

超分辨率重建在车牌定位中的应用

来沛剑, 刘 辉, 康燕妮, 朱玉艳

(昆明理工大学 信息工程与自动化学院, 云南 昆明 650051)

摘要: 针对车牌识别系统中由于采集图像分辨率低而造成的车牌定位效果较差这一问题, 提出了一种基于凸集投影(POCS)的车牌定位超分辨算法, 该方法能够提高图像的分辨率, 保留原始图像的细节信息, 准确地定位出车牌区域, 为接下来的车牌识别提供了有力保障。

关键词: 超分辨率; 车牌定位; 凸集投影

中图分类号: TP391

文献标识码: A

Application of super-resolution reconstruction in license plate location

LAI Pei Jian, LIU Hui, KANG Yan Ni, ZHU Yu Yan

(Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650051, China)

Abstract: A super-resolution algorithm based on projections onto convex sets (POCS) for vehicle license plate location was proposed, in respect to the less effective in license plate location of license plate recognition system which is a result of the low-resolution of collecting images. This method could improve the image resolution, retain the details of the original image information and accurately locate the license plate area, it provides a strong support for the license plate recognition.

Key words: super-resolution; license plate location; POCS

车辆牌照自动识别是近几年发展起来的计算机视觉与模式识别技术在智能交通领域应用的重要研究课题之一。车牌识别系统主要由图像采集、图像预处理、车牌定位^[1]、字符分割和字符识别 5 个部分组成。车牌定位是指将车牌区域从车辆图像中分割出来, 是实现整个系统的关键环节, 其定位的速度和准确程度直接影响到车牌识别系统的性能。要完成准确的车牌定位, 首先要看清楚含有车牌的图像, 提高图像的质量才能获得好的定位效果。

然而在现实应用中往往不能得到高分辨率的车牌图像, 由于成像系统本身的缺陷、车速的快慢以及图像的聚焦等多方面原因, 获取的图像都会有不同程度的退化, 主要表现为图像的模糊和受噪声污染, 在这种情况下进行的车牌定位以及识别, 其准确率是非常低的。因此, 必须提高图像的分辨率, 才能更好地进行车牌识别的下一步工作。

目前, 提高车牌图像的分辨率可以从 2 个方面着手, 一方面是通过提高成像系统的精度和稳定性来实现, 但其费用昂贵而且制作技术难度大; 另一方面是通

过对低分辨率图像进行处理来获得高分辨率图像, 这将大大降低成本。虽然传统的插值算法也可以对图像进行放大, 但图像的分辨能力并没有提高, 高频细节被丢失。图像超分辨率重建^[2-3]不但能提高图像的像素值, 而且能使图像获得更多的细节和信息, 因此, 非常适用于车牌定位的前期图像处理。

1 图像超分辨率重建技术

图像超分辨率重建是由多帧低分辨率图像来估计出一幅较高分辨率的非变形图像, 同时还能够消除加性噪声以及由有限探测器尺寸和光学元件产生的模糊。对于一段场景变化不大的图像序列来说, 这一组图像具有相似但不完全相同的信息, 其主要变化在于图像中运动物体产生了空间位移。这些图像之间的信息是冗余的, 又是互补的, 因此可以认为这一组图像从整体上记录了更高分辨率图像的信息。超分辨率重构技术就是利用多帧低分辨率图像中所带有的不同的空域和时域信息, 再引入图像的先验约束, 以软件的方法重建出更接近于原始图像的重建图像。这种方法的优点是不涉及硬件、成本低, 而且现有的图像系统还可以使用, 是一种比较经

《微型机与应用》2009 年第 24 期

图形、图像与多媒体

Image Processing and Multimedia Technology

济的方案。

超分辨率重建方法主要可以分为2类:频率域方法和空间域方法。

频域方法实际上是在频域内解决图像内插问题,其观察模型是基于傅里叶变换的移位特性。但是由于频域方法只能局限于全局平移运动和线性空间不变降质模型,包含空域先验知识的能力有限,因此目前这类方法不再成为研究的主流。

空间域方法是图像超分辨率重建应用中的一类主要的方法,它将复杂的运动模型与相应的插值、迭代及滤波重采样放在一起进行处理,主要包括非均匀样本内插法、迭代反投影方法、凸集投影(POCS)方法、统计重建方法(最大后验概率估计MAP和最大似然估计ML)等^[2-3]。由于凸集投影(POCS)算法能最大限度地利用不同图像包含的信息量,因此选择该算法作为本文的研究对象。

2 凸集投影算法

凸集投影法(POCS)^[4-5]最早由Bregman提出,其应用十分广泛,例如信号解卷积、层析成像、图像重建、带限信号外推等。凸集投影法把成像系统空间相应特性及噪声统计特性等先验知识作为进行图像重构的约束条件,每个约束条件对应于在整个成像空间中的一个含有理想高分辨率图像的凸形约束集。这组凸形约束集代表了期望的超分辨率图像的一些特性,例如正定、能量有界等,通过这些约束集就可以得到简化的解空间。总的来说,凸集投影法就是将给定空间矢量的任一点,定位到满足所有凸集约束的一点的迭代过程。凸集投影法的基本理论如下:

对应每个观察到的低分辨率像素 (n_1, n_2, k) ,定义一个各不相同的凸集:

$$C_{n_1, n_2, i, k} = \{x_i(m_1, m_2) : |r_k^{(x_i)}(n_1, n_2) - \delta_0| \leq \delta_0\} \quad (1)$$

式中:

$$r_k^{(x_i)}(n_1, n_2) = g_k(n_1, n_2) - \sum_{m_1=0}^{M-1} \sum_{m_2=0}^{M-1} x_i(m_1, m_2) h_{ik}(m_1, m_2; n_1, n_2) \quad (2)$$

其中 $h_{ik}(m_1, m_2; n_1, n_2)$ 为脉冲响应系数; δ_0 是对观察结果所具有的置信度,它被设置为等于 $c\delta v$,其中 δv 是噪声的标准偏差, $c \geq 0$ 是由一个适当的统计置信度范围来决定的。这些设置定义了高分辨率图像,在与观察噪声的变化成比例的某个置信度范围内,这些图像与所观察到的低分辨率帧是一致的。

一个任意的 $x_i(m_1, m_2)$ 在 $C_{n_1, n_2, i, k}$ 上的投影 $y(m_1, m_2)$ $\square P_{n_1, n_2, i, k}[x_i(m_1, m_2)]$ 被定义为

$$P_{n_1, n_2, i, k} =$$

$$\begin{cases} x_i(m_1, m_2) + \frac{r_k^{(x_i)}(n_1, n_2) - \delta_0}{\sum_o \sum_p h_{ik}^2(o, p; n_1, n_2)} h_{ik}(m_1, m_2; n_1, n_2) & r_k^{(x_i)}(n_1, n_2) > \delta_0 \\ x_i(m_1, m_2) & -\delta_0 < r_k^{(x_i)}(n_1, n_2) < \delta_0 \\ x_i(m_1, m_2) + \frac{r_k^{(x_i)}(n_1, n_2) + \delta_0}{\sum_o \sum_p h_{ik}^2(o, p; n_1, n_2)} h_{ik}(m_1, m_2; n_1, n_2) & r_k^{(x_i)}(n_1, n_2) < -\delta_0 \end{cases} \quad (3)$$

可以运用附加约束条件,如幅值和有限区域的约束来改善其结果。幅值约束由下式给定:

$$C_A = \{x_i(m_1, m_2) : 0 \leq x_i(m_1, m_2) \leq 255, 0 \leq m_1, m_2 \leq M-1\} \quad (4)$$

在幅值约束 C_A 上的投影 P_A 由下式定义:

$$P_A[x_i(m_1, m_2)] = \begin{cases} 0 & x_i(m_1, m_2) < 0 \\ x_i(m_1, m_2) & 0 < x_i(m_1, m_2) < 255 \\ 255 & x_i(m_1, m_2) > 255 \end{cases} \quad (5)$$

估计公式为:

$$\hat{S}_i(m_1, m_2) = T_A \prod_{\substack{(0,0) \leq (n_1, n_2) \leq (N-1, N-1) \\ k=1,2,\dots,L, j=0,1,\dots \\ 0 \leq m_1 \leq M-1, 0 \leq m_2 \leq M-1}} T_{n_1, n_2, k} [\hat{S}_i^j(m_1, m_2)] \quad (6)$$

其中 \hat{S} 为高分辨率图像估计; T 为综合投影算子; $T_i = 1 + \lambda_i(P_i + 1)$, $0 < \lambda_i < 2$ 为一松弛因子。可以在2个方向上使用双线性方法来插入低分辨率参考帧,以便作为一个初始估计值。

3 实验

采用Matlab仿真实现本文的超分辨率车牌定位算法。下面给出具体算法步骤。

首先给出如下定义:

p 为低分辨率图像帧数; g_k 为第 k 帧低分辨率图像,

其中 $0 < k < p$; f 是重建的高分辨率图像, $f^{(0)}$ 是高分辨率图像的初始值。

(1)读入一帧含有车牌的低分辨率图像 g_1 ;

(2)对该图像双线性插值得到参考帧 $f^{(0)}$;

(3)For $k=2$ to p

读入一帧含有车牌的低分辨率图像 g_k ;

与参考帧 $f^{(0)}$ 做精确到半像素的运动估计;

End

(4)For $i=1$ to n (迭代次数)

For $k=1$ to p

对帧中每像素进行POCS运算,得到差值 r ,并据

图形、图像与多媒体

Image Processing and Multimedia Technology

此对参考帧 $f^{(0)}$ 进行修正;

End

End

(5)将重建得到的高分辨率图像 f 转化为256色灰度级的DIB位图灰度图像;

(6)将灰度图像二值化;

(7)对二值化图像进行数学形态学处理;

(8)对处理后的图像进行逐行逐列扫描,确定车牌的上下左右边界,并返回车牌图像,若不存在车牌则返回失败;

(9)程序结束。

以大小为 526×350 像素的测试图像作为原始的高分辨率图像,将该图像进行水平平移、垂直平移和旋转并加入加性噪声模拟生成4幅 263×175 像素的低分辨率图像,如图1所示。



图1 模拟生成的4幅低分辨率图像

将任意一幅低分辨率图像进行普通的车牌定位,则不能正确定位出车牌区域,定位结果如图2(a)所示。用本文的方法,将4幅低分辨率图像在图像预处理的时候应用超分辨率重建,在重建图像的基础上进行的车牌定位则有很好的结果,如图2(b)所示。



(a) 普通车牌定位结果 (b) 超分辨率车牌定位结果

图2 测试图像的车牌定位结果比较

对实际采集的4幅 214×160 像素的低分辨率图像(如图3所示)进行普通的车牌定位,不能得到正确的定位图像,定位结果如图4(a)所示。运用本文方法,则能将车牌准确地定位出来,定位结果如图4(b)所示。

本文针对车牌定位前影响定位准确性的图像分辨

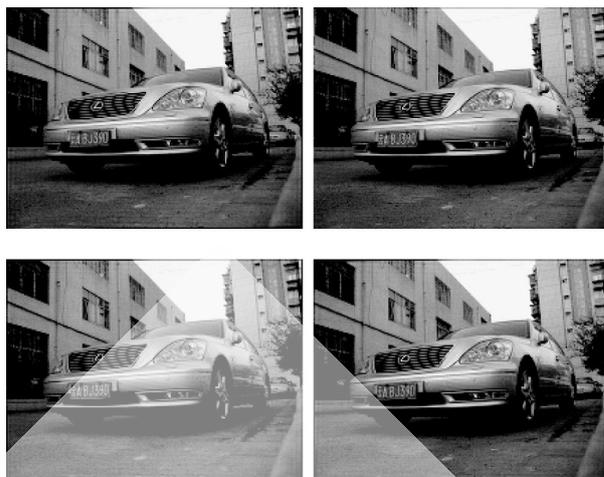


图3 实际采集的低分辨率图像



(a) 普通车牌定位结果 (b) 超分辨率车牌定位结果

图4 实际采集图像的车牌定位结果比较

率较低这一问题,对于从低分辨率的车牌图像得到高分辨率图像的方法进行了较为深入的研究,并在大量理论分析和实验基础上,提出了一种基于凸集投影(POCS)的车牌图像超分辨算法,在提高图像分辨率的同时能够保留丰富的细节,从而达到准确定位车牌的目的。

参考文献

- [1] 徐晓丹,张健.车牌定位的研究与实现[J].辽宁工业大学学报,2008,28(5):287-289.
- [2] PARK S C, PARK M K, KANG M G. Super-resolution image reconstruction: technical review [J]. IEEE signal processing magazine, 2003,(5):21-36.
- [3] ELAD M, FEUER A. Restoration of a single super-resolution image from several blurred, noisy and under sampled measured Images [J]. IEEE Transactions on Image Processing, 1997, 6(12): 1646-1658.
- [4] ODKOUI-FARD P, STRAK H. Tomographic image reconstruction using the theory of convex projections [J]. IEEE Trans. Medical Imaging, 1988, 7(1):45-58.
- [5] 邓乾国,游志胜.基于凸集投影算法的超分辨率图像重建技术[J].成都信息工程学院院报,2005,20(6):710.

(收稿日期:2009-09-15)