

文章编号: 1007-2780(2012)01-0114-07

# 基于 DSP 系统的超分辨率图像重建技术研究

邓建青<sup>1,2</sup>, 刘晶红<sup>1\*</sup>, 刘铁军

(1. 中国科学院 长春光学精密机械与物理研究所, 吉林 长春 130033, E-mail: dengjianqing@126.com;

2. 中国科学院 研究生院, 北京 100039)

**摘 要:** 由于航空光电设备造价与体积等的限制, 需要在不改变航空光电设备硬件结构的前提下, 获取尽可能清晰的图像或视频。文章提出了基于 DSP 图像处理系统的超分辨率重建方法, 首先利用 Fourier-Mellin 变换法和 Keren 算法的联合优化算法进行运动估计; 然后利用基于边缘保持的凸集投影简化方法进行超分辨率重建; 最终结合 DM642 的特征, 在不降低精度的前提下, 对算法进行优化实现。该方法在不增加系统结构体积和成本的前提下, 有效地提高了成像系统的分辨力, 进而提高系统的目标识别能力。在以 DM642 为核心嵌入式图像处理平台中实现超分辨率重建实验, 所采用的相机分辨率为  $720 \times 576$ , 整个重建的时间由传统的几分钟甚至几十分钟下降至 20 s 左右。实验结果表明, 用本文方法重建出的图像细节明显比单帧插值的图像清晰, 图像的平均梯度和信息熵有了明显提高。

**关 键 词:** 超分辨率; DSP; 图像处理系统; 图像重建

中图分类号: TN91

文献标识码: A

DOI: 10.3788/YJYXS20122701.0114

## Super-Resolution Image Reconstruction Technology Based on DSP System

DENG Jian-qing<sup>1,2</sup>, LIU Jing-hong<sup>1\*</sup>, LIU Tie-jun

(1. Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130031, China, E-mail: dengjianqing@126.com;

2. Graduate University of Chinese Academy of Sciences, Beijing 100039, China)

**Abstract:** Because of the limitations of cost and size of the navigation photo-electricity equipments, it is necessary to get images or videos as clear as possible without changing the structure of navigation photo-electricity equipments. This paper introduces a method based on DSP image processing system to realize super-resolution reconstruction which realizes motion estimation by the united optimized algorithm of Fourier-Mellin and Keren firstly, and achieves super-resolution image reconstruction based on projection onto convex sets, lastly, realizes the method on DSP by optimizes the method based on the feature of DM642 without reducing the accuracy. This method improves the resolving power of the image system without increasing the size of the system structure as well as enhance the identify ability of the system. This paper has proposed to realize super-resolution reconstruction on the embedded image processing platform which the core is DM642, the resolution of camera is  $720 \times 576$ , and the reconstruction time decreases to 20 s from several minutes or even several ten minutes. Experiment results indicate that the detail of reconstruction image is clearer than

收稿日期: 2011-08-01; 修订日期: 2011-09-01

基金项目: “973”国家自然科学基金项目(No. 2009CB72400607); “863”国家自然科学基金项目(No. 2008AA121803)

作者简介: 邓建青(1987-), 女, 江西抚州人, 硕士研究生, 主要研究方向为基于 DSP 的图像超分辨率重建。

\* 通讯联系人, E-mail: liu1577@126.com

that of the single frame interpolation image, and the average gradient and the information entropy also get improved.

**Key words:** super-resolution; DSP; imaging processing system; imaging reconstruction

## 1 引言

超分辨率重建需要精确的亚像素级运动估计和复杂的超分辨率重建迭代算法, 目前国内水平仍处于仿真阶段, 即只能做到对图像进行事后处理<sup>[1]</sup>。美国的 Aaron L. Paolinia 等针对超分辨率算法的计算强度使现有的 CPU 技术无法为中等大小的帧进行实时操作的问题, 提出了一个超分辨率实现的 GPU 加速算法, 该算法的速度比传统的 CPU 的执行速度提高了 40 倍, 并适合实时处理操作<sup>[2]</sup>。Aaron L. Paolinia 提出的方法有效地解决了实时性问题, 但是 GPU 的体积影响了其在航空光电设备中的应用。另外, 美国的 Moshe Ben-Ezra 等提出了一种超分辨率的视频摄像机, 称为抖动相机, 通过额外添加一个控制器, 可实现对探测器的精确移位来获取低分辨率图, 然后用一个自适应超分辨率算法对相机产生的视频进行超分辨率重建<sup>[3]</sup>。但 Moshe Ben-Ezra 提出的抖动相机中额外加入了控制器, 加大了成像设备的体积, 且该方法只适合于对几乎静止或运动很慢的情况, 限制了该方法的使用。

随着 DSP 芯片的集成度、稳定性、运算速度、数据吞吐量等性能不断提高, 采用 DSP 实现快速数字图像处理, 将软件算法嵌入到 DSP 中实现, 减少了设备的成本和体积, 已经成为 DSP 的一个应用方向<sup>[4-5]</sup>。本文充分利用了 TI 公司推出的 C6000 系列中 DM642 的高速处理能力, 将超分辨率重建算法嵌入至 DSP 中, 初步实现了图像超分辨率重建。

## 2 算法实现

超分辨率图像重建的前提是要准确获取图像序列之间的运动关系, 也就是要对低分辨率图像进行运动估计, 获取运动估计参数<sup>[6-8]</sup>。本文的运动估计算法结合了 Fourier-Mellin 变换法和 Keren 算法的优点, 实现了亚像素级的精确估计。

凸集投影算法 (POCS) 可以方便地加入先验信息, 很好地保持高分辨率图像上的边缘和细节,

可以应用于存在任意平滑运动、线性空域可变模糊和非均匀加性噪声的场合<sup>[9-12]</sup>。本文采用的重建算法为基于边缘保持的凸集投影算法, 重建效果良好。

### 2.1 运动估计算法实现及步骤

本文基于一个仿真模型, 假设参考图像  $f(x, y)$  是通过平移、旋转及缩放生成了待估计图像  $g(x, y)$ , 即有下式<sup>[6]</sup>。

$$g(x, y) = f(\alpha x \cos\beta + \alpha y \sin\beta - \Delta x, \\ -\alpha x \sin\beta + \alpha y \cos\beta - \Delta y)$$

其中  $\alpha$  为缩放比例因子,  $\beta$  为旋转角度,  $(\Delta x, \Delta y)$  为纵横方向相对平移量。故  $g(x, y) = f(x_1, y_1)$ , 坐标  $(x_1, y_1)$  和  $(x, y)$  之间的映射关系为:

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \alpha \begin{bmatrix} \cos\beta & \sin\beta \\ -\sin\beta & \cos\beta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

式(1)为三参数泰勒展开式计算公式。

$$X = C^{-1}V \quad (1)$$

其中:

$$X = \begin{bmatrix} \Delta x \\ \Delta y \\ \beta \end{bmatrix}$$

$$C = \begin{bmatrix} \sum \left(\frac{\partial f}{\partial x}\right)^2 & \sum \frac{\partial f}{\partial x} \frac{\partial f}{\partial y} & \sum R \frac{\partial f}{\partial x} \\ \sum \frac{\partial f}{\partial x} \frac{\partial f}{\partial y} & \sum \left(\frac{\partial f}{\partial y}\right)^2 & \sum R \frac{\partial f}{\partial y} \\ \sum R \frac{\partial f}{\partial x} & \sum R \frac{\partial f}{\partial y} & \sum R^2 \end{bmatrix}$$

$$V = \begin{bmatrix} \sum \frac{\partial f}{\partial x}(g - f) \\ \sum \frac{\partial f}{\partial y}(g - f) \\ \sum R(g - f) \end{bmatrix}, R = x \frac{\partial f}{\partial y} - y \frac{\partial f}{\partial x} \quad (2)$$

Fourier-Mellin 变换算法的优点是算法复杂度及计算量与平移及旋转大小无关, 而且, 在大平移、大旋转时不降低准确度; 缺点是在亚像素级估计方面的估计精度欠缺。而 Keren 算法的突出优点是计算简单且精度高, 还可以直接达到亚像素精度; 其缺点是算法复杂度及计算量和平移大小有关, 即大平移时迭代次数及计算量相应地增

加,而且需要采用复杂的金字塔实现,尤其不适用于大旋转角度情形。

基于 Fourier-Mellin 变换算法和 Keren 算法的优缺点,本文采用的算法是先用 Fourier-Mellin 变换算法进行粗估计,然后再用 Keren 算法进行精估计,从而结合二者的优点,避开二者的不足。运动估计步骤如下:

第一步:用 Fourier-Mellin 变换进行估计

(1)对基准图像  $f(x,y)$  和待估计图像  $g(x,y)$  进行 Fourier 变换,得到  $F(u,v)$  和  $G(u,v)$ ;

(2)求取  $F(u,v)$  和  $G(u,v)$  的幅值谱,得到  $|F(u,v)|$  和  $|G(u,v)|$ ;

(3)对幅值谱  $|F(u,v)|$  和  $|G(u,v)|$  分别进行极坐标对数变换,得到  $|F(\rho,\theta)|$  和  $|G(\rho,\theta)|$ ;

(4)对  $|F(\rho,\theta)|$  和  $|G(\rho,\theta)|$  利用相位相关算法,估计得到  $[-\ln(\alpha'),\beta']$ ,从而估计出比例因子和旋转角度  $(\alpha',\beta')$ ;

(5)将待估计图像  $g(x,y)$  按估计出来的参数进行旋转和缩放补偿,得到  $g'(x,y)$ ;

(6)用相位相关运算估计出移位参数  $(\Delta x',\Delta y')$ 。

第二步:用 Keren 算法进行估计

(1)求待估计图像  $g(x,y)$  的一阶偏导数  $\frac{\partial g}{\partial x}$  和  $\frac{\partial g}{\partial y}$ ;

(2)求参考图像  $f(x,y)$  的一阶偏导数  $\frac{\partial f}{\partial x}$  和  $\frac{\partial f}{\partial y}$ ;

(3)对参考图像和待估计图像中心  $256 \times 256$  区域,用式(1)求  $X_0$ ;

(4)利用估计出来的  $X_0$  对  $f(x,y)$  进行旋转、移位操作,得到  $f'(x,y)$ ,然后再按步骤(2)、(3)求出  $X_1$ ;

(5)判断迭代次数是否达到设定值或运动估计参数  $X_n$  是否足够小,如果是则退出,否则重复步骤(2)、(3)。

第三步:综合第一、第二步估计结果,计算出最终的运动估计参数。

## 2.2 重建算法实现及步骤

POCS 算法的基本思想是:超分辨率解空间的可行解具有多个限制条件,从而可以把每一个限制条件都定义为向量空间中的凸集合  $C_i$ ,超分

辨率重建问题的解空间就是这些凸集的交集  $C_i$ 。假设一幅图像  $H$  向凸集  $C_i$  的投影算子是  $P_i$ ,则  $H$  在  $C_i$  之上的投影可以表示为  $P_i H$ ,那么连续向所有凸集的投影就可以写为  $P_m P_{m-1} \cdots P_1 H$ 。给定初始值  $H^{(0)}$ ,可以通过  $H^{(n+1)} = P_m P_{m-1} \cdots P_1 H^{(n)}$  进行迭代,获得重建的高分辨率图像<sup>[12]</sup>。

重建步骤如下:

(1)选择低分辨率图像序列中的一幅图像作为超分辨率重建的参考图像,对其余低分辨率图像进行运动估计,并将参考图像进行插值放大,作为初始估计图像  $H^{(0)}$ ;

(2)将低分辨率图像中的像素根据求得的运动估计参数投影到高分辨率图像中,并确定点扩散函数的作用范围及系数;

(3)对  $H^{(0)}$  进行模拟低分辨率图像获取过程,得到低分辨率图像像素的估计值;

(4)计算实际像素灰度值与估计值之间的残差;

(5)对高分辨率图像中相应的像素点灰度值进行修正得到新的  $H^{(0)}$ ;

(6)检查是否达到迭代次数;

(7)程序结束。

在对低分辨率(LR)图像进行运动估计后,要将 LR 图像序列配准到高分辨率(HR)网格上。传统办法是先将图像进行缩放、旋转,然后再计算 LR 图像像素点坐标  $(x,y)$  映射到 HR 网格上的坐标  $(x_1,y_1)$ ,这样多次投影会产生插值误差。本文提出将 LR 图像序列一次反投影到 HR 网格上,可以减少重建时间,并获得更好的重建效果,其表达式如下所示。

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = 2 \begin{bmatrix} -\cos\beta & \sin\beta \\ -\sin\beta & -\cos\beta \end{bmatrix} \begin{bmatrix} x + \Delta x \\ y + \Delta y \end{bmatrix} / \alpha \quad (3)$$

## 2.3 图像评价函数

平均梯度是指能够反映出图像细微反差的程度。其计算公式为:

$$\bar{g} = \frac{1}{(M-1)(N-1)} \cdot$$

$$\sum_{i=1}^{M-1} \sum_{j=1}^{N-1} \sqrt{\frac{\nabla_i^2 f(i,j) + \nabla_j^2 f(i,j)}{2}} \quad (4)$$

式中,  $f(i,j)$ 、 $\nabla_i f(i,j)$  和  $\nabla_j f(i,j)$  分别为像点灰度及其在行、列方向上的梯度,  $M$  和  $N$  分别为图像的行、列数。平均梯度反映了图像的微小细节反差和纹理变换特征,也反映了图像的清

晰度,一般来说,平均梯度越大,表明图像越清晰,反差越好。

熵是从信息论角度反映图像信息丰富程度的一种度量方式,根据香农(Shannon)信息论原理,图像的熵定义为

$$H(X) = - \sum_{i=0}^{L-1} P_i \lg(P_i) \quad (5)$$

式中, $L$  为图像的最大灰度级, $P_i$  为图像  $X$  上像元灰度值为  $i$  的概率。

当影像中的像素在各个灰度级均匀分布,即各个灰度级出现的概率均为  $P_i = 1/L$  时,熵  $H(X)$  具有最大值  $\lg(L)$ 。此时图像的信息量最丰富,灰度分布最均匀,层次最多;当图像的所有像素只有某一灰度级,而没有其他灰度级时,熵  $H(X)$  具有最小值 0,此时图像无信息。

### 3 系统结构

本设计选用了 TMS320DM642 定点 DSP,结合解码器和编码器,完成了数字图像处理。系统的整体结构如图 1 所示:前端通过 CCD 摄像机(分辨率为  $720 \times 576$ ,帧频为 25 帧/s)获取视频图像,经过 SAA7113H 视频解码器(A/D 转换器)数字化后,以 EDMA 传输方式将数据传送至 DM642 芯片的接收缓冲区,DM642 芯片对图像进行处理后存入显示缓冲区,再以 EDMA 方式将结果输出至 SAA7121H 编码器(D/A 转换器),通过 SAA7121H 编码器编码形成码流后送到显示器显示重建图像。

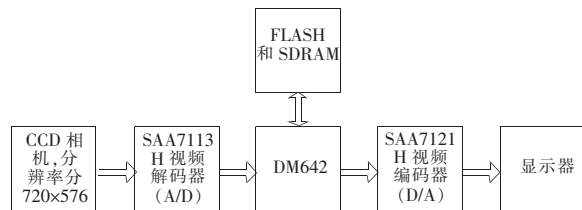


图 1 DSP 图像处理系统总体结构框图

Fig. 1 Frame diagram of DSP image processing system

本文采用连续的六幅低分辨率图像  $C_0 \sim C_5$  (大小为  $360 \times 288$ ,为相机所采图像的中间位置区域),重建出超分辨率图像  $CH$  (大小为  $720 \times 576$ ),即分辨率提高因子为 2。用变量  $M$  来表示所采集的图像为第几帧图像;用变量  $N$  表示当前是在采集图像还是在进行超分辨率重建, $N = 0$  表示在采集图像, $N = 1$  表示在重建图像。

算法流程如图 2 所示:主要分成 4 部分,开始、图像采集、图像处理和图像显示。

开始:初始化, $M = -1, N = 0$

图像采集:将 CCD 摄像机输出的模拟图像数据发送至 SAA7113H 视频编码器,视频编码器将模拟图像数据转换成数字图像数据后,以 EDMA 方式将数据传送至 DM642 的图像接收缓冲区。当图像接收缓冲区采完一帧图像后,将接收缓冲区数据搬送至处理缓冲区, $M = M + 1$ 。

图像处理:若  $M = 0$ ,则认为所采图像为重建图像序列的第一帧  $C_0$ ,对该帧图像做放大处理,得到图像  $H_0$ ,作为超分辨率重建的初始估计图像;若  $M = 5$ ,置  $N = 1$ ,以  $C_0$  为参考图像,对  $C_1 \sim C_5$  进行运动参数估计,然后用超分辨率重建算法对所采六帧图像进行超分辨率重建,重建结束后置  $M = -1, N = 0$ 。

图像显示:将超分辨率图像数据拷贝到发送缓冲区,再将发送缓冲区数据以 EDMA 方式搬送至 SAA7121H 视频编码器,由视频编码器将数字

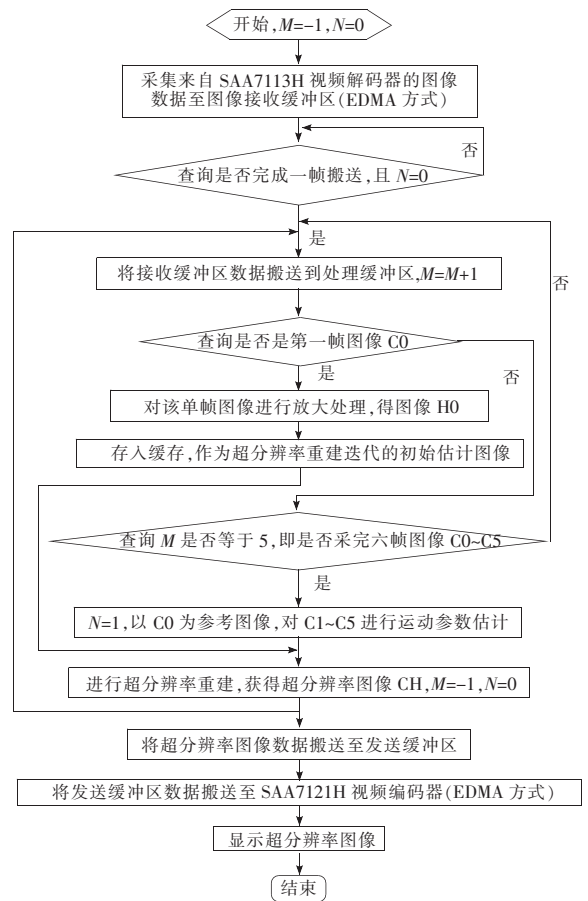


图 2 算法流程图

Fig. 2 Flow of algorithm

图像数据转换成模拟图像数据,最后发送至显示器显示超分辨率图像。

本文根据 DM642 特点将运动估计算法和重建算法按图 2 流程图移植到 DM642 嵌入式平台上,并采用 C 代码优化和线性汇编优化等技巧对重建代码进行优化实现。

## 4 实验结果及分析

本文利用 CCD 摄像机的抖动来获取具有相对位移及旋转的近景视频;用光电平台的移动拍摄来获取横向移动较大的远景视频;用嵌入式图像处理平台对图像序列进行运动参数估计和超分辨率重建。这里分别用摄像机拍摄近景视频 A 和远景视频 B 来进行重建效果实验,摄像机分辨率为  $720 \times 576$ ,选择中心位置  $360 \times 288$  大小区域进行重建,重建因子为 2,由于图片太大,本文只显示了 LR 图像序列中心  $108 \times 108$  大小区域,重建和插值图像也只显示了对应的  $216 \times 216$  大小区域。

### 4.1 用相机拍摄近景视频序列进行重建实验结果

图 3 是近景视频 A 中 6 帧连续的低分辨率图像 C0A-C5A,大小为  $108 \times 108$ ;图 4(a)是对第 1 帧低分辨率图像 C0A 进行插值(用基于梯度插值的方法)放大的图像 H0A,大小为  $216 \times 216$ ;图 4(b)为超分辨率重建图像 CHA。

表 1 为以第 1 帧图像 C0A 为参考帧,其他 5 帧图像 C1A-C5A 相对于第 1 帧进行运动参数估计的结果。其中,下表格中, $\Delta x$  为垂直位移, $\Delta y$  为水平位移,单位均为一个像素; $\beta$  为旋转角度,单位为度。若  $\Delta x > 0$ , $\Delta y > 0$ , $\beta > 0$  分别表示待估计图像相对参考图像向右移动、向下移动和顺时针旋

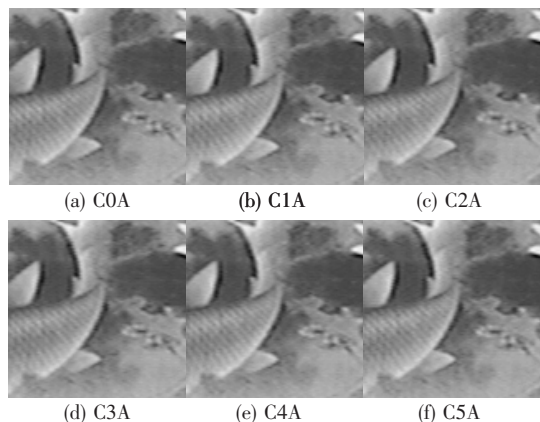


图 3 近景视频 A 中 6 帧低分辨率图像 C0A~C5A  
Fig. 3 Six frames of low resolution images C0A~C5A of close shot video A

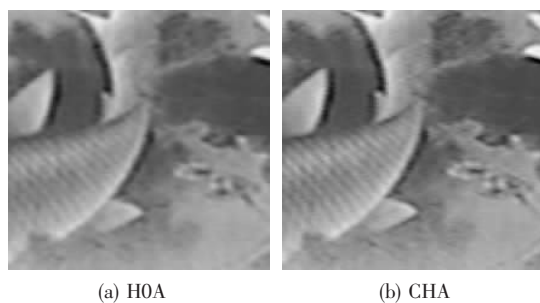


图 4 单帧插值图像 H0A 和重建图像 CHA  
Fig. 4 Single frame interpolation image H0A and reconstruction image CHA

表 1 近景视频序列运动参数

Table 1 Parameters of movement of the sequence of close shot video

	C0A	C1A	C2A	C3A	C4A	C5A
$\Delta x$	0.000	0.224	0.728	0.957	1.238	1.820
$\Delta y$	0.000	0.154	0.367	0.435	0.480	0.611
$\beta$	0.000	0.097	0.148	0.145	0.157	0.225

表 2 近景视频序列插值图像和重建图像对比

Table 2 Comparison of the interpolation image and reconstruction image of close shot video

	平均梯度	信息熵
单帧插值放大图像 H0A	3.668 9	7.000 7
重建图像 CHA	5.264 7	7.060 6

转,其中,本文要求  $\Delta y \in (0, 128)$ , $\Delta \beta \in (0, 360)$ 。

表 2 为对单帧插值放大图像 H0A 和超分辨率重建图像 CHA 用评价函数对比的结果。

### 4.2 用相机拍摄近景视频序列进行重建实验结果

图 5 是远景视频 B 中 6 帧连续的低分辨率图像序列,大小为  $108 \times 108$ ,图 6(a)是对第 1 帧图像 C0B 进行单帧插值放大的图像 H0B,图 6(b)是用该 6 帧低分辨率图像序列进行重建图像 CHB。

表 3 为以第 1 帧图像 C0B 为参考帧,其他 5 帧图像 C1B-C5B 相对于第 1 帧进行运动参数估计的结果。

表 4 为对单帧插值放大图像 H0B 和超分辨率重建图像 CHB 用评价函数对比的结果。

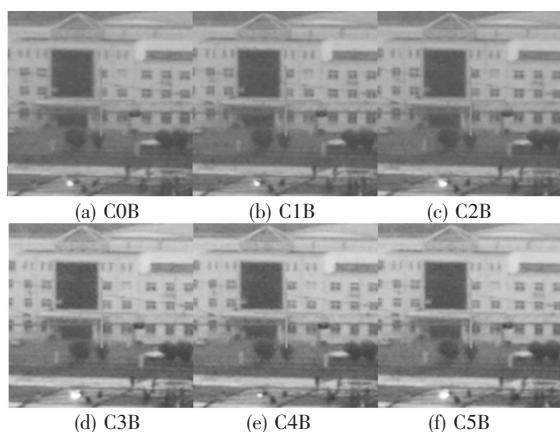


图 5 远景视频 A 中 6 帧连续的低分辨率图像 C0B~C5B

Fig. 5 Six frames of low resolution images C0B~C5B of establishing shot video B

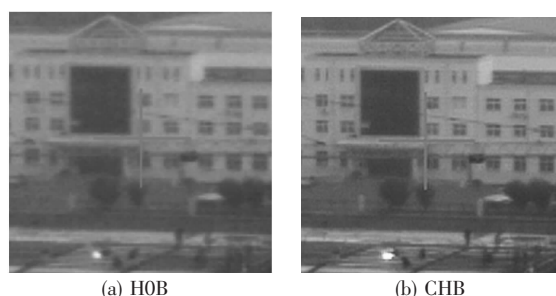


图 6 单帧插值图像 H0B 和重建图像 CHB

Fig. 6 Single frame interpolation image H0B and reconstruction image CHB

表 3 远景视频序列运动参数

Table 3 Parameters of movement sequence of establishing shot video

	C0B	C1B	C2B	C3B	C4B	C5B
$\Delta x$	0.000	-0.034	-0.080	-0.228	-0.288	-0.466
$\Delta y$	0.000	1.913	2.970	4.232	4.147	3.725
$\beta$	0.000	-0.004	-0.006	-0.010	0.039	-0.007

表 4 远景视频序列单帧插值图像和重建图像对比

Table 4 Comparison of the interpolation image and reconstruction image of establishing shot video

	平均梯度	信息熵
单帧插值放大图像 H0B	2.085 3	5.813 6
重建图像 CHB	3.020 5	5.873 9

### 4.3 实验分析

表 5 为用本文方法对视频序列 A、B 和用传统未优化的方法对 A、B 序列进行重建的时间对比。

表 5 优化方法和未优化方法对视频 A、B 的重建时间  
Table 5 Reconstruction time for video A, B with optimizing method and non-optimizing method

	本文重建时间/s	传统重建时间/s
近景视频序列 A	19.876	312.375
远景视频序列 B	21.125	352.204

目视评价 A、B 视频序列中重建图像与单帧插值图像质量,可以看出重建图像明显比单帧插值图像清晰。而且远景视频序列 B 的重建效果比近景视频序列 A 的重建质量要好,分析原因发现:B 视频由平台机械移动获得,LR 序列之间的冗余互补信息较多,A 为人工抖动相机获得,运动参数都较小,互补信息较少。

本文还用平均梯度和信息熵来评价重建效果。由表 2 和表 4 看出,重建出的图像的平均梯度与信息熵均比单帧插值图像大。同样表明,重建图像更清晰,所含信息量更丰富。

且由表 5 看出本文采用的优化重建算法大大减少了计算量,重建时间由几分钟甚至几十分钟下降到 20 s 左右,为工程化应用打下了基础。

## 5 结 论

首次在 DM642 图像处理系统上,初步实现了图像超分辨率重建,实验证明该方法对细节较多的近景视频和细节较少的远景视频重建均适用,且效果明显。然而由于运动估计算法要求达到亚像素级精度,且图像重建算法需对 6 幅图进行迭代计算,需要的运算量极大,整个重建时间需要 20 s 左右,故而还未达到实时处理,这是以后继续学习和研究的重点。在 DSP 上初步实现图像超分辨率重建为今后工程化打下了一定的基础,若想真正实现工程化,需要进一步解决大运算量下的实时性问题。

### 参 考 文 献:

[ 1 ] Baker S, Kanade T. Limits on super-resolution and how to break them [J]. *IEEE Transactions on Pattern Analysis*



and Machine Intelligence, 2002, 24(9):1167-1183.

- [ 2 ] Park S C, Park M K, Kang M G. Super-resolution image reconstruction; A technical overview [J]. *IEEE Signal Processing Magazine*, 2003, 20(3):1235-1240.
- [ 3 ] Moshe B E, Assaf Z, Shree K N. Jitter camera; a super-resolution video camera [J]. *Visual Communications and Image Processing*, 2006, 6077(4):1-13.
- [ 4 ] 张月, 马云, 王伟, 等. 基于 DSP 的 CCD 天文图像处理系统 [J]. *电光与控制*, 2005, 12(4):47-50.
- [ 5 ] 张锐, 吴婉兰, 吴庆宪, 等. DM642 在数字图像处理系统中的应用 [J]. *计算机应用与软件*, 2008, 25(1):211-212.
- [ 6 ] Ozkan M K, Sezan M I, Tekalp A M. Adaptive motion-compensated filtering of noisy image sequences [J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 1993, 3(4):277-290.
- [ 7 ] Segall C A, Molina, Katsaggelos R, et al. Signal high-resolution images from low-resolution compressed video [J]. *IEEE Processing Magazine*, 2003, 20(3):37-48.
- [ 8 ] Tsai R Y, Huang T S. Multi frame image restoration and registration [J]. *Computer Vision and Image Processing*, 1984, 1(2):317-339.
- [ 9 ] Baker S, Kanade T. Limits on super-resolution and how to break them [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002, 24(9):1167-1183.
- [10] 周春大, 张岩. 基于微位移技术提高 CCD 分辨率的方法 [J]. *光子学报*, 2006, 35(12):1969-1974.
- [11] 郑丽贤, 何小海, 吴炜, 等. 基于学习的超分辨率技术 [J]. *计算机工程*, 2008, 34(5):193-195.
- [12] 徐美芳, 刘晶红. 基于边缘保持的航拍图像凸集投影超分辨率重建算法 [J]. *液晶与显示*, 2010, 25(6):873-878.

## SI 词头

代表的因素	词头名称		符号	代表的因素	词头名称		符号
	英文	中文			英文	中文	
$10^{24}$	yotta	尧[它]	Y	$10^{-1}$	deci	分	d
$10^{21}$	zetta	泽[它]	Z	$10^{-2}$	centi	厘	c
$10^{18}$	exa	艾[可萨]	E	$10^{-3}$	milli	毫	m
$10^{15}$	peta	拍[它]	P	$10^{-6}$	micro	微	$\mu$
$10^{12}$	tera	太[拉]	T	$10^{-9}$	nano	纳[诺]	n
$10^9$	giga	吉[咖]	G	$10^{-12}$	pico	皮[可]	p
$10^6$	mega	兆	M	$10^{-15}$	femto	飞[母托]	f
$10^3$	kilo	千	k	$10^{-18}$	atto	阿[托]	a
$10^2$	hecto	百	h	$10^{-21}$	zepto	仄[普托]	z
$10^1$	deca	十	da	$10^{-24}$	yocto	幺[科托]	y