Captura y procesamiento digital de Señales e Imágenes

Trabajo Final

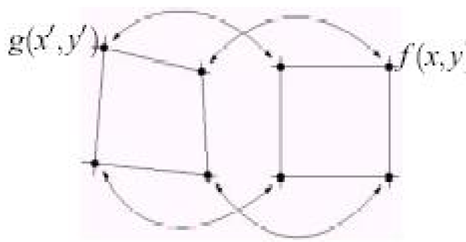
Warping

<u>Grupo</u>:

Burgener, Guillermo
Petrecca, Martín

Introducción:

El warping es una técnica que permite deformar imágenes modificando la ubicación espacial de los pixels. Para esto se necesita una función de transformación determinada:



$g(x',y')=T(f(x,y))$ o $f(x,y)=T^{1}(g(x',y'))$

- Figura 1 -

Nuestro trabajo se basa en dos aplicaciones de las técnicas de warping:

- Modificación de perspectiva
- Distorsión de imágenes

Modificación de perspectiva

Mediante técnicas de warping podemos efectuar un cambio en la forma de ver una imagen en cuanto a su perspectiva.
Para implementar este tipo de modificación utilizaremos distintas transformaciones que se incluyen dentro del warping no paramétrico (transformaciones que dan la correspondencia entre puntos de la imagen original y la distorsionada):

1. Afín
2. Proyectiva
3. Bilineal

1. Transformación Afín:
   Modelo:

$$u=ax+by+c$$
$$v=dx+ey+f$$

En notación matricial:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \text{ or } \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Permite una combinación de escalado, rotación y traslación; también distorsionar un cuadrado dentro de cualquier parelelogramo. Posee 6 grados de libertad. La transformación inversa se obtiene mediante la inversa de la matriz de 3x3. Funciona bien cuando las deformaciones siguen la figura de un triángulo, ya que mediante los 3 puntos obtenemos los 6 grados de libertad.

2. Transformación Proyectiva:

   Modelo:

$$u=(ax+by+c) / (gx+hy+i)$$
$$v=(dx+ey+f) / (gx+hy+i)$$

   En notación matricial:

$$\begin{bmatrix} uq \\ vq \\ q \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \text{ and } \begin{matrix} u = uq / q \\ v = vq / q \end{matrix}$$

   Si g=h=0, tenemos la transformación afín. Permite distorsionar un cuadrado dentro de cualquier cuadrilátero. Posee 8 grados de libertad (son necesarios 4 puntos). La transformación inversa se obtiene mediante la inversa de la matriz de 3x3. Puede ocurrir que no exista la inversa.

3. Transformación Bilineal:

   Modelo:

$$u=axy+bx+cy+d$$
$$v=exy+fx+gy+h$$

   En notación matricial:

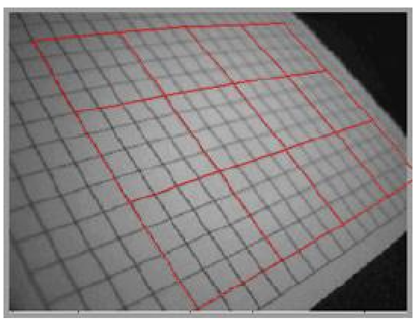
$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \end{bmatrix} \begin{bmatrix} xy \\ x \\ y \\ 1 \end{bmatrix}$$

   Si a=e=0, tenemos la transformación afín. Permite distorsionar un cuadrado dentro de cualquier cuadrilátero. Posee 8 grados de libertad (son necesarios 4 puntos). La transformación inversa no se obtiene mediante la inversa de la matriz de coeficientes.

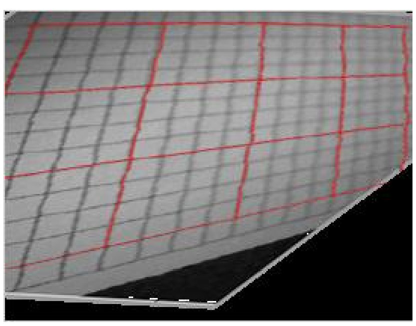
Implementación Práctica:

Se obtuvieron los puntos de referencia sobre la imagen mediante el uso de la función improfile. Con estos puntos y los puntos de destino (que forman un rectángulo –ver figura 1-) se calcularon los coeficientes, para esto se usó notación matricial y el operador \. Con los coeficientes obtenidos, se realizó el mapeo espacial; como los valores obtenidos eran reales, se procedió a interpolarlos para obtener los valores correspondientes de intensidad de gris. El tipo de interpolación usado fue la bicúbica. Con esta interpolación se obtiene un buen resultado (bajo pixelado) a costa de un alto costo computacional en el cálculo de la misma.
La dificultad que encontramos es la lentitud de procesamiento de Matlab (lenguaje interpretado) producido por los ciclos for (para el cálculo de las transformaciones espaciales). Para solucionar este problema se podría utilizar alguna notación especial de Matlab que no conocemos, o bien, programar con otro lenguaje.

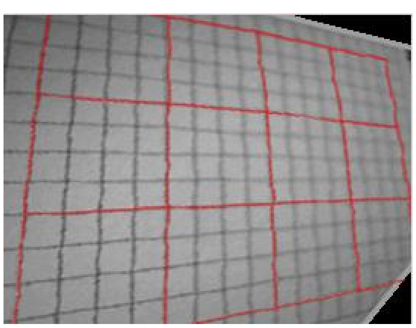
Resultados:



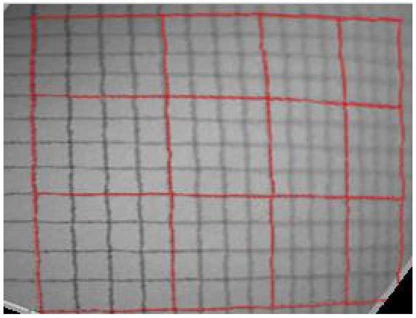
Original



Afín



Proyectiva



Bilineal

Distorsión de imágenes

Las técnicas de warping con respecto a distorsión de imágenes son muy variadas en cuanto a costo y complejidad dependiendo de la deformación que se desee practicar.
Nuestro estudio se limitó simplemente a hacer una transformación de coordenadas, modificar sus valores, y luego volver al sistema de coordenadas original, con lo que se obtuvo el objetivo deseado (deformar la imagen).

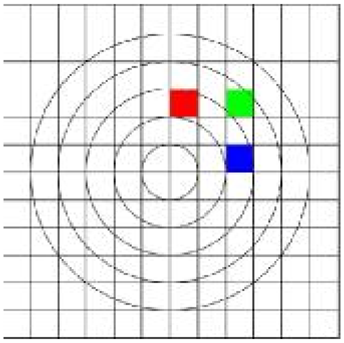
Implementación Práctica:

Partiendo de la imagen original (en el sistema cartesiano), se calculó el valor de coordenadas de cada píxel en el sistema polar (radio – ángulo). Luego, modificamos el valor del radio, con el cual calculamos los nuevos valores de x e y.
Al igual que en el caso anterior, usamos la interpolación bicúbica para obtener los valores de intensidad de gris correspondiente a cada píxel.
Pasos:

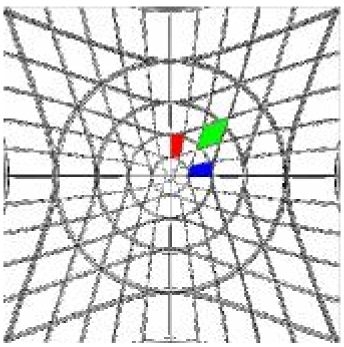
- Normalización a [-1,1]
  $x = 2\, i \,/\, ancho - 1$
  $y = 2\, j \,/\, alto - 1$
- Rectangular a polar:
  $radio = sqrt(x^2 + y^2)$
  $ang = atan2(y,x)$
- Modificación del valor de r:
  $radio = sqrt(radio)$
- Polar a rectangular:
  $x = radio\ cos(ang)$
  $y = radio\ sin(ang)$
- Desnormalización
  $i = (x+1) / 2 * ancho$
  $j = (y+1) / 2 * alto$

Nota: Los pasos de normalización y desnormalización se deben a que en la transformación de coordenadas Polares a Cartesianas intervienen términos con senos y cosenos

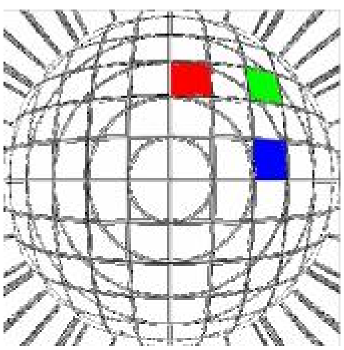
Resultados:



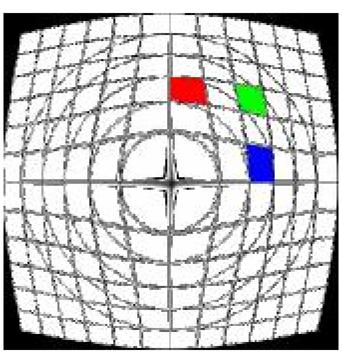
Original



r=sqrt(r)



r=2*asin(r)/pi



r=r^1.5

Conclusiones:

Modificaciones de Perspectiva:

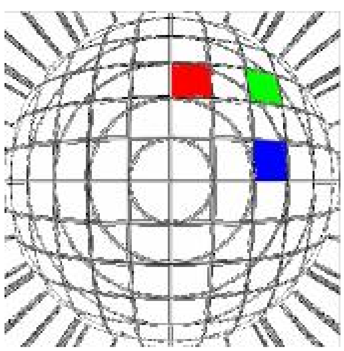
Como puede observarse, la transformación que funciona mejor cuando se desea pasar a una vista frontal es la bilineal, pero tiene el defecto de mantener las distancias con la misma relación que tenían en la imagen en perspectiva(esto puede observarse en las ventanas del edificio), para solucionar este problema deberíamos hacer una transformación que tenga en cuenta la relación distancia entre los píxeles en perspectiva y los frontales.

Bilineal

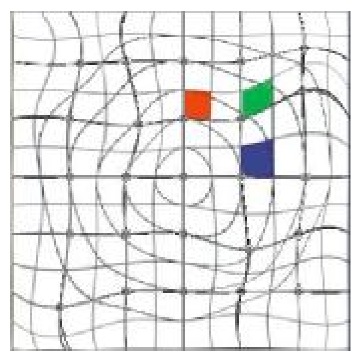
Bilineal Corregida

Distorsión de Imágenes:

Como podemos apreciar en los resultados, se pueden obtener deformaciones satisfactorias mediante este sencillo modelo de cambio y modificación de coordenadas. Dicho modelo se encuentra restringido a un mapeo fijado por las fórmulas usadas en la trasformación (ej. Todos los píxeles se modifican según r = sqrt(r)), mientras que en otros modelos cada vecindad de píxeles es afecta por distintas transformaciones.



Modelo de dependencia Global



Modelo de dependencia por Vecindad