

ALGORITMI DE RECONSTITUIRE 2D PENTRU CETĂȚI MEDIEVALE SAU ANTICE; STUDIU DE CAZ - CETĂȚI MEDIEVALE DIN TRANSILVANIA

Dragoș Nicolau, Dragoș Barbu, Laura Ciocoiu, Dragoș Smada

Antonio Cohal, Ionuț Petre, Valentin Răduț

Institutul Național de Cercetare – Dezvoltare în Informatică, ICI - București

Rezumat: Lucrarea de față prezintă algoritmi de analiză, detectare muchii și segmentare a imaginilor cetăților medievale sau antice, imagini existente în fotografii vechi, cu scopul reconstituirii virtuale 2D a imaginii cetății.

Cuvinte Cheie: analiză imagine, detectare muchii, segmentare imagine, fotografii vechi, reconstituire virtuală 2D.

Abstract: This paper presents the algorithms for image analysis, edge detection and image segmentation from old pictures in order to make the virtual 2D reconstruction.

Keywords: image analysis, edge detection, image segmentation, old pictures, 2D virtual reconstruction.

1. Introducere

Societatea modernă este o societate informațională caracterizată de fenomenul de diseminare a informației din diferite domenii sub formă electronică, diseminare facilitată de posibilitățile de comunicare prin intermediul rețelelor de calculatoare. Acestea au condus la o adevărată revoluție a conceptelor de structurare și regăsire a informației. Principalele modalități de diseminare a informației sunt: rețeaua Internet prin intermediul modelului de tip hypertext și DVD / CD-I (Compact Disc Interactiv) utilizând tehnicile multimedia, care se bazează pe combinarea diferitelor moduri de reprezentare a informației de tip imagine, sunet, film. Astfel, percepția informației este îmbunătățită prin asociativitate, prin structurarea acesteia și prin exploatarea ei în mod nelinear, dirijat interactiv de utilizator, potrivit cerințelor de informare, facilitând astfel regăsirea și înțelegerea conceptelor. În urma realizării de site-uri și CD/DVD-uri pentru accesul la patrimoniul cultural, cum ar fi muzee și arhive virtuale, s-a observat interesul manifestat pentru cunoașterea și reconstituirea trecutului. Informațiile despre evoluția așezărilor urbane, multe dintre acestea începând cu cetăți medievale, prezintă interes în educarea tinerei generații.

Astfel, în cadrul acestei lucrări se vor prezenta algoritmi pentru reconstituirea 2D a cetăților medievale. S-a elaborat un studiu de caz pentru o cetate medievală din Transilvania.

Pentru realizarea reconstituirii 2D din imagine preluată din fotografii vechi, se vor parcurge următoarele etape:

- analiza imaginii din fotografie;
- procesarea geometrică a imaginii (fotografiei), constând în descompunerea asistată a imaginii în suprafețe: detectarea muchiilor, segmentarea imaginii cetății medievale sau antice;
- generarea mesh-urilor pentru simularea spațiului 2D;
- reconstituirea virtuală 2D a imaginii cetății prin vizualizarea muchiilor nevizibile.

Produsele software necesare realizării reconstituirii 2D sunt destinate:

- digitizării și prelucrării imaginilor (Adobe Photoshop);
- generării structurii ierarhice a obiectelor (suprafețe, elemente constructive/decorative, etc.) ce definesc clădirea;
- gestionării bazelor de date (SQL ca server de baze de date);

- gestionării evenimentelor necesare parcurgerii ierarhiei de obiecte.

Utilizând baza de date cu informații statice (imagini, hărți, explicații contextuale) referitoare la cetățile medievale, a fost realizat site-ul de prezentare în Internet a cetăților medievale. Pentru refacerea muchiilor „slabe” (pereți distruși) a fost utilizată baza de date arborescentă ce conține clasele rezultate din descompunerea ierarhică a unei cetăți. Fiecare element este definit prin atributele sale (tip, coordonate, legături între obiecte / elemente).

În străinătate s-au dezvoltat proiecte ce folosesc reprezentări avansate și metode de vizualizare în dezvoltarea de software care convertește datele în modele 2D. Noua tehnologie dezvoltă aplicații avansate în astfel de domenii cum sunt grafica computerizată, animație și efecte speciale. Aceste cercetări au fost expuse la conferința SIGGRAPH 2002 despre grafică computerizată și tehnologie interactivă.

2. Algoritmi de reconstituire 2D

2.1 Procesarea imaginilor

Procesarea imaginilor este un domeniu al inteligenței artificiale, se ocupă cu modul de reprezentare, reconstituire, clasificare, recunoaștere și analiză a imaginilor cu ajutorul calculatorului. Sursa de proveniență a imaginilor poate fi un dispozitiv de achiziție (camera video, scanner, captor radar) dar poate fi la fel de bine și un ansamblu de date statistice.

Există două motive principale pentru care se apelează la procesarea imaginilor :

1. îmbunătățirea calității unei imagini având drept scop o mai bună vizualizare pentru un operator uman. Aceasta poate însemna: reducerea zgomotului și a altor defecte ce pot fi prezente în imagine (datorate, de exemplu, dispozitivului de achiziție), evidențierea unor zone de interes prin modificarea luminozității, a contrastului, accentuarea muchiilor, etc.

2. extragerea de informații dintr-o imagine, informații ce reprezintă datele de intrare pentru sistemul de recunoaștere a formelor și clasificarea acestora. Aceste informații pot fi: diferite distanțe și relații dintre obiectele prezente în imagine, momente statistice, parametrici geometrici (arie, perimetru, circularitate), coeficienți Fourier, etc.

Exemple clasice de aplicații pentru procesarea imaginilor includ: recunoașterea formelor caracterelor, recunoașterea amprentelor, prelucrarea imaginilor medicale, a imaginilor satelit.

Reprezentare a unei imagini este un tablou bi-dimensional de numere întregi (pixeli). Valoarea fiecărui pixel descrie nivelul de strălucire sau culoarea acestuia. În cel mai simplu caz, al imaginilor binare, pentru reprezentarea fiecărui pixel este folosit un singur bit. În cazul imaginilor cu niveluri de gri, valoarea fiecărui pixel reprezintă strălucirea acestuia. Cel mai comun format pentru aceste imagini are la bază reprezentarea pixelilor pe 8 biți. Astfel, gama de valori posibile este 0...255, 0 codificând culoarea negru, 255 alb iar valorile intermediare reprezentând nuanțe (niveluri) de gri. În cazul imaginilor color, pot fi folosite diferite sisteme de culori (RGB, HSI, CMY, etc.).

2.2 Modele de culori

Modelul RGB este un model aditiv, având culorile fundamentale roșu, verde și albastru, celelalte culori constituindu-se din anumite procente ale culorilor de bază. Culoarea alb are prezente toate culorile fundamentale, iar culoarea negru se caracterizează prin absența lor. Modelul RGB este adesea reprezentat sub forma unui cub unitate.

Originea (0,0,0) este considerată culoarea negru, iar colțul opus (1,1,1) culoarea alb. Diagonala ce unește aceste două puncte conține nivelurile de gri obținute prin utilizarea aceluiași procent din culorile de bază.

O altă metodă de clasificare a culorilor este prin nuanțare (**Hue**), saturație (**Saturation**) și intensitate (**Intensity**), rezultând **modelul HSI**. Nuanța reprezintă poziția relativă a culorii în

spectru și, în modelul HSI, corespunde unghiului pe care îl face culoarea în cercul de culori. Domeniul de nuanțare se întinde de la 0 la 360 de grade. Saturația specifică puritatea culorii. Valoarea saturației se exprimă în procente și variază de la 0 (fără culoare) până la 100 (culoare pură, așa cum este definită de valoarea nuanțării). Ultimul parametru specifică intensitatea culorii. Nuanța determină culoarea (unghi = 60° - Yellow, 180° - Cyan, etc.) Saturația este măsurată în procente de la centrul cercului de bază la suprafața conurilor. Intensitatea se măsoară pe linia Black-White.

Modelul CMY are la bază culorile Cyan, Magenta și Yellow, considerate culori secundare față de culorile Red, Green, Blue. Modelul este substractiv, culoarea alb fiind dată de absența culorilor CMY, iar culoarea negru fiind caracterizată de prezența acestora.

Vecinătăți. Conectivitate

Valoarea pixelului aflat la coordonatele (i,j) din matricea ce reprezintă imaginea (unde i reprezintă linia, iar j coloana, este notată cu $f(i,j)$; matricea este de dimensiune $m*n$. Deasemenea, dacă nu se precizează altfel, discuțiile vor avea în vedere o imagine cu niveluri de gri (considerăm reprezentarea pe 8 biți, deci cu 256 de valori posibile). Aceasta nu reprezintă o limitare întrucât o imagine color poate fi descompusă după culorile fundamentale (R,G,B) în trei imagini separate, fiecare astfel de imagine putând fi tratată ca alcătuită din niveluri de gri.

Un pixel p de coordonate (i,j) prezintă patru vecini în punctele: $(i-1,j)$, $(i+1,j)$, $(i,j-1)$, $(i,j+1)$. Vom nota acest ansamblu (vecinătate) de pixeli cu $N4(p)$. Pixelii diagonali de coordonate: $(i-1,j-1)$, $(i-1,j+1)$, $(i+1,j+1)$, $(i+1,j-1)$ alcătuiesc un nou ansamblu notat $ND(p)$. În final, ansamblurile $N4(p)$ și $ND(p)$ reunite, formează ansamblul notat $N8(p)$.

Având în vedere ansamblurile (vecinătățile) definite mai sus, se pot considera două tipuri de conectivitate:

1. **4-conectivitate** - doi pixeli, p_1 și p_2 , de coordonate (i_1,j_1) și (i_2,j_2) sunt în relație de 4-conectivitate dacă (i_1,j_1) se află în ansamblul $N4(p_2)$ sau echivalent, dacă (i_2,j_2) se află în ansamblul $N4(p_1)$;

2. **8-conectivitate** - doi pixeli, p_1 și p_2 , de coordonate (i_1,j_1) și (i_2,j_2) sunt în relație de 8-conectivitate dacă (i_1,j_1) se află în ansamblul $N8(p_2)$ sau echivalent, dacă (i_2,j_2) se află în ansamblul $N8(p_1)$.

Drumul este lungime n mulțimea de pixeli $D_n = \{p_1, p_2, \dots, p_n\}$ pentru care fiecare pixel p_i se află în relație de conectivitate cu pixelul următor p_{i+1} , $i = 1, \dots, n-1$.

Doi pixeli p și q sunt conectați dacă există un drum D_n de la p la q . Un set de pixeli formează o regiune dacă orice doi pixeli aparținând setului sunt conectați.

Distanțele dintre pixeli sunt numite **metrici**. Fie P mulțimea pixelilor unei imagini. O metrică, în cazul unei imagini, este o funcție care îndeplinește următoarele condiții:

Exemple de metrici des utilizate:

1. distanța euclidiană, unde p_1 și p_2 sunt pixeli de coordonate (i_1,j_1) respectiv (i_2,j_2) ;
2. distanța Manhattan (city block metric).

2.3 Histograma nivelurilor de gri

Histograma nivelurilor de gri a unei imagini reprezintă o funcție care asociază fiecărui nivel de gri prezent în imagine frecvența sa de apariție. Dacă considerăm o imagine f care prezintă un număr maxim G de niveluri de gri, atunci funcția histogramă poate fi definită astfel :

- a) imagine cu 256 niveluri gri;
- b) histograma nivelurilor de gri.

Histograma nu oferă nicio informație cu privire la locația pixelilor sau la poziția relativă a

acestora însă, dispunând de histograma unei imagini, nivelurile de gri pot fi manipulate pentru a îmbunătăți contrastul, luminozitatea, etc. De exemplu, o deplasare a nivelurilor de gri din histogramă spre stânga/dreapta conduce la o diminuare/creștere a luminozității. Acest aspect este foarte util în procesul de segmentare permițând o separare rapidă a obiectelor de fundal în funcție de luminozitatea acestora.

În cazul imaginilor color folosind modelul RGB se pot realiza histograme pentru fiecare culoare de bază în parte cu păstrarea validității celor discutate mai sus.

2.4 Pre-procesarea imaginilor

Pre-procesarea are drept scop realizarea unei îmbunătățiri a imaginii prin punerea în evidență a anumitor regiuni, schimbarea luminozității, detecția muchiilor, etc., pregătind imaginea în vederea operației de segmentare.

În această etapă, transformările aplicate unei imagini pot fi de două tipuri:

- transformări punctuale (modificarea valorii unui pixel se face independent de vecinătatea acestuia);
- transformări locale (noua valoare a pixelului depinde de valorile pixelilor înconjurători).

Transformări punctuale. Egalizarea histogramei

Din cadrul transformărilor punctuale, vom avea în vedere manipularea contrastului prin transformarea nivelurilor de gri. În acest sens, o transformare punctuală poate fi privită ca o funcție de transfer T definită pe mulțimea nivelurilor de gri originale, care atribuie fiecărui pixel o nouă valoare în funcție de luminozitatea sa.

Transformările nivelurilor de gri au drept scop o mai bună vizualizare a imaginii de către un operator uman. O tehnică din cadrul acestor transformări care dă rezultate bune în practică poartă denumirea de **egalizarea histogramei** și constă în distribuirea nivelurilor de gri existente pe întreaga gamă posibilă de valori. Această tehnică este foarte utilă în cazul unor imagini ale căror niveluri de gri sunt localizate într-un interval restrâns, iar „împrăștierea” lor pe întreaga scală conduce la o vizualizare sensibil mai bună. Funcția de transfer utilizată în acest caz are la bază histograma nivelurilor de gri, fiind reprezentată de curba cumulată a frecvențelor nivelurilor de gri. Prin aplicarea metodei egalizării histogramei originalului, se observă accentuarea contrastului ce permite o mai clară observare a detaliilor, punerea în evidență a variațiilor minore dintre regiuni care trec neobservate în imaginea inițială. Concret, metoda egalizării histogramei constă în înlocuirea fiecărui nivel de gri k din imaginea original cu valoarea frecvențelor cumulate până la acest nivel (valoare scalată bineînțeles în intervalul $0...255$).

Trebuie notat faptul că egalizarea histogramei poate fi realizată doar pentru o anumită regiune de interes și nu pentru toată imaginea (de exemplu o zonă determinată de un dreptunghi).

Transformări locale

Aceste transformări țin cont de o anumită vecinătate a pixelului al cărui nivel de gri va fi modificat. De obicei, vecinătățile cu care se lucrează sunt pătratice, de dimensiuni 3×3 , 5×5 , 7×7 , în general de forma $(2p+1) \times (2p+1)$, centrate pe pixelul supus transformării.

Transformările locale, în funcție de scopul lor, se pot clasifica în două categorii :

- transformări care vizează „netezirea” imaginii (**image smoothing**) sau a unor regiuni din aceasta, având drept scop reducerea zgomotului sau altor defecte ale imaginii;
- transformări care vizează detecția muchiilor (**edge detection**), bazate pe derivatele funcției asociată imaginii.

O altă clasificare a transformărilor locale are în vedere proprietățile de liniaritate ale acestora. Astfel, putem vorbi despre:

- transformări liniare;
- transformări neliniare.

Transformările liniare se caracterizează prin faptul că valoarea rezultată reprezintă o combinație liniară a nivelurilor de gri din vecinătatea utilizată pentru fiecare pixel. Contribuția fiecărui pixel din vecinătate în rezultatul final este ponderată cu o anumită valoare, astfel: (x_0, y_0) sunt coordonatele pixelului asupra căruia se realizează transformarea, f_i reprezintă imaginea original iar f_{ies} imaginea finală.

Reducerea zgomotului (Image Smoothing). Metodele privind reducerea zgomotului prezent în imagine (filtrare) pot fi liniare sau neliniare. **Filtrarea liniară** se bazează pe înlocuirea nivelului de gri al unui pixel cu media (ponderată sau nu) a valorilor pixelilor din vecinătatea sa. O metodă simplă care poate rezolva acest deziderat este utilizarea unor nuclee de convoluție de tipul: se observă că prin aplicarea nucleului w_1 imaginii originale, valoarea fiecărui pixel este înlocuită cu media valorilor pixelilor dintr-o vecinătate 3×3 centrată pe pixelul respectiv. Desigur, pot fi folosite vecinătăți de dimensiuni mai mari: 5×5 , 7×7 . În cazul celorlalte două exemple se atribuie o pondere mai mare pixelului central în calculul mediei, respective pixelilor din vecinătatea N_4 . O altă variantă este utilizarea unor nuclee ce reprezintă aproximări ale suprafeței lui Gauss.

Se observă ponderarea diferită a pixelilor din vecinătate în funcție de poziția lor față de pixelul central. Trebuie remarcat faptul că nucleele prezentate mai sus pot fi aplicate consecutiv imaginii, însă o aplicare excesivă conduce la efectul denumit “blur” (ceata) și va determina pierderea detaliilor.

O operație de filtrare a unei imagini la care s-a adăugat un zgomot reprezentat de un număr de pixeli albi și negri generați aleator. A fost utilizat de două ori nucleul w_1 de dimensiune 3×3 . Ultima imagine prezintă efectul de “blur” prin aplicarea nucleului w_5 de 4 ori succesiv pe jumătatea dreaptă a imaginii originale.

O metodă de **filtrare neliniară** este dată de filtrarea mediană al lui Tuckey care constă în înlocuirea valorii fiecărui pixel cu valoarea mediană a șirului de valori ale pixelilor din vecinătatea utilizată.

2.5 Detectia muchiilor (Edge detection)

Intuitiv, într-o imagine muchiile pot fi identificate acolo unde valorile pixelilor se schimbă brusc.

- a) muchie tip treaptă;
- b) muchie tip vârf;
- c) muchie reală cu zgomot.

Muchiile de tip **treaptă** se situează între regiuni caracterizate de valori medii ale nivelurilor de gri diferite. Muchiile de tip **vârf** corepund unei variații locale a intensității pixelilor, variație ce prezintă un maxim sau un minim. Schimbările de intensitate pot fi detectate aplicând imaginii un operator de tip **gradient**. Gradientul unei funcții într-un punct este un vector nivel gri. Șirul de valori sortat al nivelelor de gri din vecinătatea considerată este: 1, 20, 25, 30, 30, 40, 40, 50, 50 iar valoarea mediană este 30. Aceasta va fi noua valoare a pixelului central.

Privind imaginea ca o funcție de două variabile $f(x,y)$ care asociază fiecărui punct (x,y) o valoare a intensității (nivel de gri) și, făcând presupunerea că funcția este continuă și derivabilă, amplitudinea gradientului măsoară diferența dintre intensitățile pixelilor dintr-o vecinătate, iar orientarea relevă direcția celei mai semnificative schimbări de intensitate, care este și direcția muchiei.

În continuare sunt prezentați diferiți operatori bazați de gradient:

- **operatorul Roberts.** Acest operator presupune calculul următoarelor diferențe finite: amplitudinea și orientarea se calculează în conformitate cu definițiile de mai sus, calculul diferențelor finite se poate realiza ca un produs de convoluție între imagine și nuclee. Un dezavantaj al acestui operator îl reprezintă sensibilitatea la zgomot datorită dimensiunilor mici ale vecinătății folosite.

- **operatorul Prewitt.** Operatorul Prewitt folosește următoarea aproximare a derivatelor în cele două direcții:

Gradientul este estimat în 8 direcții posibile (în cazul unui nucleu 3x3), iar valoarea amplitudinii reprezintă maximul rezultatelor obținute prin aplicarea celor 8 nuclee. Direcția gradientului se calculează pentru situația corespunzătoare maximului amplitudinii.

- **operatorul Sobel.** Se pot folosi toate cele 8 nuclee corespunzătoare celor 8 direcții, într-o manieră similară operatorului Prewitt, însă se pot folosi doar nucleele w_1 și w_3 pentru realizarea diferențelor finite pe linii, respectiv coloane. În acest caz amplitudinea și direcția gradientului se calculează în maniera obișnuită:

Există și alți operatori ce utilizează metode derivative pentru detecția muchiilor pe care doar îi vom aminti și pentru care vom prezenta primul nucleu de convoluție (celelalte pot fi obținute prin rotații succesive): **operatorul Kirsch, operatorul Robinson, operatorul Frei & Chen.** Trebuie ținut cont de faptul că numai o valoare mare a amplitudinii gradientului poate semnifica prezența unei muchii. Astfel, se impune o filtrare a rezultatelor obținute (de exemplu, se păstrează ca potențiale componente ale muchiei acei pixeli ale căror valori depășesc un anumit prag).

Alte metode de detecție a muchiilor se bazează pe aplicarea operatorului Laplace, ținându-se cont de derivata a doua a funcției asociată imaginii: dacă în cazul precedent existența unei muchii era determinată de un maxim al gradientului (bazat pe derivate de ordin I), în acest caz muchiile sunt identificate prin trecerea prin 0 a laplacian-ului imaginii (un maxim al derivatei de ordinul I reprezintă un zero al derivatei de ordinul II).

2.6 Segmentarea imaginii

Scopul segmentării imaginilor este de a „strânge” pixelii în regiuni imagine continue, de exemplu, regiuni care corespund suprafețelor individuale, obiectelor, sau părților obiectelor naturale. O segmentare ar putea fi utilizată pentru recunoașterea obiectelor, estimarea delimitărilor ”granițelor” suprafețelor individuale în cadrul sistemelor de mișcare, compresia imaginilor, editarea de imagini sau baze de date tip căutare.

Noi considerăm segmentarea imaginilor de jos în sus. Ignorăm contribuțiile la recunoașterea obiectelor în procesul de segmentare de sus-jos (topdown). Ca intrare, se consideră în primul rând luminozitatea imaginii, deși tehnici similare pot fi utilizate utilizând culoarea, mișcarea și / sau dispersarea informației.

Gruparea pixelilor în funcție de caracteristici (culoare, intensitate)

Vectorul caracteristic a $F(x)$ este de forma:

$$F(x) = \begin{pmatrix} x \\ I(x) \\ L(x) \end{pmatrix}$$

unde x este pixel, $I(x)$ - imagine, $F(x)$ -vectorul caracteristic,
 $L(x)$ este vectorul de caracteristici locale a imaginii.
Pentru imaginile color, $F(x)$ include, de asemenea, informații despre culoarea pixelului x .

Pentru segmentarea imaginii se realizează o grupare a vectorilor caracteristici $F(x)$ detectați în imagine. O regiune compactă a imaginii cu un nivel distinct de gri sau culoare va corespunde unei regiuni în spațiul caracteristic, cu o densitate relativ mare de vectori cu caracteristica șablon.

Modelul Gaussian

O abordare naturală este modelarea distribuției vectorului de caracteristică observat utilizând un model M Gaussian (MoG), $p(F|M) = \sum_k \pi_k g(F | m_k, \Sigma_k)$.

Aici $\pi_k \geq 0$ sunt coeficienții de mixare, unde $\sum_k^K \pi_k = 1$ și m_k, Σ_k sunt mediile și covarianțele componenteii Gaussiene.

Pentru un K dat, parametrii $\{(\pi_k, m_k, \Sigma_k)\}_{k=1; K}$ a modelului MoG se pot potrivi cu datele $\{F(x)\}_{x \in X}$ folosind maximul de asemănare (aici X reprezintă mulțimea setului de pixelii). Probabilitatea penalizată (lungimea descrierii minime (MDL)) poate fi folosită pentru a selecta numărul de componente, K.

Eticheta segmentului $c(x) = k$ pentru un pixel x este k, care maximizează proprietatea lui $F(x)$ în MoG modelul M, este $c(x) = \arg \max_F [\sum_k \pi_k g(F(x) | m_k, \Sigma_k) / p(F(x) | M)]$.

Imaginea cu maxim de proprietate este procesată folosind componente conectate și regiunile mici (gri) au fost eliminate; este afișată culoarea medie a componentelor mari rămase. Lățimea granițelor segmentului se datorează utilizării caracteristicii spațiale de textură. Modelul MoG poate fi înlocuit cu mediile K sau limitat la utilizarea parametrizărilor dimensionale pentru Σ_k .

Calitatea segmentării rezultate depinde de felul în care imaginea se potrivește cu ipotezele (implicite), cu care am început, și anume:

- diferite segmente formează grupuri compacte, bine separate în vectorul F;
- componentele gaussiene în M corespund regiunilor importante.

Cu toate acestea, această facilitate - spațiu de grupare (clustering) - poate fi utilă pentru extragerea rezumatelor brute a conținutului imaginii potrivite pentru interogarea bazelor de date de imagini [27].

2.6.1 Segmentarea cu deplasare medie

Algoritmul de segmentare cu deplasare medie [29] ia în considerare, de asemenea, densitatea de probabilitate a vectorilor caracteristică $F(x)$ obținută de la o imagine dată. Cu toate acestea, este folosit un model neparametrizat al densității în loc de un MoG. În special, este utilizată o estimare a nucleului densității, cu $P_k(F) = 1/|X| * \sum_k K(F - F(x))$, unde X este un set de toți pixelii din imagine, |X| este numărul de pixeli, și $K(e)$ este un nucleu.

Alegerea comună pentru $K(e)$ au forma: $K(e) = k(e^{-T} \sum^{-1} e)$, unde $k(s)$ este o funcție concavă descrescătoare a abaterii pătrate $s = e^{-T} \sum^{-1} e$

De exemplu $k(s) = ce^{-s/2}$ (nucleul Gaussian); $k(s) = c[1-s]_+$ nucleul Epanechnikov

Aici $c = c(\sum)$ este o constantă normalizată care asigură că integrala $K(e)$ tinde la 1, și $[z]_+$ denotă rectificarea pozitivă, adică $[z]_+ = \max(z, 0)$.

În cazul estimării nucleului densității folosind Epanechnikov, parametrul covarianța \sum al nucleului $K(e)$ determină netezimea estimării densității $p_K(F)$. Se realizează un compromis între artefactele de prelevare de probe (nucleu prea îngust) comparativ cu pierderea de rezoluție în $p_K(F)$ (nucleu prea larg).

Iterațiile la segmentarea cu deplasare medie

Se folosesc nodurile (de exemplu, vârfurile) lui $p_K(F)$ pentru a fi etichete de segmentare, înlocuind utilizarea etichetelor componentelor din modelul precedent MoG. Se rezolvă la nivel local $F_* = \arg \max_F p_K(F)$.

Acest lucru este similar cu estimarea M, cu toate că aici maximizăm funcția obiectiv $p_K(F)$, nu o minimizăm. O derivare similară cu cea pentru estimare M din F trebuie să îndeplinească $F_* = [\sum \omega(F - F_*)F] / [\sum \omega(F - F_*)]$ în cazul în care $\omega(e) = -k'(e^T \sum^{-1} e)$ și $k'(s) ds = dk / ds(s)$. În cuvinte, F_* trebuie să fie media ponderată a $F(x)$ folosind ponderea $\omega(F(x) - F_*)$ centrată pe F_* .

Analog ideii de reponderare iterativă folosită în estimarea M este rezolvarea pentru F^* , prin reiterarea ecuația de deplasare medie $F_{j+1} = [\sum \omega(F - F_j) F] / [\omega(F - F_j)]$, unde F_{j+1} este chiar medie ponderată a punctelor pe caracteristica $F(x)$, cu ponderea $\omega(F - F_j)$ centrată pe valoarea anterioară F_j .

Segmentarea cu algoritmul deplasării medii. Scara nucleului deplasării medii controlează, aproximativ, mărimea și forma regiunilor extrase. Există un compromis între menținerea unor granițe liniare, dar care suferă în timpul segmentării, comparativ cu lipsa unor granițe importante și subsegmentarea imaginii.

Un sistem îmbunătățit (**Edge Detection and Image SegmentatiON - EDISON**[28]) combină algoritmul deplasării medii cu informații despre muchii. O măsură despre muchia lină este folosită pentru a modifica funcția pondere folosită în ecuația deplasării medii (29). Aceasta facilitează compromisul de mai sus, care permite granițe slabe de păstrat în segmentare, fără a genera la fel de mult supra-segmentare.

2.6.2 Metode bazate graf de similitudine

Metodele bazate graf de similitudine oferă o alternativă la funcția de grupare în spațiu a caracteristicilor. Un graf neorientat ponderat $G = (V, E)$ este format, cu set de noduri V corespunzătoare la x pixeli din imagine. Marginile E în grafic sunt luate între oricare doi pixeli x_i și x_j la o distanță mică unul de altul. Ponderea marginii $\omega(x_i, x_j) \geq 0$ reflectă disimilaritate (alternativ, similitudinea) între două imagini vecine centrate pe pixeli x_i și x_j . O formă comună a ponderii este folosirea $\omega(x_i, x_j) = 1 - a(x_i, x_j)$, unde $a(x_i, x_j)$ este dat de

$$A(x_i, x_j) = e^{-1/2(F(x_i) - F(x_j))T \Sigma^{-1} (F(x_i) - F(x_j))}, \text{ unde}$$

$F(x)$ este vectorul caracteristică asociat cu pixel x , de exemplu:

- $F(x) = I(x)$, astfel încât afinitatea este determinată numai prin diferența de culoare de nivel gri între pixeli învecinați;

- $F(x) = I(x)$, valorile RGB pentru o imagine color, sau maparea câtorva valori RGB la un spațiu de culoare mai uniformă (ex. $L^*u^*v^*$);

- $F(x)$ include primitive de textură, cum ar fi răspunsurile unui filtru local, împreună cu luminozitatea și / sau culoarea pixelului x .

Componente conectate. O abordare simplă este ștergerea tuturor marginilor între pixeli diferiți (de exemplu, cu ponderi $\omega(x_i, x_j) > \tau$ și, atunci, caută componentele conectate (CC) în graficul rămas. O singură muchie cu $\omega(x_i, x_j) \leq \tau$ ar fi suficientă pentru a determina ca cele două regiuni de interes să fie îmbinate. Prin urmare, CC nu sunt viabile să se abată de la legăturile (numite „scurgeri”) între regiuni. Consecința este că deseori nu există nicio valoare corespunzătoare τ care să ofere o segmentare utilă.

Algoritmul lui Kruskal. Este util de subliniat că o modalitate eficientă de a face clustering CC, cu o variabilă τ , este aceea de a construi, întâi, un arbore minim de expansiune a grafului (minimal spanning tree - **MST**). Algoritmul lui Kruskal poate fi folosit, pentru că este o abordare garantată pentru a da un MST optim. Începând cu graficul complet deconectat, marginile se adaugă una la un moment dat, în ordinea crescătoare a ponderii lor, atâta timp cât, adăugând o margine, nu se introduc cicluri în subgraful curent.

Componentele conectate (CC) grafului tăiat (cu margini, având $\omega(x_i, x_j) > \tau$ șterse) sunt apoi calculate eficient prin ștergerea acestor margini din arborele minim de expansiune a grafului (MST). Arborii din „pădure” rezultantă furnizează CC-urile dorite.

Metoda Variației locale. Felzenszwalb și Huttenlocher [32] introduc o modificare simplă, dar eficientă a algoritmului lui Kruskal. Ca și în algoritmul lui Kruskal, se începe cu graficul complet deconectat, marginile se adaugă câte una la un moment dat, în ordinea crescătoare a ponderii lor, menținând o „pădure” de MST-uri pentru componentele curente.

În timpul prelucrării, fiecare C_i al MST-ului este asociat cu un prag $T(C_i) = \omega(C_i) + k/I C_i I$ (1) în cazul în care $\omega(C_i)$ este ponderea maximă în arborele expandat C_i (de exemplu, variațiile locale de C_i). De asemenea, $k > 0$ este o constantă, și $|C_i|$ este numărul de pixeli în C_i . (1).

Să presupunem că marginea (x_k, x_l) urmează fi prelucrată în continuare, și cele două puncte finale sunt separate în două MST-uri C_i și C_j . Apoi, aceste MST-uri sunt îmbinate prin adăugarea marginii (x_k, x_l) numai în cazul în care $\omega(x_k, x_l) \leq \min(T(C_i), T(C_j))$. (2)

Deoarece dimensiunea lui C_i crește, (1) și (2) dictează un $T(C_i)$ ușor superior (în comparație cu cea mai mare pondere $\omega(C_i)$ în C_i) pentru o afinitate acceptabilă a unei margini care fuzionează C_i cu o altă regiune.

Sortarea marginilor în funcție de pondere provoacă ca algoritmul să crească întâi regiunile relativ omogene. Parametru k din (1) controlează aproximativ dimensiunea regiunilor în segmentarea rezultată. Un k mai mare oferă o constrângere mai flexibilă (2) și permite mai multă fuzionare.

Fuzionarea este sensibilă la variațiile locale în cadrul regiunilor de îmbinat. Datorită legării din ce în ce mai strânse, o regiune omogenă mare C_i este adăugată la cele fracționate, cu cea mai mare pondere $\omega(C_i)$, adică la cele cu cea mai mare afinitate C_i în MST, numai folosind marginile cu pondere. Cu toate acestea, această legare este mult mai flexibilă pentru regiunile mici C_i , încurajând creșterea lor.

Abordarea are tendința de a produce regiuni înguste de-a lungul granițelor segmentului „adevărat”. Abordarea este foarte ușor de implementat, deoarece necesită operațiuni $O(\log(e))$, unde e este numărul de muchii.

2.6.3 Tehnici de segmentare

Prin segmentare se înțelege divizarea imaginii într-un număr finit de regiuni de interes, în conformitate cu anumite criterii. O segmentare completă a unei imagini (privită ca o regiune globală R) constă în determinarea unui set de regiuni R_1, R_2, \dots, R_p , astfel încât să fie îndeplinite condițiile: $R = R_1 \cup R_2 \cup \dots \cup R_p$

Un exemplu de segmentare îl constituie separarea obiectelor prezente în imagine de fundal. În cazul în care atât obiectele cât și fundalul se caracterizează prin niveluri medii de gri diferite, acest lucru se poate realiza printr-o simplă filtrare a nivelurilor de gri în funcție de un anumit prag T (**threshold**), rezultând o imagine binară astfel:

- a) imagine cu 256 niveluri de gri;
- b) imagine segmentată $T=110$.

Trebuie notat faptul că performanțele segmentării sunt determinate de alegerea nivelului de prag T . Metodele privind alegerea valorii acestuia au în considerare histograma nivelurilor de gri a imaginii. Un algoritm simplu iterativ pentru alegerea automată a pragului în cazul unei histogramme bimodale presupune următorii pași:

- a) alegerea unei prime aproximări pentru T (de exemplu $T=1$);
- b) la pasul i se calculează nivelurile medii de gri ale obiectelor și fundalului: histograma imaginii este bimodală în sensul că există două concentrări importante ale nivelelor de gri - în jurul valorii 41 (nivelul mediu de gri al obiectelor) și în jurul valorii 150 (nivelul mediu de gri al fundalului). Ca urmare, nivelul pragului poate fi ales între cele două valori. ($T=110$);
- c) calcularea pragului T la pasul i : dacă valoarea T a pragului la pasul i este egală cu valoarea pragului la pasul anterior sau dacă a expirat numărul de iterații dorite, algoritmul se încheie, altfel se reia cu pasul 2.

De regulă, un număr de iterații între 4 și 10 este suficient. În cazul histogramelor multimodale (de exemplu, pot exista obiecte cu diferite niveluri medii de gri), pot fi alese mai multe niveluri de prag, corespunzătoare concentrării nivelurilor de gri în jurul unor anumite

puncte în histograma imaginii:

În cazul imaginilor color, segmentarea presupune calcularea histogramelor pentru cele trei culori de bază (R, G, B) și determinarea pragurilor T în funcție de culoarea obiectelor ce se doresc a fi separate de fundal.

Se realizează segmentarea pentru fiecare componentă, rezultând trei imagini binare finale care vor fi supuse unei operații de tip OR.

2.7 Analiza și recunoașterea formelor

Procesul de segmentare conduce la obținerea unei imagini binare care conține obiectele separate de fundal. Utilizând un algoritm de determinare a componentelor conexe se realizează clasificarea și etichetarea obiectelor, ceea ce creează premisele pentru ultima etapa în cadrul procesării imaginilor: **analiza formelor**. În urma acestei analize, pentru fiecare obiect în parte se pot calcula diverși parametri (cum ar fi, de exemplu, invarianții), parametri care pot fi utilizați într-un sistem automat de clasificare. Există o graniță foarte fină în această etapă între domeniul procesării imaginilor și domeniul recunoașterii formelor. Prin formă se înțelege reprezentarea matematică a unui obiect pe baza observațiilor și măsurărilor efectuate asupra lui.

Există două direcții principale în domeniul analizei formelor, și anume:

- abordarea funcțională;
- abordarea din perspectiva teoriei mulțimilor.

Fiecare dintre cele 2 direcții definește conceptul de formă, plecând de la diferite abordări matematice. Abordarea funcțională se bazează pe aproximarea conturilor formelor (obținute eventual printr-o metodă de detecție a muchiilor) cu diverse funcții, ceea ce permite aplicarea rezultatelor geometriei diferențiale și analitice în studiul acestora. Funcțiile contur rezultate pot fi approximate cu serii trigonometrice de funcții **sin** și **cos** ceea ce permite o caracterizare a formelor cu ajutorul coeficienților Fourier.

Abordarea din perspectiva teoriei mulțimilor privește formele ca fiind mulțimi compacte și închise și are în vedere obținerea anumitor proprietăți geometrice sau rapoarte. În acest sens se pot calcula următoarele proprietăți:

- $A(X)$ - aria formei X;
- $P(X)$ - perimetrul formei X;
- $DA(X)$, $DP(X)$ - diametrele cercurilor având aceeași arie respectiv perimetru cu forma X;
- $FMD(X)$ - Feret Maximum Diameter – cea mai mare distanță dintre două puncte aparținând formei X (este o măsură a alungirii);
- $C(X)$ - circularitatea $C(X) = 4 \cdot A(X) / (P(X))^2$ (caracterizează devierea formei X față de un disc. Valoarea maximă este 1 în cazul discului);
- $a(X)/b(X)$ - raportul lungimilor axelor unei elipse având aceeași arie și același perimetru cu forma X (acest raport este o măsură a alungirii);
- raportul de convexitate: $A(X)/A(\text{conv}(X))$ unde $\text{conv}(X)$ reprezintă poligonul dat de înfășurătoarea convexă a formei X (raportul este o măsură a convexității, pentru o formă convexă luând valoarea 1, iar în caz contrar valoarea este subunitară).

Au fost prezentate numai câteva rapoarte și proprietăți doar pentru exemplificare, alegerea și definirea lor făcându-se în funcție de specificul fiecărei aplicații. O proprietate importantă pe care parametrii pot să o dețină este invarianta (la translație, rotație, scalare), un exemplu în acest sens fiind momentele de diferite ordine. În final, o formă va putea fi caracterizată de un set de valori (x_1, x_2, \dots, x_p) , urmând aplicarea tehnicilor specifice recunoașterii formelor în vederea clasificării acesteia.

În acest capitol, au fost prezentate, sumar, metodele și tehnicile folosite în cadrul procesării imaginilor, multe aspecte fiind ignorate (cum ar fi: reconstrucția imaginilor, compresia, analiza texturii, reconstrucția 2D, etc.). Aplicațiile acestui domeniu sunt practic nelimitate: de la simple corecții ale defectelor imaginilor până la aplicații complexe în medicină, industrie, securitate.

3. Algoritmi de reconstituire 2D a unei cetăți

Pentru realizarea **reconstituirii 2D** se vor parcurge următoarele etape, și anume:

- analiza și procesarea geometrică a fotografiei;
- descompunerea imaginii în suprafețe;
- generarea mesh-urilor pentru simularea spațiului 2D;
- generarea suprafețelor ascunse, a interioarelor;
- animarea reconstituirii, ceea ce dă posibilitatea vizualizării muchiilor invizibile.

Prima fază a reconstituirii constă în **analiza grafică a imaginii cetății și descompunerea** a acesteia în suprafețe, prin trasarea de contururi de culori diferite funcție de tipul elementului (fațada frontală, lateral dreapta/stânga, principală, etc).

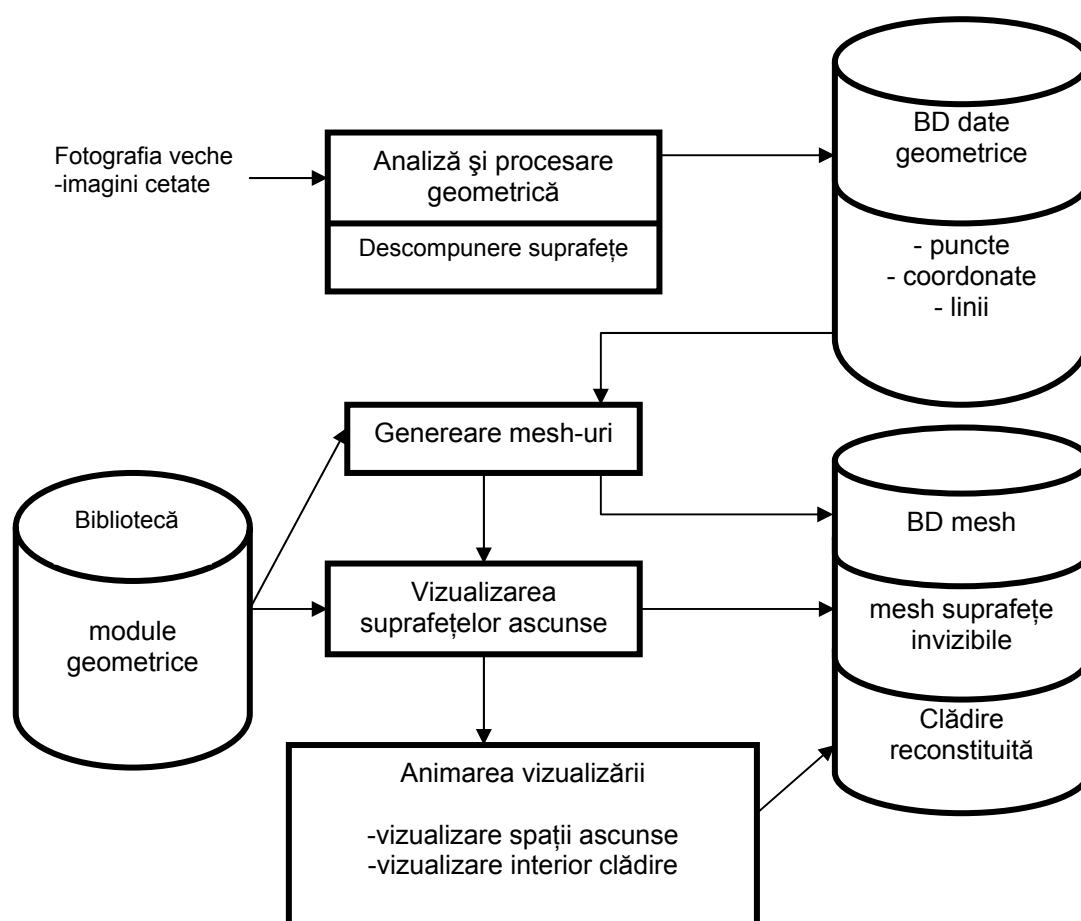


Figura 1. Schema de realizare a Reconstituirii 2D

3.1 Generarea automată a mesh-urilor

Procesul de creare a mesh-ului poate fi definit ca procesul de împărțire a unui domeniu fizic în subdomenii mai mici (elemente) pentru a facilita găsirea unei soluții numerice a unei ecuații diferențiale parțiale. Mesh-ing-ul este o parte integrată în procesul de analiză. Mesh-ul

influențează acuratețea, convergența și viteza soluțiilor. Timpul în care se creează modelul mesh-ului este cea mai semnificativă parte a timpului de creare a unei soluții; cu cât se creează mai repede mesh-ul și cu cât este mai automatizat instrumentul de mesh-ing, cu atât va fi mai bună și soluția. Cea mai răspândită metoda este **metoda elementelor finite**. Domeniile suprafeței pot fi divizate în forme de triunghiuri sau patrulatere, în timp ce volumele pot fi divizate în forme de tetraedru (piramidă triunghiulară) sau hexaedru (poliedru cu 6 fețe, cum este cubul). Forma și distribuția elementelor este definită ideal de algoritmi automați de creare a mesh-urilor.

În continuare, sunt prezentați algoritmi fundamentali folosiți în generarea automată a mesh-urilor structurate sau nestructurate, sub formă de triunghiuri, piramide triunghiulare, patrulatere și hexaedre. În plus, sunt prezentate problemele legate de post-procesarea mesh-urilor incluzând aici netezirea, curățirea și rafinarea.

3.2 Vizualizarea reconstituirii

Tehnicile de redare a imaginii permit integrarea reprezentărilor virtuale a lumii reale în aceste noi concepte media. În multe cazuri, căutarea realismului în reprezentările virtuale nu este doar de dorit, dar este și un atribut esențial. Anumite aplicații necesită un nivel suficient de asemănător cu cel original. Asemenea situații apar de multe ori în documentările despre patrimoniu sau în simulări și învățare și în aplicațiile industriale. Tehnicile de redarea imaginii au un rol esențial în atingerea acestui obiectiv prin folosirea datelor fizice ca o bază pentru acest proces.

Lucrarea de față este orientată către căutarea și dezvoltarea de astfel de tehnici. Îmbunătățiri semnificative sunt necesare în inspectarea și procesarea imaginii, în special către creșterea substanțială a flexibilității și a automatizării, pentru a permite, în general, extinderea utilizării acestei tehnologii.

Reconstituire 2D din imagini

În acest proiect se urmărește dezvoltarea de software pentru reprezentări avansate și elaborarea de metode de vizualizare pentru convertirea fotografiilor vechi în modele 2D. Noua tehnologie dezvoltă aplicații avansate în astfel de domenii cum sunt grafică computerizată, animație și efecte speciale.

Metodele tradiționale de reconstrucție 2D, bazate pe imagini, folosesc imagini multiple pentru a extrage geometria 2D. Totuși, nu este întotdeauna posibil să obținem astfel de imagini, cum ar fi la reconstrucția structurilor distruse folosind fotografiile existente sau picturile cu o perspectivă adecvată.

Există metode pentru reconstrucția 2D dintr-o singură imagine. Acestea folosesc blocuri ale formelor cunoscute pentru reconstrucția dintr-o singură imagine. Pentru reconstrucția imaginii, aceste metode folosesc linii și puncte ascunse, plecând de la un model obținut de la camera fotografică. Abordarea nu are nevoie de modele ale obiectelor sau parametrii acestora, nu are nevoie nici de linii sau puncte ascunse, care uneori nu sunt disponibile sau sunt foarte greu de extras. Se folosesc diferite tipuri de constrângeri: constrângerii de tip punct/coordonată, constrângeri de suprafețe și constrângeri topologice.

3.3 Automatizarea modelării bazată pe imagini

Automatizarea modelării bazată pe imagini cercetează modul în care noile tehnici pot oferi o utilizare pe scară largă a automatizării procesului de modelare 2D.

Modelarea bidimensională din imagini, atunci când este făcută numai de un om, poate dura foarte mult și este nerealizabilă pentru proiecte de dimensiuni mari. Pe de altă parte, metodele complet automate nu pot obține sau nu pot fi destul de precise (exacte) pentru multe aplicații cum ar fi documentarea "moștenirii". Modelarea tridimensională din imagini are nevoie de

extragerea trăsăturilor (caracteristicilor), cum ar fi colțurile și apariția lor în imaginile multiple. Totuși, în situațiile practice, aceste trăsături nu sunt întotdeauna disponibile și uneori nu apar în imaginile singulare, datorită ocluziei sau absenței texturii suprafeței.

Obiective:

- folosirea tehnicilor automate cât și a celor interactive, fiecare unde se aplică mai bine, pentru obținerea modelului obiectelor complexe cât mai precis și complet;
- concentrarea către automatizarea construcției suprafețelor nemarcate cum sunt coloanele și arcadele dintr-un număr minim de indici disponibili;
- extragerea coțurilor obstructionate sau invizibile din suprafețele și liniile existente;
- exportul coordonatelor 2D și a modelelor în diferite formate 2D;
- modelarea 2D din două sau mai multe fotografii;
- modelarea 2D dintr-o singură fotografie;
- extragerea informațiilor în diferite formate 2D;
- gestionare îmbunătățită a mulțimilor și atribuirea diferitelor tipuri de suprafețe fiecărei mulțimi;
- trasarea mesh-urilor triunghiulare pe orice imagine. Toate mulțimile triunghiulare sau fiecare mulțime triunghiulară individuală poate fi trasată pe o imagine;
- vizualizarea punctelor de ștergere;
- introducerea directă a mulțimilor de coordonate 2D;
- măsurarea dimensiunilor dintre puncte;
- combinarea modelelor create independent din diferite mulțimi de imagini într-un singur model. De exemplu, un model creat din imagini apropiate ale unei fațade a unei clădiri pot fi combinate cu modelul creat de imaginile depărtate ale aceleiași clădiri;
- după crearea unui model dintr-o mulțime de imagini, pot fi înscrise cu acest model imagini adiționale, obținute cu scopul de îmbunătățire a texturii;
- modelul structurii de legare poate fi vizualizat și editat folosind un editor 2D. Acest lucru este folositor în înlăturarea triunghiurilor nedorite. Modelul poate fi editat de această dată adăugând sau înlăturând triunghiuri din proiectul structurii de legare al oricărei imagini;
- calcularea coordonatelor 2D ale trăsăturilor extrase, din două sau mai multe imagini;
- adaptarea datelor la modelele geometrice standard: când punctele 2D cad pe suprafață care este plană, sferă, cilindru;
- modelare geometrică și trasarea texturii pentru vizualizarea datelor;
- combinarea modelelor create independent din diferite mulțimi de imagini într-un singur model. De exemplu, un model creat din imagini apropiate ale unei fațade a unei clădiri poate fi combinat cu modelul creat de imaginile depărtate ale aceleiași clădiri, sau un model al unei secțiuni a unei camere poate fi combinat cu alt model al altei secțiuni a camerei;
- modelarea este automată pentru vârfurile îmbinate;
- trasarea texturii: distorsiunea perspectivă este înlăturată din texturi.

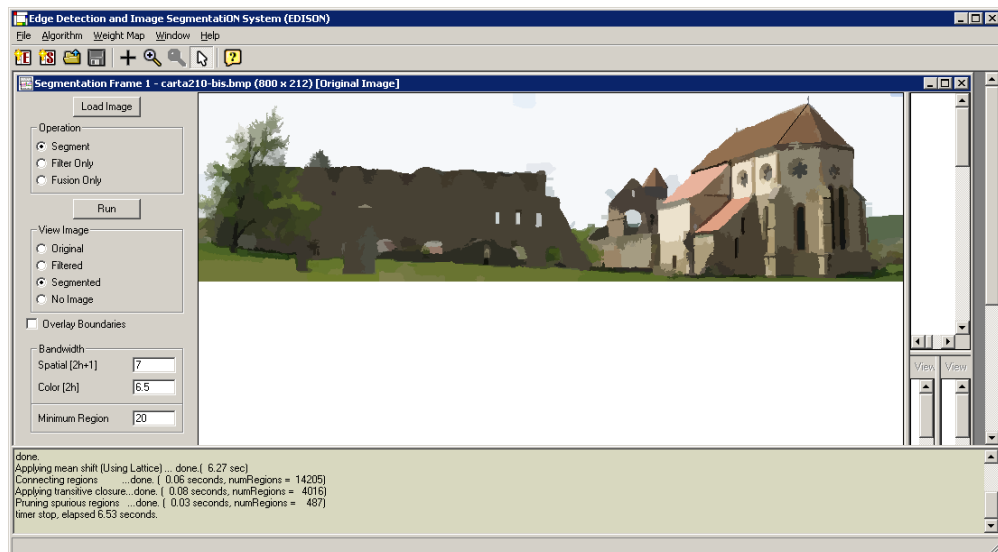
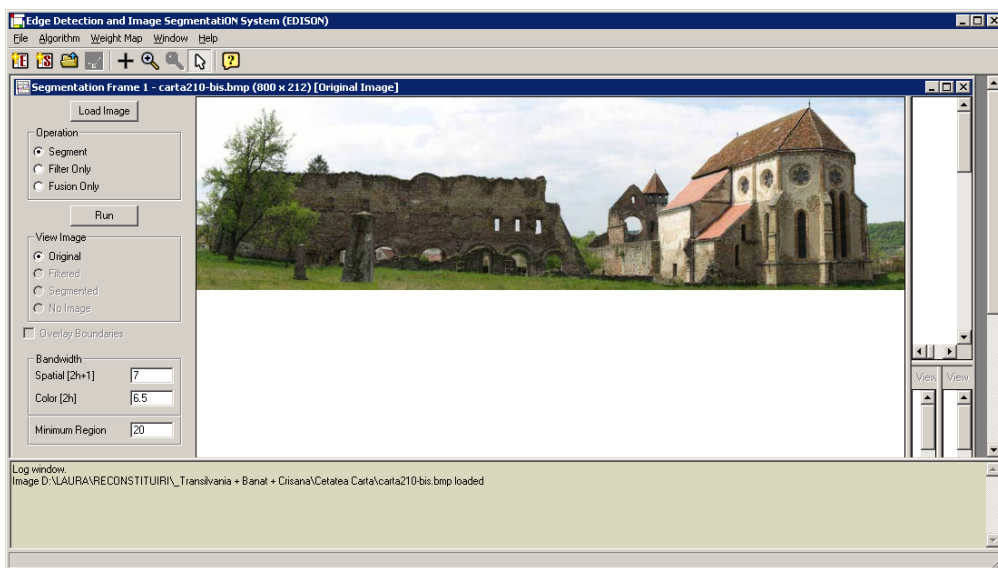
Operații de procesare a imaginii constau în:

- limitarea automată (se folosește în extragerea vârfurilor și la operații referitoare la câteva muchii);

- reducerea distorsiunii;
- ajustarea și evidențierea;
- detectarea muchiilor.

Trasarea texturii din modelele 2D este o ramură a cercetării în dezvoltarea și aplicarea tehnicilor imagistice pentru reprezentări virtuale cu scopul de a crea o experiență virtuală extrem de realistă. Aceasta se ocupă cu vizualizarea real-time și manipularea interactivă a suprafețelor extrem de texturate. În acest proiect, trasarea texturii se aplică la două concepte ale reprezentărilor virtuale: filme de vizitare foto-realistice și modele 2D care permit utilizatorului vizualizarea interactivă și manipularea.

Pentru realizarea detectării muchiilor și segmentării imaginii, se va utiliza un software **Open Source Edison**, utilizând un editor de imagine propriu, gândit pentru a executa decupări și memorări fidele; concret, în acest editor se vor decupa elementele monocrome, care se vor memora pe disc, așa cum este prezentat în imaginile de mai jos.



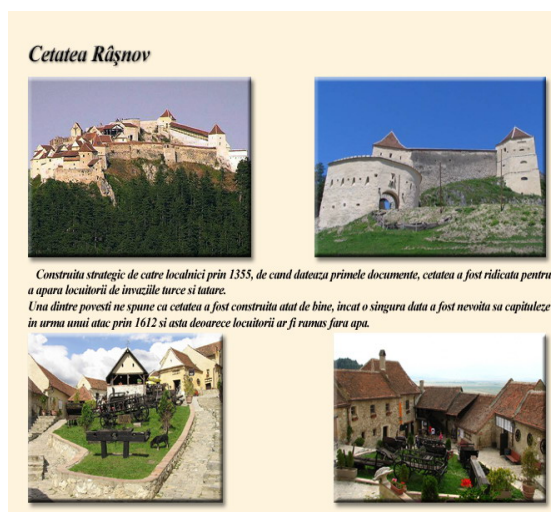
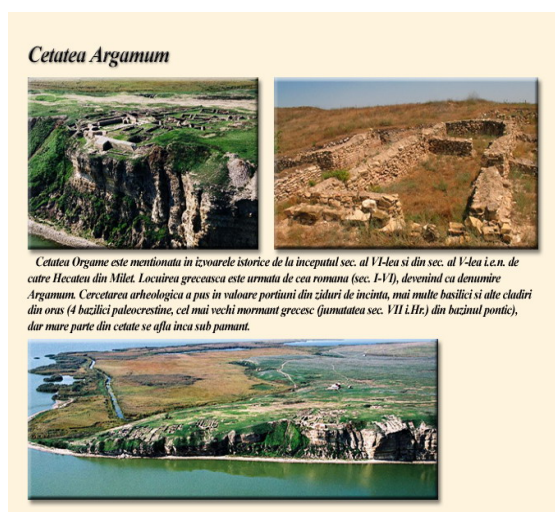
Prin utilizarea tehnologiei DirectX, pentru redarea în 2D a imaginii se crează părțile care nu se văd drept imagini în oglindă ale zonelor vizibile:

- se alege în imagine un punct, cât mai convenabil, care va reprezenta centrul sistemului cartezian Oxyz;
- pentru fiecare modul planar VIZIBIL (determinat de 3, 4 sau mai multe puncte coplanare) vom calcula coordonatele spațiale, față de sus-amintitul centru;
- se crează fiecare modul planar NEVĂZUT ca fiind imaginea în oglindă a unui anume modul planar VIZIBIL; în acest scop, se utilizează coordonatele fiecărui plan vizibil cărora li se calculează imaginile simetrice față de planele principale : Oxy, Oyz, Ozx; fiecare plan nou creat are culoarea de fond a propriului plan „sursă”;
- în final se ”assemblează” prin operații de „drag-drop” în planul final eD toate elementele planare pe care le avem la dispoziție în acest moment: cele inițial VIZIBILE și cele calculate.

4. Prezentarea cetății în Internet și pe CD multimedia

Pentru marcarea atributelor elementelor (coordoanate x-y, poziționare față de o suprafață de referință, culoare) s-a folosit **Inprise Delphi**.

Pentru generarea **structurii ierarhice** a “obiectelor” ce definesc o clădire se va folosi **Inprise Delphi**; gestionarea bazei de date ce conține descrierea elementelor componente și a legăturilor dintre ele se va realiza utilizând **SQL** server.



5. Concluzii

În cadrul lucrării a fost realizat un instrument de **Reconstituire 2D** cu scopul de a reconstitui cetăți din fotografii vechi. Componenta de **Reconstituire 2D** va completa CD-ul multimedia și va permite vizualizarea în Internet 2D.

În prezent, *sistemele multimedia* alături de rețeaua Internet au revoluționat circulația informației. Realizarea de pagini în rețeaua Internet pentru **Cetăți Medievale din România** și a **DVD-ului Interactiv** pentru reconstituire 2D este importantă pentru continuarea susținerii accesului la patrimoniului cultural.

BIBLIOGRAFIE

1. * * * Adobe Photoshop
2. * * * Delphi - A Guide to Programming
3. * * * SQL server
4. **2D From paintings and photos:** http://iit-iti.nrc-cnrc.gc.ca/projects-projets/paintings-tableaux_e.html
5. ANSYS web site: <http://www.ansys.com>
6. **BOYKOV, Y.; KOLMOGOROV, V.:** An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. submitted IEEE Trans. Pattern Anal. and Machine Intell., 2004.
7. **CARSON, C.; BELONGIE, S.; GREENSPAN, H.; BLOBWORLD, MALIK J.:** Image segmentation using expectation-maximization and its application to image querying. IEEE Trans. Pattern Anal. And Machine Intell., 24(8):1026–1038, 2002.
8. **CHRISTOUDIAS, C. M.; GEORGESCU, B.; MEER, P.:** Synergism in low level vision. In 16th International Conference on Pattern Recognition., Quebec City, Canada, volume IV, pp. 150–155, 2002.
9. **CIOCOIU, L.; BOROZAN, A-M.; COȘOIU, C.:** Tema A19 / 2001 – “Muzeu virtual privind arhitectura până la începutul sec. XX în rețeaua Internet”.
10. **CIOCOIU, L.; BOROZAN, A-M.; COȘOIU, C.:** Tema A20/2001 – “Arhive virtuale specifice muzeelor județene în rețeaua Internet”.
11. **COCQUEREZ, J.P.; PHILIPP, S.:** Analyze d’Images: Filtrage et Segmentation.
12. **COMANICIU, D.; MEER, P.:** Mean shift: A robust approach toward feature space analysis. IEEE Trans. Pattern Anal. and Machine Intell., 24:603–619, 2002.
13. **COUR, T.; YU, S.; SHI, J.:** Normalized cuts matlab code. Computer and Information Science, Penn State University. Code available at <http://www.cis.upenn.edu/~jshi/software/>.
14. **CUBIT Mesh Generation Toolkit**, web site: <http://cubit.sandia.gov/>
15. **EL-HAKIM, S.; BERARDIN, J.-A.; PICARD, M.:** 2D Modeling of Heritage Monuments. GIM International, 17(4): 13-15. April 2003. NRC 45821.
16. **EL-HAKIM, S.F.:** Semi-automatic 2D Reconstruction of Occluded and Unmarked Surfaces from Widely Separated Views. Proceedings of ISPRS Commission V Symposium, Close Range Visualization Techniques, Corfu, Greece. Sept. 1-2, 2002, pp. 143-148

NRC 44944.

17. **ESTRADA, F.J.; JEPSON, A.D.; CHENNUBHOTLA, C.:** Spectral embedding and min-cut for image segmentation. In British Machine Vision Conference, 2004.
18. **FEGS web site:** <http://fegs.co.uk>
19. **FELZENSZWALB, P.F.; HUTTENLOCHER, D.P.:** Efficient graph-based image segmentation. *Int. Journ. of Comp. Vis.*, 59(2):167–181, 2004.
20. **GEORGE, P.L.; HECHT, F.; SALTEL E.:** Automatic Mesh Generator with Specified Boundary. *Computer Methods in Applied Mechanics and Engineering*, North-Holland, vol. 92, 1991, pp. 269-288.
21. **Home page:** http://iit-iti.nrc-cnrc.gc.ca/r-d/2D-vir-reality-realite-vir-2D_e.html
22. **LOHNER, R.:** Progress in Grid Generation via the Advancing Front Technique. *Engineering with Computers*, vol 12, 1996, pp.186-210.
23. **MacNeal-Schwendler Home Page,** web site: <http://www.mssoftware.com/>
24. **MALIK, J.; BELONGIE, S.; LEUNG, T.; SHI, J.:** Contour and texture analysis for image segmentation, *Int. Journ. of Computer Vision*, 43(1):7–27, 2001.
25. **MARION, A.:** Introduction aux Tehniques de Traitement d’Images.
26. **MARTIN, D.; FOWLKES, C.:** The Berkeley Segmentation Dataset and Benchmark. <http://www.cs.berkeley.edu/projects/vision/grouping/segbench/>.
27. **MARTIN, D.; FOWLKES, C.; MALIK, J.:** Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Pattern Anal. and Machine Intell.*, 26(5): 530–549, 2004.
28. **MARTIN, D.; FOWLKES, C.; MALIK, J.:** A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int’l Conf. Computer Vision*, volume 2, pp. 416–423, July 2001.
29. **NOWOTTNY, DIETRICH:** Quadrilateral Mesh Generation via Geometrically Optimized Domain Decomposition. *Proceedings, 6th International Meshing Roundtable*, 1997, pp. 309-320.
30. **OWEN, S. J.:** Meshing Software Survey. *Structured Grid Generation Software*, web page: <http://www.andrew.cmu.edu/user/sowen/software/structured.html>
31. **OWEN, S.:** A survey of unstructured mesh generation technology. *Proceedings of the 7th International Meshing Roundtable*, pp. 239-267, <http://www.andrew.cmu.edu/user/sowen/survey>, 1998.
32. **OWEN, STEVEN J.; STATEN, MATTHEW L.; CANANN, SCOTT A.; SAIGAL, SUNIL:** Advancing Front Quad Meshing Using Local Triangle Transformations. *Proceedings, 7th International Meshing Roundtable*, 1998.
33. **PRATT, W.K.:** *Digital Image Processing*. 2nd Edition, John Wiley & Sons, New York, 1991.
34. **RUSS, J.C.:** *The Image Processing Handbook*. 2nd Edition, CRC Press.
35. **SHI, J.; FOWLKES, C.; MARTIN, D.; SHARON, E.:** Graph based image segmentation tutorial. *CVPR 2004*. <http://www.cis.upenn.edu/~jshi/GraphTutorial/>.
36. **SHI, J.; MALIK, J.:** Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. And Machine Intell.*, 22(8):888–905, 2000.
37. **STATEN, MATTHEW L.; CANANN, SCOTT A.; OWEN, STEVE J.:** BMSWEEP: Locating Interior Nodes During Sweeping. *7th International Meshing Roundtable*, 1998.
38. **TetMesh, GSH2D web site:** <http://www.simulog.fr/tetmesh/>

39. **TOUMAZET, J.J.:** Traitement de l'Image sur Micro-ordinateur.
40. **WHITE, DAVID R.:** Automated Hexahedral Mesh Generation by Virtual Decomposition. Proceedings, 4th International Meshing Roundtable, Sandia National Laboratories, 1995, pp. 165-176.
41. **WHITE, DAVID R.; KINNEY, PAUL:** Redesign of the Paving Algorithm: Robustness Enhancements through Element by Element Meshing. Proceedings, 6th International Meshing Roundtable, Sandia National Laboratories, 1997, pp. 323-335.
42. **YU, S.; SHI, J.:** Multiclass spectral clustering. In Proc. Int'l Conf. Computer Vision, 2003.