

一种新的红外焦平面阵列疵点补偿算法

王炳健, 刘上乾, 李庆, 赖睿

(西安电子科技大学技术物理学院, 陕西西安 710071)

摘要:提出了一种基于运动图像分析的红外焦平面阵列疵点补偿算法,它利用序列图像帧间的相关性,通过对相邻两帧图像进行运动分析,获取图像中目标的运动参数,再通过运动参数,将当前图像中的疵点对应到前一帧图像相应的位置,并用前一帧相应位置像素的灰度对疵点进行补偿.该算法克服了邻域疵点补偿算法无法保持目标边缘的缺点,其补偿效果优于邻域补偿算法的效果.

关键词:红外焦平面阵列;疵点补偿;运动分析;保持目标边缘

中图分类号: TN911.73 **文献标识码:** A **文章编号:** 1001-2400(2006)02-0278-03

A new blind pixels correction algorithm for IRFPA based on moving scenes

WANG Bing-jian, LIU Shang-qian, LI Qing, LAI Rui

(School of Technical Physics, Xidian Univ., Xi'an 710071, China)

Abstract: This paper presents a new blind pixels approximation algorithm based on the scene moving analysis for the infrared focal plane array (IRFPA). It is known that there exists strong relativity between neighboring images in image sequences. By moving scene analysis, moving parameters can be calculated from image sequences. Then the pixels in the previous frame image can be found which correspond to blind pixels in the current frame image. And these pixels are used to approximate blind pixels in the current frame. This algorithm can make objects in the approximated image maintain its edge completely. Theoretical analysis and experimental results show its effectiveness.

Key Words: infrared focal plane array; blind pixels compensation; moving analysis; maintain edge

红外焦平面阵列(IRFPA)是目前最先进的红外成像器件,但由于工艺、材料等原因,IRFPA器件存在疵点、非均匀性问题,从而降低了器件的输出图像质量.疵点,又称无效像元^[1],是器件在生产过程中产生的死像元和过热像元.在图像中,这些像点的灰度值变化缓慢,不能正确地反映场景的变化.在图像处理过程中,利用空间的相关性,采用邻域插值、中值滤波等^[2~4]算法对这些点进行灰度补偿,即通过对疵点周围像点的灰度值作线性或者非线性运算,得到该疵点的灰度.当图像中的疵点完全位于目标或者背景中时,邻域补偿可取得较好的效果.但是,当疵点位于目标的边缘时,邻域补偿无法判断疵点是位于目标区域还是背景之中,从而使补偿后图像中的目标边缘有凹凸现象,破坏了目标的边缘.为了解决这一问题,笔者提出了一种基于场景运动分析的疵点补偿算法,通过对目标进行运动估计^[5],利用相邻帧的图像相关性对疵点进行补偿,使补偿后的图像能够很好地保持其边缘.

1 疵点补偿算法

首先通过疵点搜索方法^[1]找到疵点的位置,并保存疵点的位置坐标.然后,对于每一帧图像判断疵点是

否位于目标的边缘。如果疵点不是图像的边缘点，采用邻域补偿方法对其进行补偿；如果疵点位于目标的边缘，为了使补偿后的目标图像能够保持其边缘，不产生伪边缘，对序列图像进行运动分析，估计其运动参数，并根据运动参数计算疵点在前一帧图像中的位置，用前一帧图像对应点的灰度补偿当前帧的疵点灰度。

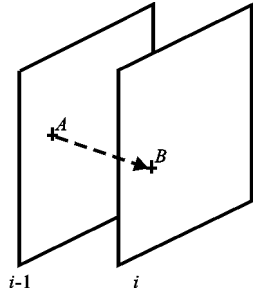


图 1 运动分析示意图

1.1 运动模型

观察图 1 所示的序列图像，第 $i-1$ 帧中的 A 点，在第 i 帧运动到 B 点。由于相邻两帧图像的时间间隔很短，可认为 A 、 B 两点具有相同的辐照度 $\mathcal{F}(A) = \mathcal{F}(B)$ ， $\mathcal{F}(x_A, y_A, t_{i-1}) = \mathcal{F}(x_B, y_B, t_i)$ ，并可认为目标仅做旋转和平移运动，此时的运动模型可表示为

$$\begin{bmatrix} x_B \\ y_B \end{bmatrix} = R \cdot \left\{ \begin{bmatrix} x_A \\ y_A \end{bmatrix} - \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \right\} + \begin{bmatrix} x_p \\ y_p \end{bmatrix}, \quad R = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}, \quad (1)$$

其中 (x_0, y_0) 为旋转轴， θ 为绕旋转轴旋转的角度（逆时针旋转角度为正，顺时针旋转角度为负）， R 称为旋转因子， (x_p, y_p) 为目标的平移量。

在这个运动模型中共有 5 个参数，分别是旋转角度 θ ，旋转轴 (x_0, y_0) 和平移量 (x_p, y_p) 。可在这两帧图像中找到多个 A 、 B 这样的匹配点，然后通过最小二乘法估计运动参数。

1.2 疵点补偿

获取了目标的运动参数后，可对位于边界的疵点进行补偿，对式 (1) 进行变换得到

$$\begin{bmatrix} x_A \\ y_A \end{bmatrix} = R^{-1} \cdot \left\{ \begin{bmatrix} x_B \\ y_B \end{bmatrix} - \begin{bmatrix} x_p \\ y_p \end{bmatrix} \right\} + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}, \quad R^{-1} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}. \quad (2)$$

如果疵点在目标内部或者边界，可通过式 (2) 计算出它在前一帧图像的位置。相邻两帧图像的采样时间间隔很短，经过非均匀性校正后，可认为这两点的灰度相同，因此用前一帧对应点的值作为当前帧的点值，即 $\mathcal{F}(x_B, y_B, t_i) = \mathcal{F}(x_A, y_A, t_{i-1})$ 。

2 运动参数估计

2.1 图像匹配

采用基于特征点的块匹配方法^[5,6]进行图像的匹配。实现方法是：(1)选择特征点；(2)选用恰当的块匹配准则；(3)找出满足匹配准则的特征点。

为了加快匹配的速度，在目标的边缘寻找特征点。条件是以特征点为中心，在各个方向的灰度方差较大；同时特征点不能为疵点。以特征点为中心的 $(2N+1) \times (2N+1)$ 块作为待匹配的块。

常用的匹配准则^[5]有均方误差函数 (MSE)、绝对平方差函数 (MAD)、最大误差最小函数 (MME)、最大匹配像素数 (MPC) 等。笔者选用均方误差函数 (MSE) 作为匹配准则，它的精度较高。

$$\delta_{\text{MSE}}(i, j) = \frac{1}{(2N+1) \times (2N+1)} \sum_{m=-N}^N \sum_{n=-N}^N (I_2(kx+m, ky+n) - I_1(i+m, j+n))^2, \quad (3)$$

$$-W \leq i, j \leq W,$$

该准则取 δ_{MSE} 最小者对应的点 (i, j) 作为 $I_2(kx, ky)$ 的匹配点。其中 $(2N+1) \times (2N+1)$ 为匹配块的大小， $(2W+1) \times (2W+1)$ 为搜索范围。

在全部特征点中，只有部分能得到正确的匹配，这是由于物体发生旋转、平移后，在前一帧中出现的特征点，在当前帧中不会出现。

2.2 运动参数估计

假设在相邻的两帧图像中找到 m 对匹配的特征点，即存在 m 组点对满足

$$\begin{bmatrix} x_{B_i} \\ y_{B_i} \end{bmatrix} = R \times \begin{bmatrix} x_{A_i} - x_0 \\ y_{A_i} - y_0 \end{bmatrix} + \begin{bmatrix} x_p \\ y_p \end{bmatrix}, \quad (4)$$

其中 $i = 1, 2, \dots, m$ 。可采用最小二乘法得到运动参数。将方程 (4) 重新写成

$$\mathbf{X}_B = \mathbf{V}_A \times \boldsymbol{\beta}_X, \quad (5)$$

$$\mathbf{Y}_B = \mathbf{V}_A \times \boldsymbol{\beta}_Y, \quad (6)$$

其中

$$X_B = [x_{B_1} \quad x_{B_2} \quad \dots \quad x_{B_m}]', \quad V_A = \begin{bmatrix} x_{A_1} & y_{A_1} & 1 & 1 & 1 \\ x_{A_2} & y_{A_2} & 1 & 1 & 1 \\ \dots & \dots & \dots & \dots & \dots \\ x_{A_m} & y_{A_m} & 1 & 1 & 1 \end{bmatrix}, \quad Y_B = [y_{B_1} \quad y_{B_2} \quad \dots \quad y_{B_m}]',$$

$$\beta_X = [\cos \theta \quad -\sin \theta \quad x_P \quad -\cos \theta \cdot x_O \quad \sin \theta \cdot y_O]'$$

$$\beta_Y = [\cos \theta \quad -\sin \theta \quad y_P \quad -\cos \theta \cdot x_O \quad \sin \theta \cdot y_O]'$$

采用最小二乘法,从式(5)和(6)得到估计参数

$$\beta_X = (V_A' \times V_A)^{-1} \times V_A' \times X_B, \tag{7}$$

$$\beta_Y = (V_A' \times V_A)^{-1} \times V_A' \times Y_B, \tag{8}$$

然后,从式(7)和(8)计算出运动参数.

事实上,可将目标的运动分为 3 步:(1)将旋转轴平移到坐标原点;(2)目标绕旋转轴旋转 θ 度;(3)将旋转后的目标平移到新的位置.如图 2 所示,在(a)中,以点 S 为旋转轴,将 S 平移到坐标原点,然后旋转 θ ,再将 S 平移到点 P,得到运动后目标的位置;在(b)中,以点 Q 为旋转轴,将 Q 平移到坐标原点,然后旋转 θ ,再将 Q 平移到点 K,得到运动后目标的位置.比较(a)(b)可发现,旋转轴不同,平移量也不同,旋转角度相同,最后目标运动到相同的位置.

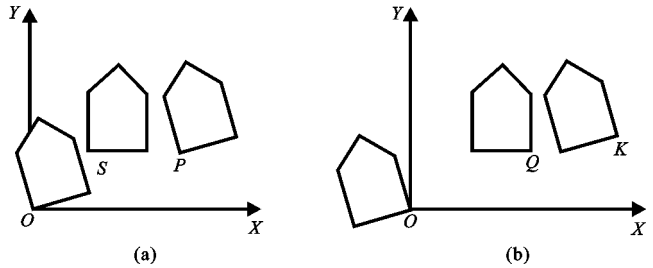


图 2 目标运动图解

因此,可在匹配点中选择最佳的一对匹配点 A、B, A 是前一帧图像中的点, B 为当前帧图像中的点.可认为目标以 A 点为旋转轴旋转,然后平移到 B 点,这样就确定了 5 个参数中的 4 个,只剩下旋转角度 θ .将式(5)和(6)重新写成

$$X_{BN} = V_{AN} \times \beta_{XN}, \tag{9}$$

$$Y_{BN} = V_{AN} \times \beta_{YN}, \tag{10}$$

其中 $X_{BN} = [x_{B_1} - x_P \quad x_{B_2} - x_P \quad \dots \quad x_{B_m} - x_P]'$, $V_{AN} = \begin{bmatrix} x_{A_1} - x_O & y_{A_1} - y_O \\ x_{A_2} - x_O & y_{A_2} - y_O \\ \dots & \dots \\ x_{A_m} - x_O & y_{A_m} - y_O \end{bmatrix}$,

$$Y_{BN} = [y_{B_1} - y_P \quad y_{B_2} - y_P \quad \dots \quad y_{B_m} - y_P]', \quad \beta_{XN} = [\cos \theta \quad -\sin \theta]', \quad \beta_{YN} = [\sin \theta \quad \cos \theta]'$$

采用最小二乘法估计,可从式(9)和(10)得到

$$\beta_{XN} = (V_{AN}' \times V_{AN})^{-1} \times V_{AN}' \times X_{BN}, \tag{11}$$

$$\beta_{YN} = (V_{AN}' \times V_{AN})^{-1} \times V_{AN}' \times Y_{BN}, \tag{12}$$

通过式(11)和(12)可得到旋转角度 θ .如果相邻两帧间的时间间隔很短,可给角度设定一个范围,这样,可加速求解和提高 θ 的精度.

3 仿真结果及分析

从一组序列图像中抽出相邻的两帧图像,如图 3 所示.图(a)(b)的圆圈内的黑点为疵点,(c)中圈内的灰点表示没有补偿好的疵点.可发现,在图(a)中,疵点位于背景中,而在图(b)中,疵点位于杯子的边缘.图(c)为对第 i 帧图像进行邻域疵点补偿的结果,可看到疵点补偿不理想,使杯子在疵点位置凹进去了,产生了伪边缘.图(d)为笔者提出的疵点补偿算法的结果,杯子的边缘保持完整.很明显,笔者提出的补偿算法效果好.图(e)(f)是两帧图像的特征点图.

在仿真过程中发现,当目标仅做平移运动或者旋转角度很小($-10^\circ \leq \alpha \leq 10^\circ$)时,图像匹配的精度很

