

Informe laboratorio - Procesamiento de Imágenes Digitales

Transformaciones geométricas, corrección gamma e histogramas

Ruben Rodriguez

19 de junio de 2022

Índice

1. Desarrollo	1
2. Anexo	6

1. Desarrollo

1. Transformaciones geométricas

- a) Aplicar las transformadas geométricas a la imagen de la letra T. Se deberán realizar sobre la imagen de la letra T las transformaciones de: identidad, rotación, escalado, distorsión horizontal, distorsión vertical y traslación. Se deberán desarrollar las seis funciones en matlab que permitan aplicar las transformaciones geométricas. El informe debe contener imágenes que evidencien las salidas de los algoritmos implementados.
 - Código en el anexo.

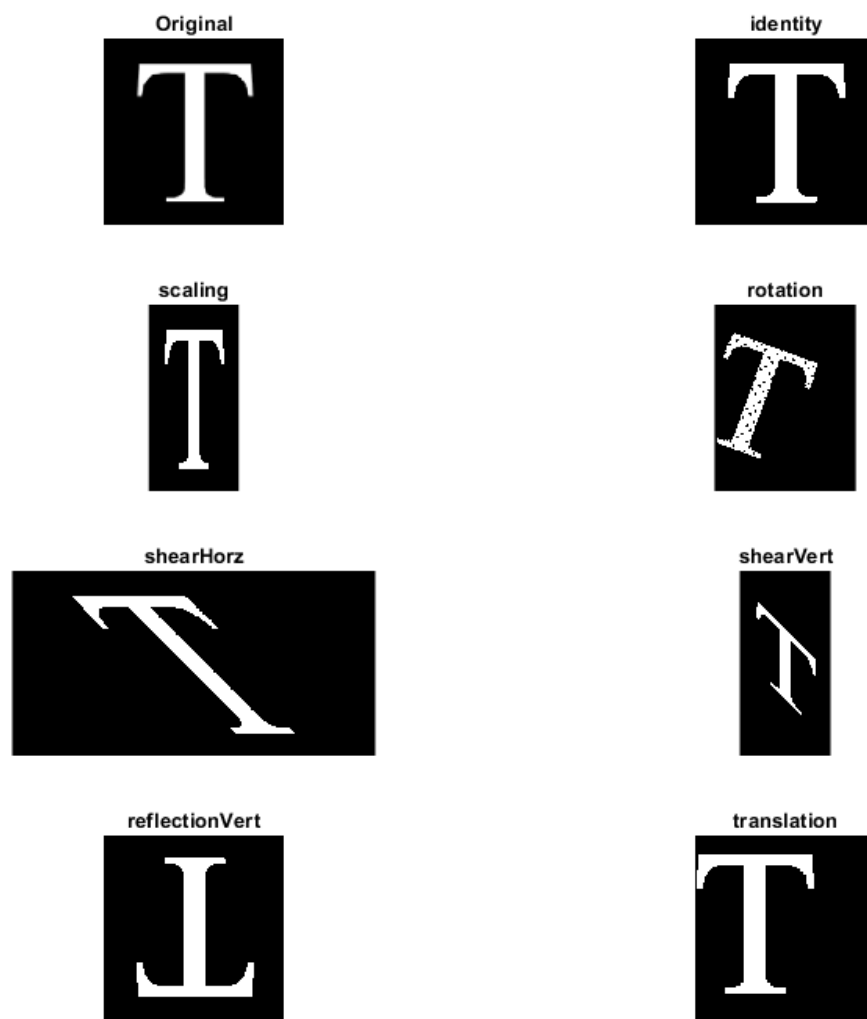


Figura 1: Transformaciones.

2. Corrección gamma

- a) Se debe realizar la corrección gamma a la imagen aérea. Adicionalmente, se debe generar la familia de curvas para diferentes valores de gamma y realizar un análisis respecto al efecto que tiene el parámetro gamma sobre la imagen transformada.

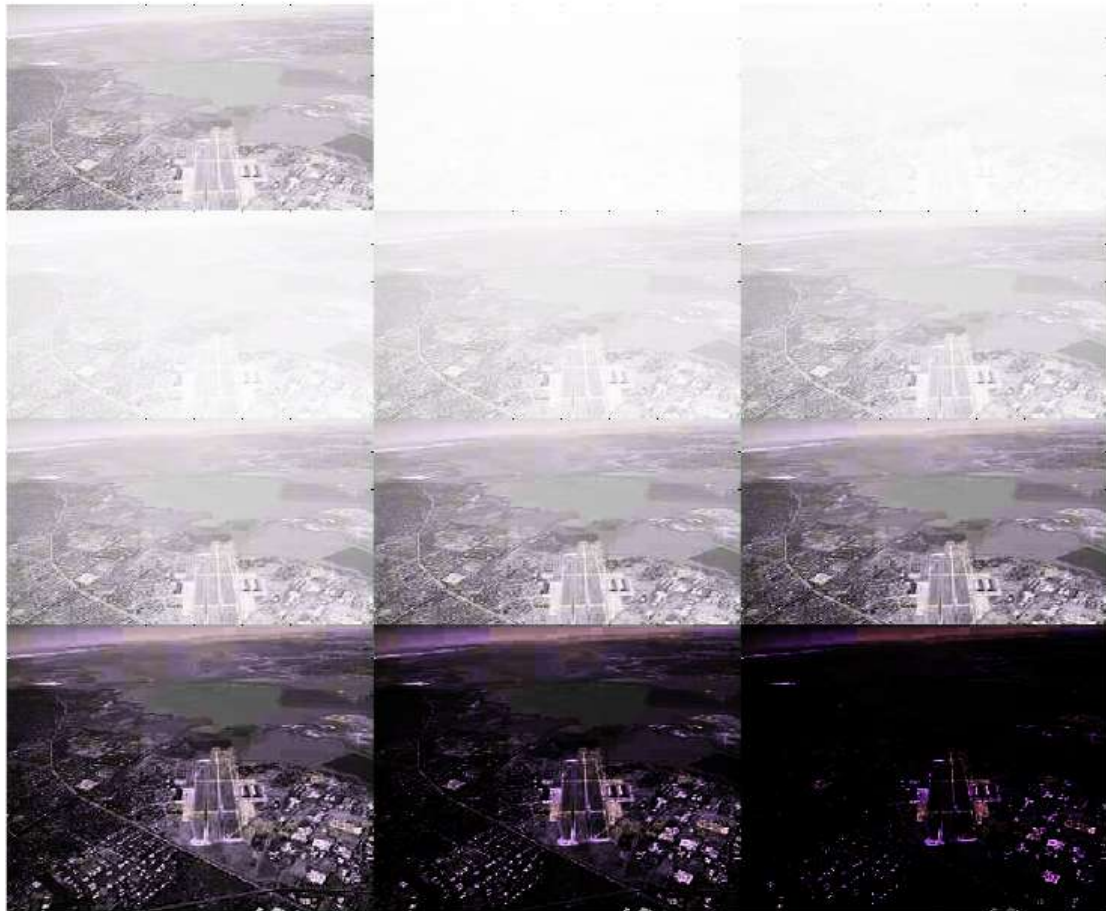


Figura 2: Original. Gamma: 0.04, 0.10, 0.20, 0.40, 0.67, 1.00, 1.50, 2.50, 5.00, 10.0, 25.0 .

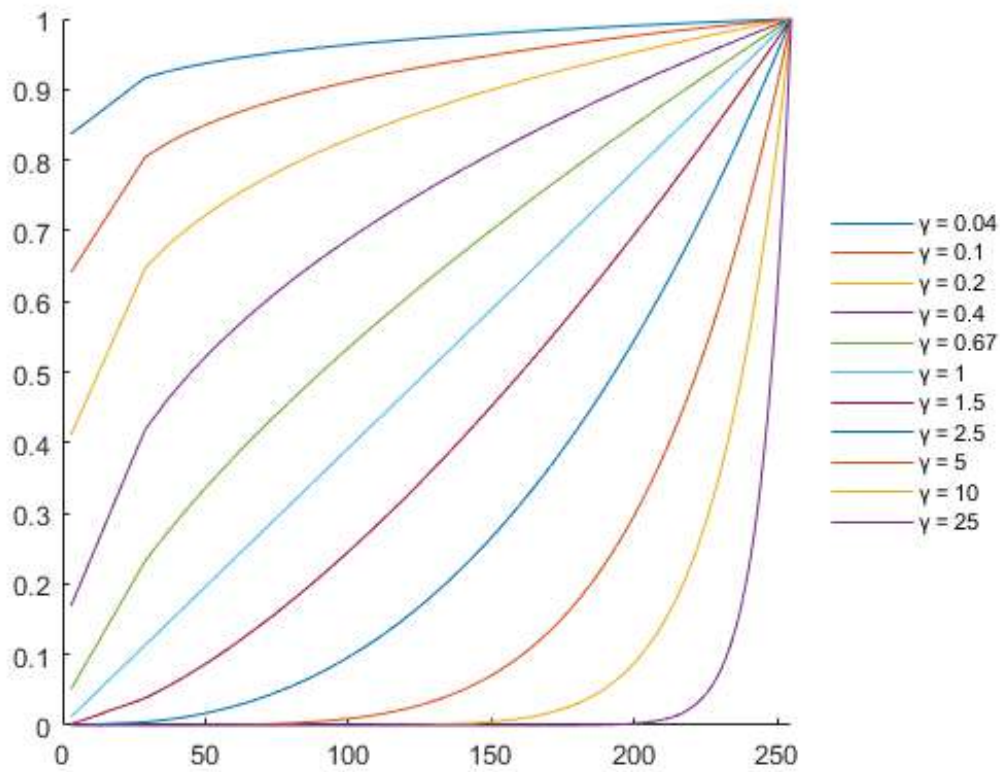


Figura 3: Familia de curvas.

Cuando gamma es pequeño la intensidad aumenta, por otro lado cuando gamma aumenta, la intensidad que toma el pixel va disminuyendo para cierto rango de valores, por ejemplo para gamma igual a 25 los valores de 0 a 220 aproximadamente su intensidad disminuye notablemente.

3. Histogramas

- a) Identificar imágenes con al menos 3 histogramas diferentes, responder ¿por qué se presentan este tipo de histogramas?

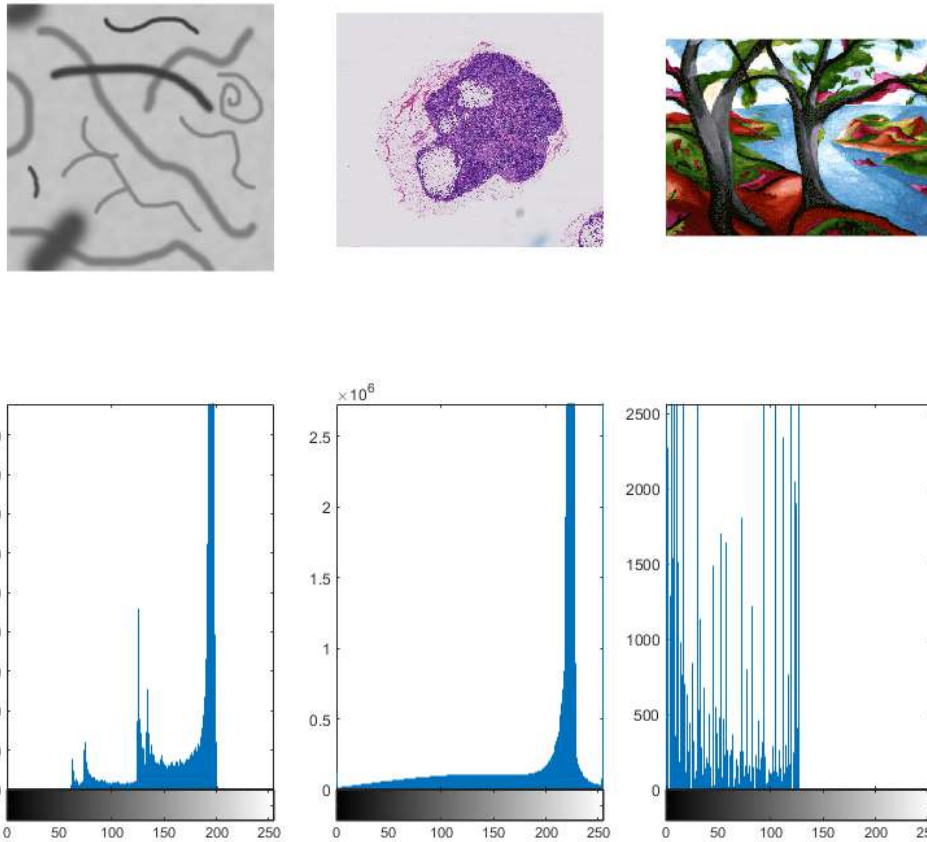


Figura 4: Imágenes y sus histogramas.

- De la primera columna tenemos valores que se encuentran en un rango de 50 a 200, lo que quiere decir que no cuenta con colores de baja y muy alta intensidad.
- En la segunda columna tenemos valores en todo el rango de 0 a 255 pero su mayor concentración se encuentra en 220 y 240.
- De la tercera columna tenemos que la imagen cuenta con un rango de colores de 0 a 128, en donde a medida que se acerca a los extremos de este rango algunos valores tienen una mayor frecuencia.

2. Anexo

Código de matlab.

```
1 figure
2 hold on
3 showaxes('boxoff')
4
5 % [haletterT, posletterT] = tight_subplot(4,2,[-0.0001 -0.0001]);
6
7 letterT = imread("imageT.png");
8 letterT = rgb2gray(letterT);
9
10 % axes(haletterT(1));
11 subplot(4,2,1);imshow(letterT), title("Original")
12
13 imgT1 = identity(letterT);
14 % axes(haletterT(2));
15 subplot(4,2,2);imshow(imgT1), title("identity")
16
17 imgT2 = scaling(2,1,letterT);
18 % axes(haletterT(3));
19 subplot(4,2,3);imshow(imgT2), title("scaling")
20
21 imgT3 = rotation(20,letterT);
22 % axes(haletterT(4));
23 subplot(4,2,4);imshow(imgT3), title("rotation")
24
25 imgT4 = shearHorz(1,letterT);
26 % axes(haletterT(5));
27 subplot(4,2,5);imshow(imgT4), title("shearHorz")
28
29 imgT5 = shearVert(1,letterT);
30 % axes(haletterT(6));
31 subplot(4,2,6);imshow(imgT5), title("shearVert")
32
33 imgT6 = reflectionVert(letterT);
34 % axes(haletterT(7));
35 subplot(4,2,7);imshow(imgT6), title("reflectionVert")
36
37 imgT7 = translation(-10,-50,letterT);
38 % axes(haletterT(8));
39 subplot(4,2,8);imshow(imgT7), title("translation")
```

```
40
41 % Gamma transformations
42 figure
43 aerea = imread("aerea.png");
44
45 gamma = [0.04 0.10 0.20 0.40 0.67 1.00 1.50 2.50 5.00 10.0 25.0];
46
47 figure
48 showaxes('boxoff')
49 [ha, pos] = tight_subplot(4,3,[-0.08 -0.08]);
50
51 axes(ha(1))
52 imshow(aerea);
53
54 for gammaIndex=1:length(gamma)
55     gammaImage = gammaTransformation(double(aerea)/255,1,gamma(gammaIndex));
56     axes(ha(gammaIndex+1));
57     imshow(gammaImage)
58 end
59
60
61 figure
62
63 hold on
64 for gammaIndex=1:length(gamma)
65     gammaImage = gammaTransformation(double(aerea)/255,1,gamma(gammaIndex));
66     uniqueValuesGammaImage = unique(gammaImage);
67     uniqueValuesOriginalImage = unique(aerea);
68     labelname = " = " + gamma(gammaIndex);
69     plot(uniqueValuesOriginalImage,uniqueValuesGammaImage,'DisplayName',labelname)
70     xlim([0 255])
71 end
72 lgd = legend('FontSize',7,'Box','off','Location','eastoutside');
73 hold off
74
75 figure
76
77 [ha, pos] = tight_subplot(2,3,[0.06 0.06]);
78
79 threads = imread("threads.png");
80 % subplot(2,3,1);
81 axes(ha(1));
```

```
82 imshow(threads)
83 % subplot(2,3,4);
84 axes(ha(4));
85 imhist(threads);
86
87 tumor_091R = imread("tumor_091R.tif");
88 % subplot(2,3,2);
89 axes(ha(2));
90 imshow(tumor_091R);
91 % subplot(2,3,5);
92 axes(ha(5));
93 imhist(tumor_091R);
94
95 [trees,map] = imread("trees.tif");
96 % subplot(2,3,3);
97 axes(ha(3));
98 imshow(trees,map);
99 % subplot(2,3,6);
100 axes(ha(6));
101 imhist(trees);
102
103 function [imgT] = gammaTransformation(image, c, gamma)
104
105     [sy,sx,sz] = size(image);
106     imgT = zeros(sy,sx,sz);
107     for z=1:sz
108         for y=1:sy
109             for x=1:sx
110                 r = image(y,x,z);
111                 s = power(c*r,gamma);
112                 imgT(y,x,z) = s;
113             end
114         end
115     end
116 end
117 -
118 function [imgT] = identity(image)
119
120     [sy,sx] = size(image);
121     imgT = zeros(sy,sx);
122
123     for v=1:sy
```



```
124     for w=1:sx
125         x = w;
126         y = v;
127         imgT(y,x) = image(v,w);
128     end
129 end
130
131 end
132 -
133 function [imgT] = scaling(cy,cx, image)
134
135     [sy,sx] = size(image);
136     imgT = zeros(sy,sx);
137
138     for v=1:sy
139         for w=1:sx
140             x = ceil(cx*w);
141             y = ceil(cy*v);
142             areaY = ceil(y-cy/2:y+cy/2);
143             areaX = ceil(x-cx/2:x+cx/2);
144             imgT(areaY,areaX) = image(v,w);
145         end
146     end
147
148 end
149 -
150 function [imgT] = rotation(degree,image)
151
152     [sy,sx] = size(image);
153     imgT = zeros(sy,sx);
154
155     for v=1:sy
156         for w=1:sx
157             x = ceil(w*cosd(degree) - v*sind(degree));
158             y = ceil(w*sind(degree) + v*cosd(degree));
159             if x > 0 && y > 0
160                 imgT(y,x) = image(v,w);
161             end
162
163
164
165         end
```

```
166     end
167
168 end
169 -
170 function [imgT] = shearHorz(alpha,image)
171
172     [sy,sx] = size(image);
173     imgT = zeros(sy,sx);
174
175     for v=1:sy
176         for w=1:sx
177
178             x = ceil(w + alpha*v);
179             y = v;
180
181             if x > 0
182                 imgT(y,x) = image(v,w);
183             end
184
185         end
186     end
187
188 end
189 -
190 function [imgT] = shearVert(beta,image)
191
192     [sy,sx] = size(image);
193     imgT = zeros(sy,sx);
194
195     for v=1:sy
196         for w=1:sx
197
198             x = w;
199             y = ceil(beta*w + v);
200
201             if y > 0
202                 imgT(y,x) = image(v,w);
203
204             end
205         end
206     end
207 end
```

```
208 -
209 % function [imgT] = reflectionHorz(image)
210 %
211 %     [sy,sx] = size(image);
212 %     imgT = zeros(sy,sx);
213 %
214 %     for v=1:sy
215 %         for w=1:sx
216 %             x = sx-w+1;
217 %             y = v;
218 %             imgT(y,x) = image(v,w);
219 %         end
220 %     end
221 % end
222 -
223 function [imgT] = reflectionVert(image)
224
225     [sy,sx] = size(image);
226     imgT = zeros(sy,sx);
227
228     for v=1:sy
229         for w=1:sx
230             x = w;
231             y = sy-v+1;
232             imgT(y,x) = image(v,w);
233         end
234     end
235 end
236 -
237 function [imgT] = translation(translation_y,translation_x,image)
238     [sy,sx] = size(image);
239     imgT = zeros(sy,sx);
240     for v=1:sy
241         for w=1:sx
242             x = w + translation_x;
243             y = v + translation_y;
244             if x > 0 && y > 0
245                 imgT(y,x) = image(v,w);
246             end
247         end
248     end
249 end
```