

Pràctica 2: Region Growing

Lluís F Collell i Miguel Agundez

Breu introducció:

La segmentació d'imatges és una tècnica fonamental en l'àmbit de l'anàlisi i el processament d'imatges, ja que permet identificar i separar les regions d'interès dins d'una imatge. En aquesta pràctica, es treballarà amb l'algorisme de creixement de regions (Region Growing), un mètode eficaç i relativament senzill d'implementar, que permet obtenir bons resultats fins i tot en escenaris complexos. L'objectiu principal és desenvolupar una implementació funcional d'aquest algorisme utilitzant MATLAB, capaç de treballar tant amb imatges en escala de grisos com amb imatges en color.

Índex de continguts

1. Introducció a la segmentació d'imatges
 - 1.1. Tipus de segmentació
 - 1.2. Què és el *Region Growing*
2. Paràmetres clau del *Region Growing*
 - 2.1. Valor *Threshold*
 - 2.2. Veïnatge (*4-connectat* i *8-connectat*)
3. Implementació i proves amb matrius artificials
 - 3.1. Representació d'imatges com a matrius
 - 3.2. Proves amb matrius en escala de grisos
 - 3.3. Proves amb matrius RGB (*3 canals*)
4. Implementació de l'algorisme *Region Growing* a color
5. Proves i temps d'execució (comparació de *Thresholds*)
 - 5.1. Segmentació en escala de grisos
 - 5.2. Segmentació en RGB
6. Problemes i possibles millores
 - 6.1. Anàlisi de la segmentació en la imatge *maduixa.jpg*
 - 6.2. Causes principals de la sobresegmentació
7. Conclusions
8. Referències

1. Introducció a la segmentació d'imatges

La segmentació d'imatges és una etapa fonamental dins el camp de l'anàlisi i processament d'imatges digitals. El seu **objectiu principal és dividir una imatge en regions significatives que facilitin la identificació i anàlisi d'estructures o objectes dins la imatge.**

Aquest procés és considerat un dels problemes clàssics i encara oberts dins la visió per computador. **Tot i que s'han proposat nombroses tècniques, l'elecció de la millor depèn sovint del domini i de la naturalesa de les imatges a processar.** En aplicacions com la detecció de tumors, la segmentació pot esdevenir una etapa crítica per a l'èxit global del sistema.

Hi ha **diverses formes d'enfocar la segmentació: des de mètodes supervisats fins a tècniques no supervisades**, i cadascun té avantatges i limitacions que s'han d'estudiar amb cura.

1.1 Tipus de segmentació

Els mètodes de segmentació es poden agrupar en diverses categories, entre les quals destaquen:

A. Mètodes basats en regions

Aquestes tècniques **cerquen identificar regions homogènies dins la imatge a partir de criteris com la intensitat, la textura o el color.** Inclouen:

- **Creixement de regions (Region Growing):** S'expandeixen regions a partir de píxels inicials (llavors), agregant-ne de nous segons una condició de similitud.
- **Divisió i fusió (Split & Merge):** Parteixen d'una divisió recursiva de la imatge (normalment en forma de quadtree) i posterior fusió de regions similars.

B. Mètodes basats en clustering

Es basen en **tècniques estadístiques que agrupen píxels segons característiques compartides (com el color o la intensitat), independentment de la seva posició espacial:**

- **Lindars (Thresholding):** S'utilitza un valor (o valors) per separar els píxels en grups.
- **K-means:** Algorisme iteratiu que agrupa els píxels en k clústers, minimitzant la distància respecte al centre del clúster.
- **EM (Expectation-Maximization):** Mètode més avançat que considera distribucions probabilístiques.

1.2 Què és el Region Growing

El **creixement de regions** és un algorisme de segmentació que parteix d'un conjunt inicial de píxels, anomenats *llavors*, i expandeix cadascuna d'aquestes regions afegint píxels veïns que compleixin un determinat criteri de similitud (normalment basat en la intensitat o el color).

Funcionament bàsic:

1. S'escullen un o més píxels com a llavors.
2. Es marquen com a considerats i s'inicia una nova regió.
3. Es revisen els píxels veïns. Si compleixen el criteri de similitud amb la mitjana de la regió (per exemple, $|f(x,y) - \mu_{Ri}| \leq \Delta$), s'afegeixen a la regió.
4. El procés es repeteix de manera recursiva o iterativa fins que no es poden afegir més píxels.

Consideracions:

- **Implementació:** es pot fer de forma recursiva (tot i que pot generar problemes per l'excés d'iteracions), seqüencial o concurrent.
- **Criteri d'agregació:** defineix si un píxel s'afegeix o no a la regió. Pot considerar intensitat, color, textura, etc.
- **Distribució de llavors:** la seva quantitat i ubicació inicial són claus per obtenir una segmentació adequada.
- **Connectivitat:** pot ser de 4 (A4) o 8 (A8) veïns.

És una tècnica molt utilitzada per la seva **simplicitat i bons resultats en imatges amb regions homogènies**, com teixits cerebrals, lesions o tumors.

La nostra implementació:

En el nostre projecte optem per un mètode **no supervisat** basat en **creixement de regions (Region Growing)**, treballant amb els dos tipus de connectivitat (A4 i A8) per adaptar-nos millor a la diversitat de les imatges a segmentar i millorar la robustesa del nostre algorisme.

2. Paràmetres clau del Region Growing

L'algorisme de *Region Growing* depèn de diversos paràmetres que condicionen fortament la qualitat i la precisió de la segmentació obtinguda. En aquesta secció es descriuen dos dels paràmetres més importants: el llindar (threshold) i el tipus de veïnatge utilitzat per a l'expansió de les regions.

2.1 Valor Threshold

El **threshold** (llindar) és el valor que defineix la **tolerància de similitud** entre el valor d'un píxel candidat i la mitjana dels valors de la regió en construcció. És a dir, determina si un píxel adjacent serà agregat o no a la regió.

El criteri més utilitzat és de la forma:

$$|f(x,y) - \mu_{R_i}| \leq \Delta$$

On:

- $f(x,y)$ és el valor d'intensitat (o mitjana del color) del píxel candidat.
- μ_{R_i} és la mitjana dels valors dels píxels actuals de la regió R_i .
- Δ és el valor de llindar.

Implicacions del valor de Δ :

- **Valors baixos** de Δ generen **regions més petites** i homogènies, però poden provocar una **subsegmentació** (no arribar a incloure totes les àrees rellevants).
- **Valors alts** poden causar una **sobresegmentació**, unint zones que realment no pertanyen a la mateixa regió semàntica.

Per tant, l'ajust d'aquest paràmetre és crític. En la nostra implementació, aquest valor serà configurable per tal de poder adaptar-lo al contingut i variabilitat de les imatges a processar.

2.2 Veïnatge (4-connectat i 8-connectat)

El segon paràmetre fonamental és el tipus de **veïnatge** considerat durant l'exploració de la imatge. Això afecta com es defineixen els píxels veïns en el procés d'expansió de la regió:

- **4-connectat (A4)**: només es consideren veïns els píxels situats **a dalt, a baix, a l'esquerra i a la dreta** del píxel actual.
- **8-connectat (A8)**: es consideren veïns **tots els píxels adjacents**, incloses les diagonals.

Diferències pràctiques:

- El veïnatge A4 tendeix a generar **regions més angulars**, útils per evitar connexions diagonals no desitjades.
- El veïnatge A8 produeix **regions més compactes i suaus**, però pot facilitar la connexió entre estructures separades només per diagonals.

Per tal de valorar com afecta aquest paràmetre als resultats, en el nostre projecte es realitzarà la segmentació amb **ambdós tipus de veïnatge**, comparant-ne els resultats visuals i quantitius.

3. Implementació i proves amb matrius artificials

Per validar l'algorisme de segmentació mitjançant *Region Growing*, abans de treballar amb imatges reals, s'ha optat per implementar proves sobre matrius artificials. **Aquesta estratègia permet observar de forma controlada el comportament de l'algorisme, així com depurar i ajustar els paràmetres com el llindar (*threshold*) i la connectivitat.**

3.1 Representació d'imatges com a matrius

En el context del processament digital, una imatge es pot representar com una matriu on cada element correspon a la intensitat (en escala de grisos) o al valor de color (en RGB) d'un píxel.

- *Imatge en escala de grisos*: representada per una única matriu de valors entre 0 i 255.
- *Imatge RGB*: representada per tres matrius (una per canal R, G i B), que combinades formen una imatge en color.

Aquesta correspondència facilita el tractament computacional i és especialment útil per prototipar algorismes de segmentació.

3.2 Proves amb matrius en escala de grisos

Per començar, s'ha creat una matriu artificial de 5×5 amb valors d'intensitat que simulen dues regions diferenciades. L'algorisme implementat aplica el creixement de regions considerant:

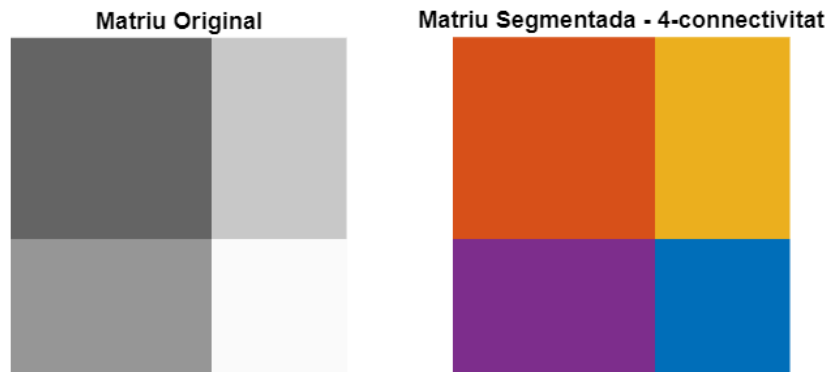
- Un llindar de similitud (*threshold*) que controla la tolerància entre valors.
- Connectivitat de 4 o 8 veïns, configurable.
- Mitjana actualitzada progressivament per cada regió.
- Ús d'una cua estàtica preallocada per optimitzar el rendiment (evitant costos de redimensionament dinàmic).

Fragment del codi (resumit):

```
matriu = double(matriu_1);
Resultat = zeros(size(matriu));
...
if abs(matriu(pixel_x, pixel_y) - mitjana) <= th
```

...
$$\text{mitjana} = (\text{mitjana} * (\text{comptador} - 1) + \text{nou_valor}) / \text{comptador};$$

Resultats obtinguts (imatge original i segmentada):



3.3 Proves amb matrius RGB (3 canals)

A continuació, es va estendre l'algorisme per tractar matrius RGB simulades. Per això, es generen tres matrius independents que representen els canals R, G i B. El criteri de similitud ja no es basa només en la diferència d'un valor, sinó en la distància euclidiana entre vectors de color.

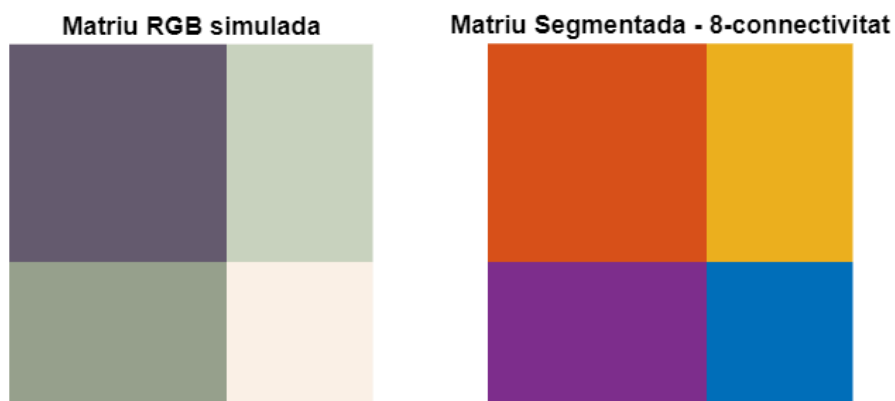
A més, es manté la mitjana de cada canal separatament, recalculada en cada pas.

Exemple de càlcul de distància RGB:

$$\text{distancia} = \sqrt{(R1 - R2)^2 + (G1 - G2)^2 + (B1 - B2)^2};$$

També s'ha utilitzat la mateixa estructura de cua estàtica i control de connectivitat per mantenir coherència i eficiència amb la versió en grisos.

Resultats obtinguts:



Consideracions d'implementació

- *Criteri d'eficiència*: l'ús de cues preallocades evita la penalització computacional que comporten les estructures dinàmiques en MATLAB.
- *Visualització*: s'ha utilitzat la funció **label2rgb** per convertir les etiquetes de regions en colors distintius.
- *Flexibilitat*: tots els paràmetres clau són fàcilment modificables (llindar, connectivitat), permetent una experimentació àgil.

4. Implementació de l'algorisme *Region Growing* a color

```
% Tipus de connectivitat (4 o 8)
Connectivitat = 8; % ← pots canviar-ho aquí

% Utilitzem una cua estàtica (preallocada) per evitar el
mida_cua = size(matriu, 1) * size(matriu, 2);
vector_x = zeros(mida_cua, 1);
vector_y = zeros(mida_cua, 1);
```

Es genera una variable **Connectivitat** on es pot escollir quin tipus de connectivitat es vol utilitzar en l'algorisme: **A4** o **A8**.

S'aprofita per crear una **cua estàtica** a partir de dos vectors, on en **vector_x** s'emmagatzema la posició **x** del píxel a analitzar i en **vector_y** la posició **y**.

```
% Bucle principal: mentre quedin píxels sense etiquetar
while ~isempty(x)|
    % Inicialitzem la mitjana RGB i sumatoris
    mitjana1 = matriu(x, y, 1);
    mitjana2 = matriu(x, y, 2);
    mitjana3 = matriu(x, y, 3);

    sumatori1 = mitjana1;
    sumatori2 = mitjana2;
    sumatori3 = mitjana3;

    comptador = 1;
    Resultat(x, y) = etiqueta;
```

S'assigna el valor numèric del píxel que conté el primer **zero** de la matriu a la variable **mitjana** (un per a cada matriu del RGB).

- Es crea la variable auxiliar **sumatori** per poder assignar la suma dels diferents valors de les mitjanes dels píxels a analitzar.
- S'etiqueta aquest píxel amb el valor inicial d'**etiqueta** i s'inicialitza un comptador per al càlcul de les mitjanes.

```

% Inicialitzem la cua amb el píxel inicial
inici = 1;
final = 1;
vector_x(final) = x;
vector_y(final) = y;

% Definim els offsets segons la connectivitat triada
if Conectivitat == 4
    offsets = [ -1, 0; % Esquerra
               0, 1; % Dalt
               1, 0; % Dreta
               0, -1]; % Baix
elseif Conectivitat == 8
    offsets = [ -1, 0; % Esquerra
               -1, 1; % Diagonal superior esquerra
               0, 1; % Dalt
               1, 1; % Diagonal superior dreta
               1, 0; % Dreta
               1, -1; % Diagonal inferior dreta
               0, -1; % Baix
               -1, -1]; % Diagonal inferior esquerra

```

Es creen dos **punters**, **inici** i **final**, inicialitzats en la primera posició de la cua, que serveixen per desplaçar-se per aquesta i comprovar que el punter **inici** no sigui més gran que **final**.

S'utilitzen **offsets** per estalviar repeticions de condicions (*if*) al codi, reduint el cost computacional.

```

% Inici del creixement de regió
while (inici <= final)
    % 'inici' indica el pròxim píxel a processar
    % 'final' indica on afegim nous píxels a la cua
    x1 = vector_x(inici);
    y1 = vector_y(inici);
    inici = inici + 1;

    for i = 1:size(offsets, 1)
        pixel_x = x1 + offsets(i, 1);
        pixel_y = y1 + offsets(i, 2);

        % Comprovem que estem dins dels límits de la imatge
        if (pixel_x > 0 && pixel_x <= size(matriu, 1) && pixel_y > 0 && pixel_y <= size(matriu,

```

També s'utilitzen dos vectors com a estructura de dades **FIFO** per guardar les coordenades dels píxels seleccionats per l'algorisme de *Region Growing* i processar-los posteriorment. En aquest cas, inicialment s'afegeix el primer píxel trobat a la matriu **Resultat**.

Es fa servir la variable **index** com a punter per desplaçar-se per la cua i comprovar que no superi mai la seva mida, ja que significaria que no hi ha posicions disponibles i no es podria continuar l'anàlisi de la imatge.

Es comprova que els veïns del píxel inicial estiguin dins dels límits de la matriu per evitar errors de compilació. Per dur a terme aquesta tasca, s'utilitzen **offsets** i un **bucle for** per verificar totes les possibilitats.

```
if (Resultat(pixel_x, pixel_y) == 0)
    % Calculem la distància euclidiana entre el color actual i la mitjana RGB
    distancia = sqrt( ...
        (matriu(pixel_x, pixel_y, 1) - mitjana1)^2 + ...
        (matriu(pixel_x, pixel_y, 2) - mitjana2)^2 + ...
        (matriu(pixel_x, pixel_y, 3) - mitjana3)^2 );

    if distancia < th
        % Etiquetem i afegim a la cua
        Resultat(pixel_x, pixel_y) = etiqueta;

        final = final + 1;
        vector_x(final) = pixel_x;
        vector_y(final) = pixel_y;

        % Actualitzem sumatoris i mitjanes
        sumatori1 = sumatori1 + matriu(pixel_x, pixel_y, 1);
        sumatori2 = sumatori2 + matriu(pixel_x, pixel_y, 2);
        sumatori3 = sumatori3 + matriu(pixel_x, pixel_y, 3);
        comptador = comptador + 1;

        mitjana1 = sumatori1 / comptador;
        mitjana2 = sumatori2 / comptador;
        mitjana3 = sumatori3 / comptador;
    end
end
```

Es calcula la **distància euclidiana** entre la mitjana de cada canal de color i el corresponent píxel de la matriu. Si aquest valor és menor que el **threshold (Th)** establert, s'assigna el píxel a la matriu **Resultat** com a part de la regió i s'afegeix a la cua per a ser analitzat posteriorment. La variable **sumatori** s'empra per optimitzar el càlcul de la mitjana per a tots els colors.

```

    % Busquem un nou píxel no etiquetat per començar una nova regió
    [x, y] = find(Resultat == 0, 1);
    if isempty(x) || isempty(y)
        break;
    end
    etiqueta = etiqueta + 1;
end

```

Si l'índex arriba a un punt on no troba cap coordenada a la cua (perquè ja no hi ha píxels etiquetats amb la mateixa regió), es busca un nou píxel sense etiquetar a la matriu **Resultat** (valor **0**). Si no es troba, el procés finalitza, ja que tota la imatge ha estat etiquetada. En cas contrari, s'incrementa la variable **etiqueta** per estudiar una nova regió de la imatge.

```

% Convertim les etiquetes en colors aleatoris per visualitzar les regions
matriu_final = label2rgb(Resultat, lines(max(Resultat(:))), 'b', 'shuffle');

% Mostrem les imatges
figure;
subplot(1, 2, 1), imshow(imatge), title('Imatge Original');
subplot(1, 2, 2), imshow(matriu_final), title(['Imatge Segmentada - ', num2str(Conectivitat), '-connectivitat']);

```

Quan ja no queden **zeros** a la matriu **Resultat**, l'algorisme surt del bucle *while*, converteix **Resultat** en una matriu d'etiquetes amb colors diferents i es mostra amb un **subplot** la comparació amb la imatge original.

5.Proves i temps d'execució (comparació de Thresholds)

coins.png

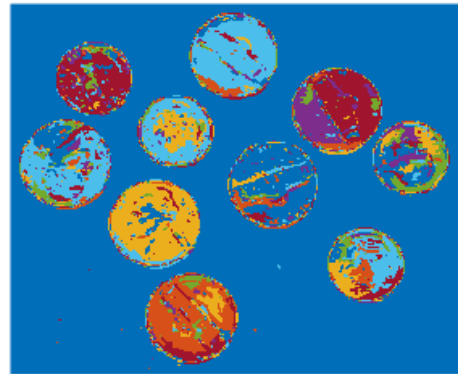
Th=15

Elapsed time is 0.235772 seconds.

Imatge Original



Imatge Segmentada - 4-connectivitat



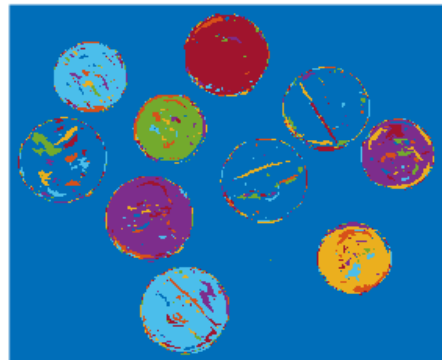
Th=25

Elapsed time is 0.188478 seconds.

Imatge Original



Imatge Segmentada - 4-connectivitat



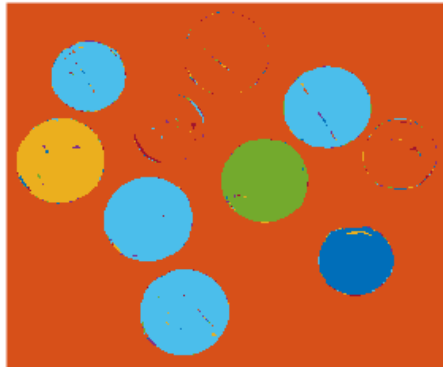
Th=50

Elapsed time is 0.147116 seconds.

Imatge Original



Imatge Segmentada - 4-connectivitat



Th=100

Elapsed time is 0.103059 seconds.

Imatge Original



Imatge Segmentada - 4-connectivitat



cameraman.tif

Th=15

Elapsed time is 0.224373 seconds.

Imatge Original



Imatge Segmentada - 8-connectivitat



Th=25

Elapsed time is 0.172064 seconds.

Imatge Original



Imatge Segmentada - 8-connectivitat



Th=50

Elapsed time is 0.129177 seconds.

Imatge Original



Imatge Segmentada - 8-connectivitat



Th=100

Elapsed time is 0.102614 seconds.

Imatge Original



Imatge Segmentada - 8-connectivitat



5.1. Segmentació en escala de grisos

En la imatge “*coins.png*”, els resultats no són excessivament equidistants, però es pot deduir que un *threshold* adequat estaria entre **25 i 50**. Això es deu al fet que, en aquests valors, la relació entre la identificació d'objectes i el nombre de regions es manté de manera més coherent.

- Amb **Th = 15**, hi ha una càrrega excessiva de regions, la qual cosa permet identificar els diferents contorns de les monedes, però provoca una **sobresegmentació**.
- En canvi, amb **Th = 100**, el nombre de regions es redueix considerablement, però alguns contorns de les monedes queden coberts i no es poden identificar tots els objectes de la imatge.

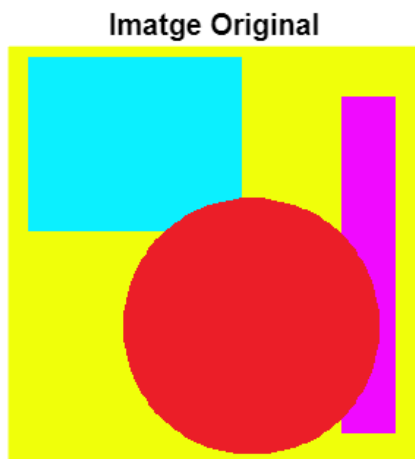
En la imatge “*cameraman.tif*”, els llinars mostrats segmenten de manera similar la persona amb la càmera i el trípod. No obstant això, la part del fons, que conté diferents formes, així com el terra amb l'herba, és el que marca la diferència entre els models generats.

- En aquest cas, el *threshold* que sembla complir millor la relació **identificació d'objectes - nombre de regions** és entre **15 i 25**.
- Això permet identificar tots els objectes de la imatge sense generar una sobresegmentació excessiva en la part de l'herba.

color.tif

Th=15

Elapsed time is 0.113349 seconds.

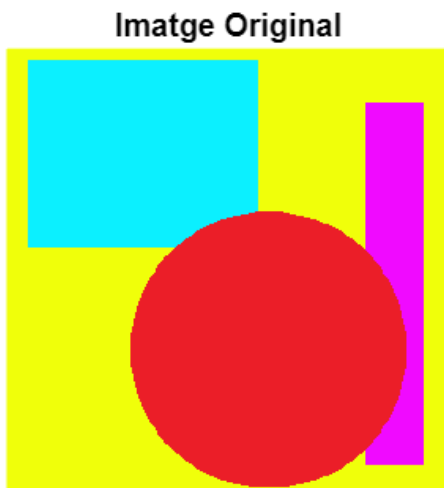


Imatge Segmentada - 4-connectivitat



Th=25

Elapsed time is 0.138226 seconds.



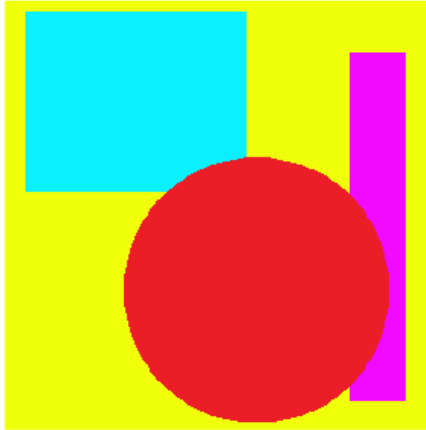
Imatge Segmentada - 4-connectivitat



Th=50

Elapsed time is 0.109254 seconds.

Imatge Original



Imatge Segmentada - 4-connectivitat



Th=100

Elapsed time is 0.104768 seconds.

Imatge Original



Imatge Segmentada - 4-connectivitat



gantrycrane.png

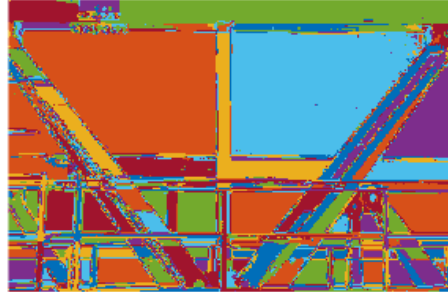
Th=15

Elapsed time is 0.421222 seconds.

Imatge Original



Imatge Segmentada - 8-connectivitat



Th=25

Elapsed time is 0.309772 seconds.

Imatge Original



Imatge Segmentada - 8-connectivitat



Th=50

Elapsed time is 0.197579 seconds.

Imatge Original



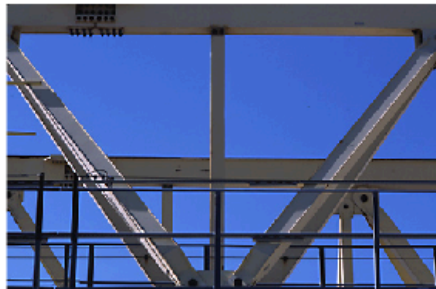
Imatge Segmentada - 8-connectivitat



Th=100

Elapsed time is 0.166535 seconds.

Imatge Original



Imatge Segmentada - 8-connectivitat



5.2. Segmentació en RGB

En la imatge “color.tif”, independentment del *threshold* seleccionat, les diferències són pràcticament indistingibles a causa de la claredat de la geometria dels objectes i

les seves tonalitats de color, que estan absents d'efectes visuals com ombres o llums.

En la imatge “*gantrycrane.png*”, com que està sotmesa a diversos efectes visuals, ombres i llums, el rang de color disponible augmenta. Això fa que, amb $Th=15$ i $Th=25$, es produeixi una **sobresegmentació**, ja que es detecten molts petits fragments. No obstant això, amb un *Threshold* més elevat ($Th=100$), aquest problema es resol, ja que l'interval de colors acceptats és més ampli i es redueix la fragmentació.

6.Problemes i possibles millores

maduixa.jpg

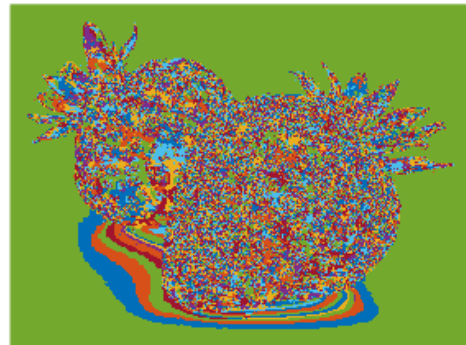
Th=15

Elapsed time is 4.697165 seconds.

Imatge Original



Imatge Segmentada - 4-connectivitat



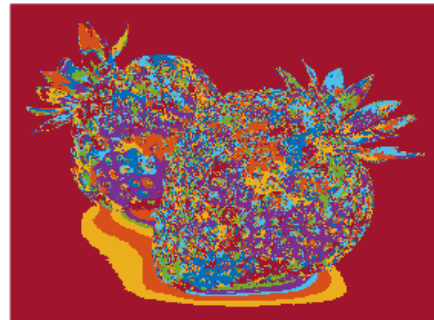
Th=25

Elapsed time is 2.792699 seconds.

Imatge Original



Imatge Segmentada - 4-connectivitat



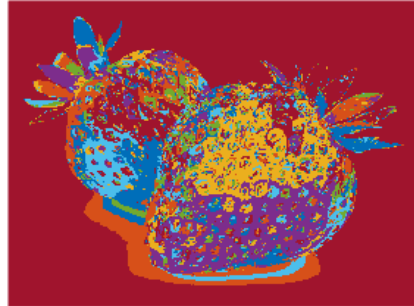
Th=50

Elapsed time is 1.073377 seconds.

Imatge Original



Imatge Segmentada - 4-connectivitat



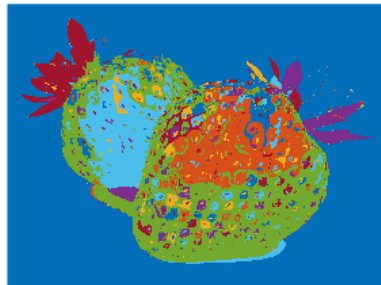
Th=100

Elapsed time is 0.450978 seconds.

Imatge Original



Imatge Segmentada - 4-connectivitat



6.1. Anàlisi de la segmentació en la imatge maduixa.jpg

En carregar la imatge de dues maduixes, s'observa un cas de **sobresegmentació** en la imatge resultant. Es generen molts fragments de diferents colors, la qual cosa suggereix una alteració significativa en la coloració de la imatge.

A més a més, el **temps d'execució** passa de ser pràcticament instantani en els exemples anteriors a una durada d'entre **0,4 i 5 segons**, depenent del *threshold*. Això significa que **l'eficiència de l'algorisme es redueix exponencialment** a mesura que es processa la imatge.

6.2. Causes principals de la sobresegmentació

Hi ha diverses **causes** que expliquen aquest fenomen:

- **Problemes de textura:** Les maduixes tenen petits aquenís (*les llavors visibles a la superfície*), que provoquen canvis subtils en el color. A causa de la seva elevada presència, l'algorisme detecta contínuament noves regions diferents.
- **Espai de color inadequat:** L'estudi s'ha realitzat en l'espai de color **RGB**, que no és òptim per comparar objectes reals. Potser seria més adequat utilitzar **HSV**, ja que separa millor la informació de lluminositat i color.
- **Presència de soroll:** Podria haver-hi diferents tipus de soroll en algunes parts de la imatge, la qual cosa afecta la segmentació.
- **Influència del *threshold*:** Com es pot observar en les imatges, el *threshold* afecta directament el grau de segmentació. En aquest estudi es dedueix que amb un *threshold* més elevat, la sobresegmentació és menys evident. Per exemple, amb **Th=100**, es poden identificar regions més àmplies de la maduixa sense tanta subdivisió provocada pels aquenís.



7. Conclusions

A partir dels resultats obtinguts en aquest projecte, podem concloure que l'algorisme Region Growing ofereix una eina potent per a la segmentació d'imatges digitals, però requereix un ajust acurat dels seus paràmetres principals per obtenir resultats òptims.

El valor del llindar (threshold) és un factor crucial que determina el grau de segmentació:

- Un llindar baix tendeix a produir regions petites i homogènies, útils per captar detalls però amb risc de sobresegmentació.
- Un llindar elevat pot simplificar excessivament la imatge, unint regions que haurien de romandre separades, i pot provocar subsegmentació.

En les proves realitzades:

- En imatges amb textures complexes, com "maduixa.jpg", la selecció del llindar és crítica. Llindars elevats redueixen la sobresegmentació provocada per petits canvis en el color.
- En imatges amb regions geomètriques clares, com "color.tif", els resultats són consistents independentment del llindar, demostrant robustesa en entorns simples.

Pel que fa al tipus de connectivitat, s'han observat diferències significatives:

- La connectivitat A4 és útil per evitar connexions diagonals no desitjades i ofereix regions més angulars, reduint el risc de sobresegmentació en llindars baixos.
- La connectivitat A8 produeix regions més compactes i suaus, adequada per imatges amb estructures variades o objectes propers entre si.

Finalment, la implementació eficient de cues estàtiques en MATLAB millora significativament el temps d'execució, encara que es percep una clara disminució d'eficiència en imatges de major complexitat o mida.

Per tant, per aconseguir resultats òptims, és essencial adaptar els paràmetres del Region Growing (threshold i connectivitat) a les característiques específiques de cada imatge.

8. Bibliografia

- [1] Gonzalez, R.C., Woods, R.E. (2017). Digital Image Processing. Pearson Education.
- [2] Shapiro, L.G., Stockman, G.C. (2001). Computer Vision. Prentice Hall.
- [3] Sonka, M., Hlavac, V., Boyle, R. (2014). Image Processing, Analysis, and Machine Vision. Cengage Learning.
- [4] Pal, N.R., Pal, S.K. (1993). "A review on image segmentation techniques." Pattern Recognition, 26(9), 1277-1294.
- [5] Adams, R., Bischof, L. (1994). "Seeded region growing." IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(6), 641-647.
- [6] Haralick, R.M., Shapiro, L.G. (1985). "Image segmentation techniques." Computer Vision, Graphics, and Image Processing, 29(1), 100-132.
- [7] Jain, A.K., Murty, M.N., Flynn, P.J. (1999). "Data clustering: a review." ACM Computing Surveys (CSUR), 31(3), 264-323.
- [8] MATLAB Documentation. (2023). MathWorks, Inc. Retrieved from <https://www.mathworks.com/help/images/index.html>.