

**MINISTERUL EDUCAȚIEI
UNIVERSITATEA "VALAHIA" DIN TÂRGOVIȘTE
IOSUD – ȘCOALA DOCTORALĂ DE ȘTIINȚE INGINEREȘTI
DOMENIUL FUNDAMENTAL: ȘTIINȚE INGINEREȘTI
DOMENIU: INGINERIE ELECTRICĂ**

**"Contribuții la dezvoltarea tehniciilor de camuflare
reversibilă a informației în semnale 1D"**

**CONDUCĂTOR DE DOCTORAT:
Prof. univ. dr. ing. Dinu COLȚUC**

**DOCTORAND :
Ing. Constantin-Alin BOBEICĂ**

**TÂRGOVIȘTE
2023**

Cuprins

1	Introducere	2
2	Metode de inserție reversibilă a datelor	6
2.1	Insertia RDH bazata pe translatarea histogramei	6
2.2	Insertia RDH bazata pe expandarea erorii de predictie	8
2.3	Insertia RDH bazata pe ordonarea valorilor pixelilor (PVO)	11
2.4	Insertia RDH bazata pe ordonarea îmbunătățită a valorii pixelilor (IPVO)	13
2.5	Insertia RDH bazata pe PPVO	14
3	Dezvoltarea de noi metode RDH pentru domeniul audio	16
3.1	Controlul capacitatii prin adaptarea pragului de inserție	16
3.1.1	Rezultate si concluzii	17
3.2	Audio RDH derivat din PVO si PPVO	19
4	Transmiterea de date de la senzori mobili	24
4.1	Analiza mersului	25
4.2	Proiectarea arhitecturii hardware și software	29
4.2.1	Arhitectura hardware a modelului experimental	29
4.2.2	Insetie RDH cu 2 biti	29
4.3	Rezultate si concluzii	31
Concluzii		32
	Contributii originale	33
	Direcții viitoare de cercetare	34
	Listă de publicații	35
Bibliografie		36

Capitolul 1

Introducere

În ultima perioadă, începând cu anii 2000, s-a observat o creștere a puterii de procesare și capacitatea de stocare a dispozitivelor inteligente (computere, tablete, telefoane inteligente), fapt ce a dus la folosirea datelor digitale într-un mod accelerat. În ziua de azi, aproape tot conținutul multimedia este trecut prin procesele de memorare, transmitere și prelucrare în format digital. Faptul ca datele, în format digital, pot fi reproduse și distribuite cu ușurință, a îngreunat în mod semnificativ securizarea conținutului original și a proprietății intelectuale. Pentru a remedia această problemă, au apărut o nouă generație de algoritmi pentru transmiterea și securizarea datelor digitale.

Marcarea digitală (*digital watermarking*) constă în inserarea unei informații ascunse într-un semnal gazdă care duce la modificarea imperceptibilă a acestuia (Osborne et al. în [1]). Pornind de la această idee au fost dezvoltăți și adaptați algoritmi specializați pentru diverse semnale gazdă (fișiere audio sau video, imagine statică etc.) și în funcție de aplicația dorită (protejarea drepturilor de autor, monitorizarea transmisiei, autentificarea conținutului, adnotarea de date).

1.1 Inserția reversibilă de date

În comparație cu inserția de date clasică, algoritmii de inserție reversibili (*RDH – reversible data hiding*) oferă posibilitatea recuperării exacte a semnalului gazdă, prin extragerea completă a datelor ascunse. Trebuie luat în calcul faptul că procesul de inserție a mesajului secret produce distorsiuni.

Algoritmii RDH pentru imaginii exploatează redundanța semnalelor (imagini, video, sunet, etc.). În mod uzual, sunt adoptate două criterii pentru a evalua eficiența unei metode RDH: distorsiunea (evaluată prin *SNR - Signal to Noise Ratio* la fișierele audio) și capacitatea de inserție (*EC - Embedding Capacity*). Comparând semnalul gazdă marcat cu originalul, primul criteriu măsoară modificarea datorată ascunderii datelor pornind de la eroarea medie pătrată

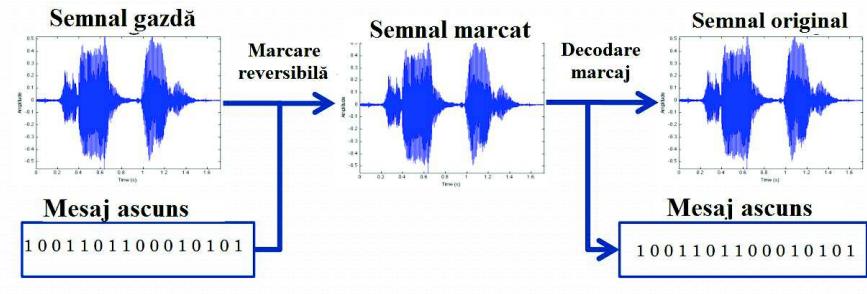


FIGURA 1.1: Marcarea reversibilă a unui semnal audio

(*MSE – Mean Square Error*) obținută. O valoare SNR ridicată indică o mai bună imperceptibilitate, adică, semnalul gazdă este asemănător cu varianta sa inițială.

1.2 Conținutul tezei

În capitolul 2 sunt prezentate și discutate cele mai relevante metode din domeniul inserției reversibile de date. Aici sunt descriși algoritmii de inserție și extragere clasici de translatarea de histogramă și expandarea erorii de predicție, precum și cei recenti derivați din ordonarea valorilor pixelor.

Trei metode noi de inserție reversibilă în fișiere audio dezvoltate în această teză sunt prezentate în capitolul 3. Prima metodă permite un control fin pentru capacitatea de inserție în funcție de fișierul gazdă. A doua metodă propusă combină două abordări existente din inserția în imagini digitale (translatarea histogramei de erori cu ordonarea valorilor pixelor), adaptându-le pentru inserția reversibilă în eșantioane audio. Această metodă obține cele mai bune rezultate raportate în literatură pentru capacitați sub 0.3 bps (bits-per-sample). A treia metodă (o abordare multi-bit cu predicție derivată din ordonarea valorilor pixelor) oferă o soluție eficientă pentru capacitați între 0.3 și 0.7 bps.

În capitolul 4 s-a propus o metodă de inserție simplă pentru o aplicație de analiza a mersului, având ca scop reducerea volumului de date transmis de aplicație. Achiziția de date de la senzori se face cu ajutorul unui microcontroler, care transmite datele prin Bluetooth către o aplicație Android de pe un telefon mobil.

Concluziile cercetării sunt prezentate în ultima parte a tezei.

1.3 Stadiul actual al schemelor de inserție reversibilă de date

Având ca punct de pornire schema clasică în [1], Hossinger et al. este dezvoltatorul primei scheme de marcă reversibilă [2]. Metoda este bazată pe inserarea unui cod de autentificare într-o imagine digitală, prin simpla operație de adunare modulo 256, în imagini, $W = (I + w) \text{ mod } 256$, unde I , w și W reprezintă imaginea originală, marcajul și imaginea marcată. Cum w este

cunoscut, imaginea originală se obține prin scăderea modulo 256, $I = (W-w) \bmod 256$. Metoda introduce un zgomot de tip sare și piper, din cauza aritmeticii modulare 256.

Friedrich et al. propune în [3], prima schemă de marcare reversibilă bazată pe compresie fără pierderi. Schema este îmbunătățită ulterior în [3, 4]. Problema acestor scheme este că, pentru a nu genera distorsiuni vizibile, inserția trebuie făcută în zona biților mai puțin semnificativi. În zona biților mai puțin semnificativi, rata de compresie este foarte scăzută ceea ce conduce la o rată de inserție foarte mică. În [5], Xuan et. al. a propus o metodă RDH de mare capacitate, metodă bazată pe transformata wavelet (IWT). Pentru metodele bazate pe compresie mai menționăm schemele raportate de Celik et al. în [6].

În [7], Tian prezintă o metodă promițătoare de mare capacitate de inserție RDH, bazată pe expandarea diferenței (DE). Această metodă operează pe perechi de pixeli și inserează informația în diferența dintre pixeli. Perechile de pixeli sunt modificate astfel încât diferența dintre pixeli să se dubleze. Folosind expandarea diferenței se atinge o capacitate maximă de 0.5 bpp pentru o etapă de marcare, ținând cont de faptul că în fiecare pereche de pixeli se introduce câte un bit de informație. Comparându-se cu scheme anterioare bazate pe compresia fără pierderi, expandarea diferenței are rezultate mai bune oferind o capacitate de inserție mai mare și o distorsiune scăzută.

Alattar în [8], a propus o nouă metodă care este bazată pe expandarea diferenței dezvoltată de Tian, dar bazându-se de această dată pe blocuri de pixeli de dimensiuni arbitrară, în loc să folosească perechi de pixeli. Astfel reușește să ajungă la capacitatea maximă de $n - 1/n$ bpp, pornind de la ideea de a insera într-un grup de n pixeli, $n - 1$ biți. PEE este propus în primul rând de Thodi și Rodriguez în [9], iar această tehnică a fost adoptată pe scară largă de multe lucrări ulterioare de RDH în [10, 11]. Contribuții recente aduse metodei PEE au fost aduse în [12] (bazate pe selecția și sortarea pixelilor gazdă). Rețelele neuronale convolutionale (CNN - *Convolutional neural network*) au fost folosite cu succes pentru predicție în [13], [14], [15] și [16]. Predicția pe bază de CNN a fost optimizată apoi prin sortare [17] și rafinarea predicției inițiale [18].

Ordonarea valorilor pixelilor (*PVO-Pixel-Value-Ordering*) este o metodă eficientă pentru marcarea reversibilă de înaltă fidelitate. Metoda PVO dezvoltată de X. Li et al. în [19], împarte imaginea în blocuri egale. F. Peng et al. în [20] prezintă metoda îmbunătățită a ordonării valorilor pixelilor, metodă ce a îmbunătățit procesul de calcul al diferenței. PVO este o abordare extrem de populară pentru inserția în imagini digitale.

Abordarea PVO a fost introdusă ca predictor pentru translatarea de histogramă și pentru PEE, primind denumirea de „*Pixel-based pixel value ordering*” (PPVO). Și această abordare a fost îmbunătățită semnificativ în ultimii ani. Cele mai semnificative contribuții au fost: optimizarea inserției, introducerea unui predictor hibrid și optimizarea selecției pixelilor gazdă.

Insetia în perechi de pixeli, introdusă în [21] și optimizată ulterior, permite o inserție mai eficientă decât cea clasică în pixelii gazdă. Această abordare a fost utilizată recent, însă rezultatele cele mai bune au fost obținute prin combinarea sa cu PVO.

În domeniul audio, metodele RDH au fost derivate din cele pentru imaginile digitale, cel mai frecvent din PEE. Metoda din [22] foloșește un predictor simplu non-causal (idee adaptată din [23]). Acest predictor este înlocuit în [24] cu unul liniar, calculat prin regresie liniară pe un număr variabil de esantioane audio. În [25], esantioanele din context sunt întâi sortate, apoi se determină predictorul liniar. În [26] se folosește o codare reversibilă bazată pe multiplexare. Metoda din [27] inserează robust în domeniul transformatei DCT cu valori întregi, abordare făcută reversibilă în [28].

Capitolul 2

Metode de inserție reversibilă a datelor

2.1 Inserția RDH bazată pe translatarea histogramei

Algoritmul ce se bazează pe translatarea histogramei, introdus pentru marcarea reversibilă a imaginilor digitale, selectează la fiecare etapă a marcării două valori ale erorii prezise: e_p și e_n . Se vor modifica pixelii bazându-ne pe aceste erori, mai exact, doar acei pixelii care au eroarea e_p sau e_n vor conține biți secerți, iar restul pixelilor din imagine vor fi lăsați neschimbați sau deplasati pentru evitarea suprapunerii lor cu acei pixeli marcati.

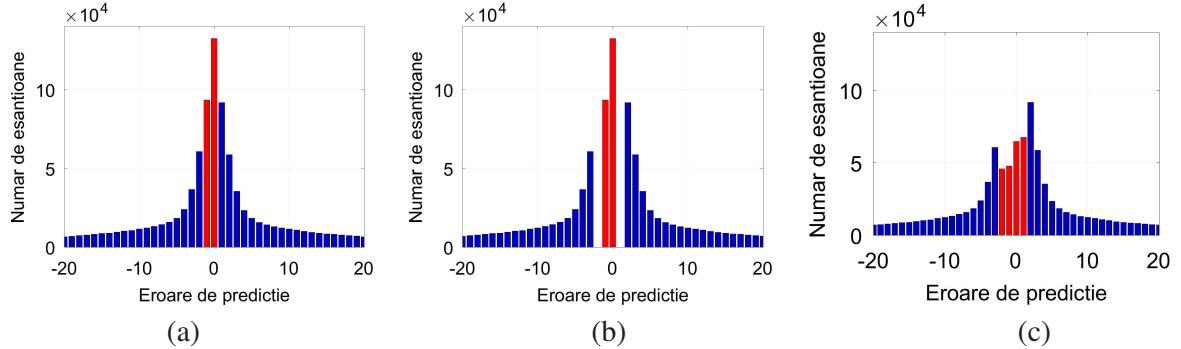


FIGURA 2.1: Etapele marcării HS.
a: selectarea lui e_p și e_n ; b: deplasarea valorilor adiacente; c: inserarea marcajului.

Algoritmul de inserție de date HS

Funcționalitatea algoritmului de marcare constă în parcurgerea imaginii gazdă, citind valoarea fiecărui pixel în ordine *rasterscan* (de la stânga la dreapta și de sus în jos). În cazul în care eroarea de predicție a pixelul analizat este egală cu e_p sau e_n , acesta poate fi marcat cu:

$$x'_{i,j} = \begin{cases} x_{i,j} + b, & \text{dacă } e_{i,j} = e_p \\ x_{i,j} - b, & \text{dacă } e_{i,j} = e_n \end{cases} \quad (2.1)$$

unde bitul ascuns este $b \in \{0, 1\}$. Atunci când $b = 0$, pixelii în care s-a introdus un bit de informație, rămân nemodificați sau când $b = 1$, pixelii sunt deplasăți astfel încât e_p să fie cu o unitate mai aproape de extremele histogramei, e_{\min} sau e_{\max} . Prezența marcajului este indicată de următoarele erori : $\{e_p, e_n, e_p + 1, e_n - 1\}$. Valorile pixelilor din imaginea marcată care au eroarea inițială $e_p + 1$ sau $e_n - 1$, trebuie deplasate astfel încât să eliminăm ambiguitatea dintre pixelii marcați și ele. Însă deplasarea acestora mută problema ambiguității la perechea $e_n - 2$, $e_p + 2$. În acest caz, pentru distingerea corectă a pixelilor marcați față de cei originali, toate valorile care au $e_{i,j} \in [e_{\min}, e_n - 1] \cup [e_p + 1, e_{\max}]$ vor fi deplasate în domeniul $[e_{\min} - 1, e_n - 2] \cup [e_p + 2, e_{\max} + 1]$:

$$x'_{i,j} = \begin{cases} x_{i,j} + 1, & \text{dacă } e_{i,j} > e_p \\ x_{i,j} - 1, & \text{dacă } e_{i,j} < e_n \end{cases} \quad (2.2)$$

În cazul unor anumite valori ale e_p și e_n vom observa ca există pixeli ce respectă condiția $e_n < e_{i,j} < e_p$, iar în acest caz erorile acestora nu se vor mai suprapune cu cele deplasate sau marcate și vor avea valoarea originală.

În cazul imaginilor pe 8 biți, nivelele de gri își au limitele în intervalul $[0, 255]$. La fiecare proces de marcăre, distorsiunea maximă introdusă în pixelul gazdă este de ± 1 , dar imaginea poate avea nivale de gri de 0 ori 255 și nu pot fi deplasate, devenind -1 sau 256 . Înainte de marcăre, pixelii problemă sunt identificați și deplasăți la 1 respectiv 254, adică spre interiorul intervalului, ceea ce menține schema de marcăre reversibilă. Semnalele audio sunt în general pe 16 sau 24 de biți, domeniul fiind între -32.768 și 32.767 , respectiv $-8.388.608$ și $8.388.607$.

Această tehnică de marcăre folosește o hartă pentru stocarea poziției pixelilor deplasăți și după aceea o compreseză fără pierderi și împreună cu valorile alocate pentru e_p și e_n formează datele auxiliare. Inserarea acestora în imagine se face prin substituirea LSB-urilor. De asemenea, biții substituți sunt stocați împreună cu marcajul.

Schema de marcăre reversibilă ce folosește translatarea histogramei, oferă o capacitate de inserție ce este oferită de numărul pixelilor care au $e_{i,j} \in \{e_p, e_n\}$:

$$C = H(e_p) + H(e_n) - D_{\text{aux}} \quad (2.3)$$

unde $H(e_n)$ și $H(e_p)$ reprezintă pixelii cu $e_{i,j} = e_n$, respectiv $e_{i,j} = e_p$ și dimensiunea datelor auxiliare măsurată în biți este notată: D_{aux} . În cazul imaginilor naturale, numărul de pixeli pur negri sau albi este foarte scăzut (poate fi nul pentru unele imagini), situație ce face ca numărul de biți al lui D_{aux} să nu fie unul mare. Însă cu fiecare etapă de marcăre, mărimea acestuia crește și devine o problemă.

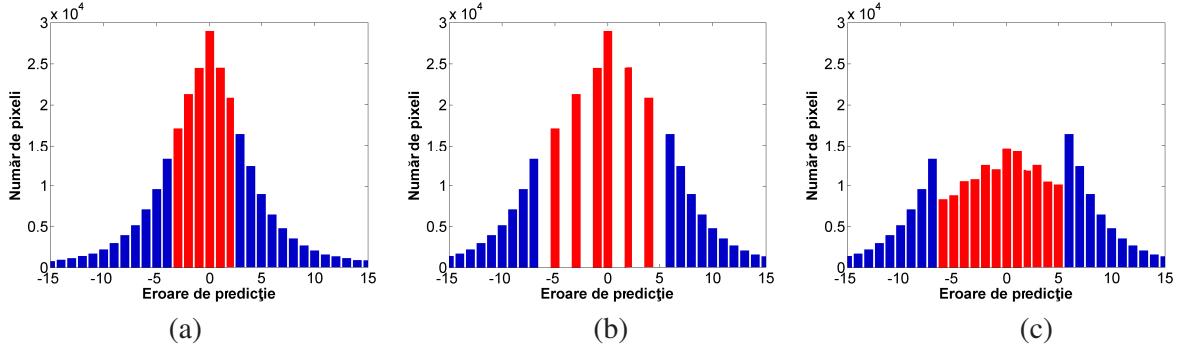


FIGURA 2.2: Vizualizarea etapelor marcării PEE pe histograma erorii de predicție.
a: selectarea intervalului $[-T, T]$; b: deplasarea valorilor vecine; c: inserția marcajului

2.2 Inserția RDH bazată pe expandarea erorii de predicție

Pentru marcarea prin expandarea erorii de predicție s-a dezvoltat un algoritm de inserare reversibilă (PEE) ce folosește un interval al erorilor de predicție, de forma $[-T, T]$, unde T este pragul de marcăre. Valorile care nu aparțin acestui interval sunt deplasate pentru crearea unui spațiu liber folosit la marcăre. Trebuie precizat faptul că distorsiunile inserate nu mai au limită la ± 1 , și ajung până la $\pm T$.

Algoritmul de inserție de date PEE

Se folosește din nou ordinea *raster scan*, pentru a fi parcursă imaginea pixel cu pixel. Apoi se compară eroarea de predicție ce corespunde pixelului actual cu pragul de marcăre T . În cazul în care $-T \leq e_{i,j} < T$ se inserează un bit de informație în $x_{i,j}$:

$$x'_{i,j} = x_{i,j} + e_{i,j} + b \quad (2.4)$$

După inserție, intervalul de valori $[-T, T]$ este expandat la $[-2T, 2T]$. Pixelii care au $e_{i,j} \in [e_{\min}, -T) \cup [T, e_{\max}]$, pentru eliminarea ambiguității dintre ei și pixelii care au fost marcați, sunt deplasati cu $\pm T$:

$$x'_{i,j} = \begin{cases} x_{i,j} + T, & \text{dacă } e_{i,j} \geq T \\ x_{i,j} - T, & \text{dacă } e_{i,j} < -T \end{cases} \quad (2.5)$$

Controlul îmbunătățit al capacitatei

Distorsiunea introdusă de schema de marcăre reversibilă a PEE este compusă din distorsiunea produsă de inserția marcajului și din distorsiunea deplasării pixelilor.

Distorsiunea dată de inserare este realizată de expandarea erorii de predicție și de adăugarea bițiilor de informație. După cum am menționat în secțiunea anterioară, în loc să creăm spațiu pentru B biți, se extind erorile de predicție pentru N pixeli în care eroarea de predicție este mai

mică decât T și pentru cei $B - N$ pixeli rămași care au o eroare de predicție de $\pm T$. Altfel spus, pentru a obține capacitatea de inserare dorită, schema de inserare cu două praguri utilizează pixelii care au cele mai mici erori de predicție posibile. Deoarece pixelii cu $|p_e| < T$ nu sunt suficienți, odată ce acești pixeli sunt inserați, se iau în considerare și pixelii cu $|p_e| = T$. Acum se observă clar că $T' = T + 1$ menține eroarea de inserare minimă. De exemplu, $T' > T + 1$, ar trebui să găsim $B - N$ biți suplimentari marcând mai puține rânduri cu acest prag decât cu $T + 1$, dar cu costul expandării erorilor de predicție mai mari decât $|T|$ și, implicit, cu costul creșterii erorii de inserare.

Eroarea de deplasare introdusă de schema de marcare PEE clasică nu este simetrică. Pixelii neinserați care au eroarea de predicție pozitivă sunt deplasați cu valoarea pragului, T , în timp ce cei care au eroarea de predicție negativă sunt decalați cu $-(T - 1)$. Eroarea pătrată a deplasării rezultă din T^2 la erori de predicție pozitive și $(T - 1)^2$ la erori de predicție negative, adică o eroare cu $2T - 1$ mai mică decât cea pentru pixelii cu valoare pozitivă.

Această asimetrie se datorează inserării datelor. Eroarea de predicție a pixelilor marcați este de $2p_e + b$, cu $|p_e| < T$. Limita pentru eroarea de predicție pozitivă, $2T - 1$, apare pentru $b = 1$, în timp ce cea pentru erori de predicție negative apare pentru $b = 0$. Cele două limite (T și $T + 1$) sunt pentru a asigura intervale de eroare de predicție disjuncte pentru pixelii marcați și nemarcați.

Erorile de deplasare simetrice pot fi obținute prin luarea în considerare în locul unui singur prag, a unor praguri distințe pentru și erori de predicție pozitive. Astfel, în loc de praguri simetrice $(-T; T)$, se pot lua în considerare $(-T; T - 1)$. Erorile de deplasare sunt reduse la $(T - 1)^2$ atât pentru valori negative, cât și pentru valori pozitive ale erorilor de predicție, dar cu prețul unei scăderi a valorii capacitații de inserare. În mod similar, luăm pragul $(-T - 1, T)$ pentru a crește capacitatea de inserare (și eroarea de deplasare). Dacă notăm cu $W(T_n, T_p)$ capacitatea schemei de marcare PEE pentru praguri distințe pozitive (T_p) și negative (T_n), avem:

$$\dots W(-T, T - 1) < W(-T, T) < W(-T - 1, T) \dots \quad (2.6)$$

Condiția pentru simetria erorilor de deplasare este:

$$T_p = |T_n| - 1 \quad (2.7)$$

Ecuația 2.6 oferă o segmentare mai fină a intervalului de capacitate decât cea oferită de pragurile simetrice (sau cele nesimetrice). Metoda poate fi extinsă imediat pentru interpolarea între limitele furnizate de ecuația 2.6.

Schema de inserție/extragere PEE necesită un prag de inserție adecvat T pentru a furniza numărul corect de eșantioane gazdă, care, la rândul său, determină capacitatea oferită. T controlează, de asemenea, distorsiunile de inserare introduse de schema PEE, prin urmare, selecția lui T este crucială pentru obținerea unor rezultate optime de ascundere a datelor.

Schemele actuale de ultimă generație PEE bazate pe RDH audio utilizează algoritmul de control al capacitatii de bază introdus pentru prima dată în [9]. Fie N numărul necesar de eșantioane gazdă pentru setul curent:

$$N = \frac{N_b}{2} + \frac{N_R}{2} + N_o \quad (2.8)$$

unde N_b este dimensiunea datelor inserate, N_R este numărul de LSB-uri stocate din R și N_o este numărul estimat de eșantioane care prezintă risc de depășire (eșantioane cu $x_i \notin [-32768 + T, 32767 - T]$). Rețineți că atât în [29] cât și în [22], controlul capacitatii este discutat doar în termeni generali, pentru simplitate vom folosi ecuația de mai sus.

Eroarea de predicție se determină cu următoarea formulă pentru fiecare pixel din setul curent:

$$e_i = x_i - \hat{x}_i \quad (2.9)$$

Acstea valori sunt colectate într-o histogramă a erorilor de predicție. Fie H histograma erorilor de predicție, unde $H(e_i)$ este egal cu numărul total de eșantioane din setul curent cu eroarea de predicție e_i . Pragul de inserare se selectează ca fiind cea mai mică valoare T care îndeplinește:

$$\sum_{e_i=-T}^{T-1} H(e_i) \geq N \quad (2.10)$$

După ce toți biții de date ascunse $\frac{N_b}{2}$ au fost introdusi împreună cu biții de date auxiliare $\frac{N_R}{2}$ și nu mai rămân biți de semnalizare, atunci pixelii rămași în set nu mai sunt prelucrați și rămân neschimbați (și se înregistrează poziția ultimului eșantion modificat).

Algoritmul de control fin al capacitatii este derivat din schema RDH pentru imagini, propusă pentru prima dată în [30] și rafinată în [29]. Spre deosebire de abordarea clasică (în care marea majoritate a eșantioanelor sunt modificate utilizând pragul selectat T), abordarea propusă utilizează două praguri distincte: T și $T - 1$, unde T a fost determinat cu (2.10).

Fie D numărul de eșantioane gazdă suplimentare care sunt necesare în cazul în care $T - 1$ a fost utilizat ca prag de inserție:

$$D = N - \sum_{e_i=-T+1}^{T-2} H(e_i) \quad (2.11)$$

unde N a fost calculat cu (3.1) și $\sum_{e_i=-T+1}^{T-2} H(e_i)$ este numărul de eșantioane gazdă furnizate de $T - 1$.

Procesul de inserare pentru setul curent începe prin utilizarea lui T ca prag de inserare. După ce un total de D biți au fost inserati cu succes în eșantioane cu $e_i \in \{-T, T - 1\}$, pragul de inserare devine $T - 1$ pentru restul eșantioanelor din set. Cu alte cuvinte, eșantioanele sunt modificate cu T până când este inserată capacitatea care nu a putut fi asigurată de $T - 1$, apoi eșantioanele rămase sunt modificate cu $T - 1$.

Principalele îmbunătățiri față de [30] și [29] sunt estimarea capacitatii bazată pe biți de semnalizare în (3.1) și modelul de implementare simplificat descris în această secțiune. În plus, ambele abordări au fost dezvoltate pentru RDH pentru imagini și, din căte știm, algoritmul clasic (inferior) de control al capacitatii este singurul model utilizat în prezent în RDH audio.

2.3 Inserția RDH bazată pe ordonarea valorilor pixelilor (PVO)

Ordonarea valorilor pixelilor (*pixel value ordering - PVO*) [19] este inserția reversibilă în funcție de ordonarea valorilor de gri a pixelilor. În inserarea reversibilă de date bazată pe expandarea erorii de predicție, histograma erorii de predicție este împărțită în două regiuni, una fiind numită regiunea interioară și cealaltă regiune exterioară. Erorile de predicție din regiunea interioară a histogramei sunt extinse pentru a conține date, în timp ce cele din regiunea exterioară sunt deplasate astfel încât interiorul și regiunile exterioare rămân separate după inserarea datelor (similar cu metodele HS și PEE discutate mai sus). Dacă există doar una sau două valori ale histogramei în regiunea interioară, adică, modificarea maximă a valorii pixelilor în inserarea datelor este 1 și se poate deduce că valoarea modificării pentru imaginea gazdă este $0.5N_c + N_s$, unde N_c și N_s reprezintă capacitatea de inserție (în biți) respectiv numărul de pixeli deplasati.

În această secțiune, ca o trecere în revistă, va fi prezentată mai întâi metoda bazată pe PVO a lui Li et al. [19], iar apoi metoda lui Lee et al. [31].

Pentru început, imaginea gazdă este împărțită în blocuri nesuprapuse de dimensiuni egale. Pentru un bloc dat, X care conține n pixeli, se sortează valorile acestora (x_1, \dots, x_n) în ordine crescătoare pentru a obține $x_{\sigma(1)}, \dots, x_{\sigma(n)}$, unde $\sigma : 1, \dots, n \rightarrow 1, \dots, n$ reprezintă o mapare unică unu la unu cum ar fi: $x_{\sigma(1)} \leq \dots \leq x_{\sigma(n)}$, $\sigma(i) < \sigma(j)$ dacă $x_{\sigma(i)} = x_{\sigma(j)}$ și $(i) < (j)$. Apoi se folosește a doua cea mai mare valoare, $x_{\sigma(n-1)}$, pentru a prezice maximul $x_{\sigma(n)}$. Eroarea de predicție corespunzătoare este:

$$PE_{\max} = x_{\sigma(n)} - x_{\sigma(n-1)} \quad (2.12)$$

În timp ce înălțimea care are $PE_{\max} = 1$ este în mod ușor valoarea de vârf a histogramei, similar cu eroarea de predicție PE_{\max} este modificată în:

$$PE_{\max} = \begin{cases} PE_{\max} & \text{dacă } PE_{\max} = 0, \\ PE_{\max} + b & \text{dacă } PE_{\max} = 1, \\ PE_{\max} + 1 & \text{dacă } PE_{\max} > 1. \end{cases} \quad (2.13)$$

unde $b \in \{0, 1\}$ reprezintă bitul de informație care va fi inserat. În conformitate cu valoarea maximă $x_{\sigma(n)}$ este modificată în:

$$x = x_{\sigma(n-1)} + PE_{\max} = \begin{cases} x_{\sigma(n)} & \text{dacă } PE_{\max} = 0, \\ x_{\sigma(n)} + b & \text{dacă } PE_{\max} = 1, \\ x_{\sigma(n)} + 1 & \text{dacă } PE_{\max} > 1. \end{cases} \quad (2.14)$$

Celelalte valori $x_{\sigma(1)}, \dots, x_{\sigma(n-1)}$ rămân neschimbate. Prin urmare, valoarea marcată a lui x este (y_1, \dots, y_n) , unde $y_{\sigma(n)} = \tilde{x}$ și $y_i = x_i$ pentru fiecare $i \neq \sigma(n)$. În procedura de mai sus, deoarece maximul $x_{\sigma(n)}$ este fie neschimbat, fie mărit, ordinea valorilor pixelilor (maparea σ) rămâne neschimbată. Prin urmare, pentru un bloc marcat a cărui valoare este (y_1, \dots, y_n) , prin calcularea erorii de predicție:

$$PE_{\max} = y_{\sigma(n)} - y_{\sigma(n-1)} \quad (2.15)$$

extragerea datelor și restaurarea imaginii poate fi realizată astfel:

- dacă $PE_{\max} = 0$, blocul rămâne neschimbat la inserția datelor iar valoarea sa originală este doar (y_1, \dots, y_n)
- dacă $PE_{\max} \in \{1, 2\}$, blocul este expandat pentru a incorpora datele de inserat. Bitul de date de inserat $b = PE_{\max} - 1$ și valoarea originală este (x_1, \dots, x_n) , unde $x_{\sigma(n)} = y_{\sigma(n)} - b$ și $x_i = y_i$ pentru orice $i \neq \sigma(n)$.
- dacă $PE_{\max} > 2$, blocul este deplasat de datele inserate iar valoarea lor originală este (x_1, \dots, x_{σ}) , unde $x_{\sigma(n)} = y_{\sigma(n)} - 1$ și $x_i = y_i$, pentru $i \neq \sigma(n)$.

Pentru PVO cantitatea de valori deplasate (*shift*-ate) poate fi reformulată sub forma:

$$P_{shifted} = \frac{\#\{PE_{\max} > 1\}}{\#\{PE_{\max} \geq 1\}} \quad (2.16)$$

unde PE_{\max} este eroarea de predicție a imaginii gazdă definită în formula (2.15)

Pe lângă inserția în valoarea maximă a blocului, PVO permite și inserția în cea minimă. Mai exact, a doua cea mai mică valoare, $x_{\sigma(2)}$, poate fi utilizată pentru a prezice minimul

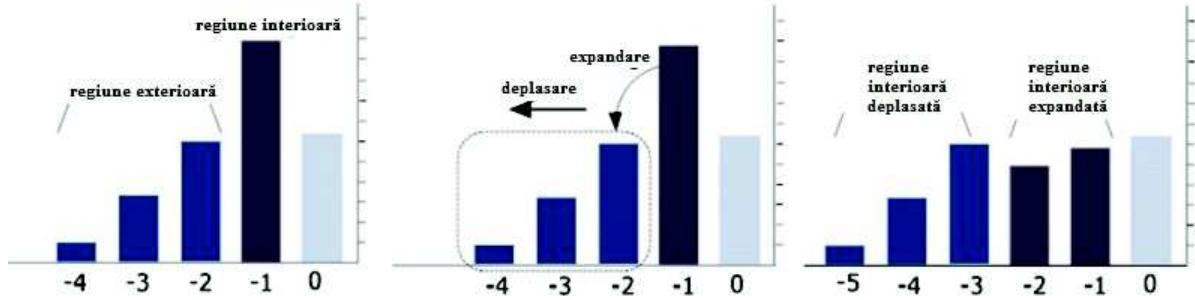


FIGURA 2.3: Insertia PVO prin modificarea minimului unui bloc

$x_{\sigma(1)}$, iar inserția PEE poate fi implementată prin modificarea erorii de predicție corespunzătoare $PE_{\min} = x_{\sigma(1)} - x_{\sigma(2)}$. În acest caz, eroarea de predicție PE_{\min} se modifică astfel:

$$\widetilde{PE}_{\min} = \begin{cases} PE_{\min} & \text{dacă } PE_{\min} = 0, \\ PE_{\min} - b & \text{dacă } PE_{\min} = -1, \\ PE_{\min} - 1 & \text{dacă } PE_{\min} < -1. \end{cases} \quad (2.17)$$

unde $b \in \{0, 1\}$ reprezintă bitul de informație de inserat.

Luând în considerare atât maximul, cât și minimul, cel mult doi biți pot fi inserați în același bloc gazdă. De exemplu, atunci când condițiile $x_{\sigma(n)} - x_{\sigma(n-1)} = 1$ și $x_{\sigma(1)} - x_{\sigma(2)} = -1$ sunt ambele îndeplinite, se pot insera doi biți în blocul curent.

În cele din urmă, observăm că utilizarea blocurilor uniforme este mai favorabilă pentru ascunderea reversibilă a datelor. Complexitatea unui bloc se măsoară prin diferența $x_{\sigma(n-1)} - x_{\sigma(2)}$, iar un bloc este considerat ca fiind plat dacă complexitatea sa este mai mică decât un prag predefinit T . De exemplu, pentru trei blocuri din Fig. 2(b), complexitățile lor sunt $x_1 - x_2 = 43 - 42 = 1$ și $x_2 - x_4 = 119 - 117 = 2$, respectiv $x_1 - x_4 = 154 - 153 = 1$. Se observă că mărimea $x_{\sigma(n-1)} - x_{\sigma(2)}$ este neschimbată după inserția datelor, astfel codorul și decodorul pot găsi exact aceleași blocuri. Acest lucru garantează reversibilitatea metodei PVO, în cazul utilizării blocurilor uniforme.

2.4 Inserția RDH bazată pe ordonarea îmbunătățită a valorii pixelilor (IPVO)

Pentru o capacitate de inserție dată, metoda îmbunătățită PVO (IPVO - Improved Pixel Value Ordering) poate utiliza blocuri de dimensiuni mai mari pentru inserția datelor și, prin

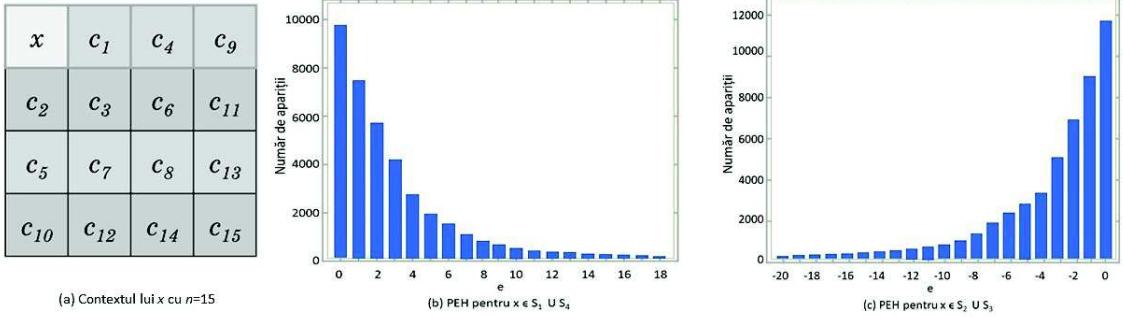


FIGURA 2.4: Contextul de predicție și histogramele erorii de predicție pentru imaginea Lena cu pixeli în $S_1 \cup S_4$ și $S_2 \cup S_3$, pentru PPVO

urmare, poate exploata mai bine redundanța imaginii, ceea ce este util pentru performanța încorporării.

$$d_{\max} = x_u - x_v \quad (2.18)$$

unde

$$\begin{cases} u = \min(\sigma(n), \sigma(n-1)), \\ v = \max(\sigma(n), \sigma(n-1)). \end{cases} \quad (2.19)$$

Aici, în loc de diferența $PE_{\max} = x_{\sigma(n)} - x_{\sigma(n-1)}$ utilizată de PVO și care are întotdeauna o valoare pozitivă, în această secțiune nouă diferență d_{\max} ia valori între $(-\infty, +\infty)$. Si, spre deosebire de cazul lui PE_{\max} , histograma lui d_{\max} este o distribuție de tip laplacian centrată la 0, deoarece ambele diferențe $x_{\sigma(n)} - x_{\sigma(n-1)}$ și $x_{\sigma(n-1)} - x_{\sigma(n)}$ sunt numărate luând în considerare ordinea $\sigma(n)$ și $\sigma(n-1)$. În acest fel, blocurile cu $x_{\sigma(n)} = x_{\sigma(n-1)}$ pot fi exploataate de metoda descrisă pentru a insera date prin modificarea lui d_{\max} .

2.5 Inserția RDH bazată pe PPVO

Mai departe, metoda PPVO este analizată pe scurt. Spre deosebire de metodele anterioare bazate pe PVO [19], PPVO [20] (*Pixel-based Pixel Value Ordering*) realizează predicția în funcție de pixel. Procedurile specifice de inserție și extragere a datelor sunt prezentate în cele ce urmează.

Așa cum este ilustrat în figura 2.4, contextul de predicție al unui pixel țintă este determinat de pixelii din apropiere din colțul din dreapta jos (a).

Se construiește un context de predicție $C = \{c_1, \dots, c_n\}$, unde n este numărul de pixeli de context. Pixelii de context vor fi apoi sortați, iar patru scenarii pot duce la predicția lui x . Apoi se definește eroarea de predicție.

Pentru o imagine gazdă, procesul de inserție a PPVO se realizează în ordinea de scanare raster, în timp ce procesul de extragere se realizează în mod invers. Ca urmare, fiecare pixel marcat și pixelul gazdă corespunzător au contexte de predicție identice.

Apoi, eroarea de predicție marcată este calculată prin $\tilde{e} = \tilde{x} - \hat{x}$. Bitul de date este extras ca $b = 0$ dacă $\tilde{e} = 0$ și $b = 1$ dacă $\tilde{e} \in \{-1, 1\}$. În cele din urmă, pixelul original este recuperat prin $x = \hat{x} + e$.

În plus, PPVO utilizează o abordare de selecție a pixelilor. Un pixel va fi utilizat pentru inserția datelor dacă complexitatea sa locală este mai mică decât un prag specificat, ceea ce indică faptul că este mai simplu să se facă o predicție precisă.

Dimensiunea contextului de predicție pentru fiecare pixel în PPVO este fixă. Cu toate acestea, performanța etapei de inserție poate fluctua în funcție de dimensiunea contextului de predicție. Pe baza complexității locale, în această parte explorăm efectele diferitelor dimensiuni ale contextului de predicție asupra performanței de inserție.

În concluzie, capacitatea mare poate fi obținută cu un context de predicție mic, dar distorsiunea introdusă de shift-area aferentă crește dramatic din cauza predicției inexacte. Atunci când se utilizează un context de predicție de dimensiuni mari, distorsiunea de deplasare este redusă, deoarece o mulțime de pixeli de acoperire sunt săriți, în timp ce se poate obține doar o capacitate scăzută de inserție.

Capitolul 3

Dezvoltarea de noi metode RDH pentru domeniul audio

În acest capitol sunt introduse o serie de noi metode de inserție reversibilă în domeniul audio. Se urmărește atât rafinarea metodelor existente, cât și formularea unor noi abordări care să permită ascunderea datelor cu un grad mai bun de imperceptibilitate.

3.1 Controlul capacitatii prin adaptarea pragului de inserție

Metodele actuale de audio RDH bazate pe PEE utilizează un algoritm de bază pentru controlul capacitatii de inserție introdus pentru prima dată în [19]. Fie N numărul necesar de eșantioane gazdă pentru setul curent:

$$N = \frac{N_b}{2} + \frac{N_R}{2} + N_o \quad (3.1)$$

unde N_b este dimensiunea datelor ascunse, N_R este numărul de LSB-uri stocate de la R și N_o este numărul estimat de eșantioane care sunt expuse riscului de depășire (eșantioane cu $x_i \notin [-32768 + T, 32767 - T]$).

Algoritmul propus de control a capacitatii este derivat din schema RDH de imagine, propusă mai întâi în [30] și rafinată în continuare în [29]. Spre deosebire de abordarea clasică, abordarea propusă utilizează două praguri distincte: T și $T - 1$.

Fie D numărul de eșantioane gazdă suplimentare care sunt necesare pentru a ascunde un total de N biți dacă $T - 1$ a fost folosit ca prag de inserție:

$$D = N - \sum_{e_i=-T+1}^{T-2} H(e_i) \quad (3.2)$$

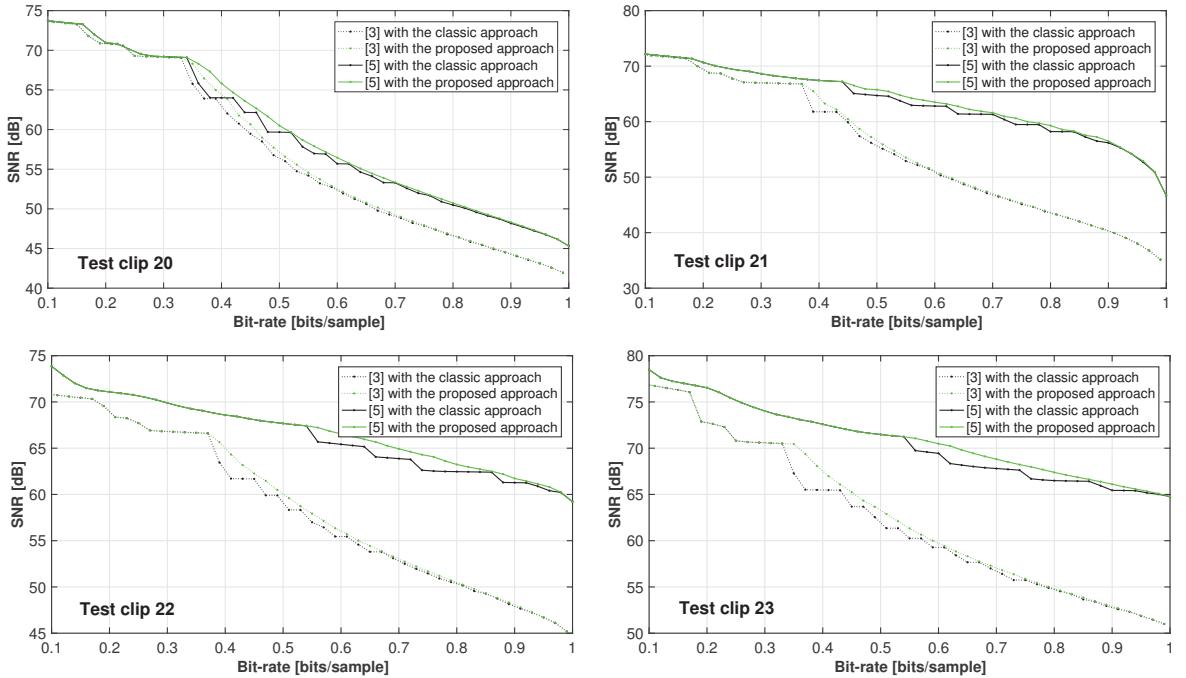


FIGURA 3.1: Rezultatele SNR/Bit-rate pentru controlul capacitatei

Procesul de inserție pentru setul curent începe prin utilizarea lui T ca prag de marcăre. După ce un număr total de D biți a fost inserat cu succes în eșantioane cu $e_i \in \{-T, T - 1\}$, pragul de marcăre devine $T - 1$ pentru valorile rămase din set. Cu alte cuvinte, eșantioanele sunt modificate folosind T până când a fost inserată capacitatea necesară care nu a putut fi furnizată de $T - 1$, apoi valorile rămase pot fi modificate cu $T - 1$.

3.1.1 Rezultate și concluzii

În această secțiune, algoritmii prezențați în capitolul anterior sunt comparați cu omologii lor clasici PEE pentru RDH în domeniul audio. Performanțele lor sunt evaluate pe 70 de fișiere audio standard cu o rată de eșantionare de 44,1 kHz.

Cele două abordări de control al capacitatei (clasică și cea propusă) sunt comparate mai întâi pe patru dintre clipurile de testare, rezultatele SNR/bit-rate sunt prezentate în Figura 3.1. Se observă un câștig clar în performanță pe toate cele patru clipuri de testare. Acest câștig este mai pronunțat atunci când se utilizează predicția liniară.

Performanța schemei propuse este evaluată în Tabelul 3.1. Controlul capacitatei propus aduce o creștere medie a performanței de 0.31 dB în SNR pentru o capacitate fixă de 0.6 bps. O îmbunătățire mai mare față de abordarea clasică este observată la predicția liniară cu o creștere medie a performanței de 0.51 dB.

TABELUL 3.1: Comparație în SNR între control clasic al capacitatei și abordarea propusă pentru 0.6 bps [dB].

Fișier de test	Medie aritmetică		Predicție liniară		Fișier de test	Medie aritmetică		Predicție liniară	
	Clasic	Propus	Clasic	Propus		Clasic	Propus	Clasic	Propus
1	51.17	51.25	86.92	86.92	36	52.59	54.39	53.64	55.51
2	11.97	12.03	18.05	18.25	37	58.03	58.20	64.85	66.22
3	82.22	82.21	83.33	83.34	38	55.31	55.31	55.57	55.57
4	80.22	80.22	83.25	83.25	39	59.32	60.51	64.27	64.76
5	45.33	45.37	82.86	82.87	40	41.38	41.58	40.86	40.95
6	78.27	78.26	80.68	80.69	41	60.18	61.22	64.52	65.89
7	74.32	75.10	80.74	80.75	42	38.67	38.79	45.91	46.19
8	40.35	40.46	47.59	48.05	43	36.69	36.76	47.61	47.73
9	44.30	44.38	52.66	52.69	44	48.24	48.44	54.78	55.31
10	46.54	46.85	52.33	52.78	45	43.73	43.88	50.64	51.38
11	59.36	59.55	60.19	60.53	46	41.69	41.94	53.22	54.13
12	39.20	39.30	50.28	50.69	47	47.09	47.37	57.25	57.76
13	50.59	50.91	58.78	59.46	48	44.75	44.90	53.41	53.84
14	41.96	42.01	55.74	56.03	49	48.85	49.24	48.47	48.56
15	45.87	46.01	52.91	53.38	50	50	50.37	52.82	53.22
16	50.13	50.27	62.59	63.01	51	45.31	45.47	44.60	44.89
17	47.56	47.71	55.40	55.75	52	50.52	50.91	50.98	51.46
18	56.61	57.12	61.16	62.46	53	48.56	48.91	48.73	49.07
19	56.29	56.63	58.80	59.56	54	47.80	48.20	50.99	51.12
20	51.95	52.52	55.69	56.46	55	45.90	46.02	60.11	60.53
21	50.90	51.12	62.81	63.50	56	33.15	33.17	51.88	52.08
22	55.45	55.99	65.42	66.56	57	25.47	25.53	38.70	38.99
23	59.28	59.74	69.45	70.49	58	49.65	49.98	52.18	53.43
24	69.07	69.76	74.17	74.17	59	40.61	40.65	43.54	43.66
25	63.27	64.72	65.01	65.55	60	59.11	59.31	60.57	61.47
26	58.22	58.31	61.01	61.89	61	42.27	42.47	52.99	53.46
27	34.97	35.05	38.74	39.12	62	52.62	53.13	55.02	55.82
28	60.48	61.95	60.48	61.96	63	44.74	44.76	50.95	51.55
29	66.31	66.31	66.31	66.30	64	36.17	36.23	46	46.13
30	60.48	60.47	60.76	60.76	65	51.29	51.75	59.27	60.44
31	56.45	57.01	56.36	56.76	66	39.91	39.98	51.50	51.70
32	44.12	44.59	47.27	48.20	67	49.15	49.37	53.51	54.54
33	59.11	59.99	58.02	58.66	68	54.24	54.57	54.25	54.58
34	54.78	54.84	59.95	61.37	69	40.43	40.71	44.87	45.34
35	41.97	42.07	55.84	56.18	70	42.90	43.08	44.18	44.30
		Medie pe set	50.22	50.53		56.63	57.14		

3.2 Audio RDH derivat din PVO și PPVO

În această subcapitol, sunt introduse două metode RDH pentru fișiere audio. Aceste noi metode au la bază principiile de inserție PE - HS, IPVO și PPVO, abordări intens utilizate pentru inserția în imagini digitale, prezentate în secțiunile 2.1, 2.4 și 2.5.

Audio RDH prin sortarea eșantioanelor audio (SVO)

Regiunile perfect uniforme sunt rare în imaginile naturale, dar eșantioanele audio pot avea secțiuni de tăcere. Aceste regiuni sunt ideale pentru predicție și prin urmare, ele pot fi ușor de exploatați pentru RDH. Toate abordările bazate pe blocuri PVO necesită ca referințele și valorile sortate între ele să rămână neschimbate. Capacitatea de inserție se pierde dacă nu se iau în considerare toate eșantioanele din secțiunile de liniște. Acest efect este într-o oarecare măsură atenuat de o mai bună performanță a PVO pe zone mai complexe. Cu toate acestea, schemele bazate pe PVO nu permit în mod corespunzător exploatarea secțiunilor uniforme.

Pentru a rezolva problema menționată mai sus, a fost creat un nou sistem de cadru bazat pe PVO pentru RDH audio, și anume ordonarea valorii eșantionului (*Sample Value Ordering-SVO*). Această abordare începe prin împărțirea fișierului audio gazdă în blocuri de dimensiunea n . Pentru fiecare bloc, se calculează complexitatea locală a blocului numai pe baza valorilor eșantioanelor vecine:

$$l_c = \max(x_{i+n}, \dots, x_{i+2n}) - \min(x_{i+n}, \dots, x_{i+2n}) \quad (3.3)$$

Toate valorile dintr-un bloc pot fi acum modificate fără a modifica complexitatea locală a blocului respectiv. Din cauza dimensiunii mici a blocului ($n \in [3, 24]$), blocurile vecine ar trebui să aibă același tip de complexitate.

În continuare, blocurile sunt clasificate pe baza complexității locale. Blocurile netede (cu $l_c \leq t_1$) sunt acum procesate ca eșantioane individuale utilizând inserția RFH bazată pe translatarea histogramei din secțiunea 2.1. Valorile prezise sunt calculate cu:

$$\hat{x}'_i = \left\lfloor \frac{3x_{i+2} + x_{i+2}}{4} + \frac{1}{2} \right\rfloor \quad (3.4)$$

Trebuie reținut faptul că valorile din blocurile învecinate sunt utilizate pentru a prezice eșantioanele de limită din blocul curent. Eroarea de predicție se calculează cu:

$$e_i = x_i - \hat{x}_i \quad (3.5)$$

Audio RDH derivat din PPVO

Metoda PPVO (discutată în secțiunea 2.5) reprezintă o soluție eficientă de a insera date în imagini digitale. Principala contribuție a acestei abordări este selecția dintre două valori de predicție distincte pentru fiecare pixel gazdă. În continuare se urmărește menținere acestui avantaj și adaptarea sa pentru domeniul audio. Noua metodă folosește trei valori posibile de predicție pentru fiecare eșantion audio: o medie simplă a eșantioanelor învecinate din jurul poziției curente și două medii ponderate. Diferențele între valori învecinate sunt folosite pentru a selecta dintre cele două medii ponderate. Inserția reversibilă PPVO este apoi folosită pentru a selecta predictorul final pentru fiecare eșantion din cei doi predictori rămași (media simplă și o medie ponderată). Algoritmul de inserție a datelor este, de asemenea, adaptat pentru a permite inserția a mai multor biți într-o singură valoare gazdă. Această abordare exploatează mai bine histograma erorilor de predicție întâlnită în domeniul audio.

Metoda RDH introdusă utilizează un context de predicție centrat pe eșantionul curent, prin urmare, atât valorile originale, cât și cele modificate sunt utilizate pentru predicție. Rețineți că ordinea inversă în etapa de decodare menține reversibilitatea schemei. Etapa de inserție începe prin calcularea unei perechi de valori absolute ale diferenței pentru fiecare eșantion gazdă:

$$\begin{cases} d'_i = |x_{i-1} - x_{i-2}| + |x_{i-2} - x_{i-3}| + |x_{i-3} - x_{i-4}| \\ d''_i = |x_{i+1} - x_{i+2}| + |x_{i+2} - x_{i+3}| + |x_{i+3} - x_{i+4}| \end{cases} \quad (3.6)$$

unde $|x| = \begin{cases} x & \text{dacă } x \geq 0 \\ -x & \text{dacă } x < 0 \end{cases}$

Complexitatea locală a eșantionului curent este calculată cu:

$$l_c = d'_i + d''_i \quad (3.7)$$

În continuare se folosește o valoare prag pentru complexitate, T_c , pentru a controla capacitatea oferită de metodă. Doar eșantioanele x_i ce îndeplinesc $l_c \leq T_c$ pot fi folosite pentru ascunderea datelor.

Prima valoare prezisă \hat{x}'_i este calculată ca:

$$\hat{x}'_i = \left\lfloor \frac{x_{i-2} + x_{i-1} + x_{i+1} + x_{i+2}}{4} + \frac{1}{2} \right\rfloor \quad (3.8)$$

A doua valoare prezisă (necesară pentru inserția PPVO), \hat{x}_i'' , este selectat dintre două valori de predicție (mediile ponderate ale celor mai apropiati vecini în cele două direcții disponibile). Selecția se bazează pe valorile diferențelor absolute calculate cu ecuația (3.6).

$$\hat{x}_i'' = \begin{cases} \left\lfloor \frac{3x_{i-1}+x_{i-2}}{4} + \frac{1}{2} \right\rfloor & \text{dacă } d'_i \leq d''_i \\ \left\lfloor \frac{3x_{i+1}+x_{i+2}}{4} + \frac{1}{2} \right\rfloor & \text{altfel} \end{cases} \quad (3.9)$$

Dacă $\hat{x}'_i = \hat{x}''_i$, calculăm un nou \hat{x}''_i pentru a avea două valori distincte de predicție:

$$\hat{x}''_i = \begin{cases} \hat{x}'_i + 1 & \text{dacă } \hat{x}'_i = \hat{x}''_i \\ \hat{x}''_i & \text{altfel} \end{cases} \quad (3.10)$$

Eșantioanele cu $\hat{x}'_i < x_i < \hat{x}''_i$ nu pot fi folosite pentru ascunderea datelor și sunt lăsate nemodificate. Pentru restul de eșantioane se determină eroarea de predicție:

$$e_i = \begin{cases} x_i - \hat{x}'_i & \text{dacă } x_i \leq \hat{x}'_i \\ x_i - \hat{x}''_i & \text{dacă } x_i \geq \hat{x}''_i \end{cases} \quad (3.11)$$

Inserția reversibilă de date se face printr-o nouă ecuație de tip multi-bit derivată din PEE:

$$x_i = \begin{cases} x_i + (n-1)e_i - d_j & \text{dacă } e_i > -T_e \text{ și } x_i \leq \hat{x}'_i \\ x_i + (n-1)e_i + d_j & \text{dacă } e_i < T_e \text{ și } x_i \geq \hat{x}''_i \\ x_i - (n-1)T_e & \text{dacă } e_i \leq -T_e \text{ și } x_i \leq \hat{x}'_i \\ x_i + (n-1)T_e & \text{dacă } e_i \geq T_e \text{ și } x_i \geq \hat{x}''_i \end{cases} \quad (3.12)$$

unde n controlează numărul de biți inserați în x_i ; T_e reprezintă pragul de marcare, iar d_j este informația ascunsă. Un eșantion marcat conține $\log_2 n$ biți ascunși.

Rezultate și concluzii

Același set de test alcătuit din 70 de fișiere audio este utilizat pentru a evalua eficiența metodelor propuse. Figura 3.2 oferă o comparație între SVO, schema propusă bazată pe PPVO și rezultatele raportate în [25, 26] pe cele șase clipuri de testare utilizate în [26]. În timp ce schema bazată pe SVO oferă rezultate mai bune pentru capacitate scăzute, capacitatea sa maximă este limitată la 0.3 bps pentru majoritatea fișierelor audio. Schema bazată pe PPVO are cel mai bun rezultat pentru capacitați medii, depășind semnificativ atât pe [25] cât și pe [26]. Metoda PPVO nu permite inserarea de date în pixeli cu $\hat{x}'_i < x_i < \hat{x}''_i$, prin urmare capacitatea sa maximă este

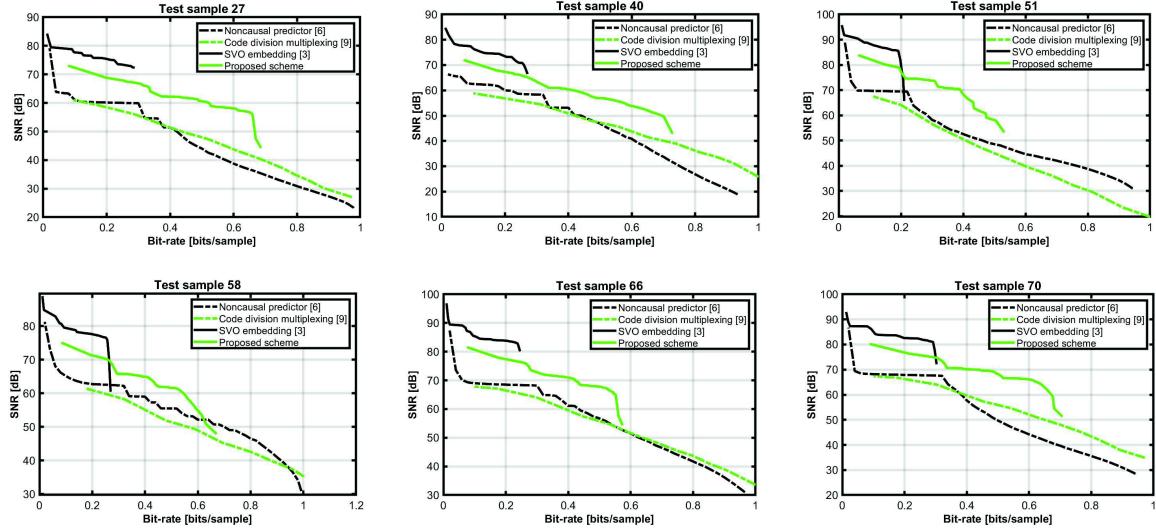


FIGURA 3.2: Rezultatele SNR/Bit-rate predictorul necauzal, codarea prin multiplexare, SVO și metoda derivată din PPVO.

limitată de acest aspect. Pe de altă parte, predicția mai bună oferită de două valori de predicție permite inserarea de mai mulți biți atunci când încorporarea este posibilă.

Comparațiile SNR între schemele propuse SVO, PPVO și [25] sunt furnizate în tabelele din 3.2. Menționăm că mediile setului de testare au fost calculate folosind numai valorile existente în care schema RDH a furnizat capacitatea necesară. De exemplu, în tabelul 3.2, SNR-ul mediu pentru schema PPVO a fost obținut ca medie a 69 de valori, deoarece Clipul 61 nu a putut furniza rata de biți necesară de 0.2 bps.

Schema PPVO a fost dezvoltată pentru rate medii de biți (sub 0.7 bps), dar performanța sa este determinată de eficiența predicției. Din această cauză, ea nu a putut furniza o rată de biți de 0.2 bps pe Clipul 61 menționat mai sus (cu o rată de biți maximă de 0.17 bits/sample), așa cum se poate vedea în Tabelul 3.2. Pe cinci dintre fișierele audio de test (clipuri de testare cu o corelație mai slabă între valorile învecinate) schema PPVO nu a putut oferi o capacitate de 0.3 bps. Pentru trei dintre aceste fișiere audio (clipurile 1, 2 și 67) capacitatea maximă este de aproximativ 0.24 bps Clipul 63 oferă o rată de biți puțin mai mare de 0.28 bps. Setul de clipuri problematice este extins la zece pentru rate de biți de 0.4 bps. Cu toate acestea, schema RDH bazată pe PPVO a depășit rezultatele raportate în [25], cu un câștig mediu în SNR de peste 5 dB. Schema de predicție noncauzală din [25] a avut rezultate mai bune pe 3 clipuri audio de test (clipurile 5, 24, 39) cu o diferență de 1 dB.

Capacitatea mai limitată a metodei SVO este vizibilă în tabelul 3.2 (sub 0.2 bps pentru 19 clipuri de test). A obținut o capacitate maximă de 0.56 bps pe Clip 6, unde schema PPVO a depășit 1 bps. Capacități similare au fost observate și pe celelalte clipuri în care SVO a furnizat

TABELUL 3.2: Comparație în SNR între metoda derivată din PPVO, SVO și predictorul necauzal pentru o capacitate de 0.2 bps [dB].

Fișier de test	Predictor noncausal	SVO	Metoda propusă	Fișier de test	Predictor noncausal	SVO	Metoda propusă	Fișier de test	Predictor noncausal	SVO	Metoda propusă
1	91.74	-	87.75	24	78.63	88.30	81.82	47	68.62	83.96	77.40
2	30.96	-	30.29	25	72.35	81.02	77.03	48	69.63	85.22	78.68
3	88.17	94.77	88.82	26	67.61	77.90	71.01	49	67.38	-	74.11
4	88.08	94.66	88.90	27	60.16	75.40	68.82	50	68.73	60.57	75.99
5	87.64	93.64	88.77	28	69.70	80.05	72.93	51	69.45	80.09	77.42
6	85.51	91.40	86.01	29	70.71	80.29	73.10	52	67.98	72.66	74.10
7	85.54	93.68	86.96	30	65.06	74.55	68.05	53	66.58	81.60	75.41
8	69.26	-	75.05	31	63.98	72.63	68.84	54	66.60	81.89	75.44
9	72.52	56.00	81.11	32	59.04	-	64.83	55	68.47	-	72.36
10	66.13	-	72.87	33	65.52	72.06	68.89	56	67.26	-	76.93
11	71.95	-	76.41	34	67.29	79.09	72.35	57	58.08	73.86	67.30
12	64.81	77.90	73.30	35	66.74	79.37	73.80	58	62.77	77.61	71.31
13	72.40	86.99	80.49	36	63.17	73.94	66.83	59	58.34	-	68.09
14	73.16	85.54	82.04	37	72.42	83.80	77.98	60	67.26	69.69	69.70
15	71.18	86.50	80.27	38	60.12	69.46	62.10	61	63.58	-	-
16	76.50	90.21	83.59	39	71.69	77.89	75.48	62	65.52	-	70.61
17	73.67	89.04	82.68	40	59.99	73.47	67.31	63	63.55	-	64.52
18	74.16	87.48	80.94	41	73.56	85.44	78.86	64	70.03	-	80.46
19	69.32	83.83	77.88	42	64.53	78.42	73.28	65	72.15	-	76.50
20	70.95	84.36	77.86	43	73.39	-	81.76	66	68.50	83.85	77.38
21	70.66	82.25	75.90	44	68.17	77.97	76.87	67	64.08	-	56.11
22	71.08	82.05	75.39	45	63.71	79.00	72.47	68	63.64	-	65.98
23	76.53	86.00	79.45	46	64.54	79.59	73.26	69	60.57	-	70.41
								70	67.89	82.56	76.79
								Medie pe set	69.01	80.77	74.69

o rată de biți peste 0.4 bps. În ceea ce privește SNR, abordarea bazată pe blocuri a SVO oferă cele mai bune rezultate pentru capacități mici (sub 0.3 bps). Pe de altă parte, capacitatea sa redusă limitează gama de aplicații posibile. Pentru capacități între 0.3 și 0.7 bps, schema RDH bazată pe PPVO oferă capacități stabile cu rezultate bune în SNR.

Rezultatele experimentale prezentate în acest capitol au demonstrat că metoda propusă de control al capacității aduce un câștig în SNR comparativ cu abordarea clasică. Metoda propusă de inserție prin SVO obține cele mai bune rezultate pentru capacități mici (sub 0.3 bps), iar metoda propusă, derivată din PPVO este cea mai performantă la capacități medii (între 0.3 și 0.7 bps). Astfel toate metodele introduse în această teză au avantaje clare față de metodele RDH existente, dar au și aplicabilitatea lor distinctă când sunt comparate între ele.

Capitolul 4

Transmiterea de date de la senzori mobili

Există o multitudine de aplicații care achiziționează date ce trebuie transmise la distanță, constrângerile fiind legate de cantitatea acestora, viteză, întârziere etc. În ultimii ani, detecția și analiza mersului reprezintă un subiect care se dezvoltă în cercetare, mai ales datorită utilizării *Unităților de Măsurare Inerțiale (IMU)*. Datele achiziționate sunt ulterior transmise prin diferite soluții de transport de date – Bluetooth, WiFi, GPRS, LTE etc.

Contribuțiile la inserția reversibilă de date în semnale 1D, aduse în această lucrare, pot să fie utilizate și în acest domeniu cu ajutorul unui algoritm simplu de inserție și recuperare a informației.

Intenția nu este de a proiecta un algoritm pentru detecția și analiza mișcării ci de a prelua, a modifica și a transmite datele achiziționate de la IMU respectiv de a recepționa datele și a recuperă informația ascunsă. În acest sens *sursa* va reprezenta un echipament purtabil ce are la bază plăci de dezvoltare prototip, semnalele fiind achiziționate prin intermediul unui IMU de tipul 9DOF, parametrii de bază fiind *accelerația, viteză unghiulară și magnetizarea*. Destinația o va reprezenta o aplicație software ce rulează pe un telefon mobil (sistem Android), aplicație care va evidenția semnalele originale, două fiind alterate iar al treilea fiind transmis prin intermediul biților primelor două semnale.

Ideea este de a insera eșantioanele unui semnal prin RDH în celelalte semnale. Cu o astfel de împachetare se reduce memoria necesară stocării (echivalent, timpul de transmisie).

Este prezentat scenariul în care semnalele se eșantionează cu aceeași frecvență, caz în care eșantioanele unui semnal se serializează și se inserează, bit cu bit. Astfel, la 8 semnale se vor stoca/transmite doar 7.

4.1 Analiza mersului

Mersul pe jos este o activitate zilnică care necesită o interacțiune complexă între nervi, mușchi și sistemul cardiorespirator. Există diferite forme de mers care se consideră a fi sănătoase acestea depinzând de sexul, vârsta și conformația corpului fiecărui individ.

Parametrii de mers, precum lungimea pasului, numărul de pași etc, măsurăți cu sau fără un echipament electronic permit, printr-un proces de analiză, identificarea unor forme anormale de mers care se regăsesc sub denumirea de mers dezordonat. Tulburările (dificultățile) de mers au fie o cauză neurologică, fie non-neurologică, iar debutul lor acut poate indica o leziune a sistemului nervos sau o boală sistemică. Un individ poate avea una sau mai multe tulburări, sau nu va avea dacă mersul său este sănătos. Diagnosticul tulburărilor de mers este realizat de un specialist care observă „calitatea” mersului unui pacient în timp ce merge.

O alternativă ar fi ca pacienții să poarte senzori care colectează date, cum ar fi unitățile de măsură inerțiale care măsoară mișcarea, utilizate pentru a estima valorile parametrilor de mers și efectuează analiza mersului. Stocarea și procesarea datelor înregistrate este ușoară și rapidă, deoarece se poate conecta un senzor la un microcontroler la fel de ieftin cu interfețe fără fir, cum ar fi Bluetooth și Wi-Fi, și poate trimite datele către un smartphone care poate fie să le stocheze și să le proceseze, fie să le transfere unui server pentru o prelucrare detaliată.

Există două faze, evidențiate în figura 4.1, care descriu mersul uman: *faza de poziționare (stance)* și *faza de balansare (swing)*. Faza de poziționare constituie aproximativ 60% din ciclul de mers și este subdivizată în contact initial (*heel-strike, HS*), răspuns la încărcare și poziție de mijloc (*foot-flat, FF*), poziție terminală (*heel-off, HO*) și pre-balansare. Ambele picioare sunt pe pământ la începutul și la sfârșitul fazei de poziție. Fiecare dintre aceste două perioade de sprijin durează aproximativ 10-12% din ciclul de mers. Faza de balansare ocupă aproximativ 40% din ciclul de mers și este subdivizată în balansare initială (*toe-off, TO*), balansare la mijloc și balansare terminală, încheiată prin lovirea călcâiului cu solul.

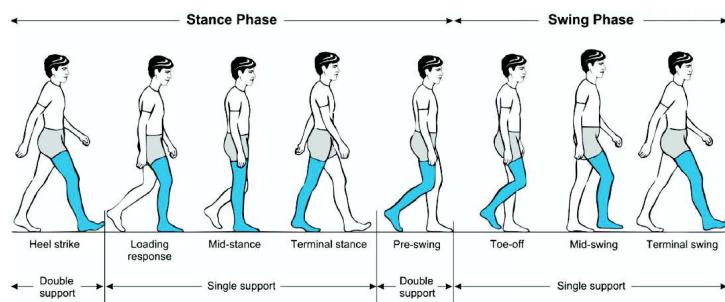


FIGURA 4.1: Fazele ciclului unui mers normal

Dacă se consideră doar patru evenimente identificate prin HS, FF, HO și TO se poate considera că faza de *poziție* este reprezentată de perioada dintre HS și TO iar faza de *balansare* este reprezentată de perioada dintre TO și HS (figura 4.2) [33].

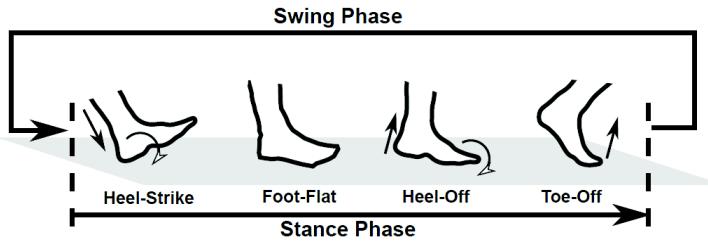


FIGURA 4.2: Detectarea evenimentelor și fazelor mersului

Un pas este reprezentat de avansul unui picior (figura 4.3). Un ciclu de mișcare este echivalent cu un pas făcut de piciorul stâng și un pas făcut de piciorul drept (*lungimea ciclului*).

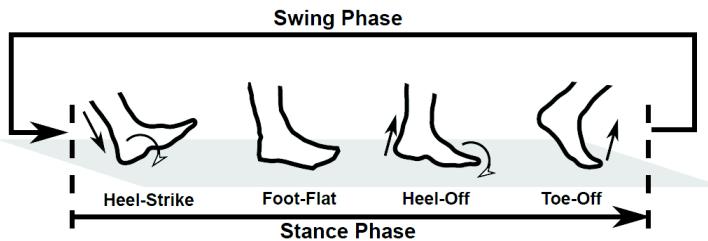


FIGURA 4.3: Terminologia de bază care descrie un ciclu de mers

Diferențele dintre parametrii mișcării picioarelor evidențiază asimetriile, iar poziționarea senzorilor pe ambele picioare poate să fie o soluție pentru evaluarea separată a parametrilor. De cele mai multe ori, chiar dacă distanța parcursă de fiecare picior separat este diferită, distanța parcursă într-un ciclu de mișcare este aceeași. Dacă se va considera un singur senzor atunci se va presupune că simetria există.

Pașii anormali sunt adesea prezenti în tulburările de mers, câteva exemple fiind evidențiate în figura 4.4. Printre tipurile de mers abnormal regăsim, mersul adinamic, mersul legănat, mersul rigid, mersul târât.

Detectia mersului și analiza mersului, utilizând senzori portabili, vor putea să ofere parametrii necesari unui kinetoterapeut / podolog pentru identificarea unei soluții de corecție.

În general, se utilizează senzori inertiali de tip accelerometru, giroscop, magnetometru, combinații într-o unitate de măsurare inertială (IMU). Astfel de exemple sunt MPU-6050 sau MPU-9250. În aplicația dezvoltată s-a utilizat BMX055.

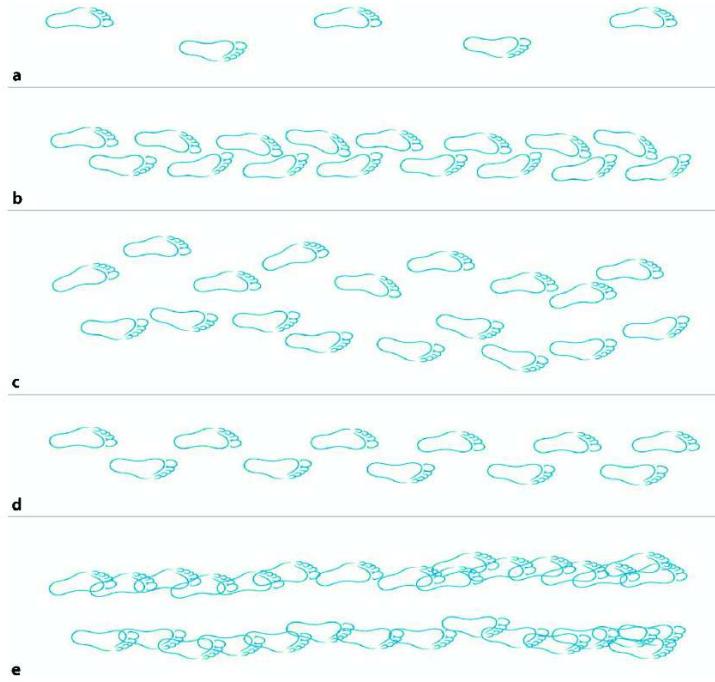


FIGURA 4.4: Secvențe de pași în tulburări clasice de mers.
a) normal; b) adynamic; c) legănat; d) parkinsonian (rigid); e) frontal (târât).

Identificarea evenimentelor unui ciclu de mișcare implică algoritmi precum al lui Tjhai [33] sau Trojaniello [34]. În cazul primului algoritm, axele giroscopului trebuie aliniate cu axele planurilor frontal, sagital și transversal ale corpului uman și cunoscând că faza de legănat începe cu TO și se încheie cu HS (figura 4.5), pentru măsurarea vitezei unghiulare sagitale se preiau trei valori consecutive de evoluție a semnalului, plecând de la primul TO și încheind cu al treilea HS. Dacă semnalul este slab, se trece la urmatorul ciclu de mers care include 3 perioade.

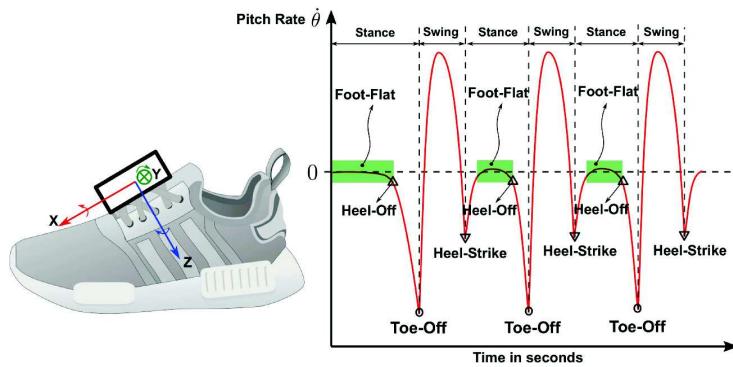


FIGURA 4.5: Semnalul sagital recepționat.

Metoda propusă de Trojaniello [34] se dovedește mai bine adaptată detectării evenimentelor mersului *pathologic* (figura 4.6). Viteza unghiulară, ω , în planul sagital (Oy) este mai mare în zona intermediană a fazei de balansare. Se identifică maximul acestor valori și se definește un interval de încredere care cuprinde această valoare, acesta fiind definit ca intervalul unde viteza unghiulară este mai mare decât un prag, ex. 20% din valoarea maximă. Dacă ω depășește acest prag de mai multe ori, aşa cum se poate întâmpla în unele mersuri patologice, (T_{SW}) este definit ca intervalul dintre prima și ultima trecere. Sunt stabilite două constrângeri suplimentare: (T_{SW}) trebuie să fie de cel puțin 100 ms, iar două (T_{SW}) consecutive ale aceluiași picior trebuie să apară la cel puțin 200 ms una de cealaltă. Acest interval poate fi de încredere că aparține fazei de balansare a unui ciclu de mers.

Atunci când nu apar evenimente de mers (GE) în timpul (T_{SW}), piciorul se va balansa prin aer. Deoarece cele două membre inferioare își alternează fază de balansare în timp ce piciorul opus este în contact cu solul, (T_{SW}) – ul unui membru inferior poate fi folosit ca interval de poziție de încredere (T_{ST}) al celuilalt membru. Astfel, atunci când sunt combinate, cele două (T_{SW}) permit identificarea atât a T_{ST} , cât și a (T_{SW}) pentru fiecare membru, îngustând intervalul de căutare pentru găsirea contactului inițial HS în zona (T_{IC}) și a contactului final TO în zona T_{FC} .

Lovirea călcâiului de sol este identificată ca momentul minimului ω în zona (T_{IC}) înainte de momentul accelerării maxime. TO este momentul de accelerare minimă identificat în zona (T_{FC}), deoarece este de așteptat să apară în momentul unei mișcări bruște înainte de momentul accelerării maxime din zona (T_{FC}).

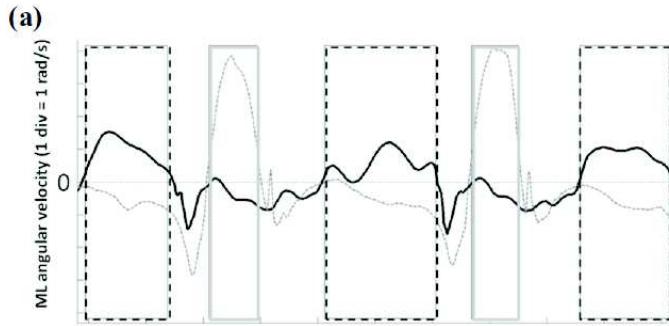


FIGURA 4.6: Viteza unghiulară și accelerarea pentru un subiect cu mobilitate redusă.

În figura 4.6 dreptunghiurile realizate cu linie punctată (a) reprezintă (T_{SW}) iar cele cu linie continuă reprezintă (T_{ST}). În zona (b) dreptunghiurile hașurate reprezintă (T_{IC}) (gri deschis) și (T_{FC}) (gri închis) iar liniile verticale se referă la perioadele GE.

4.2 Proiectarea arhitecturii hardware și software

4.2.1 Arhitectura hardware a modelului experimental

Soluția propusă (figura 4.7) permite achiziționarea datelor de la senzori (accelerometru, giroscop, magnetometru) acestea fiind transmise prin Bluetooth când acesta se împrechează cu dispozitivul Bluetooth al telefonului mobil. Aplicația de pe telefonul mobil permite afișarea rezultatelor fără să realizeze o analiză de semnal pentru detectarea sau analiza mersului. Aplicația se poate completa cu două module software, primul care să realizeze o detecție rapidă a mersului, pe baza analizei datelor curente respectiv un al doilea modul care să transmită aceste date către un server pe care să ruleze un mecanism de “*machine learning*” pe baza unor seturi de date, în final rezultatul fiind expediat aplicației de pe telefonul mobil sau unei aplicații externe gestionate de un kinetoterapeut / podolog.

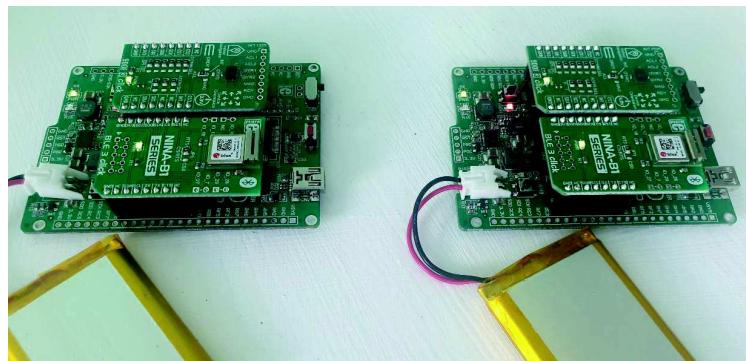


FIGURA 4.7: Module de achiziție de date

În această fază, vorbim despre un concept, dezvoltat prin implementarea soluției utilizând plăci de dezvoltare (figura 4.7). S-au utilizat trei module hardware: clicker 2 for PIC18FJ având la bază microcontrolerul PIC18F87J50 ce lucrează la frecvența de 48 kHz, BLE 3 click ce are la bază modulul NINA-B1 și 9DOF 3 click ce are la bază senzorul BMX055. Aplicația din microcontroler permite transmiterea datelor în varianta clasică respectiv prin *substituție simplă de LSB*. Într-o primă fază s-a utilizat o aplicație de tip Serial Bluetooth Monitor pentru receptionarea datelor iar în final s-a dezvoltat o aplicație pentru un sistem Android care să permită vizualizarea numerică a datelor din sistem respectiv sub formă de grafic.

4.2.2 Inserție RDH cu 2 biți

Factorul limitator la această aplicație este memoria/puterea de procesare limitată a microcontrolerului. Algoritmul de inserție RDH trebuie să fie de complexitate redusă. Valorile

transmise prin Bluetooth sunt cu virgulă mobilă, însă algoritmii RDH folosesc valori gazdă întregi. Pentru a remedia această problemă, algoritmul va efectua modificări doar la partea întreagă a semnalului. Distorsiunea introdusă este controlată cu parametrul de marcă u . Aceasta controlează deplasarea la stânga a zecimalelor valorii gazdă:

$$\bar{x}_i = \lfloor 10^u x_i \rfloor \quad (4.1)$$

Partea fracționară fiind stocată temporar: $f_i = x_i - \frac{\bar{x}_i}{10^u}$.

Eșantionul anterior este folosit pentru a estima eșantionul curent (primul set de măsurători este transmis integral, inserția RDH începe cu setul 2 de citiri de la senzori). Eroarea de predicție e_i pentru eșantionul x_i este calculată ca:

$$e_i = \bar{x}_i - \bar{x}_{i-i} \quad (4.2)$$

În scenariul 1 se inserează o valoare pe 16 biți în 8 eșantioane de aceași mărime, fiind necesar să se ascundă 2 biți în fiecare eșantion gazdă. La scenariul 2 considerăm că semnalul lent (magnetizația) are o frecvență de 4 ori mai mică decât semnalele rapide (accelerația și viteza unghiulară). Astfel, în ambele scenarii este nevoie să se insereze 2 biți de informație în fiecare eșantion gazdă. Un eșantion este marcat dacă eroarea lui de predicție îndeplinește $-T \leq e_i < T$, unde T este pragul de marcă (inserție multi-bit PEE din secțiunea 3.2 când $n = 4$):

$$\bar{x}'_i = \bar{x}_i + 3e_i + 2b_k + b_{k+i} \quad (4.3)$$

unde b_k și b_{k+i} reprezintă cei 2 biți inserți în x'_i .

Inserția este considerată validă doar când setul integral de 16 biți a fost inserat în 8 eșantioane gazdă. Valoarile gazdă curente sunt transmise ca:

$$x'_i = \frac{\bar{x}'_i}{10^u} + f_i \quad (4.4)$$

Altfel, marcarea a eșuat și eșantioanele și valoarea curentă pentru magnetizație sunt transmise direct fără inserție RDH.

Pentru a reduce complexitatea algoritmului, s-a eliminat cazul de deplasare. Eșantioanele ce nu îndeplinesc condiția $-T \leq e_i < T$ sunt considerate ca fiind parte din cazul de marcă eșuată.

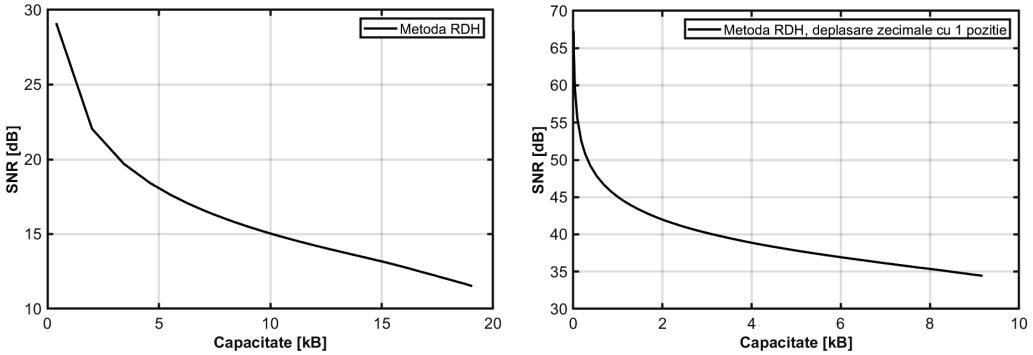


FIGURA 4.8: Rezultatele SNR/Capacitate, scenariu 1 (semnale cu aceeași frecvență de eșantionare)

4.3 Rezultate și concluzii

Pentru cele 3 axe se citesc 3 valori cu virgulă mobilă fiecare valoare fiind reprezentată pe 16 biți. Sistemul va achiziționa 10 citiri/s (la fiecare 100ms) conduce la un volum de date transmis de $9 \times 16 \text{ biți} \times 10 \text{ esantioane/s} \times 60 \text{ s/min} = 86400 \text{ biți/min} \simeq 10.8 \text{ kB/min}$ pentru transmisia clasică în scenariul 1.

Pentru validarea rezultatelor se consideră și cazul marcării prin subtituție de LSB. Acest tip de inserție introduce o distorsiune permanentă în semnalul gazdă. În acest caz, ultimele două straturi LSB ale eșantioanelor gazdă sunt înlocuite cu biții ascunși. Substituția de LSB reduce volumul datelor cu 1.2 kB/min, dar distorsionând semnalul gazdă.

În continuare se consideră transmisia a unui set de 11 înregistări de câte 2 minute cu 10 citiri/s (în total 9×13200 eșantioane, care vor fi separate în eșantioane gazdă și date ascunse). Figura 4.8 prezintă rezultatele metodei RDH propuse. Se observă că SNR-ul obținut depinde de u (variabila de control a deplasării zecimalelor). SNR-ul cel mai bun este obținut când folosit la marcarea cât mai multe zecimale, însă acolo avem și capacitatea de inserție mai mică (domeniul erorilor marcabile devenind mai mic). La transmisia clasică volumul de date trasmise este de 237.6 kB acesta poate fi redus cu până la 16 kB (la un SNR de 12 dB, necesitând extragerea datelor pentru vizualizarea semnalului gazdă) sau cu 6 kB (57dB, distorsiunea semnalului gazdă este imperceptibile și poate fi eliminată complet prin extragerea datelor). Substituția de LSB reduce volumul cu 26.4 kB, însă distorsiunile introduse (cu un SNR de 120 dB) sunt permanente.

Metoda simplă de inserție RDH a redus cu succes volumul de date transmis la o aplicație de analiză a mersului. O parte din valorile citite de la senzori au fost inserate prin RDH în celelalte măsurători pentru a reduce numărul de biți memorati/transmiși. Aceste date pot fi extrase mai târziu, eliminând în totalitate distorsiunea introdusă în eșantioanele gazdă.

Concluzii

În această teză a fost discutată inserția reversibilă de date în fișiere audio. În Capitolul 1 s-a prezentat un scurt istoric al domeniului de cercetare. Cele mai relevante metode din domeniu au fost prezentate și discutate în Capitolul 2.

Apoi s-au propus noi metode de inserție reversibilă în fișiere audio în Capitolul 3. Inițial s-a adaptat o metodă de control a capacitatii din domeniul imaginilor digitale în cel audio. S-a demonstrat că această abordare conduce la un raport mai bun SNR/capacitate pentru două metode RDH existente în domeniu. Metoda PVO (*Pixel Value Ordering*), intens utilizată la inserția în imagini, a fost adaptată pentru domeniul audio sub numele de SVO (*Sample Value Ordering*), aducând cea mai eficientă marcarea pentru capacitați de sub 0.3 bps. Tot în Capitolul 3 s-a introdus o nouă metodă RDH adaptată din PPVO. Pentru imagini digitale PPVO este de capacitate mică, fiind în competiție directă cu PVO. Însă abordare propusă se foloseste de predicția să îmbunătățească împreună cu inserția multi-bit pentru a obține cele mai bune rezultate de capacitate medie. Eficiența metodelor propuse este demonstrată experimental, testele fiind efectuate pe un set de fișiere audio intens utilizate în domeniul de audio RDH. Metodele RDH propuse aduc un câștig în performanță comparativ cu abordările existente în domeniu: SVO oferă o creștere de 7 dB comparativ cu rezultatele din [26] la capacitați sub 0.3 bps, metoda derivată din PPVO oferă în medie 5 dB față de [26] pentru domeniul 0.3 - 0.7 bps (aspect discutat în secțiunea 3.2), iar controlul capacitați propus aduce un câștig de 0.5 dB la metodele existente de marcarea reversibilă în domeniul audio (secțiunea 3.1.1).

În capitolul 4 s-a propus o metodă RDH simplă de camuflare a datelor pentru o aplicație de analiza a mersului. Achiziția de date de la senzori se face cu ajutorul unui microcontroler, care transmite datele prin *Bluetooth* către o aplicație Android de pe un telefon mobil. Inserția reversibilă este folosită pentru a reduce datele transmise prin *Bluetooth*. Factorul limitator la o astfel de aplicație este memoria și puterea de calcul limitată a unui microcontroler. S-a implementat o metodă simplă de inserție RDH ce foloseste măsurătorile anterioare pentru predicție, inserând valorile pentru magnetizare în cele pentru accelerare și viteza unghiulară. Pentru comparație s-a implementat o abordare simplă (nereversibilă) prin substituție de LSB în aceleași valori gazdă.

Contribuții originale

În capitolul 3 au fost introduse trei metode noi de inserție reversibilă în domeniul audio. Prima metodă permite controlul fin al capacitatii prin adaptarea pragului de inserție. Abordarea este de complexitate redusă, ideală pentru fișierele audio. Aceasta nu necesită sortarea eșantioanelor și poate fi folosită indiferent de numărul etapelor de marcăre. Astfel, metoda este extrem de flexibilă și poate fi ușor încorporată în metodele existente de audio RDH.

Metoda SVO adaptează pentru domeniul audio avantajele oferite de PVO în domeniul imaginilor. Spre deosebire de abordarea clasică (care lucrează doar pe blocuri de pixeli), SVO combină inserția pe blocuri (în acest caz blocuri de eșantioane audio) cu cea în eșantioane individuale prin inserția PEE. Astfel, inserția de capacitate redusă pe blocuri este folosită doar în regiuni complexe, iar în cele uniforme se utilizează complet eșantioanele disponibile.

Abordarea derivată din PPVO a adus o serie de contribuții semnificative domeniului. Metoda clasică PPVO folosește doi predictori fizici (minimul și maximul pe contextul de predicție). În schimb, metoda propusă alege dinamic între doi predictori ponderați. Perechea de predicție PPVO este cu predictorul ales și cel medie pe contextul ce înconjoară eșantionul gazdă. Predicția este de complexitate redusă (adunări și împărțiri cu multipli de 2), iar marcarea se face într-o singură etapă. Ecuațiile de inserție PPVO au fost adaptate pentru a permite marcarea multi-bit.

Algoritmii audio RDH dezvoltăți în această teză oferă cele mai bune rezultate pentru capacitați mici (SVO), medii (metoda derivată din PPVO) și îmbunătățesc metodele existente de capacitate mare (prin metoda de control a capacitatii prin adaptarea pragului de inserție).

În capitolul 4 propunem utilizarea algoritmilor RDH în sistemele de achiziție a datelor ca soluție pentru reducerea traficului de date prin inserția unui semnal lent într-un semnal rapid. Inserția reversibilă de date a fost implementată la un sistem cu microcontroler pentru analiza mersului.

Direcții viitoare de cercetare

Rezultatele cercetării din această teză pot fi dezvoltate și extinse prin următoarele abordări:

Rafinarea algoritmul de control al capacitatei

Controlul capacitatei propus în această lucrare foloseste două praguri consecutive de marcăre. Metodele PEE din domeniul imaginilor utilizează sortarea bazată pe complexitatea locală. Soluția optimală ar trebui să fie o combinație dintre cele două metode. Complexitatea locală este folosită pentru eliminarea eșantioanelor din regiuni zgomotoase, iar cele două praguri sunt folosite pentru rafinarea capacitatei pentru valorile rămase.

Optimizarea inserției prin utilizarea histogramei 2D

Histograma 2D a fost folosită cu succes la inserția în imagini (în perechi de pixeli și mai recent combinată cu algoritmul PVO). Numărul mare de eșantioane audio întâlnite într-un fișier audio nu permite implementarea directă a metodei. Însă se pot genera abordări noi 2D de complexitate redusă care să aducă avantajele metodei din domeniul imaginilor în cel audio.

Adaptarea metodei derivate din PPVO pentru domeniul criptat

Stocarea datelor cu ajutorul sistemelor de *cloud* au dus la apariția unor noi metode RDH specializate pentru domeniul criptat. Predicția propusă derivată din PPVO poate fi adaptată pentru o astfel de metodă, permitând inserția reversibilă a datelor în fișiere audio criptate.

Listă de publicații

1. **Bobeica A.**, Dragoi, I. C., Caciula, I. Coltuc, D., Albu, F. și Yang, F., "Capacity control for prediction error expansion based audio reversible data hiding." In 2018 22nd International Conference on System Theory, Control and Computing (ICSTCC), pagini 810-815, IEEE, an 2018 (**17 citări**).
2. **Bobeica A.**, Dragoi, I. C., Caciula, I. și Coltuc, D., "Sample Value Ordering for Audio Reversible Data Hiding." In 2019 6th International Symposium on Electrical and Electronics Engineering (ISEEE), pagini. 1-6, IEEE, an 2019 (**2 citări**).
3. **Bobeica A.**, Dragoi, I. C., Coandă H.-G.. și Coltuc, D., "Adapting the PPVO framework for Audio Reversible Data Hiding." In 15th Edition International conference on Electronics, Computers and Artificial Intelligence (ECAI), an 2023.

Bibliografie selectivă

- [1] Osborne C.F., van Schyndel R.G., Tirkel A.Z. "A digital watermark". In *IEEE Proceedings of International Conference on Image Processing, ICIP-94*, pagini 86–90, an 1994.
- [2] Honsinger C. W., Jones P.W., Rabbani M., Stoffel J.C. "Lossless recovery of an original image containing embedded data". In *US Patent number 6278791*, an 2001.
- [3] Fridrich J., Goljan J. "Lossless data embedding for all image formats". In *In SPIE Proceedings of Photonics West, Electronic Imaging, Security and Watermarking of Multimedia Contents*, page pagini 572–583, an 2002.
- [4] Goljan M., Fridrich J., Du R. "Distortion-free data embedding for images". In *Information Hiding*, page pagini 27–41, an 2001.
- [5] Xuan G., Shi Y.Q., Yang C., Zheng Y., Zou D., Chai P. "Lossless data hiding using integer wavelet transform and threshold embedding technique". In *IEEE International Conference on Multimedia and Expo*, an 2005.
- [6] Celik M. U., Sharma G., Tekalp A. M., Saber E. "Lossless generalized-lsb data embedding". In *IEEE Trans. Image Process*, pagini 253 – 266, an 2005.
- [7] Tian J. "Reversible data embedding using a difference expansion". In *IEEE Trans. on Circuits and Syst. for Video Technol.*, pagini 890 – 896, an 2003.
- [8] Alattar A. M. "Reversible watermark using the difference expansion of a generalized integer transform". In *IEEE Trans. Image Process*, page pagini 1147–1156, an 2004.
- [9] Thodi D. M., Rodriguez J. J. "Prediction-error based reversible watermarking". In *Proc. IEEE Int. Conf. Inf. Process*, page pagini 1549–1552, an 2004.
- [10] Fallahpour M. "Reversible image data hiding based on gradient adjusted prediction". In *IEICE Electron. Exp*, page pagini 870–876, an 2008.

- [11] Xiang S., Wang Y. "Non-integer expansion embedding techniques for reversible image watermarking". In *EURASIP J. Adv. Signal Process*, page pagina 56, an 2015.
- [12] Hui Shi, Baoyue Hu, Yanli Li, Jianing Geng, Yonggong Ren. "Adaptive multi-predictor based reversible data hiding with superpixel irregular block sorting and optimization". *The Imaging Science Journal*, 1–22, 2023.
- [13] Xiaoya Zhang, Yuanzhi Yao, Nenghai Yu. "Convolutional neural network-driven optimal prediction for image reversible data hiding". In *2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSP)*, 1–6. IEEE, 2021.
- [14] Runwen Hu, Shijun Xiang. "Cnn prediction based reversible data hiding". *IEEE Signal Processing Letters*, 28:464–468, 2021.
- [15] Runwen Hu, Shijun Xiang. "Reversible data hiding by using cnn prediction and adaptive embedding". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):10196–10208, 2021.
- [16] Xie Yang, Fangjun Huang. "New cnn-based predictor for reversible data hiding". *IEEE Signal Processing Letters*, 29:2627–2631, 2022.
- [17] Junying Yuan, Huicheng Zheng, Jiangqun Ni. "Reversible data hiding with neighboring-prediction-errors aided sorting and cnn prediction". *Journal of Imaging Science and Technology*, 67(4):1–13, 2023.
- [18] Praveen Kumar, Prateek Ravi, Rajeev Kumar. "Reversible data hiding for interpolated images using cnn based nearest neighbour interpolation". In *2023 Third International Conference on Secure Cyber Computing and Communication (ICSCCC)*, 475–480. IEEE, 2023.
- [19] Li X., Li J., Li B., Yang B. "High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion". In *Signal Process*, pagini 198 – 205, an 2013.
- [20] Peng F., Li X., Yang B. "Improved pvo-based reversible data hiding". In *Digit. Signal Process.*, pagini 255 – 265, an 2014.
- [21] Bo Ou, Xiaolong Li, Yao Zhao, Rongrong Ni, Yun-Qing Shi. "Pairwise prediction-error expansion for efficient reversible data hiding". *IEEE Transactions on image processing*, 22(12):5010–5021, 2013.

- [22] Huo Y., Xiang S., Liu S., Luo X., Bai Z. "Reversible audio watermarking algorithm using non-causal prediction". In *Wuhan University Journal of Natural Sciences*, Vol. 18, No. 5, pagini 455–460, an 2013.
- [23] Sachnev V., Kim H. J., Nam J., Suresh S., Shi Y. Q. "Reversible watermarking algorithm using sorting and prediction". In *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 19, No. 7, pagini 989–999, an 2009.
- [24] Wang F., Xie Z., Chen Z. "High capacity reversible watermarking for audio by histogram shifting and predicted error expansion". In *The Scientific World Journal*, an 2014.
- [25] Xiang S., Li Z. "Reversible audio data hiding algorithm using noncausal prediction of alterable orders". In *EURASIP Journal on Audio, Speech, and Music Processing*, an 2017.
- [26] Ma B. et al. "A reversible data hiding algorithm for audio files based on code division multiplexing". In *Multimedia Tools and Applications*, Vol. 80, pagini 17569–17581, an 2021.
- [27] Geiger R., Yokotani Y., Schuller G. "Audio data hiding with high data rates based on intmdct". In *In Proc. IEEE Int. Conf. Acoust. and Speech, Signal Proc.*, 2006.
- [28] Huang X., Ono N., Echizen I., Nishimura A. "Reversible audio information hiding based on integer dct coefficients with adaptive hiding". In *Lecture Notes in Computer Science*, Vol 8389, an 2013.
- [29] Caciula I., Coltuc D. "Capacity control of reversible watermarking by two-thresholds embedding: Further results". In *International Symposium on Signals, Circuits and Systems (ISSCS2013)*, an 2014.
- [30] Caciula I., Coltuc D. "Capacity control of reversible watermarking by two-thresholds embedding". In *IEEE International Workshop on Information Forensics and Security (WIFS2012)*, an 2013.
- [31] Chin-Feng Lee, Jau-Ji Shen, Yi-Jhen Wu, Somya Agrawal. "Pvo-based reversible data hiding exploiting two-layer embedding for enhancing image fidelity". *Symmetry*, 12(7):1164, 2020.
- [32] Lin C. C., Chen S. C., Hwang K. F. "An information hiding scheme based on pixel value-ordering and prediction-error expansion with reversibility". In *International Journal of Advanced Computer Science and Applications (IJACSA)*, pagini 1–7, an 2014.

- [33] Chandra Tjhai, Kyle O'Keefe. "Using step size and lower limb segment orientation from multiple low-cost wearable inertial/magnetic sensors for pedestrian navigation". *Sensors*, 19(14):3140, 2019.
- [34] Diana Trojaniello, Andrea Cereatti, Elisa Pelosin, Laura Avanzino, Anat Mirelman, Jeffrey M Hausdorff, Ugo Della Croce. "Estimation of step-by-step spatio-temporal parameters of normal and impaired gait using shank-mounted magneto-inertial sensors: application to elderly, hemiparetic, parkinsonian and choreic gait". *Journal of neuroengineering and rehabilitation*, 11:1–12, 2014.
- [35] Solescu I. "Dificultăți de mers, tipuri de tulburări ale mersului". <https://www.catenapascupas.ro/informatii-medicale/dificultati-de-mers-tipuri-de-tulburari-ale-mersului>, 2022.
- [36] Mikroelektronika. "Clicker2 for pic 18f". <https://www.mikroe.com/clicker-2-pic18f>.
- [37] Mikroelektronika. "BLE 3 click". <https://www.mikroe.com/ble-3-click>.
- [38] Ublox. "NINA-B1 series stand-alone bluetooth low energy modules". <https://content.u-blox.com/sites/default/files/NINA-B1-DataSheet-UBX-15019243.pdf>.
- [39] Mikroelektronika. "9DOF 3 click". <https://www.mikroe.com/9dof-3-click>.
- [40] Bosch Sensortec. "BMX055 small, versatile 9-axis sensor module". <https://download.mikroe.com/documents/datasheets/bst-bmx055-ds000-datasheet.pdf>.
- [41] Xinpeng Zhang. "Reversible data hiding in encrypted image". *IEEE signal processing letters*, 18(4):255–258, 2011.
- [42] Y. Hu, H.K. Lee, J. Li. "DE-based reversible data hiding with improved overflow location map". In *IEEE Trans. Circuits Syst. Video Technol.*, 19(2), pp. 250–260, an 2009.
- [43] T. Zen g X. Li, B. Yang. "Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection". In *IEEE Trans. Image Process.* 20(12), pp. 3524–3533, an 2011.
- [44] L. Luo, Z. Chen, M. Chen, X. Zeng, Z. Xiong. "Reveresible image watermarking using interpolation technique". In *IEEE Trans. Inf. Forensics Secur* 5 (1), pp. 187–193, an 2010.
- [45] B. Ou, X. Li, J. Wang. "High-fidelity reversible data hiding based on pixel-value-ordering and pairwise prediction-error expansion". In *Journal of Visual Communication and Image Representation*, 39, pp. 12–23, an 2016.

**MINISTRY OF EDUCATION
UNIVERSITY "VALAHIA" OF TÂRGOVISTE
IOSUD - DOCTORAL SCHOOL OF ENGINEERING SCIENCES
BASIC FIELD: ENGINEERING SCIENCES
FIELD: ELECTRICAL ENGINEERING**

**"Contributions to the development of reversible
information hiding techniques in 1D signals"**

**PHD SUPERVISOR:
Prof. univ. dr. ing. Dinu COLȚUC**

**DOCTORAL STUDENT :
Ing. Constantin-Alin BOBEICĂ**

**TÂRGOVIŞTE
2023**

Table of contents

1	Introduction	2
2	Reversible data hiding methods	6
2.1	RDH insertion based on histogram translation	6
2.2	RDH insertion based on prediction error expansion	8
2.3	RDH insertion based on pixel value ordering (PVO)	11
2.4	RDH insertion based on enhanced pixel value ordering (IPVO)	13
2.5	RDH insertion based on PPVO	14
3	Development of new RDH methods for audio	16
3.1	Capacity control by adjusting the insertion threshold	16
3.1.1	Results and conclusions	17
3.2	Audio RDH derived from PVO and PPVO	19
4	Transmission of data to mobile sensors	24
4.1	Walk analysis	25
4.2	Hardware and software architecture design	29
4.2.1	Hardware architecture of the experimental model	29
4.2.2	2-bit RDH insertion	29
4.3	Results and conclusions	30
	Conclusions	32
	Original contributions	33
	Future directions for research	34
	List of publications	35
	Selective bibliography	36

Chapter 1

Introduction

Recently, since the 2000s, there has been an increase in the processing power and storage capacity of smart devices (computers, tablets, smartphones), which has led to an accelerated use of digital data. Today, almost all multimedia content is being digitally stored, transmitted and processed. The fact that data, in digital format, can be easily reproduced and distributed has made it significantly more difficult to secure original content and intellectual property. To remedy this problem, a new generation of algorithms for transmitting and securing digital data has emerged.

Digital watermarking is the insertion of hidden information into a host signal that results in an imperceptible change in the signal (Osborne et al. in [1]). Based on this idea, specialised algorithms have been developed and adapted for various host signals (audio or video files, still images, etc.) and depending on the desired application (copyright protection, transmission monitoring, content authentication, data annotation).

1.1 Reversible data hiding

Compared to classical data insertion, reversible insertion algorithms (RDH - reversible data hiding) offer the possibility of accurate recovery of the host signal by fully extracting the hidden data. It must be taken into account that the secret message insertion process produces distortions.

RDH algorithms for images exploit the redundancy of signals (images, video, sound, etc.). Usually, two criteria are adopted to evaluate the efficiency of an RDH method: distortion (evaluated by SNR - Signal to Noise Ratio on audio files) and embedding capacity (EC). Comparing the marked host signal with the original, the first criterion measures the change due to data hiding based on the Mean Square Error (MSE) obtained. A high SNR value indicates better imperceptibility, i.e., the host signal is similar to its original version.

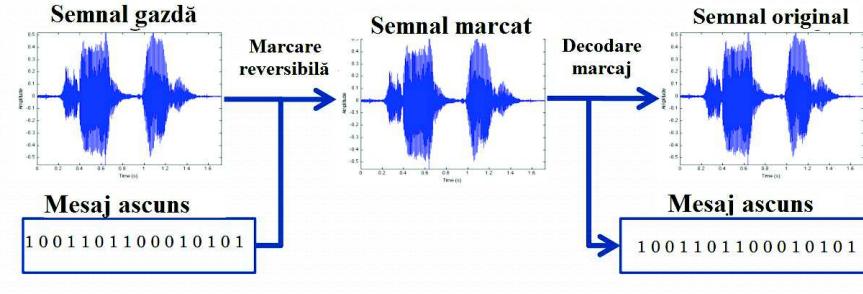


FIGURE 1.1: Audio signal reversible data hiding

1.2 Thesis content

In chapter 2 the most relevant methods in the field of reversible data insertion are presented and discussed. Here the classical insertion and extraction algorithms of histogram translation and prediction error expansion are described, as well as recent ones derived from the ordering of pixel values.

Three new methods of reversible insertion into audio files developed in this thesis are presented in chapter 3. The first method allows fine control for the insertion capacity depending on the host file. The second proposed method combines two existing approaches from insertion into digital images (error histogram translation with pixel value ordering), adapting them for reversible insertion into audio samples. This method achieves the best results reported in the literature for capacities below 0.3 bps (bits-per-sample). The third method (a multi-bit approach with derivative prediction from the ordering of pixel values) provides an efficient solution for capacities between 0.3 and 0.7 bps.

In chapter 4 a simple insertion method for a walking analysis application was proposed, aiming to reduce the amount of data transmitted by the application. Data acquisition from sensors is done using a microcontroller, which transmits the data via Bluetooth to an Android app on a mobile phone.

The research findings are presented in the last part of the thesis.

1.3 Current status of reversible data insertion schemes

Starting from the classical scheme in [1], Hossinger et al. is the developer of the first reversible marking scheme [2]. The method is based on inserting an authentication code into a digital image by simply adding modulo 256, into images, $W = (I+w) \bmod 256$, where I , w and W represent the original image, the mark and the marked image. As w is known, the original image is obtained by subtracting modulo 256, $I = (W-w) \bmod 256$. The method introduces salt and pepper noise due to the modulo 256 arithmetic.

Friedrich et al. propose in [3], the first reversible marking scheme based on lossless compression. The scheme is further improved in [3, 4]. The problem with these schemes is that, in order not to generate visible distortions, the insertion must be done in the area of the less significant bits. In the region of the least significant bits, the compression rate is very low leading to a very low insertion rate. In [5], Xuan et. al. proposed a high-throughput RDH method based on the wavelet transform (IWT). For compression-based methods we also mention the schemes reported by Celik et. al. in [6].

In [7], Tian presents a promising high-throughput RDH insertion method based on difference expansion (DE). This method operates on pairs of pixels and inserts information into the difference between pixels. The pixel pairs are modified so that the difference between pixels doubles. Using difference expansion, a maximum capacity of 0.5 bpp is achieved for one marking step, taking into account that one bit of information is inserted in each pixel pair. Compared to previous schemes based on lossless compression, difference expansion performs better by providing higher insertion capacity and lower distortion.

Alattar in [8], proposed a new method that is based on the difference expansion developed by Tian, but this time relying on blocks of pixels of arbitrary size instead of using pairs of pixels. Thus he manages to reach the maximum capacity of $n - 1/n$ bpp, starting from the idea of inserting in a group of n pixels, $n - 1$ bits. PEE is first proposed by Thodi and Rodriguez in [9], and this technique was widely adopted by many subsequent RDH works in [10, 11]. Recent contributions to the PEE method were made in [12] (based on host pixel selection and sorting). Convolutional neural networks (CNN) have been successfully used for prediction in [13], [14], [15] and [16]. The CNN-based prediction was then optimized by sorting [17] and refining the original prediction [18].

Pixel-Value-Ordering is an efficient method for high-fidelity reversible marking. The PVO method developed by X. Li et al. in [19], divides the image into equal blocks. F. Peng et al. in [20] present the improved pixel value ordering method, which improved the difference calculation process. PVO is an extremely popular approach for insertion into digital images.

The PVO approach has been introduced as a predictor for histogram translation and for PEE, receiving the name "Pixel-based pixel value ordering" (PPVO). This approach has also been significantly improved in recent years. The most significant contributions have been: optimizing the insertion, introducing a hybrid predictor and optimizing the host pixel selection.

Paired-pixel insertion, introduced in [21] and further optimized, allows more efficient insertion than classical insertion in host pixels. This approach has been used recently, but the best results have been obtained by combining it with PVO.

In the audio domain, RDH methods have been derived from those for digital images, most commonly from PEE. The method from [22] uses a simple non-causal predictor (idea adapted

from [23]). This predictor is replaced in [24] by a linear one, computed by linear regression on a variable number of audio samples. In [25], the context samples are first sorted, then the linear predictor is determined. In [26] a reversible coding based on multiplexing is used. The method in [27] inserts robust in the domain of the integer-valued DCT transform, an approach made reversible in [28].

Chapter 2

Reversible data hiding methods

2.1 RDH insertion based on histogram translation

The histogram translation algorithm introduced for reversible marking of digital images selects at each marking step two values of the predicted error: e_p and e_n . It will modify pixels based on these errors, specifically, only those pixels that have the error e_p or e_n will contain secret bits, and the rest of the pixels in the image will be left unchanged or shifted to avoid overlapping with those marked pixels.

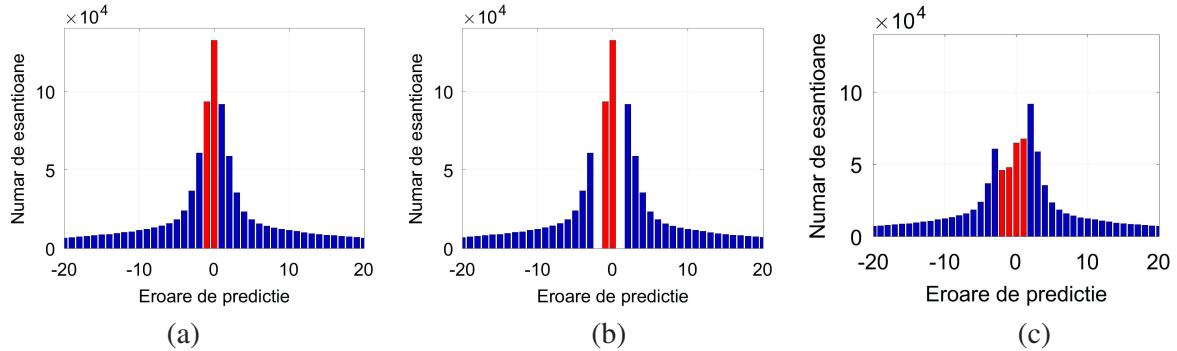


FIGURE 2.1: Stages of HS marking.

a: selection of e_p and e_n ; b: displacement of adjacent values; c: insertion of the mark.

HS data insertion algorithm

The functionality of the tagging algorithm consists of scanning the host image, reading the value of each pixel in *rasterscan* order (left to right and top to bottom). If the prediction error of the analyzed pixel is equal to e_p or e_n , it can be tagged with:

$$x'_{i,j} = \begin{cases} x_{i,j} + b, & \text{if } e_{i,j} = e_p \\ x_{i,j} - b, & \text{if } e_{i,j} = e_n \end{cases} \quad (2.1)$$

where the hidden bit is $b \in \{0, 1\}$. When $b = 0$, the pixels in which a bit of information has been inserted remain unchanged, or when $b = 1$, the pixels are shifted so that e_p is one unit closer to the extremes of the histogram, e_{\min} or e_{\max} . The presence of the marking is indicated by the following errors : $\{e_p, e_n, e_p + 1, e_n - 1\}$.The pixel values in the marked image that have the initial error $e_p + 1$ or $e_n - 1$, must be shifted so as to eliminate ambiguity between the marked pixels and themselves. But shifting them moves the ambiguity problem to the pair $e_n - 2, e_p + 2$. In this case, for the correct distinction of the marked pixels from the original ones, all values that have $e_{i,j} \in [e_{\min}, e_n - 1] \cup [e_p + 1, e_{\max}]$ will be shifted into the domain $[e_{\min} - 1, e_n - 2] \cup [e_p + 2, e_{\max} + 1]$:

$$x'_{i,j} = \begin{cases} x_{i,j} + 1, & \text{if } e_{i,j} > e_p \\ x_{i,j} - 1, & \text{if } e_{i,j} < e_n \end{cases} \quad (2.2)$$

In the case of certain values of e_p and e_n we will notice that there are pixels satisfying the condition $e_n < e_{i,j} < e_p$, and in this case their errors will no longer overlap with the displaced or marked ones and will have the original value.

In the case of 8-bit images, grey levels have their limits in the range $[0, 255]$. At each marking process, the maximum distortion introduced in the host pixel is 1, but the image can have grey levels of 0 or 255 and cannot be shifted, becoming -1 or 256 . Before marking, the problem pixels are identified and shifted to 1 or 254 respectively, i.e. to the inside of the range, which keeps the marking scheme reversible. The audio signals are generally 16 or 24 bits, with the range being $-32,768$ to $32,767$ and $-8,388,608$ to $8,388,607$ respectively.

This marking technique uses a map to store the position of the displaced pixels and then compresses it losslessly and together with the values assigned to e_p and e_n forms the auxiliary data. Inserting these into the image is done by substituting LSBs. The substituted bits are also stored together with the markup.

The reversible marking scheme using histogram translation provides an insertion capacity that is given by the number of pixels that have $e_{i,j} \in \{e_p, e_n\}$:

$$C = H(e_p) + H(e_n) - D_{\text{aux}} \quad (2.3)$$

where $H(e_n)$ and $H(e_p)$ represent pixels with $e_{i,j} = e_n$ and $e_{i,j} = e_p$ respectively and the size of the auxiliary data measured in bits is denoted: D_{aux} . In the case of natural images, the number of pure black or white pixels is very low (it may be zero for some images), which makes the bit number of D_{aux} not large. But with each marking step, its size increases and becomes a problem.

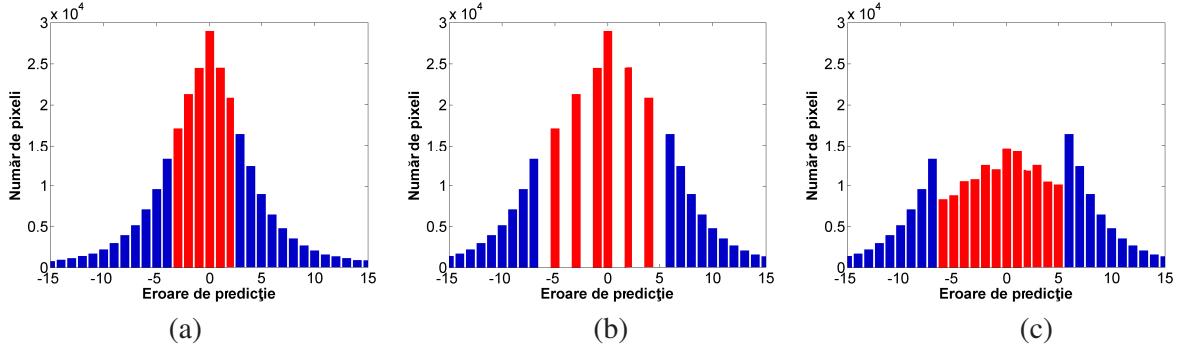


FIGURE 2.2: Viewing the stages of PEE marking on the prediction error histogram.
a: selecting the interval $[-T, T]$; b: displacement of neighbouring values; c: mark insertion

2.2 RDH insertion based on prediction error expansion

For tagging by expanding the prediction error, a reversible embedding algorithm (PEE) has been developed using a range of prediction errors of the form $[-T, T]$, where T is the tagging threshold. Values not belonging to this interval are shifted to create a free space used for marking. It should be noted that the inserted distortions are no longer bounded at ± 1 , and go up to $\pm T$.

PEE data insertion algorithm

Use the em raster scan order again to go through the image pixel by pixel. Then compare the prediction error corresponding to the current pixel with the threshold mark T . If $-T \leq e_{i,j} < T$ a bit of information is inserted into $x_{i,j}$:

$$x'_{i,j} = x_{i,j} + e_{i,j} + b \quad (2.4)$$

After insertion, the range of values $[-T, T]$ is expanded to $[-2T, 2T]$. Pixels that have $e_{i,j} \in [e_{\min}, -T) \cup [T, e_{\max}]$, to eliminate ambiguity between them and the pixels that have been marked, are shifted by $\pm T$:

$$x'_{i,j} = \begin{cases} x_{i,j} + T, & \text{if } e_{i,j} \geq T \\ x_{i,j} - T, & \text{if } e_{i,j} < -T \end{cases} \quad (2.5)$$

Improved capacity control

The distortion introduced by the PEE's reversible data hiding scheme is composed of the distortion produced by the marking insertion and the pixel displacement distortion.

The distortion given by the insertion is achieved by expanding the prediction error and adding bits of information. As mentioned in the previous section, instead of creating space for B bits, we expand the prediction error for N pixels where the prediction error is less than T and

for the remaining $B - N$ pixels that have a prediction error of $\pm T$. In other words, to achieve the desired insertion capacity, the two-threshold insertion scheme uses the pixels that have the smallest possible prediction errors. Since pixels with $|p_e| < T$ are not enough, once these pixels are inserted, pixels with $|p_e| = T$ are also considered. It is now clear that $T' = T + 1$ keeps the insertion error to a minimum. For example, $T' > T + 1$, we should find $B - N$ additional bits marking fewer rows with this threshold than with $T + 1$, but at the cost of expanding the prediction error larger than $|T|$ and, implicitly, at the cost of increasing the insertion error.

The displacement error introduced by the classical PEE marking scheme is not symmetrical. Uninserted pixels with positive prediction error are shifted by the threshold value, T , while those with negative prediction error are shifted by $-(T - 1)$. The squared error of the displacement results from T^2 at positive prediction errors and $(T - 1)^2$ at negative prediction errors, i.e. an error $2T - 1$ smaller than that for pixels with a positive value.

This asymmetry is due to data insertion. The prediction error of the marked pixels is $2p_e + b$, with $|p_e| < T$. The limit for positive prediction error, $2T - 1$, occurs for $b = 1$, while that for negative prediction error occurs for $b = 0$. The two bounds (T and $-T + 1$) are to provide disjoint prediction error ranges for the marked and unmarked pixels.

Symmetric displacement errors can be obtained by considering distinct thresholds for and positive prediction errors instead of a single threshold. Thus, instead of symmetric thresholds $(-T; T)$, one can consider $(-T; T - 1)$. The shift errors are reduced to $(T - 1)^2$ for both negative and positive values of the prediction errors, but at the cost of a decrease in the value of the insertion capacity. Similarly, we take the threshold $(-T - 1, T)$ to increase the insertion capacity (and displacement error). If we denote by $W(T_n, T_p)$ the capacity of the PEE marking scheme for distinct positive (T_p) and negative (T_n) thresholds, we have:

$$\dots W(-T, T - 1) < W(-T, T) < W(-T - 1, T) \dots \quad (2.6)$$

The condition for displacement error symmetry is:

$$T_p = |T_n| - 1 \quad (2.7)$$

The equation 2.6 provides a finer segmentation of the capacity range than that provided by the symmetric thresholds (or non-symmetric ones). The method can be immediately extended for interpolation between the boundaries provided by the equation 2.6.

The PEE insertion/extraction scheme requires an appropriate insertion threshold T to provide the correct number of host samples, which in turn determines the capacity provided. The

T also controls the insertion distortions introduced by the PEE scheme, therefore the selection of T is crucial for optimal data hiding results.

Current state-of-the-art PEE schemes based on audio RDH use the basic capacity control algorithm first introduced in [9]. Let N be the required number of host samples for the current set:

$$N = \frac{N_b}{2} + \frac{N_R}{2} + N_o \quad (2.8)$$

where N_b is the size of the inserted data, N_R is the number of LSB's stored from R and N_o is the estimated number of samples at risk of overflow (samples with $x_i \notin [-32768 + T, 32767 - T]$). Note that in both [29] and [22], capacity control is discussed only in general terms, for simplicity we will use the equation above.

The prediction error is determined with the following formula for each pixel in the current set:

$$e_i = x_i - \hat{x}_i \quad (2.9)$$

These values are collected in a histogram of prediction errors. Let H be the prediction error histogram, where $H(e_i)$ equals the total number of samples in the current set with prediction error e_i . The insertion threshold is selected as the smallest value T that satisfies:

$$\sum_{e_i=-T}^{T-1} H(e_i) \geq N \quad (2.10)$$

After all hidden data bits $\frac{N_b}{2}$ have been inserted together with the auxiliary data bits $\frac{N_R}{2}$ and no signal bits remain, then the remaining pixels in the set are no longer processed and remain unchanged (and the position of the last modified sample is recorded).

The fine capacity control algorithm is derived from the RDH scheme for images, first proposed in [30] and refined in [29]. Unlike the classical approach (where the vast majority of samples are modified using the selected threshold T), the proposed approach uses two distinct thresholds: T and $T - 1$, where T was determined with (2.10).

Let D be the number of additional host samples that are needed if $T - 1$ was used as the insertion threshold:

$$D = N - \sum_{e_i=-T+1}^{T-2} H(e_i) \quad (2.11)$$

where N was calculated with (3.1) and $\sum_{e_i=-T+1}^{T-2} H(e_i)$ is the number of host samples provided by $T - 1$.

The insertion process for the current set starts by using T as the insertion threshold. After a total of D bits have been successfully inserted into samples with $e_i \in \{-T, T - 1\}$, the insertion threshold becomes $T - 1$ for the remaining samples in the set. In other words, the samples are modified by T until the capacity that could not be provided by $T - 1$ is inserted, then the remaining samples are modified by $T - 1$.

The main improvements over [30] and [29] are the capacity estimation based on the signalling bits in (3.1) and the simplified implementation model described in this section. In addition, both approaches have been developed for image RDH and, to the best of our knowledge, the classical (lower) capacity control algorithm is the only model currently used in audio RDH.

2.3 RDH insertion based on pixel value ordering (PVO)

Pixel value ordering (PVO) [19] is the reversible insertion according to the grey value ordering of pixels. In reversible data insertion based on prediction error expansion, the prediction error histogram is divided into two regions, one is called the inner region and the other the outer region. The prediction errors in the inner region of the histogram are expanded to contain data, while those in the outer region are shifted so that the inner and outer regions remain separate after data insertion (similar to the HS and PEE methods discussed above). If there are only one or two histogram values in the inner region, i.e., the maximum change in pixel value in data insertion is 1, and it can be inferred that the change value for the host image is $0.5N_c + N_s$, where N_c and N_s represent the insertion capacity (in bits) respectively the number of shifted pixels.

In this section, as a review, the PVO-based method of Li et al. [19] will be presented first, and then the method of Lee et al. [31]

To begin, the host image is divided into non-overlapping blocks of equal size. For a given block X containing n pixels, sort their values (x_1, \dots, x_n) in ascending order to get $x_{\sigma(1)}, \dots, x_{\sigma(n)}$, where $\sigma : 1, \dots, n \rightarrow 1, \dots, n$ represents a unique one-to-one mapping such that $x_{\sigma(1)} \leq \dots \leq x_{\sigma(n)}$, $\sigma(i) < \sigma(j)$ if $x_{\sigma(i)} = x_{\sigma(j)}$ and $(i) < (j)$. Then use the second largest value, $x_{\sigma(n-1)}$, to predict the maximum $x_{\sigma(n)}$. The corresponding prediction error is:

$$PE_{\max} = x_{\sigma(n)} - x_{\sigma(n-1)} \quad (2.12)$$

While the height that has $PE_{\max} = 1$ is usually the peak value of the histogram, similar to the prediction error PE_{\max} is changed to:

$$PE_{\max} = \begin{cases} PE_{\max} & \text{if } PE_{\max} = 0, \\ PE_{\max} + b & \text{if } PE_{\max} = 1, \\ PE_{\max} + 1 & \text{if } PE_{\max} > 1. \end{cases} \quad (2.13)$$

where $b \in \{0, 1\}$ represents the bit of information to be inserted. According to the maximum value $x_{\sigma(n)}$ is changed to:

$$x = x_{\sigma(n-1)} + PE_{\max} = \begin{cases} x_{\sigma(n)} & \text{if } PE_{\max} = 0, \\ x_{\sigma(n)} + b & \text{if } PE_{\max} = 1, \\ x_{\sigma(n)} + 1 & \text{if } PE_{\max} > 1. \end{cases} \quad (2.14)$$

The other values $x_{\sigma(1)}, \dots, x_{\sigma(n-1)}$ remain unchanged. Therefore, the marked value of x is (y_1, \dots, y_n) , where $y_{\sigma(n)} = \tilde{x}$ and $y_i = x_i$ for each $i \neq \sigma(n)$. In the above procedure, since the maximum $x_{\sigma(n)}$ is either unchanged or increased, the order of pixel values (the σ mapping) remains unchanged. Therefore, for a marked block whose value is (y_1, \dots, y_n) , by calculating the prediction error:

$$PE_{\max} = y_{\sigma(n)} - y_{\sigma(n-1)} \quad (2.15)$$

data extraction and image restoration can be performed as follows:

- if $PE_{\max} = 0$, the block remains unchanged when data is inserted and its original value is only (y_1, \dots, y_n)
- if $PE_{\max} \in \{1, 2\}$, the block is expanded to incorporate the insert data. Insert data bit $b = PE_{\max} - 1$ and the original value is (x_1, \dots, x_n) , where $x_{\sigma(n)} = y_{\sigma(n)} - b$ și $x_i = y_i$ for any $i \neq \sigma(n)$.
- if $PE_{\max} > 2$, the block is displaced by the inserted data and its original value is (x_1, \dots, x_{σ}) , where $x_{\sigma(n)} = y_{\sigma(n)} - 1$ și $x_i = y_i$, for $i \neq \sigma(n)$.

For PVO the amount of displaced values can be reformulated as:

$$P_{shifted} = \frac{\#\{PE_{\max} > 1\}}{\#\{PE_{\max} \geq 1\}} \quad (2.16)$$

where PE_{\max} is the host image prediction error defined in the formula (2.15)

In addition to inserting at the maximum block value, PVO also allows insertion at the minimum. Specifically, the second smallest value, $x_{\sigma(2)}$, can be used to predict the minimum

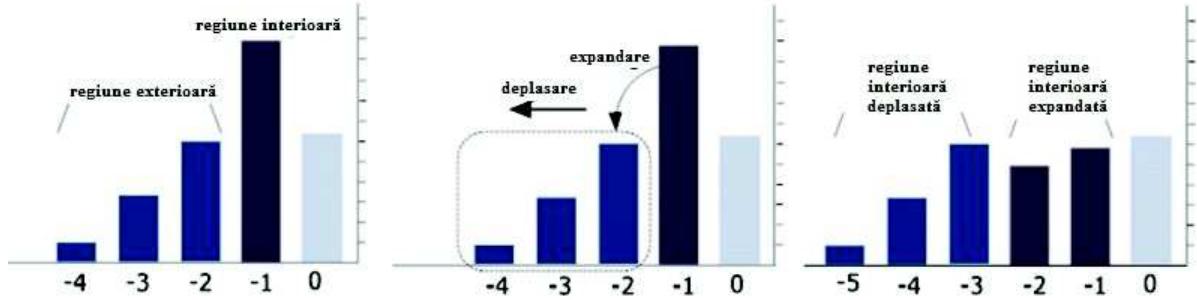


FIGURE 2.3: PVO insertion by changing the minimum of a block

$x_{\sigma(1)}$, and the PEE insertion can be implemented by modifying the corresponding prediction error $PE_{\min} = x_{\sigma(1)} - x_{\sigma(2)}$. In this case, the prediction error PE_{\min} is changed as follows:

$$\widetilde{PE}_{\min} = \begin{cases} PE_{\min} & \text{if } PE_{\min} = 0, \\ PE_{\min} - b & \text{if } PE_{\min} = -1, \\ PE_{\min} - 1 & \text{if } PE_{\min} < -1. \end{cases} \quad (2.17)$$

where $b \in \{0, 1\}$ represents the bit of information to insert.

Considering both maximum and minimum, at most two bits can be inserted in the same host block. For example, when the conditions $x_{\sigma(n)} - x_{\sigma(n-1)} = 1$ and $x_{\sigma(1)} - x_{\sigma(2)} = -1$ are both satisfied, two bits can be inserted into the current block.

Finally, we observe that the use of uniform blocks is more favorable for reversible data hiding. The complexity of a block is measured by the difference $x_{\sigma(n-1)} - x_{\sigma(2)}$, and a block is considered flat if its complexity is less than a predefined threshold T . For example, for three blocks in Fig. 2(b), their complexities are $x_1 - x_2 = 43 - 42 = 1$ and $x_2 - x_4 = 119 - 117 = 2$, respectively $x_1 - x_4 = 154 - 153 = 1$. Note that the size $x_{\sigma(n-1)} - x_{\sigma(2)}$ is unchanged after data insertion, so the encoder and decoder can find exactly the same blocks. This guarantees the reversibility of the PVO method when using uniform blocks.

2.4 RDH insertion based on enhanced pixel value ordering (IPVO)

For a given insertion capacity, the Improved Pixel Value Ordering (IPVO) method can use larger block sizes for data insertion and therefore better exploit image redundancy, which is useful for embedding performance.

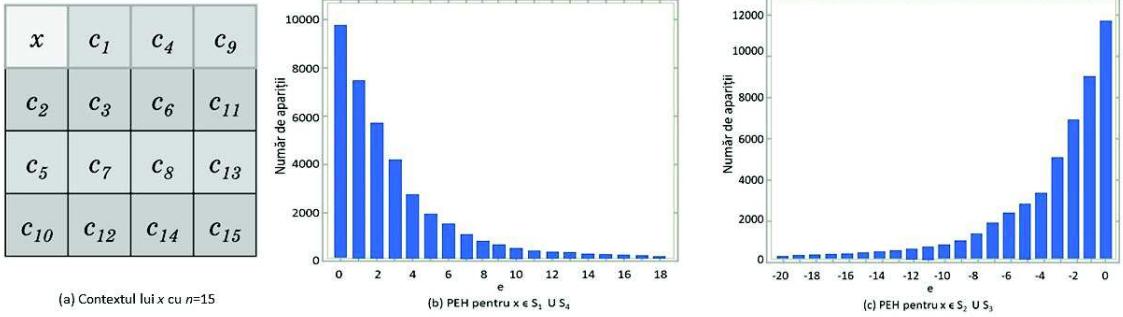


FIGURE 2.4: Prediction context and prediction error histograms for the Lena image with pixels in $S_1 \cup S_4$ și $S_2 \cup S_3$, for PPVO

$$d_{\max} = x_u - x_v \quad (2.18)$$

where

$$\begin{cases} u = \min(\sigma(n), \sigma(n-1)), \\ v = \max(\sigma(n), \sigma(n-1)). \end{cases} \quad (2.19)$$

Here, instead of the difference $PE_{\max} = x_{\sigma(n)} - x_{\sigma(n-1)}$ used by PVO which always has a positive value, in this section the new difference d_{\max} takes values between $(-\infty, +\infty)$. And unlike the case of PE_{\max} , the histogram of d_{\max} is a Laplacian distribution centered at 0, because both $x_{\sigma(n)} - x_{\sigma(n-1)}$ and $x_{\sigma(n-1)} - x_{\sigma(n)}$ are counted taking into account the order of $\sigma(n)$ and $\sigma(n-1)$. In this way, the blocks with $x_{\sigma(n)} = x_{\sigma(n-1)}$ can be exploited by the described method to insert data by modifying d_{\max} .

2.5 RDH insertion based on PPVO

Further on, the PPVO method is briefly analysed. In contrast to previous PVO-based methods [19] Pixel Value Ordering, PPVO [20] Pixel-based Pixel Value Ordering performs pixel-based prediction. The specific data insertion and extraction procedures are presented below.

As illustrated in Figure 2.4, the prediction context of a target pixel is determined by the nearby pixels in the lower right corner (a).

It builds a prediction context $C = c_1, \dots, c_n$, where n is the number of context pixels. The context pixels will then be sorted, and four scenarios can lead to the prediction of x . Then the prediction error is defined.

For a host image, the PPVO insertion process is performed in raster scan order, while the extraction process is performed in reverse. As a result, each marked pixel and the corresponding host pixel have identical prediction contexts.

Then, the marked prediction error is calculated by $\tilde{e} = \tilde{x} - \hat{x}$. The data bit is extracted as $b = 0$ if $\tilde{e} = 0$ and $b = 1$ if $\tilde{e} \in \{-1, 1\}$. Finally, the original pixel is recovered by $x = \hat{x} + e$.

In addition, PPVO uses a pixel selection approach. A pixel will be used for data insertion if its local complexity is less than a specified threshold, indicating that it is easier to make an accurate prediction.

The size of the prediction context for each pixel in PPVO is fixed. However, the performance of the insertion stage may fluctuate depending on the size of the prediction context. Based on the local complexity, in this part we explore the effects of different prediction context sizes on the insertion performance.

In summary, high capacity can be achieved with a small prediction context, but the distortion introduced by the related shift-area increases dramatically due to inaccurate prediction. When a large prediction context is used, the shift-shift distortion is reduced because a lot of the cover pixels are skipped, while only low insertion capacity can be obtained.

Chapter 3

Development of new RDH methods for audio

This chapter introduces a number of new reversible insertion methods in audio. The aim is both to refine existing methods and to formulate new approaches that allow data hiding with a better degree of imperceptibility.

3.1 Capacity control by adjusting the insertion threshold

Current RDH audio methods based on PEE use a basic algorithm for insertion capacity control first introduced in [19]. Let N be the required number of host samples for the current set:

$$N = \frac{N_b}{2} + \frac{N_R}{2} + N_o \quad (3.1)$$

where N_b is the size of the hidden data, N_R is the number of LSB's stored from R and N_o is the estimated number of samples that are at risk of overflow (samples with $x_i \in [-32768 + T, 32767 - T]$).

The proposed capacity control algorithm is derived from the image RDH scheme first proposed in [30] and further refined in [29]. Unlike the classical approach, the proposed approach uses two distinct thresholds: T and $T - 1$.

Let D be the number of additional host samples that are needed to hide a total of N bits if $T - 1$ was used as the insertion threshold:

$$D = N - \sum_{e_i=-T+1}^{T-2} H(e_i) \quad (3.2)$$

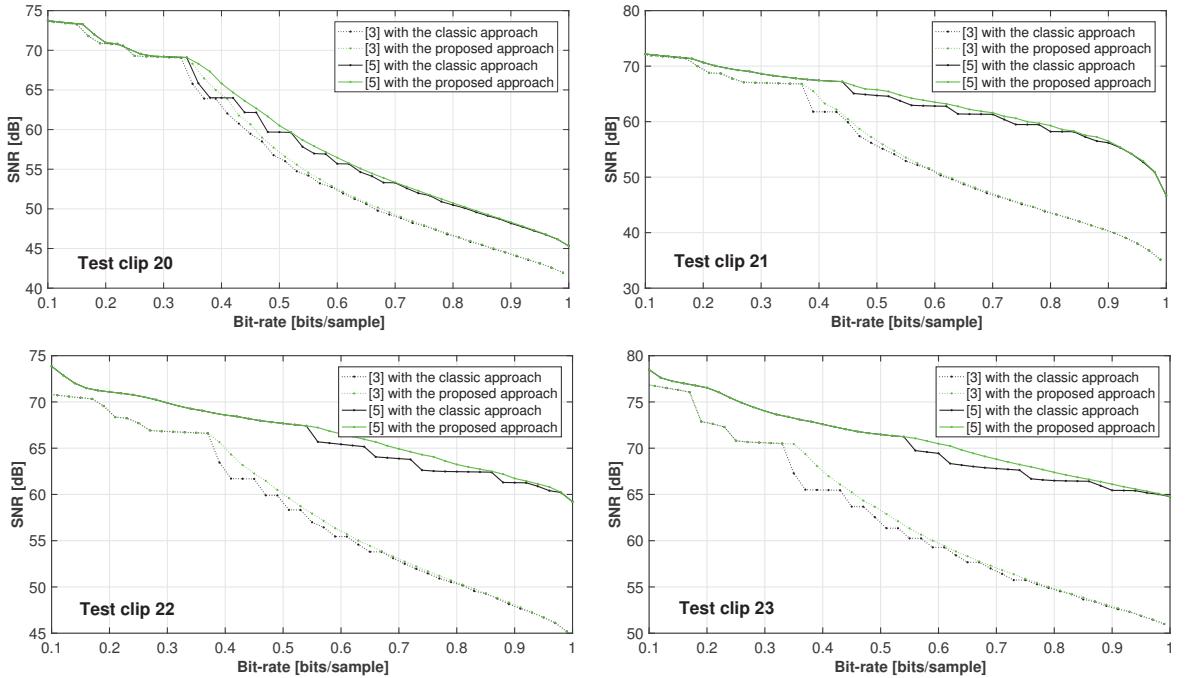


FIGURE 3.1: SNR/Bit-rate results for capacity control

The insertion process for the current set starts by using T as the marking threshold. After a total number of D bits has been successfully inserted into samples with $e_i \in \{-T, T - 1\}$, the marking threshold becomes $T - 1$ for the remaining values in the set. In other words, the samples are modified using T until the required capacity that could not be provided by $T - 1$ has been inserted, then the remaining values can be modified with $T - 1$.

3.1.1 Results and conclusions

In this section, the algorithms presented in the previous chapter are compared with their classical PEE counterparts for RDH in the audio domain. Their performance is evaluated on 70 standard audio files with a sampling rate of 44.1 kHz.

The two capacity control approaches (classical and proposed) are first compared on four of the test clips, the SNR/bit-rate results are shown in Figure 3.1. A clear gain in performance is observed on all four test clips. This gain is more pronounced when using linear prediction.

The performance of the proposed scheme is evaluated in Table 3.1. The proposed capacity control yields an average performance increase of 0.31 dB in SNR for a fixed capacity of 0.6 bps. A larger improvement over the classical approach is observed in linear prediction with an average performance increase of 0.51 dB.

TABLE 3.1: SNR comparison between classical capacity control and the proposed 0.6 bps approach [dB].

File test	Arithmetic average		Linear prediction		File test	Arithmetic average		Linear prediction	
	Clasic	Proposed	Clasic	Proposed		Clasic	Proposed	Clasic	Proposed
1	51.17	51.25	86.92	86.92	36	52.59	54.39	53.64	55.51
2	11.97	12.03	18.05	18.25	37	58.03	58.20	64.85	66.22
3	82.22	82.21	83.33	83.34	38	55.31	55.31	55.57	55.57
4	80.22	80.22	83.25	83.25	39	59.32	60.51	64.27	64.76
5	45.33	45.37	82.86	82.87	40	41.38	41.58	40.86	40.95
6	78.27	78.26	80.68	80.69	41	60.18	61.22	64.52	65.89
7	74.32	75.10	80.74	80.75	42	38.67	38.79	45.91	46.19
8	40.35	40.46	47.59	48.05	43	36.69	36.76	47.61	47.73
9	44.30	44.38	52.66	52.69	44	48.24	48.44	54.78	55.31
10	46.54	46.85	52.33	52.78	45	43.73	43.88	50.64	51.38
11	59.36	59.55	60.19	60.53	46	41.69	41.94	53.22	54.13
12	39.20	39.30	50.28	50.69	47	47.09	47.37	57.25	57.76
13	50.59	50.91	58.78	59.46	48	44.75	44.90	53.41	53.84
14	41.96	42.01	55.74	56.03	49	48.85	49.24	48.47	48.56
15	45.87	46.01	52.91	53.38	50	50	50.37	52.82	53.22
16	50.13	50.27	62.59	63.01	51	45.31	45.47	44.60	44.89
17	47.56	47.71	55.40	55.75	52	50.52	50.91	50.98	51.46
18	56.61	57.12	61.16	62.46	53	48.56	48.91	48.73	49.07
19	56.29	56.63	58.80	59.56	54	47.80	48.20	50.99	51.12
20	51.95	52.52	55.69	56.46	55	45.90	46.02	60.11	60.53
21	50.90	51.12	62.81	63.50	56	33.15	33.17	51.88	52.08
22	55.45	55.99	65.42	66.56	57	25.47	25.53	38.70	38.99
23	59.28	59.74	69.45	70.49	58	49.65	49.98	52.18	53.43
24	69.07	69.76	74.17	74.17	59	40.61	40.65	43.54	43.66
25	63.27	64.72	65.01	65.55	60	59.11	59.31	60.57	61.47
26	58.22	58.31	61.01	61.89	61	42.27	42.47	52.99	53.46
27	34.97	35.05	38.74	39.12	62	52.62	53.13	55.02	55.82
28	60.48	61.95	60.48	61.96	63	44.74	44.76	50.95	51.55
29	66.31	66.31	66.31	66.30	64	36.17	36.23	46	46.13
30	60.48	60.47	60.76	60.76	65	51.29	51.75	59.27	60.44
31	56.45	57.01	56.36	56.76	66	39.91	39.98	51.50	51.70
32	44.12	44.59	47.27	48.20	67	49.15	49.37	53.51	54.54
33	59.11	59.99	58.02	58.66	68	54.24	54.57	54.25	54.58
34	54.78	54.84	59.95	61.37	69	40.43	40.71	44.87	45.34
35	41.97	42.07	55.84	56.18	70	42.90	43.08	44.18	44.30
					Average per set	50.22	50.53	56.63	57.14

3.2 Audio RDH derived from PVO and PPVO

In this subchapter, two RDH methods for audio files are introduced. These new methods are based on the PE insertion principles - HS, IPVO and PPVO, which are widely used approaches for insertion into digital images, presented in the sections 2.1, 2.4 and 2.5.

Audio RDH by sorting audio samples (SVO)

Perfectly uniform regions are rare in natural images, but audio samples may have sections of silence. These regions are ideal for prediction and therefore can be easily exploited for RDH. All PVO block-based approaches require that the references and values sorted between them remain unchanged. Insertion capacity is lost if not all samples in the linear sections are considered. This effect is somewhat mitigated by better PVO performance on more complex areas. However, PVO-based schemes do not adequately allow the exploitation of uniform sections.

In order to solve the above mentioned problem, a new PVO-based framing scheme for audio RDH, namely Sample Value Ordering, was created. This approach starts by dividing the host audio file into blocks of size n . For each block, the local complexity of the block is calculated based only on the values of the neighbouring samples:

$$l_c = \max(x_{i+n}, \dots, x_{i+2n}) - \min(x_{i+n}, \dots, x_{i+2n}) \quad (3.3)$$

All values in a block can now be changed without changing the l_c of that block. Because of the small block size ($n \in [3, 24]$), neighboring blocks should have the same type of complexity.

Next, blocks are classified based on local complexity. The smooth blocks (with $l_c \leq t_1$) are now processed as individual samples using the RFH insertion based on the histogram translation in Section 2.1. The predicted values are computed with:

$$\hat{x}'_i = \left\lfloor \frac{3x_{i+2} + x_{i+2}}{4} + \frac{1}{2} \right\rfloor \quad (3.4)$$

Note that the values in neighbouring blocks are used to predict the boundary samples in the current block. The prediction error is calculated with:

$$e_i = x_i - \hat{x}_i \quad (3.5)$$

Audio RDH derived from PPVO

The PPVO method (discussed in section 2.5) is an efficient solution to insert data into digital images. The main contribution of this approach is the selection between two distinct

prediction values for each host pixel. In the following we aim to maintain this advantage and adapt it for the audio domain. The new method uses three possible prediction values for each audio sample: a simple average of neighbouring samples around the current position and two weighted averages. The differences between the neighbouring values are used to select between the two weighted averages. The reversible PPVO insertion is then used to select the final predictor for each sample from the two remaining predictors (the simple average and a weighted average). The data insertion algorithm is also adapted to allow the insertion of multiple bits into a single host value. This approach better exploits the histogram of prediction errors found in audio.

The RDH method introduced uses a prediction context centred on the current sample, therefore both the original and modified values are used for prediction. Note that the reverse order in the decoding step maintains the reversibility of the scheme. The insertion step starts by calculating a pair of absolute difference values for each host sample:

$$\begin{cases} d'_i = |x_{i-1} - x_{i-2}| + |x_{i-2} - x_{i-3}| + |x_{i-3} - x_{i-4}| \\ d''_i = |x_{i+1} - x_{i+2}| + |x_{i+2} - x_{i+3}| + |x_{i+3} - x_{i+4}| \end{cases} \quad (3.6)$$

$$\text{unde } |x| = \begin{cases} x & \text{if } x \geq 0 \\ -x & \text{if } x < 0 \end{cases}$$

The local complexity of the current sample is calculated with:

$$l_c = d'_i + d''_i \quad (3.7)$$

Next, a threshold value for complexity, T_c , is used to control the capacity provided by the method. Only samples x_i satisfying $l_c \leq T_c$ can be used for data hiding.

The first predicted value \hat{x}'_i is computed as:

$$\hat{x}'_i = \left\lfloor \frac{x_{i-2} + x_{i-1} + x_{i+1} + x_{i+2}}{4} + \frac{1}{2} \right\rfloor \quad (3.8)$$

The second predicted value (required for PPVO insertion), \hat{x}''_i , is selected from two predicted values (weighted averages of the nearest neighbors in the two available directions). The

selection is based on the absolute difference values calculated with equation (3.6).

$$\hat{x}_i'' = \begin{cases} \lfloor \frac{3x_{i-1}+x_{i-2}}{4} + \frac{1}{2} \rfloor & \text{if } d'_i \leq d''_i \\ \lfloor \frac{3x_{i+1}+x_{i+2}}{4} + \frac{1}{2} \rfloor & \text{else} \end{cases} \quad (3.9)$$

If $\hat{x}'_i = \hat{x}''_i$, we compute a new \hat{x}''_i to have two distinct prediction values:

$$\hat{x}_i'' = \begin{cases} \hat{x}'_i + 1 & \text{if } \hat{x}'_i = \hat{x}''_i \\ \hat{x}''_i & \text{else} \end{cases} \quad (3.10)$$

Samples with $\hat{x}'_i < x_i < \hat{x}''_i$ cannot be used for data hiding and are left unchanged. For the remaining samples the prediction error is determined:

$$e_i = \begin{cases} x_i - \hat{x}'_i & \text{if } x_i \leq \hat{x}'_i \\ x_i - \hat{x}''_i & \text{if } x_i \geq \hat{x}''_i \end{cases} \quad (3.11)$$

Reversible data hiding is done by a new multi-bit equation derived from PEE:

$$x_i = \begin{cases} x_i + (n-1)e_i - d_j & \text{if } e_i > -T_e \text{ and } x_i \leq \hat{x}'_i \\ x_i + (n-1)e_i + d_j & \text{if } e_i < T_e \text{ and } x_i \geq \hat{x}''_i \\ x_i - (n-1)T_e & \text{if } e_i \leq -T_e \text{ and } x_i \leq \hat{x}'_i \\ x_i + (n-1)T_e & \text{if } e_i \geq T_e \text{ and } x_i \geq \hat{x}''_i \end{cases} \quad (3.12)$$

where n controls the number of bits inserted in x_i ; T_e is the tagging threshold, and d_j is the hidden information. A tagged sample contains $\log_2 n$ hidden bits.

Results and conclusions

The same test set of 70 audio files is used to evaluate the effectiveness of the proposed methods. Figure 3.2 provides a comparison between SVO, the proposed PPVO-based scheme and the results reported in [25, 26] on the six test clips used in [26]. While the SVO-based scheme gives better results for low capacities, its maximum capacity is limited to 0.3 bps for most audio files. The PPVO-based scheme performs best for medium capacities, significantly outperforming both [25] and [26]. The PPVO method does not allow data to be inserted into $\hat{x}'_i < x_i < \hat{x}''_i$ pixels, so its maximum capacity is limited by this. On the other hand, the better prediction provided by two prediction values allows the insertion of more bits when embedding is possible.

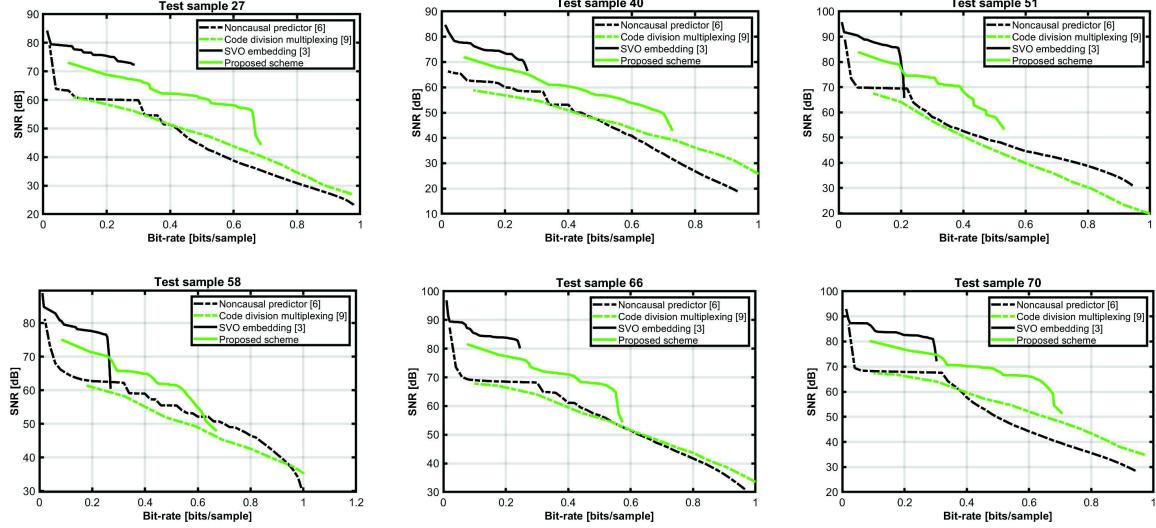


FIGURE 3.2: SNR/Bit-rate results of the non-random predictor, multiplex coding, SVO and PPVO-derived method.

SNR comparisons between the proposed SVO, PPVO and [25] schemes are provided in the tables in 3.2. Note that the test set averages were calculated using only the existing values where the RDH scheme provided the required capacity. For example, in table 3.2, the average SNR for the PPVO scheme was obtained as the average of 69 values because Clip 61 could not provide the required bit rate of 0.2 bps.

The PPVO scheme was developed for medium bit rates (below 0.7 bps), but its performance is determined by the prediction efficiency. Because of this, it could not deliver a bit rate of 0.2 bps on the above-mentioned Clip 61 (with a maximum bit rate of 0.17 bits/sample), as can be seen in Table 3.2. On five of the test audio files (test clips with lower correlation between neighboring values) the PPVO scheme could not deliver 0.3 bps. For three of these audio files (clips 1, 2 and 67) the maximum capacity is about 0.24 bps Clip 63 offers a bit rate slightly higher than 0.28 bps. The set of problem clips is extended to ten for bit rates of 0.4 bps. However, the PPVO-based RDH scheme outperformed the results reported in [25], with an average gain in SNR of over 5 dB. The noncausal prediction scheme in [25] performed better on 3 test audio clips (clips 5, 24, 39) with a difference of 1 dB.

The more limited capacity of the SVO method is visible in the table 3.2 (below 0.2 bps for 19 test clips). It obtained a maximum capacity of 0.56 bps on Clip 6, where the PPVO scheme exceeded 1 bps. Similar capacities were observed on the other clips where SVO provided a bit rate above 0.4 bps. In terms of SNR, SVO's block-based approach provides the best results for low capacities (below 0.3 bps). On the other hand, its low capacity limits the range of possible

TABLE 3.2: Comparison in SNR between the method derived from PPVO, SVO and the non-aural predictor for a capacity of 0.2 bps [dB].

File test	Noncausal SVO predictor	Proposed approach	File de test	Noncausal SVO predictor	Proposed approach	File test	Noncausal SVO predictor	Proposed approach			
1	91.74	-	87.75	24	78.63	88.30	81.82	47	68.62	83.96	77.40
2	30.96	-	30.29	25	72.35	81.02	77.03	48	69.63	85.22	78.68
3	88.17	94.77	88.82	26	67.61	77.90	71.01	49	67.38	-	74.11
4	88.08	94.66	88.90	27	60.16	75.40	68.82	50	68.73	60.57	75.99
5	87.64	93.64	88.77	28	69.70	80.05	72.93	51	69.45	80.09	77.42
6	85.51	91.40	86.01	29	70.71	80.29	73.10	52	67.98	72.66	74.10
7	85.54	93.68	86.96	30	65.06	74.55	68.05	53	66.58	81.60	75.41
8	69.26	-	75.05	31	63.98	72.63	68.84	54	66.60	81.89	75.44
9	72.52	56.00	81.11	32	59.04	-	64.83	55	68.47	-	72.36
10	66.13	-	72.87	33	65.52	72.06	68.89	56	67.26	-	76.93
11	71.95	-	76.41	34	67.29	79.09	72.35	57	58.08	73.86	67.30
12	64.81	77.90	73.30	35	66.74	79.37	73.80	58	62.77	77.61	71.31
13	72.40	86.99	80.49	36	63.17	73.94	66.83	59	58.34	-	68.09
14	73.16	85.54	82.04	37	72.42	83.80	77.98	60	67.26	69.69	69.70
15	71.18	86.50	80.27	38	60.12	69.46	62.10	61	63.58	-	-
16	76.50	90.21	83.59	39	71.69	77.89	75.48	62	65.52	-	70.61
17	73.67	89.04	82.68	40	59.99	73.47	67.31	63	63.55	-	64.52
18	74.16	87.48	80.94	41	73.56	85.44	78.86	64	70.03	-	80.46
19	69.32	83.83	77.88	42	64.53	78.42	73.28	65	72.15	-	76.50
20	70.95	84.36	77.86	43	73.39	-	81.76	66	68.50	83.85	77.38
21	70.66	82.25	75.90	44	68.17	77.97	76.87	67	64.08	-	56.11
22	71.08	82.05	75.39	45	63.71	79.00	72.47	68	63.64	-	65.98
23	76.53	86.00	79.45	46	64.54	79.59	73.26	69	60.57	-	70.41
								70	67.89	82.56	76.79
							Average per set		69.01	80.77	74.69

applications. For capacities between 0.3 and 0.7 bps, the PPVO-based RDH scheme offers stable capacities with good SNR results.

The experimental results presented in this chapter demonstrated that the proposed capacity control method brings a gain in SNR compared to the classical approach. The proposed SVO insertion method achieves the best results for low capacities (below 0.3 bps) and the proposed PPVO-derived method performs best at medium capacities (between 0.3 and 0.7 bps). Thus all methods introduced in this thesis have clear advantages over existing RDH methods, but also have distinct applicability when compared to each other.

Chapter 4

Transmission of data to mobile sensors

There are a multitude of applications that acquire data to be transmitted remotely, the constraints being quantity, speed, delay, etc. In recent years, gait detection and analysis is a growing research topic, especially due to the use of Inertial Measurement Units (IMUs). The acquired data is then transmitted via different data transport solutions - Bluetooth, WiFi, GPRS, LTE, etc.

The contributions to reversible data insertion in 1D signals made in this paper can also be used in this area using a simple insertion and information retrieval algorithm.

The intention is not to design an algorithm for motion detection and analysis but to retrieve, modify and transmit the data acquired from the IMU or to receive the data and retrieve the hidden information. In this sense *source* will represent a wearable equipment based on prototype development boards, the signals being acquired through a 9DOF IMU, the basic parameters being *acceleration, angular velocity and magnetization*. The destination will be a software application running on a mobile phone (Android system), which will highlight the original signals, two being altered and the third being transmitted via the bits of the first two signals. The idea is to insert the samples of one signal through the HDR into the other signals. With such packing, the memory required for storage (equivalently, transmission time) is reduced.

The scenario where signals are sampled at the same frequency is shown, in which case the samples of a signal are serialized and inserted, bit by bit. Thus, for 8 signals only 7 will be stored/transmitted.

4.1 Walk analysis

Walking is a daily activity that requires a complex interaction between nerves, muscles and the cardiorespiratory system. There are different forms of walking that are considered to be healthy depending on the gender, age and body shape of each individual.

Walking parameters such as step length, number of steps, etc., measured with or without electronic equipment allow, through a process of analysis, the identification of abnormal gait patterns that are referred to as disordered gait. Gait disturbances (difficulties) have either a neurological or a non-neurological cause, and their acute onset may indicate a nervous system lesion or a systemic disease. An individual may have one or more disorders, or will not have them if their gait is healthy. The diagnosis of gait disorders is made by a specialist who observes the 'quality' of a patient's gait as they walk.

An alternative would be for patients to wear sensors that collect data, such as inertial measurement units that measure motion, used to estimate gait parameter values and perform gait analysis. Storing and processing the recorded data is easy and fast, as one can connect a sensor to an equally inexpensive microcontroller with wireless interfaces such as Bluetooth and Wi-Fi, and send the data to a smartphone that can either store and process it or transfer it to a server for detailed processing.

There are two phases, highlighted in the figure 4.1, that describe human gait: *stance* and *swing*. The stance phase constitutes approximately 60% of the gait cycle and is subdivided into initial contact (*heel-strike*, *HS*), response to loading and midstance (*foot-flat*, *FF*), terminal stance (*heel-off*, *HO*) and pre-swing. Both feet are on the ground at the beginning and end of the stance phase. Each of these two stance periods lasts about 10-12% of the gait cycle. The swing phase occupies about 40 % of the gait cycle and is subdivided into initial swing (*toe-off*, *TO*), mid-swing and terminal swing, ending with the heel striking the ground.

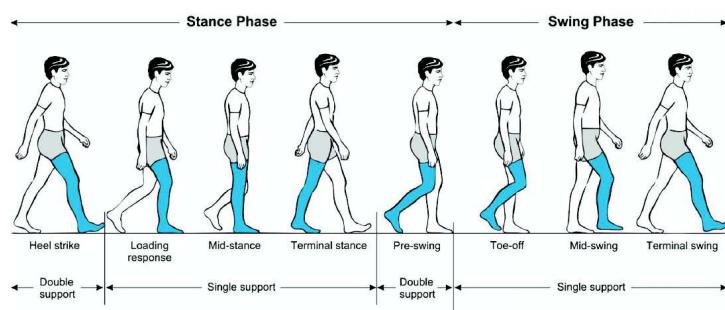


FIGURE 4.1: The phases of a normal walking cycle

If we consider only four events identified by HS, FF, HO and TO we can consider that the *position* phase is represented by the period between HS and TO and the *swing* phase is represented by the period between TO and HS (figure 4.2) [33].

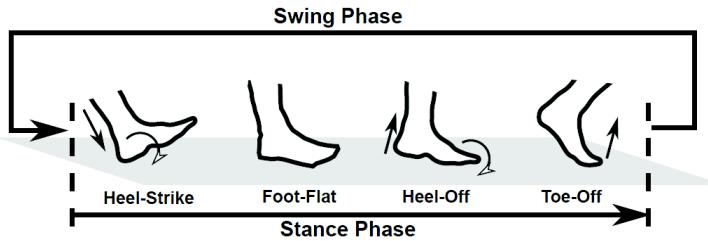


FIGURE 4.2: Event and walk phase detection

A step is represented by the advance of a foot (figure 4.3). One cycle of movement is equivalent to one step taken by the left foot and one step taken by the right foot (*cycle length*).

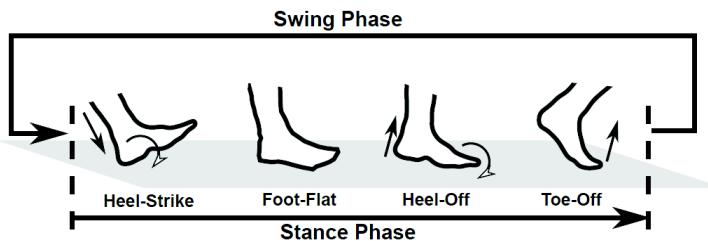


FIGURE 4.3: Basic terminology describing a walking cycle

Differences in leg movement parameters highlight asymmetries, and positioning the sensors on both legs may be a solution for separate evaluation of the parameters. Most of the time, even if the distance travelled by each leg separately is different, the distance travelled in a movement cycle is the same. If only one sensor is considered then it will be assumed that symmetry exists.

Abnormal steps are often present in gait disorders, some examples are highlighted in figure 4.4. Among the types of abnormal gait we find, adynamic gait, swaying gait, stiff gait, shuffling gait.

Gait detection and gait analysis, using wearable sensors, will be able to provide the necessary parameters for a physiotherapist/pediatrician to identify a corrective solution..

In general, inertial sensors such as accelerometer, gyroscope, magnetometer, combined in an inertial measurement unit (IMU) are used. Such examples are MPU-6050 or MPU-9250. In the developed application BMX055 was used..

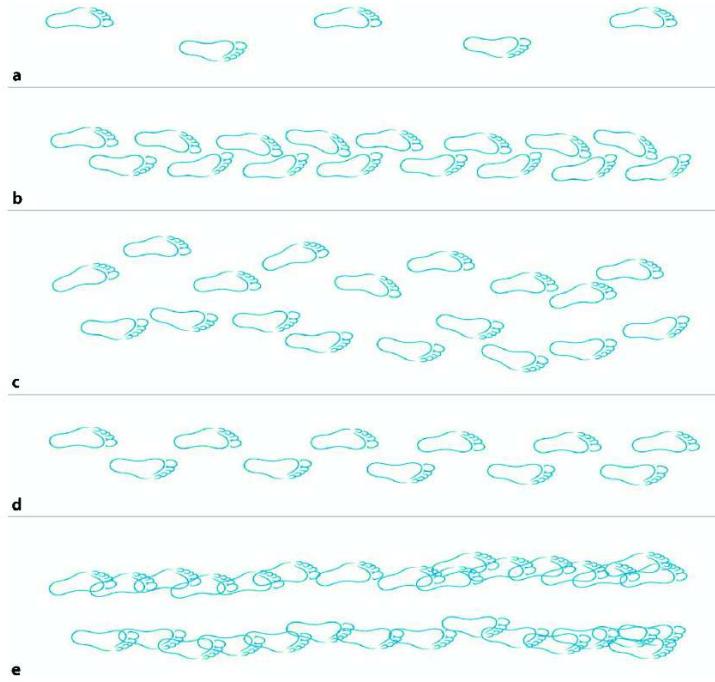


FIGURE 4.4: Step sequences in classical walking disorders.
 a) normal; b) adynamic; c)swinging; d) parkinsonian (inflexible); e) frontal (crawling).

Identifying the events of a motion cycle involves algorithms such as Tjhai's [33] or Trojaniello's [34]. In the case of the first algorithm, the axes of the gyroscope must be aligned with the axes of the frontal, sagittal and transverse planes of the human body, and knowing that the swing phase starts with TO and ends with HS (figure 4.5), three consecutive signal evolution values are taken for the sagittal angular velocity measurement, starting with the first TO and ending with the third HS. If the signal is weak, the next run cycle including 3 periods is performed.

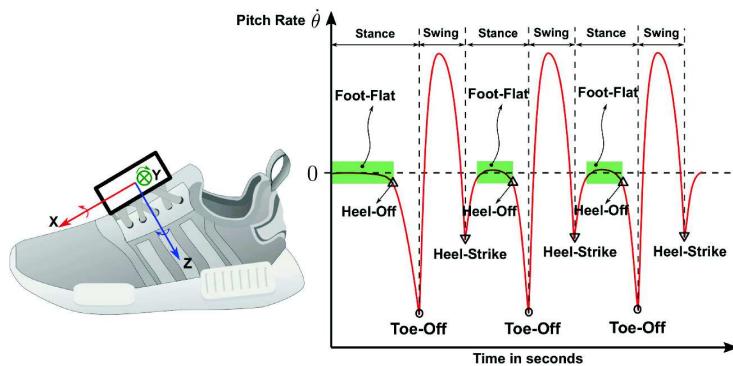


FIGURE 4.5: Sagittal signal receive.

Trojaniello's proposed method [34] proves better suited to detecting *pathological* gait events (figure 4.6). The angular velocity, ω , in the sagittal plane (Oy) is higher in the intermediate zone of the swing phase. Identify the maximum of these values and define a confidence interval encompassing this value, this being defined as the interval where the angular velocity is greater than a threshold, e.g. 20 % of the maximum value. If ω exceeds this threshold several times, as may happen in some pathological gait, (T_{SW}) is defined as the interval between the first and last pass. Two additional constraints are set: (T_{SW}) must be at least 100 ms, and two consecutive (T_{SW}) of the same leg must occur at least 200 ms apart. This interval can be trusted to belong to the swing phase of a gait cycle.

When no gait events (GE) occur during (T_{SW}), the leg will swing through the air. Since the two lower limbs alternate their swing phase while the opposite leg is in contact with the ground, the (T_{SW})— of one lower limb can be used as the reliable (T_{ST}) position interval of the other limb. Thus, when combined, the two (T_{SW}) allow the identification of both T_{ST} and (T_{SW}) for each limb, narrowing the search interval for finding the initial contact HS in the area (T_{IC}) and the final contact TO in the area T_{FC} .

The heel strike is identified as the moment of minimum ω in the area (T_{IC}) before the moment of maximum acceleration. TO is the moment of minimum acceleration identified in area (T_{FC}), as it is expected to occur at the moment of sudden motion before the moment of maximum acceleration in area (T_{FC}).

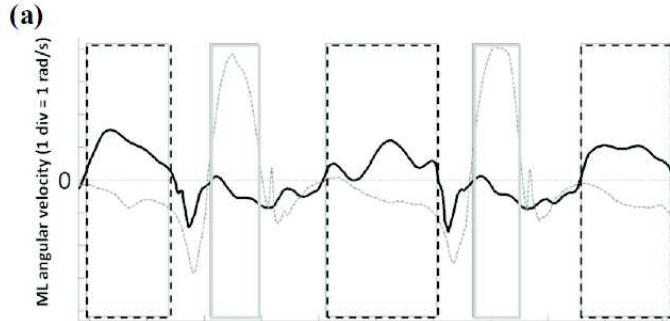


FIGURE 4.6: Angular speed and acceleration for a subject with reduced mobility.

In the figure 4.6 the rectangles made with dotted line (a) represent (T_{SW}) and those with solid line represent (T_{ST}). In area (b) the hatched rectangles represent (T_{IC}) (light grey) and (T_{FC}) (dark grey) and the vertical lines refer to GE periods.

4.2 Hardware and software architecture design

4.2.1 Hardware architecture of the experimental model

The proposed solution (figure 4.7) allows data acquisition from sensors (accelerometer, gyroscope, magnetometer) to be transmitted via Bluetooth when paired with the Bluetooth device of the mobile phone. The application on the mobile phone allows results to be displayed without performing a signal analysis for gait detection or analysis. The application can be completed with two software modules, the first one performing a quick gait detection based on the analysis of the current data and the second one transmitting this data to a server running a "machine learning" mechanism based on some data sets, finally the result is sent to the application on the mobile phone or to an external application managed by a physiotherapist/foot specialist.

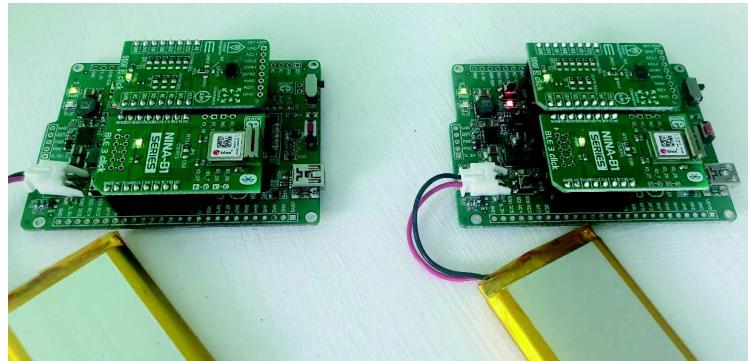


FIGURE 4.7: Data acquisition modules

At this stage, we are talking about a concept, developed by implementing the solution using development boards (figure 4.7). Three hardware modules were used: clicker 2 for PIC18FJ based on PIC18F87J50 microcontroller working at 48 kHz frequency, BLE 3 click based on NINA-B1 module and 9DOF 3 click based on BMX055 sensor. The application in the microcontroller allows data to be transmitted in the classical way, i.e. by *simple substitution of LSB*. Initially, a Serial Bluetooth Monitor application was used to receive the data, and finally an application was developed for an Android system to allow the numerical display of the data in the system in graphical form.

4.2.2 2-bit RDH insertion

The limiting factor in this application is the limited memory/processing power of the microcontroller. The RDH insertion algorithm must be of low complexity. Values transmitted via Bluetooth are floating point, but RDH algorithms use integer host values. To remedy this

problem, the algorithm will only make changes to the integer part of the signal. The introduced distortion is controlled with the u tagging parameter. It controls the left shift of the decimal of the host value:

$$\bar{x}_i = \lfloor 10^u x_i \rfloor \quad (4.1)$$

The fractional part being stored temporarily: $f_i = x_i - \frac{\bar{x}_i}{10^u}$.

The previous sample is used to estimate the current sample (the first set of measurements is transmitted in full, the RDH insertion starts with set 2 of sensor readings). The prediction error e_i for sample x_i is calculated as:

$$e_i = \bar{x}_i - \bar{x}_{i-1} \quad (4.2)$$

In scenario 1, a 16-bit value is inserted into 8 samples of the same size, requiring 2 bits to be hidden in each host sample. In scenario 2 we consider that the slow signal (magnetization) has a frequency 4 times lower than the fast signals (acceleration and angular velocity). Thus, in both scenarios it is necessary to insert 2 bits of information in each host sample. A sample is flagged if its prediction error meets $-T \leq e_i < T$, where T is the flagging threshold (PEE multi-bit insertion from section 3.2 when $n = 4$):

$$\bar{x}'_i = \bar{x}_i + 3e_i + 2b_k + b_{k+i} \quad (4.3)$$

unde b_k and b_{k+i} represents the 2 bits inserted into x'_i .

The insertion is considered valid only when the full 16-bit set has been inserted into 8 host samples. The current host values are transmitted as:

$$x'_i = \frac{\bar{x}'_i}{10^u} + f_i \quad (4.4)$$

Otherwise, the marking has failed and the samples and the current value for magnetization are transmitted directly without RDH insertion.

To reduce the complexity of the algorithm, the displacement case has been removed. Samples that do not satisfy the $-T \leq e_i < T$ condition are considered to be part of the failed marking case.

4.3 Results and conclusions

For the 3 axes 3 floating point values are read, each value being represented on 16 bits. The system will acquire 10 reads/s (every 100ms) leading to a transmitted data volume of 9×16

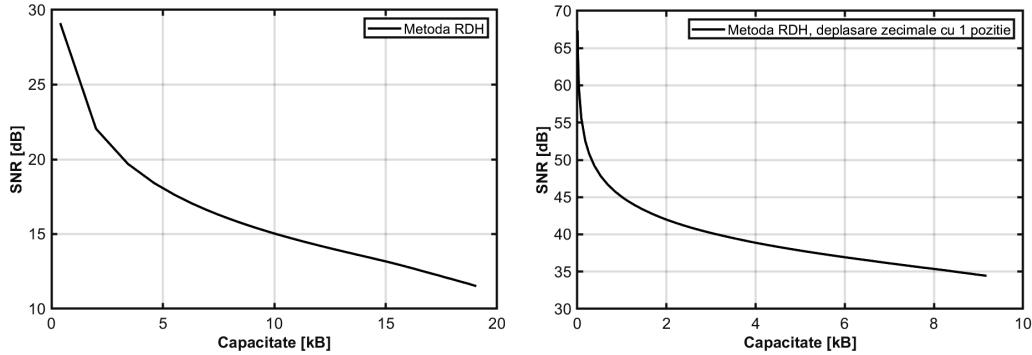


FIGURE 4.8: SNR/Capacity results, scenario 1 (signals with the same sampling frequency)

bits \times 10 samples/s \times 60 s/min = 86400 bits/min \simeq 10.8 kB/min for classical transmission in scenario 1.

For the validation of the results, the case of LSB substitution labelling is also considered. This type of insertion introduces a permanent distortion in the host signal. In this case, the last two LSB layers of the host samples are replaced by hidden bits. The LSB substitution reduces the data volume by 1.2 kB/min, but distorts the host signal.

The following considers the transmission of a set of 11 recordings of 2 minutes each with 10 reads/s (in total 9×13200 samples, which will be separated into host samples and hidden data). Figure 4.8 shows the results of the proposed RDH method. It can be seen that the SNR obtained depends on u (the decimal shift control variable). The best SNR is obtained when used to mark as many decimal places as possible, but there we also have lower insertion capacity (the range of markable errors becomes smaller). In classical transmission the volume of data transmitted is 237.6 kB, this can be reduced by up to 16 kB (at an SNR of 12 dB, requiring data extraction to view the host signal) or by 6 kB (57dB, host signal distortion is imperceptible and can be completely eliminated by data extraction). LSB substitution reduces the volume by 26.4 kB, but the introduced distortion (with an SNR of 120 dB) is permanent.

The simple RDH insertion method successfully reduced the amount of data transmitted to a gait analysis application. Some of the readings from the sensors were inserted via RDH into the other measurements to reduce the number of bits stored/transmitted. This data can be extracted later, completely eliminating the distortion introduced into the host samples.

Conclusions

In this thesis the reversible insertion of data into audio files was discussed. In Chapter 1 a brief history of the research area was presented. The most relevant methods in the field were presented and discussed in Chapter 2.

Then new methods of reversible insertion into audio files were proposed in Chapter 3. Initially a method of capacity control was adapted from the digital image domain to the audio domain. It was shown that this approach leads to a better SNR/capacity ratio for two existing RDH methods in the field. The PVO (*Pixel Value Ordering*) method, extensively used in image insertion, was adapted for the audio domain under the name SVO (*Sample Value Ordering*), bringing the most efficient marking for capacities below 0.3 bps.

Also in Chapter 3 a new RDH method adapted from PPVO was introduced. For digital images PPVO is low capacity, being in direct competition with PVO. But the proposed approach uses its improved prediction together with multi-bit insertion to achieve the best medium capacity results. The efficiency of the proposed methods is experimentally demonstrated, with tests performed on a set of audio files intensively used in the field of audio RDH. The proposed RDH methods bring a gain in performance compared to existing approaches in the field: SVO provides a 7 dB gain compared to the results from [26] at capacities below 0.3 bps, the method derived from PPVO provides on average 5 dB over [26] for the range 0.3 - 0.7 bps (discussed in section 3.2), and the proposed capacity control provides a 0.5 dB gain over existing reversible marking methods in the audio domain (section 3.1.1).

In chapter 4 a simple RDH method for data cloaking for a gait analysis application was proposed. Data acquisition from sensors is done using a microcontroller, which transmits the data via *Bluetooth* to an Android application on a mobile phone. Reversible insertion is used to reduce the data transmitted via *Bluetooth*. The limiting factor in such an application is the limited memory and computing power of a microcontroller. A simple RDH insertion method has been implemented that uses previous measurements for prediction, inserting the values for magnetization into those for acceleration and angular velocity. For comparison a simple (non-reversible) LSB substitution approach was implemented into the same host values.

Original contributions

Three new reversible insertion methods have been introduced in the audio domain in 3. The first method allows fine control of the capacitance by adjusting the insertion threshold. The approach is of low complexity, ideal for audio files. It does not require sample sorting and can be used regardless of the number of marking steps. Thus, the method is highly flexible and can be easily incorporated into existing audio RDH methods.

The SVO method adapts the advantages offered by PVO in the image domain to the audio domain. In contrast to the classical approach (working only on blocks of pixels), SVO combines block insertion (in this case blocks of audio samples) with single-sample insertion via PEE insertion. Thus, low-capacity block insertion is used only in complex regions, and in uniform regions the full available samples are used.

The PPVO-derived approach has made a number of significant contributions to the field. The classical PPVO method uses two dummy predictors (minimum and maximum per prediction context). In contrast, the proposed method dynamically chooses between two weighted predictors. The PPVO prediction pair is with the chosen predictor and the one averaged over the context surrounding the host sample. The prediction is of low complexity (addition and division by multiples of 2) and the scoring is done in one step. The PPVO insertion equations have been adapted to allow multi-bit tagging.

The audio RDH algorithms developed in this thesis provide the best results for small (SVO), medium (PPVO-derived method) and improve on existing large capacity methods (via the capacity control method by adapting the insertion threshold).

In chapter 4 we propose the use of RDH algorithms in data acquisition systems as a solution to reduce data traffic by inserting a slow signal into a fast signal. Reversible data insertion has been implemented on a microcontroller system for gait analysis.

Future directions for research

The research results in this thesis can be developed and extended through the following approaches:

Refinement of the capacity control algorithm

The capacity control proposed in this paper uses two consecutive marking thresholds. PEE methods in the image domain use sorting based on local complexity. The optimal solution should be a combination of the two methods. The local complexity is used to remove samples from noisy regions, and the two thresholds are used to refine the capacity for the remaining values.

Optimization of insertion using 2D histogram

The 2D histogram has been successfully used for insertion into images (in pixel pairs and more recently combined with the PVO algorithm). The large number of audio samples encountered in an audio file does not allow direct implementation of the method. However, new low complexity 2D approaches can be generated that bring the advantages of the method from the image domain to the audio domain..

Adaptation of the PPVO-derived method for the encrypted domain

Data storage using *cloud* systems have led to the emergence of new specialised RDH methods for the encrypted domain. The proposed prediction derived from PPVO can be adapted for such a method, allowing reversible insertion of data into encrypted audio files.

List of publications

1. **Bobeica A.**, Dragoi, I. C., Caciula, I. Coltuc, D., Albu, F. și Yang, F., "Capacity control for prediction error expansion based audio reversible data hiding." In 2018 22nd International Conference on System Theory, Control and Computing (ICSTCC), pagini 810-815, IEEE, an 2018 (**17 citations**).
2. **Bobeica A.**, Dragoi, I. C., Caciula, I. și Coltuc, D., "Sample Value Ordering for Audio Reversible Data Hiding." In 2019 6th International Symposium on Electrical and Electronics Engineering (ISEEE), pagini. 1-6, IEEE, year 2019 (**2 citations**).
3. **Bobeica A.**, Dragoi, I. C., Coandă H.-G.. și Coltuc, D., "Adapting the PPVO framework for Audio Reversible Data Hiding." In 15th Edition International conference on Electronics, Computers and Artificial Intelligence (ECAI), year 2023.

Selective bibliography

- [1] Osborne C.F., van Schyndel R.G., Tirkel A.Z. "A digital watermark". In *IEEE Proceedings of International Conference on Image Processing, ICIP-94*, pag. 86–90, year 1994.
- [2] Honsinger C. W., Jones P.W., Rabbani M., Stoffel J.C. "Lossless recovery of an original image containing embedded data". In *US Patent number 6278791*, year 2001.
- [3] Fridrich J., Goljan J. "Lossless data embedding for all image formats". In *In SPIE Proceedings of Photonics West, Electronic Imaging, Security and Watermarking of Multimedia Contents*, page pag. 572–583, year 2002.
- [4] Goljan M., Fridrich J., Du R. "Distortion-free data embedding for images". In *Information Hiding*, page pag. 27–41, year 2001.
- [5] Xuan G., Shi Y.Q., Yang C., Zheng Y., Zou D., Chai P. "Lossless data hiding using integer wavelet transform and threshold embedding technique". In *IEEE International Conference on Multimedia and Expo*, year 2005.
- [6] Celik M. U., Sharma G., Tekalp A. M., Saber E. "Lossless generalized-lsb data embedding". In *IEEE Trans. Image Process*, pag. 253 – 266, year 2005.
- [7] Tian J. "Reversible data embedding using a difference expansion". In *IEEE Trans. on Circuits and Syst. for Video Technol.*, pag. 890 – 896, year 2003.
- [8] Alattar A. M. "Reversible watermark using the difference expansion of a generalized integer transform". In *IEEE Trans. Image Process*, page pag. 1147–1156, year 2004.
- [9] Thodi D. M., Rodriguez J. J. "Prediction-error based reversible watermarking". In *Proc. IEEE Int. Conf. Inf. Process*, page pag. 1549–1552, year 2004.
- [10] Fallahpour M. "Reversible image data hiding based on gradient adjusted prediction". In *IEICE Electron. Exp*, page pag. 870–876, year 2008.

- [11] Xiang S., Wang Y. "Non-integer expansion embedding techniques for reversible image watermarking". In *EURASIP J. Adv. Signal Process*, page pagina 56, year 2015.
- [12] Hui Shi, Baoyue Hu, Yanli Li, Jianing Geng, Yonggong Ren. "Adaptive multi-predictor based reversible data hiding with superpixel irregular block sorting and optimization". *The Imaging Science Journal*, 1–22, 2023.
- [13] Xiaoya Zhang, Yuanzhi Yao, Nenghai Yu. "Convolutional neural network-driven optimal prediction for image reversible data hiding". In *2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSP)*, 1–6. IEEE, 2021.
- [14] Runwen Hu, Shijun Xiang. "Cnn prediction based reversible data hiding". *IEEE Signal Processing Letters*, 28:464–468, 2021.
- [15] Runwen Hu, Shijun Xiang. "Reversible data hiding by using cnn prediction and adaptive embedding". *IEEE Transactions on Pattern analysis and Machine Intelligence*, 44(12):10196–10208, 2021.
- [16] Xie Yang, Fangjun Huang. "New cnn-based predictor for reversible data hiding". *IEEE Signal Processing Letters*, 29:2627–2631, 2022.
- [17] Junying Yuan, Huicheng Zheng, Jiangqun Ni. "Reversible data hiding with neighboring-prediction-errors aided sorting and cnn prediction". *Journal of Imaging Science and Technology*, 67(4):1–13, 2023.
- [18] Praveen Kumar, Prateek Ravi, Rajeev Kumar. "Reversible data hiding for interpolated images using cnn based nearest neighbour interpolation". In *2023 Third International Conference on Secure Cyber Computing and Communication (ICSCCC)*, 475–480. IEEE, 2023.
- [19] Li X., Li J., Li B., Yang B. "High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion". In *Signal Process*, pag. 198 – 205, year 2013.
- [20] Peng F., Li X., Yang B. "Improved pvo-based reversible data hiding". In *Digit. Signal Process.*, pag. 255 – 265, year 2014.
- [21] Bo Ou, Xiaolong Li, Yao Zhao, Rongrong Ni, Yun-Qing Shi. "Pairwise prediction-error expansion for efficient reversible data hiding". *IEEE Transactions on image processing*, 22(12):5010–5021, 2013.

- [22] Huo Y., Xiang S., Liu S., Luo X., Bai Z. "Reversible audio watermarking algorithm using non-causal prediction". In *Wuhan University Journal of Natural Sciences*, Vol. 18, No. 5, pag. 455–460, year 2013.
- [23] Sachnev V., Kim H. J., Nam J., Suresh S., Shi Y. Q. "Reversible watermarking algorithm using sorting and prediction". In *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 19, No. 7, pag. 989–999, year 2009.
- [24] Wang F., Xie Z., Chen Z. "High capacity reversible watermarking for audio by histogram shifting and predicted error expansion". In *The Scientific World Journal*, year 2014.
- [25] Xiang S., Li Z. "Reversible audio data hiding algorithm using noncausal prediction of alterable orders". In *EURASIP Journal on Audio, Speech, and Music Processing*, year 2017.
- [26] Ma B. et al. "A reversible data hiding algorithm for audio files based on code division multiplexing". In *Multimedia Tools and Applications*, Vol. 80, pag. 17569–17581, year 2021.
- [27] Geiger R., Yokotani Y., Schuller G. "Audio data hiding with high data rates based on intmdct". In *In Proc. IEEE Int. Conf. Acoust. and Speech, Signal Proc.*, 2006.
- [28] Huang X., Ono N., Echizen I., Nishimura A. "Reversible audio information hiding based on integer dct coefficients with adaptive hiding". In *Lecture Notes in Computer Science*, Vol 8389, year 2013.
- [29] Caciula I., Coltuc D. "Capacity control of reversible watermarking by two-thresholds embedding: Further results". In *International Symposium on Signals, Circuits and Systems (ISSCS2013)*, year 2014.
- [30] Caciula I., Coltuc D. "Capacity control of reversible watermarking by two-thresholds embedding". In *IEEE International Workshop on Information Forensics and Security (WIFS2012)*, year 2013.
- [31] Chin-Feng Lee, Jau-Ji Shen, Yi-Jhen Wu, Somya Agrawal. "Pvo-based reversible data hiding exploiting two-layer embedding for enhancing image fidelity". *Symmetry*, 12(7):1164, 2020.
- [32] Lin C. C., Chen S. C., Hwang K. F. "An information hiding scheme based on pixel value-ordering and prediction-error expansion with reversibility". In *International Journal of Advanced Computer Science and Applications (IJACSA)*, pag. 1–7, year 2014.

- [33] Chandra Tjhai, Kyle O'Keefe. "Using step size and lower limb segment orientation from multiple low-cost wearable inertial/magnetic sensors for pedestrian navigation". *Sensors*, 19(14):3140, 2019.
- [34] Diana Trojaniello, andrea Cereatti, Elisa Pelosin, Laura Avanzino, anat Mirelman, Jeffrey M Hausdorff, Ugo Della Croce. "Estimation of step-by-step spatio-temporal parameters of normal and impaired gait using shank-mounted magneto-inertial sensors: application to elderly, hemiparetic, parkinsonian and choreic gait". *Journal of neuroengineering and rehabilitation*, 11:1–12, 2014.
- [35] Solescu I. "Dificultăți de mers, tipuri de tulburări ale mersului". <https://www.catenapascupas.ro/informatii-medicale/dificultati-de-mers-tipuri-de-tulburari-ale-mersului>, 2022.
- [36] Mikroelektronika. "Clicker2 for pic 18fj". <https://www.mikroe.com/clicker-2-pic18fj>.
- [37] Mikroelektronika. "BLE 3 click". <https://www.mikroe.com/ble-3-click>.
- [38] Ublox. "NINA-B1 series stand-alone bluetooth low energy modules". <https://content.u-blox.com/sites/default/files/NINA-B1-DataSheet-UBX-15019243.pdf>.
- [39] Mikroelektronika. "9DOF 3 click". <https://www.mikroe.com/9dof-3-click>.
- [40] Bosch Sensortec. "BMX055 small, versatile 9-axis sensor module". <https://download.mikroe.com/documents/datasheets/bst-bmx055-ds000-datasheet.pdf>.
- [41] Xinpeng Zhang. "Reversible data hiding in encrypted image". *IEEE signal processing letters*, 18(4):255–258, 2011.
- [42] Y. Hu, H.K. Lee, J. Li. "DE-based reversible data hiding with improved overflow location map". In *IEEE Trans. Circuits Syst. Video Technol.*, 19(2), pp. 250–260, year 2009.
- [43] T. Zen g X. Li, B. Yang. "Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection". In *IEEE Trans. Image Process.* 20(12), pp. 3524–3533, year 2011.
- [44] L. Luo, Z. Chen, M. Chen, X. Zeng, Z. Xiong. "Reveresible image watermarking using interpolation technique". In *IEEE Trans. Inf. Forensics Secur* 5 (1), pp. 187–193, year 2010.

- [45] B. Ou, X. Li, J. Wang. "High-fidelity reversible data hiding based on pixel-value-ordering and pairwise prediction-error expansion". In *Journal of Visual Communication and Image Representation*, 39, pp. 12–23, year 2016.