

```

%-----
%                                     基于中位数算法的运动目标检测
%                                     第 1 种实现方法
%-----

%*****读取视频数据*****
% 调用 aviread 函数读取视频文件
vid = aviread('WalkingMan.avi')

size(vid(1).cdata)    % 查看第 1 帧的大小，也是每一帧的大小

vid(1).colormap    % 查看第 1 帧的 colormap 值

%*****数据类型的转换*****
% 把 vid 的 cdata 字段的取值转换成一个  $240 \times 360 \times 3 \times 80$  的四维数组 IM
IM = cat(4, vid.cdata);
size(IM)    % 查看 IM 的大小

[m, n, k, h] = size(IM);    % 把 IM 的大小赋给四个变量
% 把 IM 转换成一个 80 行， $240 \times 360 \times 3$  列的矩阵
I = reshape(IM, [m*n*k, h])';

%*****调用 median 函数求中位数，进行视频图像分割*****
I = median(I);    % 求 I 矩阵中各列元素的中位数
I = reshape(I, [m, n, k]);    % 将向量 I 转成  $240 \times 360 \times 3$  的三维数组，得到背景图像
figure;    % 新建一个图形窗口
imshow(I);    % 显示背景图像
figure;    % 新建一个图形窗口
imshow(IM(:, :, :, 1) - I);    % 显示第 1 帧中的目标图像

```

```

%-----
%                                     基于中位数算法的运动目标检测
%                                     第 1 种实现方法
%-----

```

```

% 调用 mmreader 函数创建读取视频文件的多媒体阅读对象 WalkManObj
WalkManObj = mmreader('WalkingMan.avi');
% 根据多媒体阅读对象 WalkManObj，读取视频的各帧图像数据
IM = read(WalkManObj, [1, inf]);
[m, n, k, h] = size(IM);    % 把 IM 的大小赋给四个变量

```

```

% 把 IM 转换成一个 80 行，240×360×3 列的矩阵
I = reshape(IM, [m*n*k, h])';
I = median(I);    % 求 I 矩阵中各列元素的中位数
I = reshape(I, [m, n, k]);    % 将向量 I 转成 240×360×3 的三维数组，得到背景图像
figure;    % 新建一个图形窗口
imshow(I);    % 显示背景图像
figure;    % 新建一个图形窗口
imshow(IM(:, :, :, 1) - I);    % 显示第 1 帧中的目标图像

```

如下图效果



