

## 重建高分辨率图像的实时串行迭代算法\*

邵凌<sup>1)</sup> 丁佩律<sup>2)</sup> 张立明<sup>2)</sup> 胡波<sup>2)</sup>

(<sup>1)</sup>IBM 中国研究中心, 北京, 100085; (<sup>2)</sup>复旦大学电子工程系智能与图像实验室, 上海, 200433)

**摘要** 提出了一种基于多帧低分辨率图像重建高分辨率图像的实时串行迭代算法。在得到与传统多帧算法图像质量相近的同时, 大大节省了计算所需的存储空间, 给出了该算法的收敛性的证明。计算机模拟结果表明该方法和实验的一致性。最后在一个实际系统验证该方法。

**关键词** 分辨率, 图像插值, 图像超分辨率, 实时串行迭代法。

## REAL-TIME SERIAL ITERATIVE ALGORITHM TO RECONSTRUCT SUPERRESOLUTION IMAGE\*

SHAO Ling<sup>1)</sup> DING Pei-Lu<sup>2)</sup> ZHANG Li-Ming<sup>2)</sup> HU Bo<sup>2)</sup>

(<sup>1)</sup>IBM China Research Laboratory, Beijing 100085, China;

<sup>2)</sup>Department of Electronic Engineering, Fudan University, Shanghai 200433, China)

**Abstract** A new real-time serial iterative algorithm to reconstruct an image with higher resolution from several low resolution pictures having slight displacement was proposed. By comparing with conventional algorithms, it greatly spares the computation memory for the similar reconstruction quality. Its convergence has been proved. Computer simulation shows that the results tally with the theoretical analysis. Finally this method was verified in a real system.

**Key words** resolution, image interpolation, image superresolution, real-time serial iterative algorithm.

### 引言

近年来, 图像超分辨率 (Image Superresolution), 又称图像插值 (Image Interpolation) 被广泛应用于从工业到民用的各个领域。传统的图像插值方法是在一帧图像上利用局部信息进行线性插值或样条函数插值, 这些方法使图像的高频细节被丢失, 且无唯一解。1981 年 Tsay<sup>[1]</sup>首次提出了从多帧互有位移的图像序列中插值产生一帧高分辨率图像的概念, 根本上解决了图像超分辨率无唯一解的问题。Kim<sup>[2]</sup>将 Tsay 和 Huang 建立在频域上的算法模型扩展到包含噪声的情况, 但是由于频域模型只能表示平移不变的传递函数和只能表示全局平移等, 以后的研究者都直接在时域上研究基于多帧低分辨率图像的图像超分辨率问题。M. Irani<sup>[3]</sup>提出了迭代反向投影算法 (Iterative back-projection) IBP 算法。该算法运算量小, 收敛速度较快, 是目前为止最有实用

意义的方法之一。Stark<sup>[4]</sup>从集投影理论发展出凸集投影算法 POCS, 加快了求解的速度, 但是一些非线性约束条件的加入会影响算法的收敛性。Tekalp<sup>[5]</sup>发展了 POCS 方法, 使之适用于更广泛的超分辨率模型。Schultz<sup>[6]</sup>提出了基于 Bayesian 估计的多帧图像超分辨率算法, 提高了超分辨率图像的质量。缺点是模型复杂, 计算量大。这些现有的算法, 在计算时都要求存储所有的低分辨率采样图像和相应的观察矩阵, 即以批处理方式进行, 这就占用了巨大的存储空间, 使它们无法应用到实际系统中, 如从低分辨率摄像头获得高分辨率图像等。

本文基于节省存储空间的考虑, 提出了一种串行迭代算法, 在达到与传统批处理方法相近效果和计算复杂度的同时, 节省了所需的存储空间, 使其具有实时的可能性。我们还构造了硬件平台, 将它应用到实际系统中, 从计算机模拟和实际系统的应用中证明了其可行性和有效性。

\* 国家自然科学基金 (批准号 39870194) 资助项目  
稿件收到日期 2001-04-11, 修改稿收到日期 2001-08-20

The project supported by the National Natural Science Foundation of China (No. 39870194)  
Received 2001-04-11, revised 2001-08-20

## 1 多帧图像超分辨率问题

设待求的高分辨率图像为  $I \in R^{N \times N}$ , 将  $I$  从行方向堆叠(stack)得到的矢量记为  $X, X \in R^{N^2 \times 1}$ . 定义第  $i$  次采样观察得到的低分辨率图像为  $y_i \in R^{M \times M}, M < N$  同样从行方向堆叠(stack)得到的矢量记为  $Y_i \in R^{M^2 \times 1}$ . 则  $Y_i$  可以写成

$$Y_i = H_i X + V_i, \quad i=0, 1, 2, \dots, T \quad (1)$$

其中  $H_i \in R^{M^2 \times N^2}$  表示第  $i$  帧低分辨率图像与高分辨率图像间的关系, 称为观察矩阵.  $V_i$  是噪声矩阵.  $T$  为低分辨率图像的帧数. 如果定义  $\bar{Y} = [Y_1, Y_2, \dots, Y_T] \in R^{TM^2 \times 1}, \bar{H} = [H_1, H_2, \dots, H_T] \in R^{TM^2 \times N^2}, \bar{V} = [V_1, V_2, \dots, V_T] \in R^{TM^2 \times 1}$ , 则式(1)可以写成

$$\bar{Y} = \bar{H}\bar{X} + \bar{V}. \quad (2)$$

式(2)是考虑了各种实际情况后的通用数学模型, 将基于多帧超分辨率重建转化为解线性方程组. 传统的方法都需要存储  $\bar{Y} \in R^{TM^2 \times 1}$  矢量和  $\bar{H} \in R^{TM^2 \times N^2}$ , 当图像尺寸较大时, 存储空间消耗过大而无法实用. 第二节中我们从节省存储空间的角度出发, 提出了一种串行的算法, 只要存储当前帧图像  $Y_i \in R^{M^2 \times 1}$  和  $\bar{H}_i \in R^{M^2 \times N^2}$  相应的观察矩阵即可.

## 2 基于多帧图像的串行迭代算法

### 2.1 串行的最速下降迭代法

对于式(2), 一般的最速下降迭代法为(1)初始值  $\hat{X}^0$  任意, (2)  $\hat{X}^{k+1} = \hat{X}^k + \lambda \bar{H}^T (\bar{Y} - \bar{H}\hat{X}^k)$ . 这里  $\bar{Y} \in R^{TM^2 \times 1}, \bar{H} \in R^{TM^2 \times N^2}, \lambda$  为步长. 而我们提出的串行的最速下降迭代法可以写为

(1) 初始值任意.

$$(2) \hat{X}_i^{k+1} = \hat{X}_i^k + \lambda \bar{H}_i^T (Y_i - \bar{H}_i \hat{X}_i^k) \\ = \hat{X}_i^k + \lambda \Delta X_i^k, \quad (3a)$$

$$(3) \hat{X}_{i+1}^0 = \hat{X}_i^\infty \quad (3b)$$

这里  $Y_i \in R^{M^2 \times 1}$  表示第  $i$  帧采样得到的低分辨率图像,  $i = 1, 2, \dots, T$ ,  $T$  是采样的总帧数.  $\bar{H}_i \in R^{M^2 \times N^2}$  表示与第  $i$  帧低分辨率图像相对应的观察矩阵,  $\hat{X}_i^k$  表示使用第  $i$  帧低分辨率图像进行  $k$  次迭代后, 高分辨率图像的估计值. (3a)表示对于当前输

入的低分辨率图像  $i$ , 我们使用最速下降迭代法求解高分辨率图像的估计值. 其中  $\lambda$  是迭代系数. (3b)表示对于下一帧输入的低分辨率图像  $i+1$ , 将上一帧迭代的结果作为下一帧迭代的初始值. 依次类推从第一帧处理到最后一帧. 以下说明这样的迭代能否保证其收敛性.

### 2.2 算法的收敛性和有效性

设  $X^*$  是正确解, 第  $i$  帧采样图像的第  $k$  次迭代的平均迭代误差为  $E(e_i^k) = X^* - \hat{X}_i^k$ , 由式(2)和式(3)且利用白噪声的零均值条件可得, 当前帧迭代误差为

$$E(e_i^{k+1}) = (I - \lambda \bar{H}_i^T \bar{H}_i)^{k+1} e_i^0 \quad (4)$$

由于  $\bar{H}_i$  中包含了降采样因子, 所以它必然不满秩, 即无论如何选取迭代系数  $\lambda$ , 迭代误差都不能趋向于零. 但我们可以看到设  $\bar{H}$  的秩为  $r$ , 非零奇异值为  $\sigma_1, \sigma_2, \dots, \sigma_r$ , 则可以写成  $\bar{H}_i^T \bar{H}_i = V_i \begin{bmatrix} S_i^2 & 0 \\ 0 & 0 \end{bmatrix} V_i^T$ , 其中  $V_i^T$  为单位正交阵  $V_i^T V_i = I, S_i^2 = \text{diag}(\sigma_{i1}^2, \sigma_{i2}^2, \dots, \sigma_{ir}^2)$  所以

$$I - \lambda \bar{H}_i^T \bar{H}_i = V_i V_i^T - \lambda \bar{H}_i^T \bar{H}_i = V_i M_i V_i^T, \quad (5)$$

$$\text{其中 } M_i = \begin{bmatrix} I - \lambda S_i^2 & 0 \\ 0 & I \end{bmatrix}.$$

将式(5)代入式(4), 并考虑到  $V_i M_i V_i^T V_i M_i V_i^T = V_i M_i^2 V_i^T$ , 得

$$E(e_i^{k+1}) = (V_i M_i V_i^T)^{k+1} e_i^0 = V_i M_i^{k+1} V_i^T e_i^0, \quad (6)$$

将正交矩阵写成分块矩阵形式  $V_i = [V_{i1} \quad V_{i2}]$ , 则式(6)为

$$E(e_i^{k+1}) = V_i M_i^{k+1} V_i^T e_i^0 \\ = [V_{i1} \quad V_{i2}] \begin{bmatrix} I - \lambda S_i^2 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} V_{i1}^T \\ V_{i2}^T \end{bmatrix} e_i^0 \\ = V_{i1} (I - \lambda S_i^2)^{k+1} V_{i1}^T e_i^0 + V_{i2}^T V_{i2} e_i^0. \quad (7)$$

迭代的平均误差可以分成两项, 第一项只要选择适当的迭代系数  $\lambda$  会随着迭代次数  $k$  的增加而单调减小, 第二项是一个常量和迭代无关. 所以当迭代次数  $k \rightarrow \infty$  时, 有

$$E(e_i^\infty) \rightarrow V_{i2} V_{i2}^T e_i^0, \quad (8)$$

观察  $V_{i2} V_{i2}^T$ , 发现它满足  $(V_{i2} V_{i2}^T)^T = V_{i2} V_{i2}^T$  和  $(V_{i2} V_{i2}^T)^2 = V_{i2} V_{i2}^T$ , 是一个正交投影算子, 记为

$P_i$ ,式(8)又可以写为

$$E(e_i^{\infty}) \rightarrow P_i e_i^{\infty}, \tag{9}$$

由于  $\hat{X}_{i+1}^0 = \hat{X}_i^{\infty}$ ,所以第  $i+1$  帧的初始平均迭代误差  $E(e_{i+1}^0) = E(\hat{X}_i^{\infty} - \hat{X}_{i+1}^0) = E(\hat{X}_i^{\infty} - \hat{X}_i^{\infty}) = E(e_i^{\infty}) = P_i e_i^{\infty}$ ,依次类推可得对于一组图像按图像顺序进行串行迭代的平均残留误差为

$$E(e_{i+1}^{\infty}) = P_{i+1} E(e_{i+1}^0) = P_{i+1} P_i e_i^0 \\ = \dots = P_{i+1} \dots P_1 e_0. \tag{10}$$

因此串行算法的迭代过程,可以看成是对误差空间不断进行正交投影的过程.由于正交投影满足  $\|Px\| \leq \|x\|$ ,所以在整个迭代的过程中误差矢量的范围只可能减少,而不会导致发散,因此串行的最速下降迭代法是收敛的.

### 3 串行迭代算法在实际中的应用

#### 3.1 观察矩阵的生成

观察矩阵  $H$  的本质在于明确未知高分辨率图像像素和低分辨率图像像素间的对应关系.假设每次采样这个关系保持不变.对于图像旋转,我们采用了以下模型直接产生观察矩阵  $H$ .图1中实线的小矩形表示高分辨率图像的像素,虚线的大矩形表示传感器的感光单元,即低分辨率图像的像素.

以图1中左边的低分辨率图像的感光单元1为

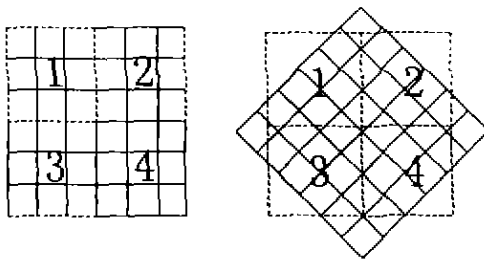


图1 旋转情况下的像素关系

Fig.1 The relationship of pixels during rotation

例,有9个高分辨率图像的像素落在该单元,假设感光单元1是均匀的,则感光单元1的值为这9个高分辨率图像像素的平均值,在图1的右图中有12个高分辨率图像像素部分或完整的落在感光单元1的范围中,则感光单元1的值为这12个高分辨率图像像素的加权平均值.不失一般性,对于高分辨率图像中的任何一个像素  $x_j \in X$ ,我们定义

$$w_j(i) = \begin{cases} 0, & \text{如果 } x_j \text{ 完整地落在} \\ & \text{第 } j \text{ 个传感单元外} \\ 1, & \text{如果 } x_j \text{ 完整地落在} \\ & \text{第 } j \text{ 个传感单元内} \\ r_j(i), & (0 < r_j(i) < 1) \text{ 如果 } x_j \\ & \text{部分地落在第 } j \text{ 个传感单元内} \end{cases} \tag{11}$$

设感光单元是均匀的,则

$$R_y(j) = \frac{\sum w_j(i) X(i)}{\sum w_j(i)}, \tag{12}$$

由此我们可以求得观察矩阵

$$H(y, x) = \frac{R_y(x)}{\sum_{k \in X} R_y(k)}. \tag{13}$$

式(13)表示第  $y$  个低分辨率图像像素和第  $x$  个等效高分辨率图像像素间的关系.其中分母表示第  $y$  个低分辨率图像像素与所有等效高分辨率图像像素关系的总和,起归一化作用.有了观察矩阵就可以按式(4)迭代求得高分辨率图像.

#### 3.2 计算机模拟实验结果

我们在计算机上模拟了从一幅高分辨率图像多次采样得到一组略有平移的低分辨率图像的过程.为了模仿成像设备镜头的转动,我们将一幅  $256 \cdot 256$  lena 图像每次旋转  $9^\circ$ ,旋转4次,再做降采样,得到4帧  $128 \cdot 128$  的低分辨率图像,如图2所示.图3~图5是各种方法插值的结果,在迭代时我们设定了初值,能较好的消除边缘效应.表1所列是各种方法 PSNR 的比较,表2所列为所需存储空间的

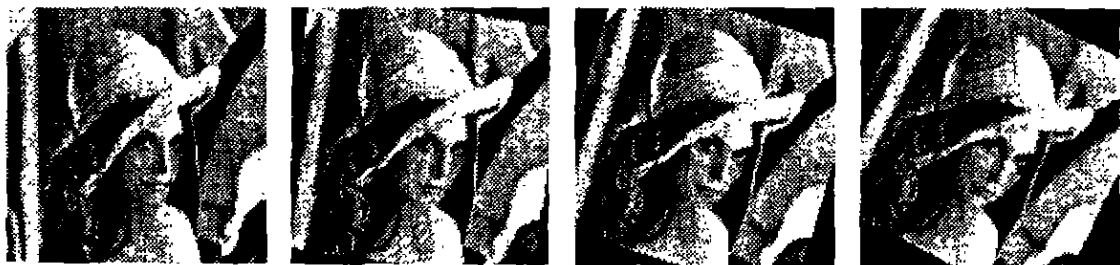


图2 低分辨率观察图像

Fig.2 The observed low resolution images



图 3 双线性插值的结果

Fig 3 bilinear interpolating result



图 4 IBP 算法的结果

Fig.4 IBP result



图 5 串行迭代的结果

Fig 5 serial iteration result

表 1 各种恢复方法的 PSNR 的比较  
Table 1 The PSNR comparison of different reconstruction methods

算法名称	PSNR(dB)
双线性插值	20.0875
三次样条插值	20.4507
IBP 迭代算法(有初值)	25.4036
串行迭代算法(有初值)	25.2339

表 2 所需存储空间的比较  
Table 2 Memory comparison of reconstruction method

	IBP 算法	串行算法
观察矩阵系数的存储空间/(字节)	354,496	98,804
低分辨率图像的存储空间/(字节)	65,536	16,384
等效高分辨率图像的存储空间/(字节)	131,072	131,072
总计/(字节)	551,104	246,260

比较.从表中可以看出,基于多帧的效果比传统帧内插值的效果好得多,本文中串行迭代算法与 IBP 算法的结果接近,但所需存储量要少一半,且参与计算的低分辨率图像越多,串行算法节省存储空间的效果就越明显.

### 3.3 系统实现

为了检验本算法在实际应用中的可能性,我们实现了一个实际的系统,其框图如图 6 所示.

通过控制摄像头的旋转,我们采集 4 帧略有旋转的图像,由于不能严格控制摄像头运动,我们使用的基于全局仿射变换模型的三步法来估计每两帧图像间的运动,进而构造出观察矩阵.图 7~图 9 是三种方法的结果.从中发现,在实际系统中其效果远没有计算机模拟的好,主要原因是运动估计中的误差,

影响了运动模型的精度,使得一些低分辨率像素不能投影到正确位置导致重影.总之,如果要将多帧算法实用化,如何设计系统使其能容易且精确地得到两帧图像间的运动模型和参数是多帧算法能否成功的关键.

### 4 结语

本文对基于多帧的图像插值方法进行了研究,从节省存储空间的角度提出了串行的迭代算法,给出了它的收敛性和有效性.计算机模拟结果表明串行迭代算法能够达到与传统批处理算法相近的效果,而所需的存储空间比传统批处理算法小许多.在实际系统上的实验表明,在无法精确控制摄像头移动的应用场合,我们需要有多帧图像之间运动矢量的精确描述.

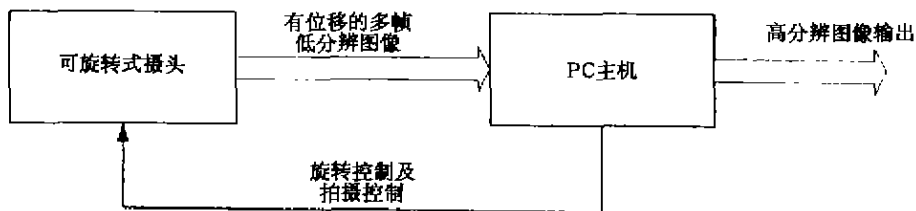


图 6 系统框图

Fig.6 Block diagram of the authors' system



图7 双线性插值的结果  
Fig 7 Bilinear interpolating result



图8 IBP算法的结果  
Fig.8 IBP result

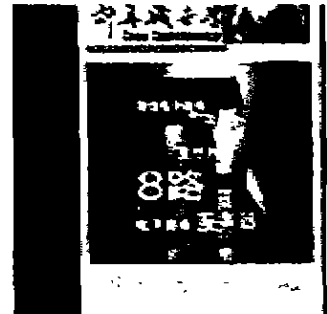


图9 串行迭代算法的结果  
Fig 9 Serial iteration result

### REFERENCES

- [1] Tsay R Y, Huang T S. Multiframe image restoration and registration. *Advances in Computer Vision and Image Processing*, 1984, 1: 317—339.
- [2] Kim S P, Bose N K, Valenzuela H V. Recursive reconstruction of high resolution image from noisy under-sampled multiframe. *IEEE Trans. ASSP*, 1990, 38(6): 1013—1027.
- [3] Irani M, Peleg S. Improving resolution by image registration. *CVGIP: Graphical Models and Image Processing*, 1991, 53(3), 231—239.
- [4] Stark H, Oskoui P. High-resolution image recovery from image-plane arrays using convex projections. *J. Opt. Soc. Am. A*, 1989, 11(6): 1715—1726.
- [5] Patti A J, Sezan M I, Tekalp A M. High-resolution image reconstruction from a low-resolution image sequence in the presence of time-varying motion blur. In: *Proc. IEEE ICIP*, Austin, TX, 1994, 343—347.
- [6] Schultz R R, Stevenson R L. Extraction of high-resolution frames from video sequences. *IEEE Trans. Image Processing*, 1996, 6(5): 996—1011.
- [7] Elad N, Feuer A. Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images. *IEEE Tran. on Image Processing*, 1997, 12(6): 1646—1657.