MAC0317/MAC5920

Introdução ao Processamento de Sinais Digitais

Seção 3.6: A DCT 2D

Motivações para a DCT bidimensional

- permitir a compressão de imagens eliminando as descontinuidade de borda da DFT;
- a ideia é usar o mesmo princípio de espelhamento do caso 1D, mas nos eixos x e y;
- podemos definir a transformada 2D usando a transformada 1D (DCT das linhas / DCT das colunas)

In [3]:
```
# função auxiliar: converte imagem colorida para cinza
def rgb2gray(rgb):
    fil = [0.299, 0.587, 0.144]
    return np.dot(rgb, fil)
# Carrega a imagem da internet e converte para nível de cinza
url = "http://sutherncharm.files.wordpress.com/2009/09/double-ferris.jpg"
M = rgb2gray(imread(urlopen(url).read()))
MM_upper = np.hstack((M, np.flip(M,1)))
MM = np.vstack((MM_upper, np.flip(MM_upper,0)))
plt.figure(figsize=(5,8)); plt.imshow(MM, cmap='gray'); plt.axis("off")
plt.title("Figura 3.11: Reflexões da imagem da roda-gigante"); plt.show()
```

Figura 3.11: Reflexões da imagem da roda-gigante

Considerando $DCT(x) = C_N x$ e $IDCT(C) = C_N^T C$ podemos definir uma transformação em 2D para uma matriz $A \in M_{M\times N}(\mathbb{C})$ qualquer assim

$$\hat{A} = DCT(A) = C_M A C_N^T (\in M_{M\times N}(\mathbb{C}))$$

cuja inversa será dada naturalmente por

$$A = IDCT(\hat{A}) = C_M^T \hat{A} C_N$$

Lembrando dos coeficientes das matrizes C_M e C_N^T, temos:

$$(C_M)_{k,m} = \begin{cases} \sqrt{\frac{1}{M}} & \text{se } k = 0 \\ \sqrt{\frac{2}{M}} \cos\left(\frac{\pi k\left(m+\frac{1}{2}\right)}{M}\right) & \text{se } k = 1, 2, \ldots, M-1 \end{cases}$$

$$(C_N)_{n,l}^T = \begin{cases} \sqrt{\frac{1}{N}} & \text{se } l = 0 \\ \sqrt{\frac{2}{N}} \cos\left(\frac{\pi l\left(n+\frac{1}{2}\right)}{N}\right) & \text{se } l = 1, 2, \ldots, N-1 \end{cases}$$

Assim

$$\hat{A}_{k,l} = u_k v_l \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mn} \cos\left(\frac{\pi k \left(m + \frac{1}{2}\right)}{M} \right) \cos\left(\frac{\pi l \left(n + \frac{1}{2}\right)}{N} \right)$$

onde

$$u_k = \begin{cases} \sqrt{\frac{1}{M}} & \text{se } k = 0 \\ \sqrt{\frac{2}{M}} & \text{se } k \neq 0 \end{cases} \qquad v_l = \begin{cases} \sqrt{\frac{1}{N}} & \text{se } l = 0 \\ \sqrt{\frac{2}{N}} & \text{se } l \neq 0 \end{cases}$$

Observação: $A \in M_{M\times N}(\mathbb{R}) \Leftrightarrow \hat{A} \in M_{M\times N}(\mathbb{R})$.

Formas básicas 2D para a DCT

Note que é possível escrever

$$\hat{A}_{k,l} = \left(A, C_{k,l}\right)$$

onde

$$\left(C_{k,l}\right)_{m,n} = u_k v_l \cos\left(\frac{\pi k \left(m + \frac{1}{2}\right)}{M}\right) \cos\left(\frac{\pi l \left(n + \frac{1}{2}\right)}{N}\right).$$

Portanto, a DCT pode ser vista como a mudança de base da matriz A para a base formada pelas matrizes $\{C_{k,l} \mid k = 0, \dots, M-1, \quad l = 0, \dots, N-1\}$.

In [4]:

```
M=4;N=8; fig, ax = plt.subplots(M,N,figsize=(15,5))
i = np.arange(0,M); j = np.arange(0,N); i, j = np.meshgrid(i,j,indexing='ij')
for k in range(M):
  for l in range(N):
    f =  m.sqrt((1+(k>0))/M)*m.sqrt((1+(l>0))/N)*np.cos((m.pi/M)*k*(i+0.5))*np.cos((m.pi/N)*l*(j+0.5))
    ax[k][l].imshow(f,cmap='gray',vmin=-m.sqrt(2/M)*m.sqrt(2/N),vmax=+m.sqrt(2/M)*m.sqrt(2/N));ax[k][l].axis("off")
    ax[k][l].set_title(r"$\mathcal{{C}}_{{{}{}}}$".format(k,l))
fig.suptitle(r'Formas de onda básicas para imagens {}×{}'.format(M,N), fontsize=16, y=1)
plt.show()
```