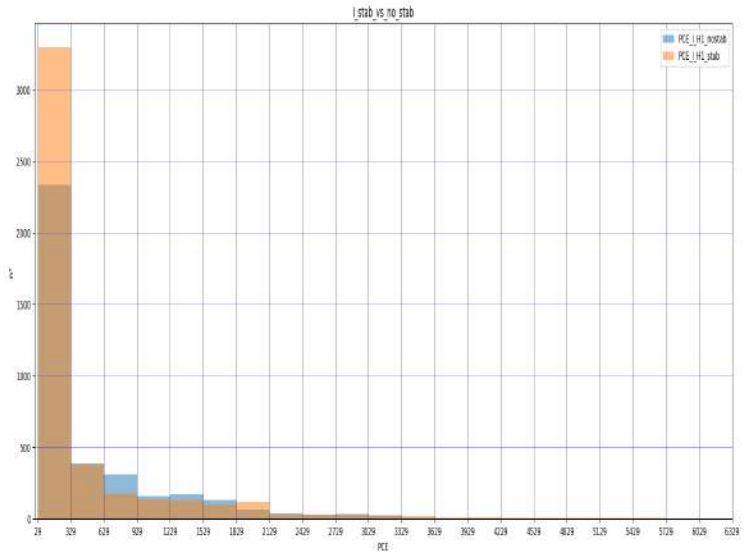


Figura 1.1: Distribuție PCE H_1 pentru I - video stabilizat & nestabilizat

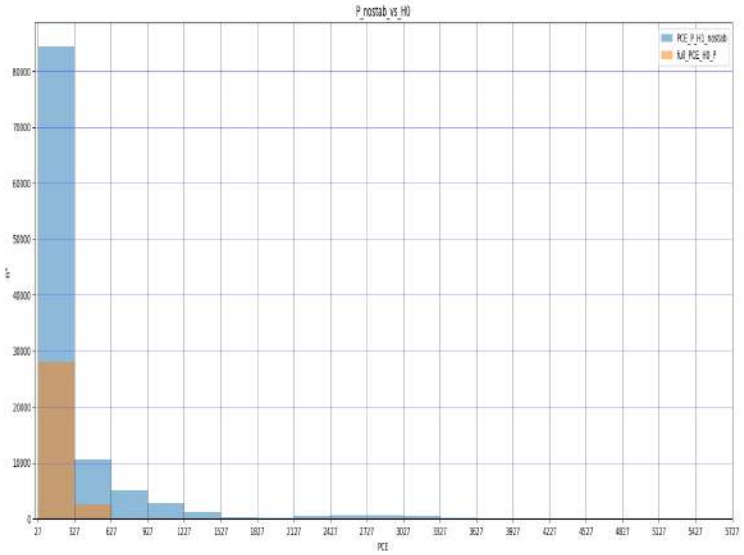


Figura 1.2: Distribuție PCE H_1 pentru P și distribuție PCE H_0 pentru I & P - video nestabilizat

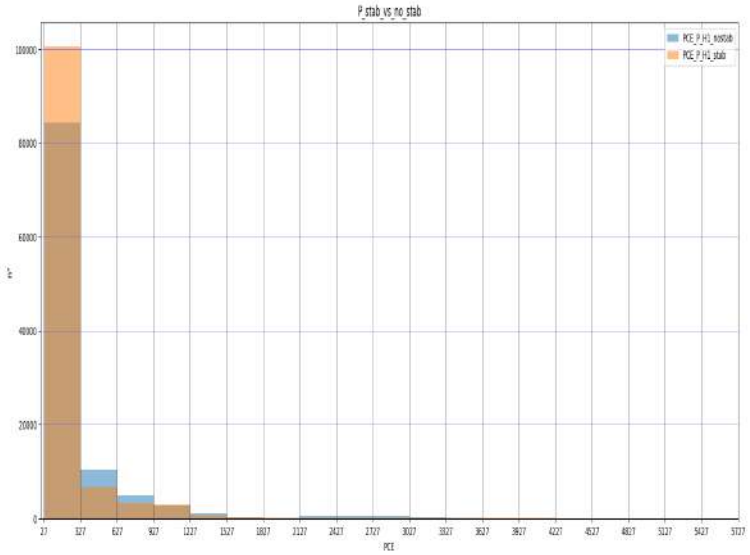


Figura 1.3: Distribuție PCE H_1 pentru I - video stabilizat & nestabilizat

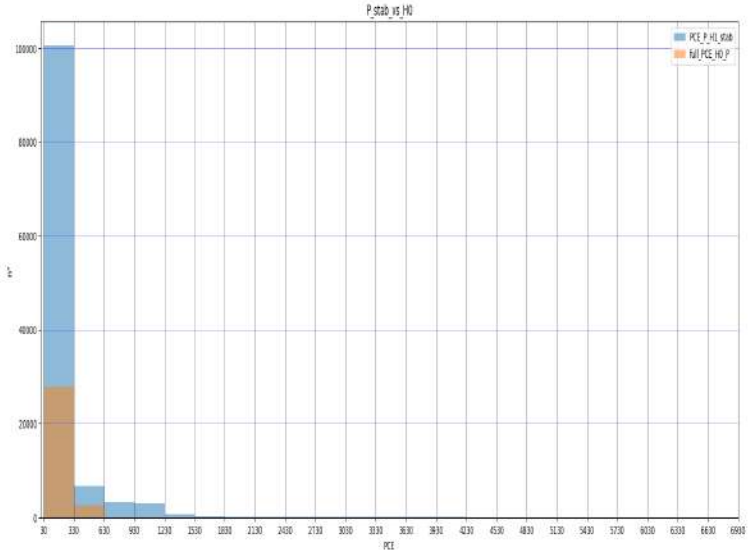


Figura 1.4: Distribuție PCE H_1 pentru P și distribuție PCE H_0 pentru I & P - video stabilizat

Anexa 2
Implementarea algoritmului

Aici este prezentat codul utilizat pentru calcularea valorii PCE:

```
import os
import warnings
import numpy as np



#cc -> cross-corr
# ranges -> they are different than [1,1] if the noise residual and the image fingerprint are not aligned and needs to be computed

def pce(cc: np.ndarray, ranges, neigh_radius: int = 2) -> dict:
    """
    PCE VALUE
    """
    assert (cc.ndim == 2)
    assert (isinstance(neigh_radius, int))
    out=dict()

    cc_inrange = cc[:ranges[0]+1, :ranges[1]+1]; #print('shape:', np.shape(cc_inrange))
    max_idx = np.argmax(cc_inrange.flatten())
    max_y, max_x = np.unravel_index(max_idx, cc_inrange.shape); #print('location:', max_y, max_x)

    peak_height = cc[max_y, max_x]

    cc_nopeaks = cc.copy()
    if (ranges[0]>=neigh_radius) & (ranges[1]>=neigh_radius):
        cc_nopeaks[max_y - neigh_radius:max_y + neigh_radius+1, max_x - neigh_radius:max_x + neigh_radius+1] = 0
    else:
        if(ranges[0]<neigh_radius) & (ranges[1] >= neigh_radius):
            cc_nopeaks[max_y - neigh_radius:max_y + neigh_radius+1,0:5]=0;
        if(ranges[0]>=neigh_radius) & (ranges[1] < neigh_radius):
            cc_nopeaks[0:5,max_x - neigh_radius:max_x + neigh_radius+1,]=0;
        if(ranges[0]<neigh_radius) & (ranges[1] < neigh_radius):
            cc_nopeaks[0:5,0:5]=0;


    pce_energy = np.mean(cc_nopeaks.flatten() ** 2)

    out['peak'] = (max_y, max_x)
    out['pce'] = abs((peak_height ** 2) / pce_energy )
    out['cc'] = peak_height


    return out
```

Aici este prezentat codul pentru inversarea corecției radiale:

```
#import tensorflow_addons as tfa
import matplotlib.pyplot as plt
from scipy.interpolate import interp1d
import numpy as np
from skimage import io,color
from PIL import Image
import os


def radial_cordinates(M, N):
    center=[M//2,N//2]
    xi,yi=np.meshgrid(np.arange(N),np.arange(M))
    xt=xi-center[1]
```

```
    yt=yi-center[0]
    r=np.sqrt(xt**2 + yt**2)
    theta=np.arctan2(xt,yt)
    R=np.sqrt(center[0]**2 + center[1]**2)
    r=r/R
    return r, theta, R, xi, yi, center, xt, yt

def distortfct(r,k):
    s = r*(1 - k*(r**2) + 3*(k**2)*(r**4))
    return s

def imdistcorrect(img, k, r, theta, R, xi, yi, M, N):
    s=distortfct(r, k)
    s2=s*R
    v =s2 * np.cos(theta)
    u= s2 * np.sin(theta)
    #PIPELINE BARRELL
    if np.amin(np.round(v + np.abs(np.amax((M)) // 2))) < 0:
        # print('barrell')
        v = np.round(v + np.abs(np.amax((M))//2))
        u = np.round(u + np.abs(np.amax((N))//2))
        u = u.astype(np.int32)
        v = v.astype(np.int32)
        dist = np.zeros([np.max(v+1), np.max(u+1)])
        dist[yi[np.logical_and(v< M-1, v>0)*np.logical_and(u< N-1, u>0)], xi[np.logical_and(v<
    M-1, v>0)*np.logical_and(u< N-1, u>0)]] = img[v[np.logical_and(v< M-1, v>0)*np.logical_and
    (u< N-1, u>0)], u[np.logical_and(v< M-1, v>0)*np.logical_and(u< N-1, u>0)]]
    else:
    #PIPELINE PINCUSHION
        # print('pincushion')
        v = np.round(v + np.abs(np.amax((M)) // 2))
        u = np.round(u + np.abs(np.amax((N)) // 2))
        u = u.astype(int)
        v = v.astype(int)
        dist = np.zeros([np.max(yi+1), np.max(xi+1)])
        dist[yi, xi] = img[v, u]

    dist = bilinear_interpolation(dist)
    auxiliary=np.where(np.logical_and(v< M-1, v>0)*np.logical_and(u< N-1, u>0)==True)
    dist=dist[np.amin(auxiliary[0]):np.amax(auxiliary[0]),np.amin(auxiliary[1]):np.amax(
    auxiliary[1])]
    return dist

def bilinear_interpolation(img):
  x = np.arange(len(img))
  img_new=np.zeros(img.shape)
  aux_x=[]
  aux_y=[]
  for i in range(len(img[0])-1):
      ix = np.where(img[:,i] != 0)
      if (len(ix[0])!=0):
          f = interp1d(x[ix],img[ix,i], fill_value='extrapolate')
          img_new[x[ix[0][0]:ix[0][-1]],i] = f(x[ix[0][0]:ix[0][-1]])
          aux_y.append(x[0:ix[0][0]])
          aux_y.append(x[ix[0][-1]:len(img[0])])
          aux_x.append(i*np.ones(len(x[0:ix[0][0]])+len(x[ix[0][-1]:len(img[0])])))
  x=np.arange(len(img[0]))
  for i in range(0,(len(img)-1)):
      ix = np.where(img[i,:] != 0)
      if (len(ix[0])!=0):
          f = interp1d(x[ix],img[i,ix], fill_value='extrapolate')
          img_new[i,x[ix[0][0]:ix[0][-1]]] = f(x[ix[0][0]:ix[0][-1]])
          aux_x.append(x[0:ix[0][0]])
          aux_x.append(x[ix[0][-1]:len(img[0])])
          aux_y.append(i*np.ones(len(x[0:ix[0][0]])+len(x[ix[0][-1]:len(img[0])])))
  aux_x=np.concatenate(np.array(aux_x)).astype(int)
  aux_y=np.concatenate(np.array(aux_y)).astype(int)
  img_new[aux_y,aux_x]=0
  return img_new

def crop_center(img,cropx,cropy):
    startx = 0
    starty = 0
    return img[starty:starty+cropy,startx:startx+cropx]

def residual():
```

```
    img= Image.open(r'C:\Users\PC\Desktop\LICENTA\prnu_aron\images\fingerprint_source\Agfa_DC-504_0_1.jpg')
    im_arr = np.asarray(img)
    res=(color.rgb2gray(im_arr)*255).astype(np.uint8)
    M, N = [res.shape[0],res.shape[1]]
    M = np.asarray(M)
    N = np.asarray(N)
    r, theta, R, xi, yi, center, _, _ = radial_cordinates(M, N)
    k = -0.22
    while k < 0.23:
        W_ld = imdistcorrect(res, k, r, theta, R, xi, yi, M, N)
        plt.figure(),plt.imshow(W_ld)
        k = k + 0.01
```

Aici este prezentat codul folosit pentru încărcarea secvențială a cadrelor în memorie:

```
import os
import cv2
import subprocess
from skimage import color
import matplotlib.pyplot as plt
import numpy as np


def get_frame_types(video_fn):
    command = 'ffprobe -v error -show_entries frame=pict_type -of default=noprint_wrappers=1'
    out = subprocess.check_output(command + ' ' + video_fn,shell=True).decode()
    frame_types = out.replace('pict_type=','').split()
    return zip(range(len(frame_types)), frame_types)

def save_i_keyframes(video_fn,flag,count,batch_size):
    file_name_save = '~\\aron_prnu\\videos_PRNU\\video_repo'+ video_fn.split('\\')[-1].split('.')[0]
    i_frames=[]
    p_frames=[]
    all_frames=[]
    if not os.path.isfile(file_name_save):
        frame_types = get_frame_types(video_fn)
        for x in frame_types:
            if x[1]=='I':
                i_frames.append(x[0])
            if x[1]=='P':
                p_frames.append(x[0])
            all_frames.append(x[0])
        asd = all_frames
    if flag == 1:
        asd=[]
        max_size = len(all_frames)
        begin = count*(15*batch_size)
        if begin + 15*batch_size > max_size:
            end = max_size
        else:
            end = begin + 15*batch_size
        if len(all_frames):
            cap = cv2.VideoCapture(video_fn)
            for frame_no in np.arange(begin,end):
                cap.set(cv2.CAP_PROP_POS_FRAMES, int(frame_no))
                ret, frame = cap.read()
                asd.append(frame)
            cap.release()
        else:
            print ('Noframes in '+video_fn)
        i_frames = np.array(i_frames)
        p_frames = np.array(p_frames)

    return asd, i_frames, p_frames
```

Aici este prezentat codul pentru analiza semnalelor video exclusiv cu SDA:

```
from multiprocessing import cpu_count, Pool
import numpy as np
import prnu
import pce
from scipy.ndimage import zoom
import sys
import save_all_video_frames_to_npy
from scipy.ndimage import rotate
```

```
from skimage import color,io
import os

def main():
    folder_path="//home//andreamontibeller//vision//"+sys.argv[1]
    for i in sorted(os.listdir(folder_path)):
        if ('indoor' in i) & ('still' in i):
            name = i
            break
    video_fn = folder_path + '//' + name
    max_size , _ , _ = save_all_video_frames_to_npy.save_i_keyframes(video_fn , 0 , 0 , 0 )
    m = len(max_size)
    d = np.ceil(m/150)
    imgs_for_sda_fingerprint = []
    ir = []
    for count in range(int(np.ceil(m/(d*15)))):
        print(count)
        frames,i_frames_positions,p_frames_positions = save_all_video_frames_to_npy.
    save_i_keyframes(video_fn, 1 , count , d)
        ir.append([i_frames_positions,p_frames_positions])
        for i in range(15):
            if ((i+1)*d > len(frames)):
                imgs_for_sda_fingerprint.append((np.sum(frames[int(i*d) : int(len(frames))],
    axis=0)/int(m  - int(i*d))).astype(np.uint8))
                break
            else:
                imgs_for_sda_fingerprint.append((np.sum(frames[int(i*d) : int(i*d + d)], axis
    =0)/d).astype(np.uint8))

    sda_fingerprint_all_frames = prnu.extract_multiple_aligned(imgs_for_sda_fingerprint, sigma
     = 2,  processes = 2)
    # pce_sda_fingerprint_all_frames, best_scale_guess,best_rot_guess =  scale_crop_rotate(
    image_fingerprint ,sda_fingerprint_all_frames,0,0)
    # sigma = 2


    fingerprint_path = "//home//aronlatis//aron_prnu//videos_PRNU//video_repo"
    results = []
    for fingerprint_name in sorted(os.listdir(fingerprint_path)):
        device_fingerprint = fingerprint_path + '//' + fingerprint_name
        image_fingerprint = np.load(device_fingerprint)
        image_fingerprint = image_fingerprint.astype(np.float64)

        s = min(np.array(image_fingerprint.shape)/np.array(sda_fingerprint_all_frames.shape))
        upscaled_fingerprint = zoom(sda_fingerprint_all_frames,s)
        ranges = np.array(upscaled_fingerprint.shape) - np.array(image_fingerprint.shape)
        if (ranges[0] >= 0) & (ranges[1] >=0):
            cc2d = prnu.crosscorr_2d(upscaled_fingerprint, image_fingerprint)
        if (ranges[0] <= 0) & (ranges[1] <= 0):
            cc2d = prnu.crosscorr_2d(image_fingerprint, upscaled_fingerprint)
        pce_KSDA_K = pce.pce(cc2d,ranges)['pce']

        print(upscaled_fingerprint.shape)
        print(image_fingerprint.shape)
        print('******************')
        if pce_KSDA_K > 100 :
            pce_max = pce_KSDA_K
            scale_max = 0
        else:
            pce_max = 0
            scale_max = 0
            for ks in np.arange(1,1.5,0.005):
                Krs = zoom(image_fingerprint,1/ks)
                s2 = min(np.array(Krs.shape)/np.array(sda_fingerprint_all_frames.shape))
                step_size = (s2 - 1)/25
                if (s2 <= 1):
                    break
                for s3 in np.arange(1, s2+step_size, step_size):
                    upscaled_fingerprint = zoom(sda_fingerprint_all_frames,s3)
                    ranges = np.array(upscaled_fingerprint.shape) - np.array(Krs.shape)
                    print(Krs.shape)
                    print(upscaled_fingerprint.shape)
                    print('-----------------')
                    if (ranges[0] <= 0) & (ranges[1] <= 0):
                        cc2d = prnu.crosscorr_2d(Krs, upscaled_fingerprint)
                    else:
```

```
80                          continue
81                      pce_for_downscaled_K = pce.pce(cc2d,abs(ranges))['pce']
82                      if pce_for_downscaled_K > pce_max:
83                          pce_max = pce_for_downscaled_K
84                          scale_max = ks
85              print('0000000000000')
86          results.append([sys.argv[1] ,sorted(os.listdir(folder_path))[0] , scale_max, pce_max
    ])
87
88      save_name = sys.argv[1] + '_results'
89      np.save(save_name, np.array(results))
90
91      return 0
92
93  if __name__ == '__main__':
94      main()
```

Codul pentru analiza semnalelor video cu sub-eșantionarea amprentei originale nu va fi prezentat deoarece ocupă prea mult spațiu.