Transformaciones geométricas

Extracto de "Visión por Computador. Fundamentos y métodos". Arturo de la Escalera. Prentice Hall, 2001

Copia para el alumno con fines didácticos

4.4.2. Transformaciones geométricas.

Las transformaciones geométricas modifican las relaciones espaciales entre los pixeles. Se necesitan dos tipos distintos de algoritmos. El primero es el que determina la relación entre las coordenadas de la imagen original y la imagen resultante. Como caso general, las transformaciones de números enteros no tienen por qué dar enteros, así que será necesario un algoritmo de interpolación que determine el nivel de gris de la imagen final a partir de uno o varios pixeles de la imagen original.

Transformaciones

<u>Traslación.</u>

Si se quiere trasladar el origen de la imagen se aplica la ecuación:

$$xf = xi + xo$$

$$yf = yi + yo$$

Que en coordenadas homogéneas es:

$$\begin{bmatrix} xf \\ yf \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & xo \\ 0 & 1 & yo \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} xi \\ yi \\ 1 \end{bmatrix}$$

<u>Magnificación</u>

$$xf = \frac{xi}{a}$$

$$yf = \frac{yi}{b}$$

o lo que es lo mismo:

$$\begin{bmatrix} xf \\ yf \\ 1 \end{bmatrix} = \begin{bmatrix} 1/a & 0 & 0 \\ 0 & 1/b & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} xi \\ yi \\ 1 \end{bmatrix}$$

<u>Rotación respecto al origen.</u>

$$\begin{bmatrix} xf \\ yf \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\operatorname{sen}\theta & 0 \\ \operatorname{sen}\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} xi \\ yi \\ 1 \end{bmatrix}$$

<u>Rotación respecto a un punto cualquiera.</u>

Aquí se manifiesta la ventaja de la notación matricial, ya que es igual a trasladar la imagen al nuevo centro, rotar y deshacer la traslación:

$$\begin{bmatrix} xf \\ yf \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & xo \\ 0 & 1 & yo \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\operatorname{sen}\theta & 0 \\ \operatorname{sen}\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -xo \\ 0 & 1 & -yo \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} xi \\ yi \\ 1 \end{bmatrix}$$

Interpolación.

La interpolación es necesaria porque, en el caso general para un píxel, no se va a obtener de la imagen original un valor entero en la imagen destino. De hecho, a varios pixeles de la imagen original les corresponde el mismo en la imagen destino. Una mayor rapidez o exactitud dependerá del método de interpolación que se elija. Con carácter general, la transformación no se realiza de la imagen inicial a la final, sino al revés; dado un píxel de la imagen final se localizan los pixeles de la inicial, que se utilizan para calcular el nuevo nivel de gris. La interpolación puede entenderse como una nueva digitalización de la imagen continua para obtener la discreta.

$$f(x,y)=\sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty} g(i,j)\ h(x-i,y-j)$$

Donde:

- g(x,y) es la imagen continua.
- f(x,y) es la imagen discreta.
- h(x,y) representa la interpolación.

Si h(x,y) fuera la delta de Dirac, la imagen inicial y la final serían la misma.

Vecino más próximo.

Es el método más sencillo. A partir de la transformación inversa se redondean los valores. Esto es lo mismo que ver cuál es el píxel más cercano. La función h(x,y) sería:

$$h(x)=\begin{cases}1 & si\ 0<|x|<0.5\\ 0 & en\ cualquier\ otro\ caso\end{cases}$$

Interpolación Bilineal.

Se denomina así porque se supone que la variación entre un píxel y el siguiente sigue un plano (figura 4.17):

$$f(x,y) = ax+by+cxy+d$$

Si una de las coordenadas -x o y- permanece constante, la ecuación anterior es la de una recta. Los coeficientes a,b,c,d deben fijarse de manera que los valores de la función coincidan con los cuatro pixeles vecinos. Las ecuaciones de las rectas son:

$$f(x,0) = f(0,0)+(f(1,0)-f(0,0))\,x$$

$$f(x,1) = f(0,1)+(f(1,1)-f(0,1))\,x$$

con estos dos puntos el valor respecto al otro eje será:

$$f(x,y) = f(x,0)+(f(x,1)-f(x,1))\,y$$

y despejando obtenemos la ecuación final:

$$f(x,y) =(f(1,0)-f(0,0))\,x+(f(0,1)-f(0,0))\,y+(f(1,1)+f(0,0)-f(0,1)-f(1,0))\,x\,y+f(0,0)$$

Los resultado son mejores que con el método anterior aunque, al usar una recta, se produce un desdibujado de los objetos. Si se usase la fórmula de la convolución la máscara vendría definida por:

$$h(x)=\begin{cases}1-|x| & si\ 0<|x|<1\\ 0 & en\ cualquier\ otro\ caso\end{cases}$$

Interpolación bicúbica.

Para evitar el inconveniente anterior se utiliza la interpolación cúbica, en la que se toman en consideración 16 puntos de la imagen. La convolución viene definida por:

$$h(x)=\begin{cases}1-2|x|^2+|x|^3 & si\ 0<|x|<1\\ 4-8|x|+5|x|^2-|x|^3 & si\ 1<|x|<2\\ 0 & en\ cualquier\ otro\ caso\end{cases}$$

<u>Caso general de magnificación.</u>

Cuando se quiera magnificar una imagen *f*(*x*,*y*) de dimensiones NxM, para obtener otra *g*(*x*,*y*) de dimensiones pNxpM, se puede aplicar el siguiente algoritmo.

Primero se obtiene la imagen *g*(*x*,*y*) utilizando la fórmula:

$$g(x,y)=\begin{cases}f(i,j) & \text{si } \frac{\text{x}}{\text{p}}=i\ \frac{\text{y}}{\text{p}}=j\\ 0 & \text{en cualquier otrocaso}\end{cases}$$

similar a la del vecino más próximo; después se convoluciona *g*(*x*,*y*) p-veces con la máscara:

$$\begin{bmatrix}\frac{1}{4} & \frac{1}{2} & \frac{1}{4}\\ \frac{1}{2} & 1 & \frac{1}{2}\\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4}\end{bmatrix}$$

Los resultados pueden verse en la figura 4.18

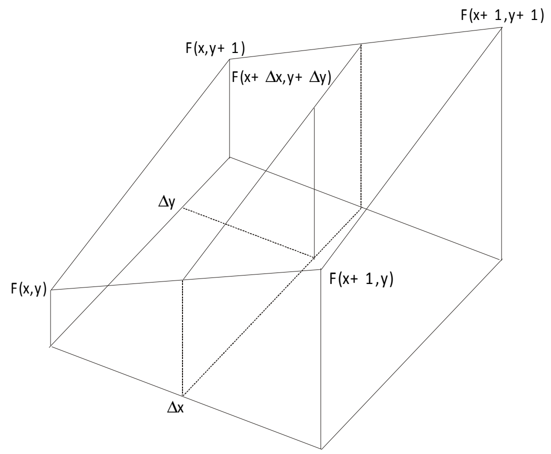
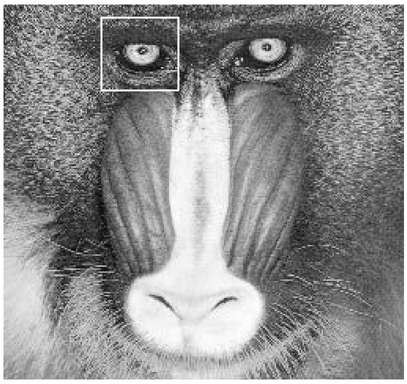
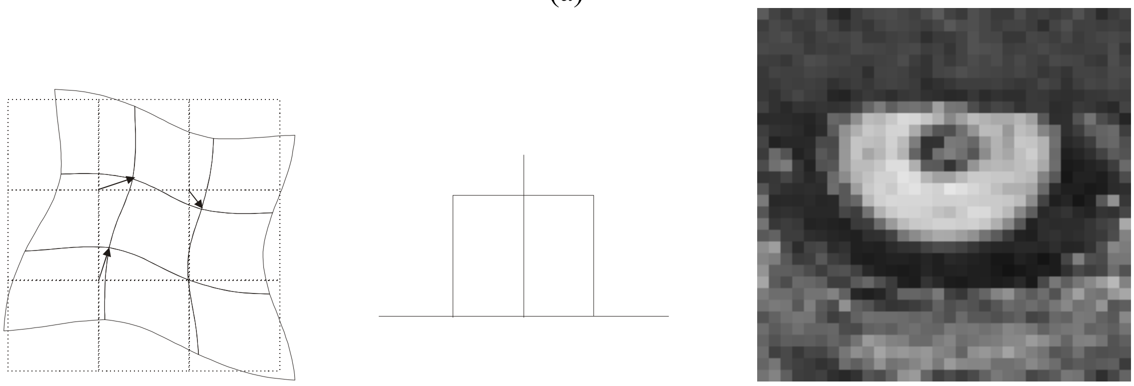


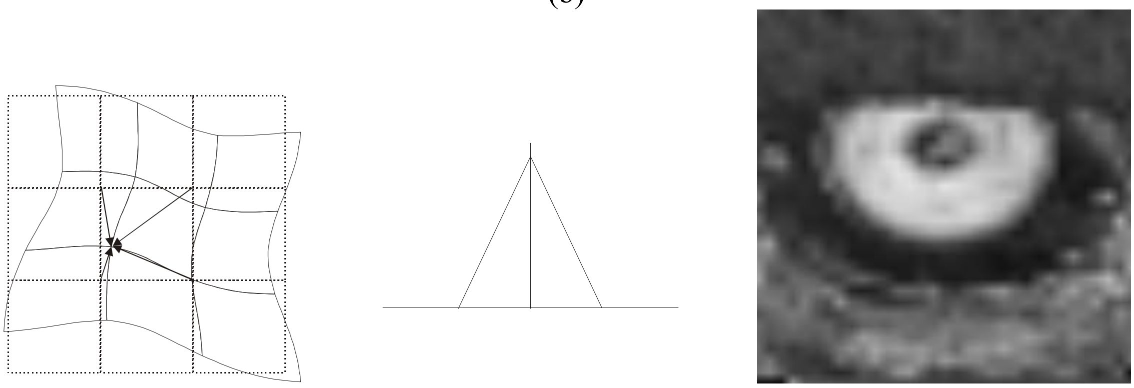
Figura 4.17. Interpolación bilineal.



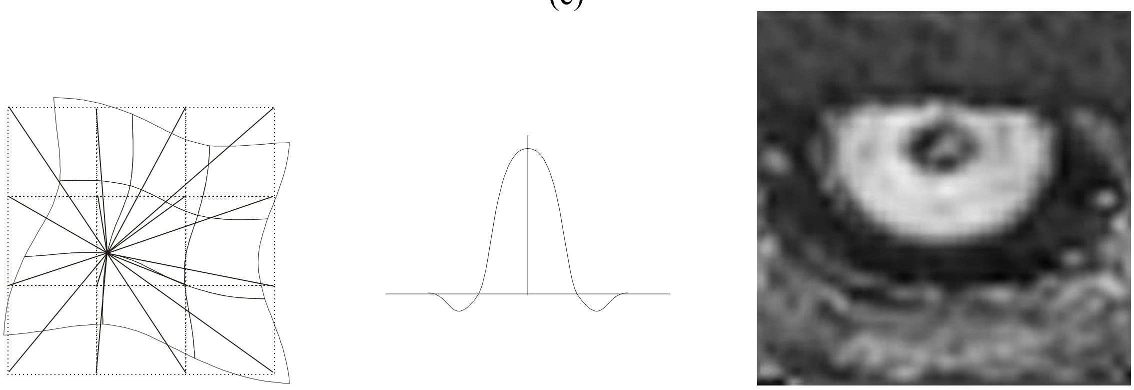
(a)



(b)



(c)



(d)

Figura 4.18. Interpolación mediante convoluciones. (a) imagen original (b) vecino más próximo (c) bilineal (d) bicúbica