

Super-resolución

Martín Santa María y Sebastián Rojas Fredini

Trabajo práctico final de “Captura y Procesamiento Digital de Imágenes”, II-FICH-UNL.

Resumen—Este trabajo presenta una técnica contenida dentro del conjunto de técnicas de super-resolución. La misma consiste en generar una imagen de alta resolución a partir de un conjunto de imágenes de baja resolución rotadas y trasladadas en el espacio. Esta técnica es alternativa a los procedimientos de interpolación y demuestra lograr mejores resultados para un conjunto de imágenes suficientemente grande.

Dentro del conjunto de técnicas de super-resolución, se utiliza el modelo teórico presentado en [1] y se muestran resultados obtenidos para una imagen de prueba propia.

Palabras clave—super-resolución, procesamiento de imágenes, gradiente conjugado, registración de imágenes, reconstrucción de imágenes.

I. INTRODUCCIÓN

EXISTEN aplicaciones que requieren de imágenes de alta resolución para su correcto desempeño, sin embargo, la obtención de dichas imágenes se encuentra limitada por las características del dispositivo de captura. Muchas veces, la construcción de dispositivos que permitan capturar con una resolución suficiente es costosa o impracticable.

Los métodos de super-resolución explotan el contenido de información de un conjunto de imágenes de un mismo fenómeno aprovechándose del alias contenido en cada una. Se requiere que el punto de captura sea diferente en cada imagen de manera que existan desplazamientos sub-píxel y los sensores capturen información que en otras imágenes representa alias [2].

II. MODELO DE OBSERVACIÓN

Se utiliza un modelo de observación que describe el proceso de captura de una imagen (Fig. 1). Supongamos $o(x, y)$ una imagen continua que es muestreada a $z(n_1, n_2)$ con una frecuencia que satisface el teorema de Nyquist. El modelo considera que las p imágenes de baja resolución (LR) $y_k(n_1, n_2)$ son versiones rotaciones, traslaciones y sub-muestreadas de $z(n_1, n_2)$.

Además, se considera un efecto de desenfoque introducido por el tamaño finito de los sensores y el efecto de la óptica. Estos efectos son modelados por una función

de dispersión (PSF) $h_d(n_1, n_2)$.

El método consiste en estimar estas transformaciones para obtener z , a partir de las p imágenes y_k . No se consideran oclusiones parciales, cambios de escala, ni cambios en la perspectiva. Otros modelos pueden utilizarse para incluirlas, así como también, cambios temporales de la imagen de alta resolución (HR) [3] [4].

El modelo expresado en notación lexicográfica se define como

$$y_m = \sum_{r=1}^N w_{m,r} z_r + \eta_m \quad (1)$$

donde

$$\mathbf{y} = [\mathbf{y}_1, \dots, \mathbf{y}_p] \quad (2)$$

$$\mathbf{y}_k = [y_{1,1}^k, y_{2,1}^k, \dots, y_{N_1, N_2}^k] \quad (3)$$

$$\mathbf{z} = [z_1, \dots, z_N] = [z_{1,1}, z_{2,1}, \dots, z_{L_1 N_1, L_2 N_2}] \quad (4)$$

Con (L_1, L_2) el factor de sub-muestreo, z la imagen HR, y_k la k -ésima imagen LR, $M = N_1 \times N_2$ el tamaño en píxeles de las imágenes LR y $N = L_1 M_1 \times L_2 M_2$. La matriz $w_{m,r}$ representa la contribución del k -ésimo píxel de HR en el m -ésimo píxel de \mathbf{y} . η_m es el m -ésimo píxel de una imagen con ruido gausseano.

Se utilizará la notación lexicográfica en lo que resta del trabajo, salvo que se especifique lo contrario.

El procedimiento se resume en dos etapas. En la primera se registran las imágenes \mathbf{y}_k y en la segunda se construye la imagen HR a partir de las imágenes LR registradas. Los procedimientos de registración y reconstrucción se explican más adelante.

A. Función de dispersión de punto

La PSF modela el efecto de los sensores y la óptica en la captura de la imagen. El efecto producido puede modelarse como una convolución con la respuesta al impulso de un único sensor (Fig.2).

La PSF se calcula por la convolución de una ventana

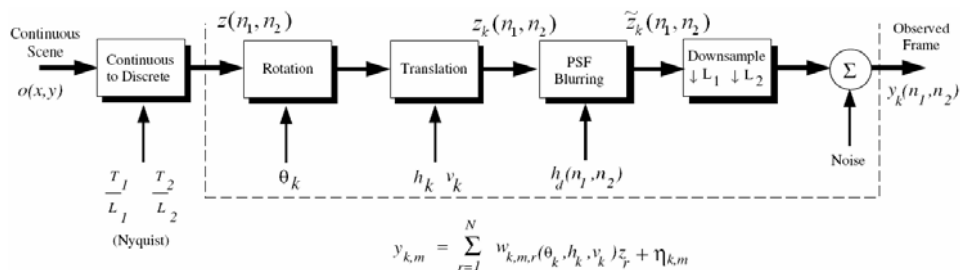


Fig. 1: Modelo de observación discreto

rectangular de $L_1 \times L_2$ dada por

$$\begin{aligned} h_d(n_1, n_2) &= \frac{T_1 T_2}{L_1 L_2} h_c\left(\frac{n_1 T_1}{L_1}, \frac{n_2 T_2}{L_2}\right) \\ h_c(n_1, n_2) &= \mathfrak{F}^{-1}\{H(u, v)\} \\ H(u, v) &= D(u, v) H_o(u, v) \end{aligned} \quad (5)$$

donde

$$D(u, v) = \text{sync}(au, bv) \quad (6)$$

$$H_o(u, v) = \begin{cases} \frac{2}{\pi} \left[\cos^{-1}\left(\frac{\rho}{\rho_c}\right) - \left(\frac{\rho}{\rho_c}\right) \sqrt{1 - \left(\frac{\rho}{\rho_c}\right)^2} \right] & \rho < \rho_c \\ 0 & \rho \geq \rho_c \end{cases} \quad (7)$$

(a, b) son las dimensiones de la región activa de los sensores, (T_1, T_2) son los períodos de muestreo, o lo que es lo mismo, la separación entre sensores (Fig. 3), $\rho = \sqrt{u^2 + v^2}$, $\rho_c = \frac{1}{\lambda \times f\text{-number}}$ la frecuencia de corte del óptica, λ la longitud de onda de la óptica, $f\text{-number}$ es el número focal de la óptica.

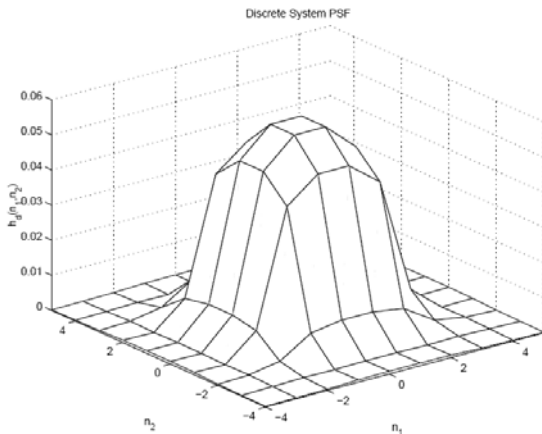


Fig. 2: PSF discreta en espacio para $L_1 = L_2 = 5$.

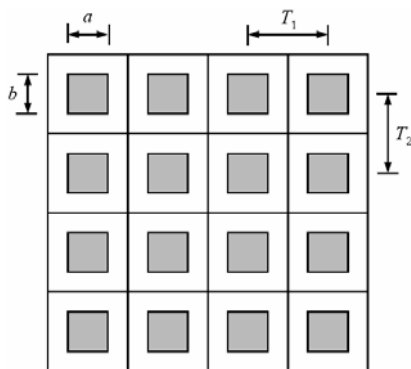


Fig. 3: Arreglo uniforme de sensores del dispositivo de captura

III. REGISTRACIÓN DE IMÁGENES

En este procedimiento se busca estimar las transformaciones relativas entre las imágenes LR con respecto a una de ellas, que denominaremos \mathbf{y}_1 .

Suponiendo desplazamientos y rotaciones pequeñas, el modelo se puede expresar como un sistema de ecuaciones de orden 3 [1]

$$\mathbf{R}_k^n = M^{-1} \mathbf{V}_k^n + \mathbf{R}_k^{n-1} \quad (6)$$

Donde M es una matriz que depende del gradiente de \mathbf{y}_1 y puede ser calculada con un operador de Prewitt. \mathbf{V}_k^n es un vector que depende del error $(\mathbf{y}_1 - \mathbf{y}_k^n)$ y $\mathbf{R}_k^n = [h_k, v_k, \theta_k]$ el vector de transformación $\mathbf{y}_k \rightarrow \mathbf{y}_k^n$. Para permitir transformaciones más grandes, se suma \mathbf{R}_k^{n-1} . El algoritmo de registración se resume en la tabla 1.

TABLA I
ALGORITMO DE REGISTRACIÓN

1- calcular M
2- inicializar $\mathbf{R}_k^0 = [0,0,0]$
3- calcular \mathbf{V}_k^{n+1}
4- calcular $\mathbf{R}_k^{n+1} = M^{-1} \mathbf{V}_k^{n+1} + \mathbf{R}_k^n$
5- si $\frac{\ \mathbf{R}_k^{n+1} - \mathbf{R}_k^n\ }{\ \mathbf{R}_k^n\ } < TOL$, $\mathbf{R}_k^n = \mathbf{R}_k^{n+1}$ y parar
6- $\mathbf{y}_k \xrightarrow{\mathbf{R}_k^{n+1}} \mathbf{y}_k^n$
7- ir a 3

Las principales fuentes de error surgen de la transformación aplicada en 6 debido a que involucra una interpolación y en las aproximaciones realizadas al gradiente para formular el sistema de ecuaciones. Aún así, basado en observaciones empíricas, el algoritmo converge a soluciones confiables para desplazamientos menores a 10 píxeles y rotaciones menores a 10 grados [1].

Con las estimaciones \mathbf{R}_k^n es posible calcular la matriz de pesos $w_{m,k}$ como la PSF centrada en la posición del píxel m en HR según la transformación \mathbf{R}_k^n (Fig. 4).

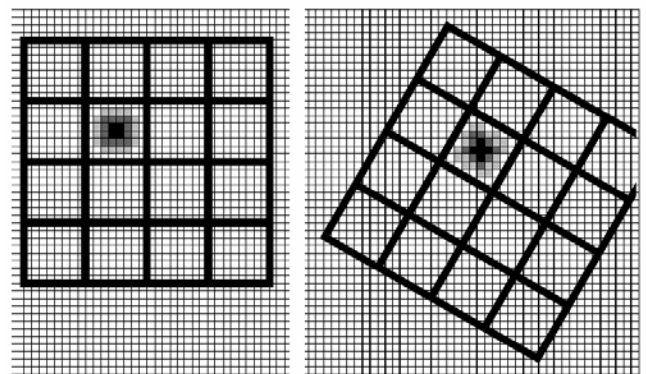


Fig. 4: Contribución de los píxeles z_r en y_m para la PSF definida en (5)

IV. RECONSTRUCCIÓN DE LA IMAGEN DE ALTA RESOLUCIÓN

Luego de la etapa de registración el modelo definido en (1) queda totalmente definido y sólo queda estimar z_r . Para ello se minimiza la función

$$C(z) = \frac{1}{2} \sum_{m=1}^{pM} \left(y_m - \sum_{r=1}^N w_{m,r} z_r \right)^2 + \frac{\lambda}{2} \sum_{i=1}^N \left(\sum_{j=1}^N \alpha_{i,j} z_j \right)^2 \quad (7)$$

Mediante el método del gradiente conjugado[1]. Inicializando \mathbf{z}^0 como la interpolación bilineal de \mathbf{y}_1 y actualizándolo según

$$\hat{\mathbf{z}}^{n+1} = \hat{\mathbf{z}}^n - \varepsilon^n \mathbf{g}^n \quad (8)$$

Con

$$\mathbf{g}_k(\hat{\mathbf{z}}^n) = \sum_{m=1}^{pM} w_{m,k} \left(\sum_{r=1}^N w_{m,r} \hat{z}_r^n - y_m \right) + \lambda \sum_{i=1}^N \alpha_{i,k} \sum_{j=1}^N \alpha_{i,j} \hat{z}_j^n \quad (9)$$

$$\varepsilon^n = \frac{\sum_{m=1}^{pM} \gamma_m \left(\sum_{r=1}^N w_{m,r} \hat{z}_r^n - y_m \right) + \lambda \sum_{i=1}^N \bar{g}_i \left(\sum_{j=1}^N \alpha_{i,j} \hat{z}_j^n \right)}{\sum_{m=1}^{pM} \gamma_m^2 + \lambda \sum_{i=1}^N \bar{g}_i^2} \quad (10)$$

$$\gamma_m = \sum_{r=1}^N w_{m,r} g_r(\hat{\mathbf{z}}^n) \quad \bar{g}_i = \sum_{j=1}^N \alpha_{i,j} g_j(\hat{\mathbf{z}}^n)$$

$$\alpha_{i,j} = \begin{cases} 1 & i = j \\ 0.25 & z_i \text{ vecino cardinal } z_j \\ 0 & \text{en otro caso} \end{cases} \quad (11)$$

El resumen del algoritmo se muestra en la tabla 2.

TABLA II
ALGORITMO DE RECONSTRUCCIÓN

1- calcular $\mathbf{z}^0, \mathbf{d}^0 = -\mathbf{g}^0$
2- calcular $\varepsilon^n, \hat{\mathbf{z}}^{n+1} = \hat{\mathbf{z}}^n + \varepsilon^n \mathbf{d}^n$
3- si $\frac{\ \hat{\mathbf{z}}^{n+1} - \hat{\mathbf{z}}^n\ }{\ \hat{\mathbf{z}}^n\ } < TOL, \hat{\mathbf{z}}^n = \hat{\mathbf{z}}^{n+1}$ parar
4- calcular $\mathbf{g}^{n+1}, \mathbf{d}^{n+1} = -\mathbf{g}^{n+1} + \beta^n \mathbf{d}^n, \beta = \frac{(\mathbf{g}^{n+1})^T (\mathbf{g}^{n+1})}{(\mathbf{g}^n)^T (\mathbf{g}^n)}$
5- ir a 3

A. Matrices Ralas y otras optimizaciones

Si consideramos 20 imágenes LR de 150x100 píxeles y un factor de ampliación de 5x5 con precisión simple la matriz $w_{m,r}$ ocuparía aproximadamente 419Gb y $\alpha_{i,j}$ 523Gb lo cual es inviable. Aprovechando la estructura rara de éstas matrices se utiliza un formato de almacenamiento *Compressed Row Storage* (CRS) [] guardando solo los índices y el valor cuando este es no nulo. Además cada fila se almacena en un *hash map* para optimizar la búsqueda que trae como consecuencia el formato elegido.

Al utilizar el formato de almacenamiento CRS también es posible realizar las iteraciones sobre los valores no nulos de la matriz de una forma muy eficiente. Las transformaciones geométricas con interpolación bilineal también fueron optimizadas para realizar la transformación solo de los valores no nulos de las matrices.

V. PRUEBAS

A. Simulación de aplicación real

No se logró la convergencia ni resultados mucho mejores que los obtenidos para los métodos de interpolación para la implementación realizada. Por ello, se presenta los resultados obtenidos en [1].

Se utilizó un conjunto de 20 imágenes de $50 \times 50 \times 8$ rotadas, trasladadas y desenfocadas y sub-muestreadas por un factor $(L_1, L_2) = (5, 5)$, generadas a partir de una imagen original de 250×250 . El desenfoco se realizó con una máscara gausseana de 5×5 . Finalmente, se adicionó ruido gausseano con $\sigma_x^2 = \sigma_y^2 = 25$.

Los parámetros de la cámara óptica utilizados fueron:

$$\lambda = 0.004 \text{mm}$$

$$f \text{- number} = 3$$

$$a = b = 0.04 \text{mm}$$

$$T_1 = T_2 = 0.05 \text{mm}$$

$$\rho_c = 83.3 \text{ciclos/mm}$$

Para la optimización del gradiente conjugado, se utilizó $\lambda = 0.1$

La Fig. 5 muestra el resultado obtenido de una interpolación bicúbica. Los resultados obtenidos por súper-resolución se muestran en la Fig. 6. El método muestra mejores resultados, los detalles son más notorios. Aún con una única imagen LR, se logran mejores resultados que con el método de interpolación bicúbica

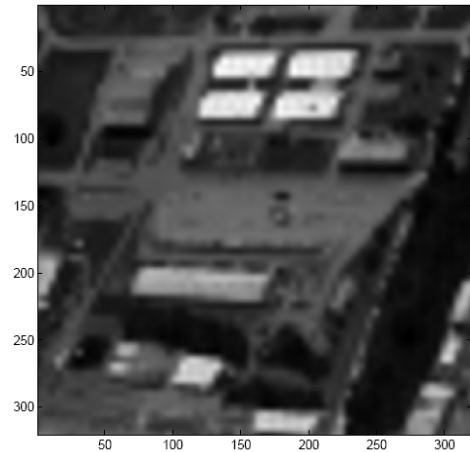


Fig. 5: Interpolación bicúbica de \mathbf{Y}_1

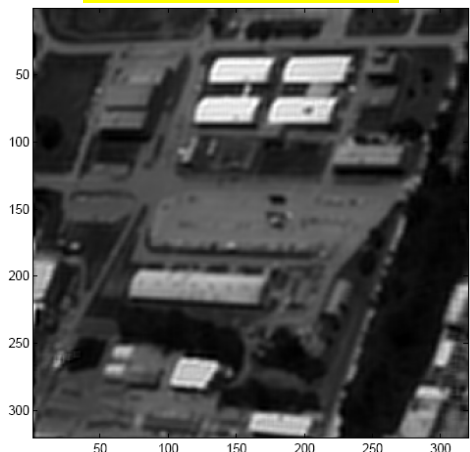


Fig. 6: Imagen reconstruida mediante Súper Resolución

B. Influencia del tamaño del conjunto de imágenes LR en la estimación

La Fig. 7 muestra el Error Medio Absoluto (MAE) en función de la cantidad de imágenes LR utilizadas. El error es medido respecto a la imagen original utilizada para generar las imágenes LR.

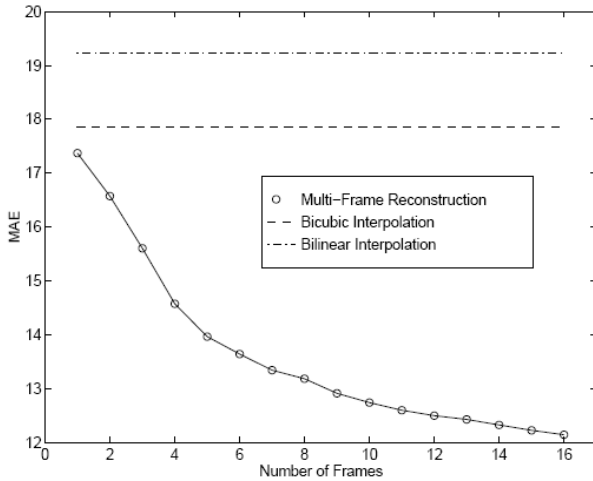


Fig. 7: MAE versus p

C. Costo computacional del algoritmo

Utilizando el formato comprimido para almacenar las matrices, se logró un algoritmo escalable tanto en costo computación como en costo de almacenamiento. Los cálculos para la resolución de los sistemas de ecuaciones, la matriz de coeficientes $\alpha_{i,j}$ y las iteraciones del algoritmo de reconstrucción son despreciables respecto al cálculo de la matriz de coeficientes $w_{m,r}$. Aproximadamente el 99,99% del tiempo de cálculo se utiliza para esto último.

Igualmente, con todas las optimizaciones posibles, no se logró el tiempo de cálculo de 5 minutos que presenta [1]. Nuestro algoritmo requirió de varias horas para la construcción de la matriz $w_{m,r}$.

D. Efectos del ruido en la estimación

Debido a las horas de cálculo requeridas para las pruebas, nos remitiremos a mostrar el efecto del ruido en la reconstrucción basándonos en las pruebas de [1].

En la Fig. 9 muestra el error medio absoluto entre la estimación y la imagen original utilizada para generar el conjunto de imágenes LR para distinto σ^2 .

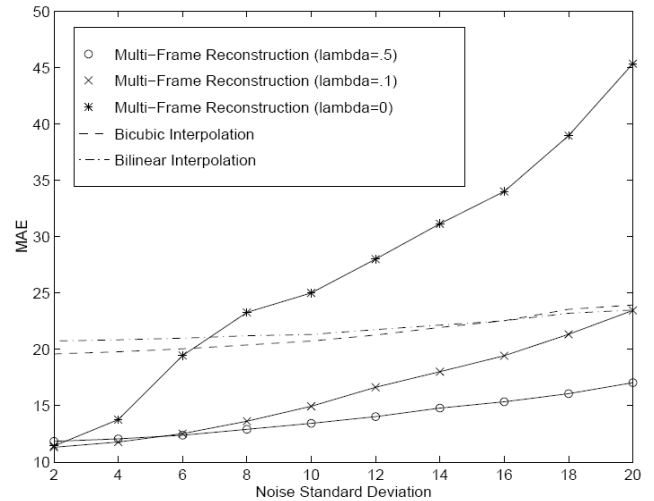


Fig. 9: efecto en el error medio absoluto (MAE) para distinto σ^2

VI. CONCLUSIONES

Se implementó y probó un método comprendido dentro de las técnicas de súper resolución basado en la estimación de las transformaciones sub-píxel. El mismo se mostró computacionalmente costoso respecto a los métodos de interpolación pero los resultados obtenidos lograron reconstruir con mayor grado de detalle la imagen de alta resolución.

Para pocas imágenes LR el algoritmo se comporta de manera similar a un interpolador bilinear, pero aumenta la calidad de la imagen HR a medida que el conjunto de imágenes LR crece.

REFERENCIAS

- [1] Russell C. Hardie, Kenneth J. Barnard, John G. Bognar, Ernest E. Armstrong and Edward A. Watson, "High Resolution Image Reconstruction From a Sequence of Rotated and translated Frames and its Application to an Infrared Imaging System", en *Optical Engineering* April 1997.
- [2] Sung Cheol Park, Min Kyu Park and Moon Gi Kang, "Super-resolution image reconstruction: A technical overview", *IEEE Signal processing magazine*, mayo 2003.
- [3] A. Van Eekeren, K. Schutte, J. Dijk y D. J. J. de Lange, "Super-resolution on moving objects and background", *IEEE International Conference on Image Processing*, 2006.
- [4] Eli Shechtman, Yaron Caspi, and Michal Irani, "Increasing Space-Time Resolution in Video", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 4, abril 2005.
- [5]