

问题一：

编写 MATLAB 函数，实现将多帧 BMP 格式的静止图像自动转换为 AVI 文件。

源程序代码：

```
function BMP_AVI(framesPath,videoName,quality,fps,startFrame,endFrame)
    %BMP_AVI 函数参数表：
    %framesPath: 图像序列所在路径，
    %videoName: 表示将要创建的视频文件的名字
    %quality: 生成视频的质量 0-100
    %fps: 帧率
    %startFrame,endFrame: 表示从哪一帧开始，哪一帧结束
    if(exist('videoName','file'))
        delete videoName.avi
    end
    %生成视频的参数设定
    aviobj=VideoWriter(videoName);    %创建一个 avi 视频文件对象，开始时其为空
    aviobj.Quality=quality;    %设置视频质量
    aviobj.FrameRate=fps;    %设置视频帧率
    %读入图像
    for i=startFrame:endFrame
        fileName=sprintf('%d',i);    %sprintf 函数将图像名字的数字转换为字符格式
        frames=imread([framesPath,fileName,'.bmp']);    %根据文件路径读取图像
        open(aviobj);    %打开设置好格式的空白.avi 文件
        writeVideo(aviobj,uint8(frames));    %输出视频
    end
    close(aviobj);    % 关闭视频
end
```

在命令行输入命令：

```
Path = 'E:\MATLAB exercise\BMP_AVI\bmp\';
    %bmp 格式的图像存储路径
BMP_AVI(Path,'test',100,24,1,591)
    %设置 BMP 图像的路径
    %创建视频文件名称为 movie,
    %设置视频质量为 100,
    %设置视频帧率 24fps,
    %设置视频从第一帧开始，第 591 帧结束
```



图 1.1 多帧 BMP 转 AVI 文件源程序



图 1.2 运行程序后生成 AVI 文件

问题二：

编写 MATLAB 函数, 实现将 AVI 中的视频帧单独存储为 BMP 格式的静止图像。

源程序代码：

```
function AVI_BMP(Path)

mov = VideoReader(Path);      %读取视频

number = mov.NumberOfFrames; %读取帧数

for i = 1 : number

    frame = read(mov,i);      %每帧读取一次

    name=sprintf('%d',i);      %sprintf 函数将图像名字的数字转换为字符格式

    imwrite(frame,strcat(name,'.bmp'),'bmp'); %将图片以 bmp 格式保存

end

end
```

在命令行输入命令：

AVI_BMP('E:\MATLAB exercise\AVI_BMP\test.avi') %输入 AVI 文件存储路径

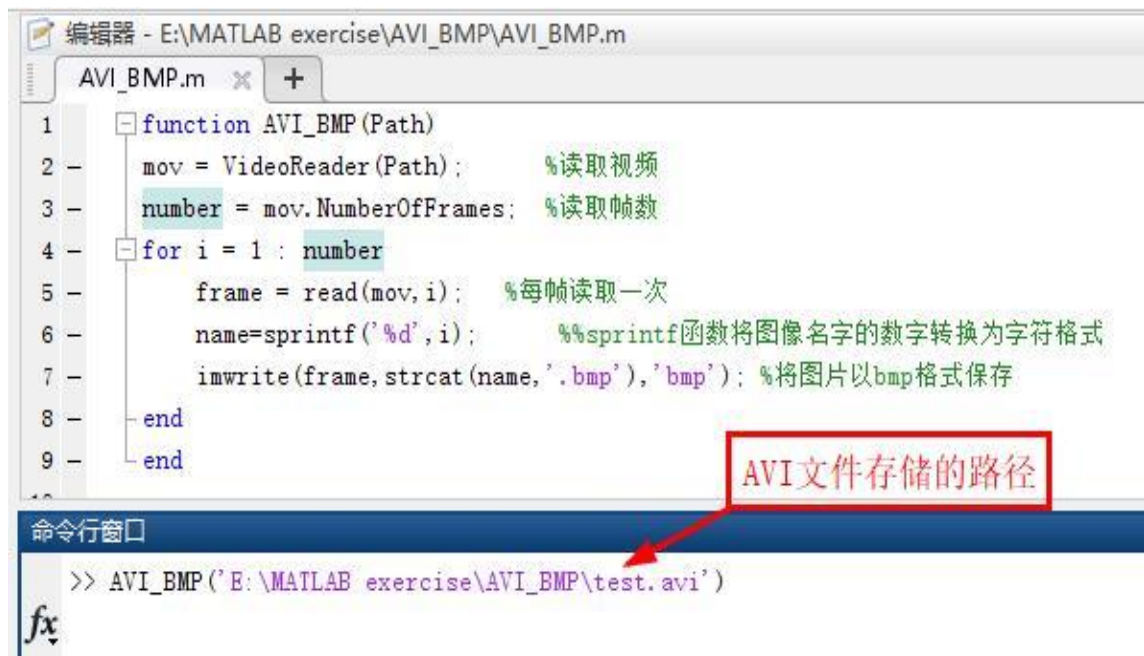


图 2.1 AVI 文件转 BMP 图像源程序

运行结果：

将 AVI 文件中的视频每一帧均保存，累计保存 591 张

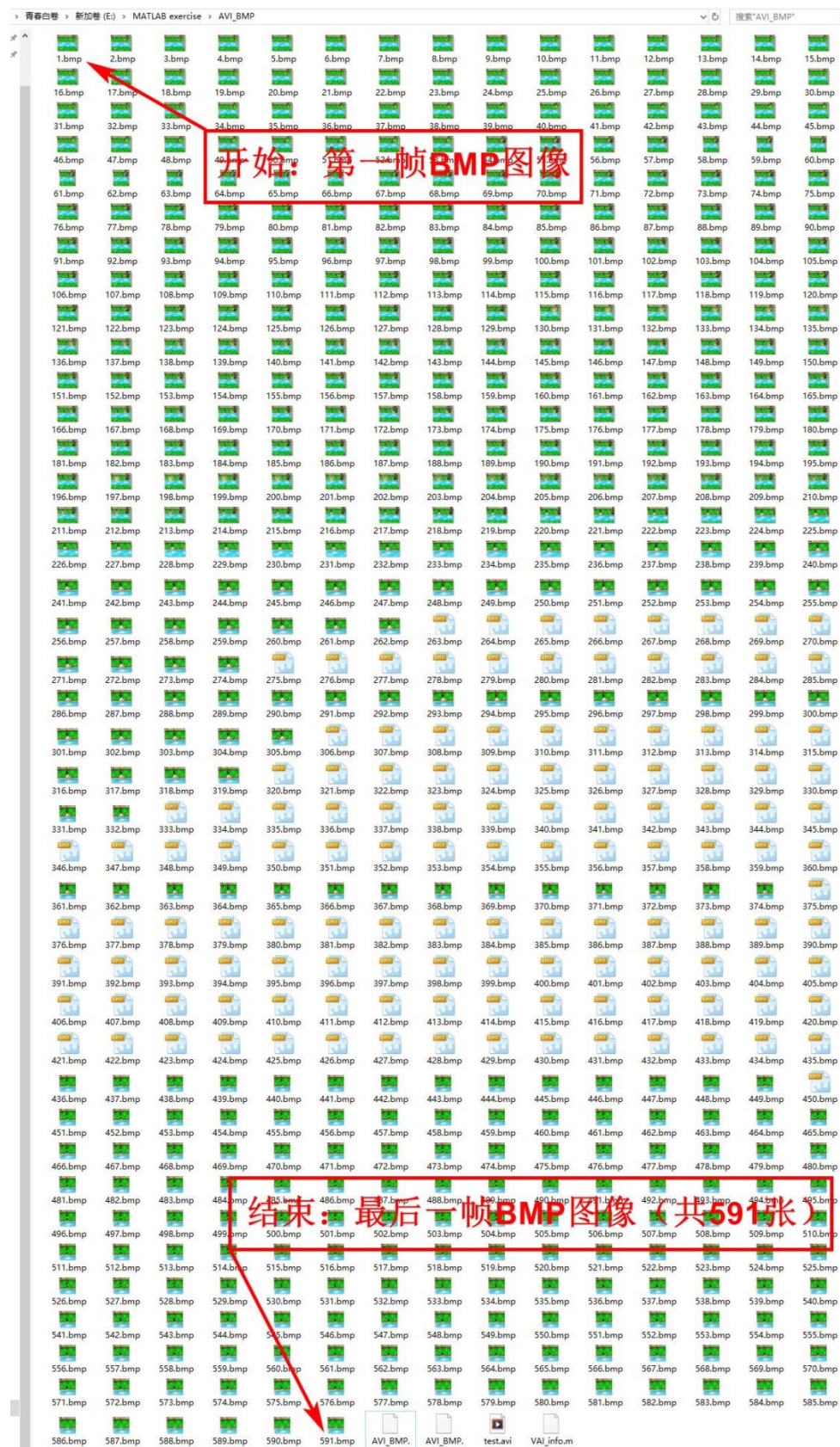


图 2.2 运行程序后生成 BMP 格式的静止图像

验证结果：

- 1、编写 AVI_info 程序，获取 AVI 文件相关信息。

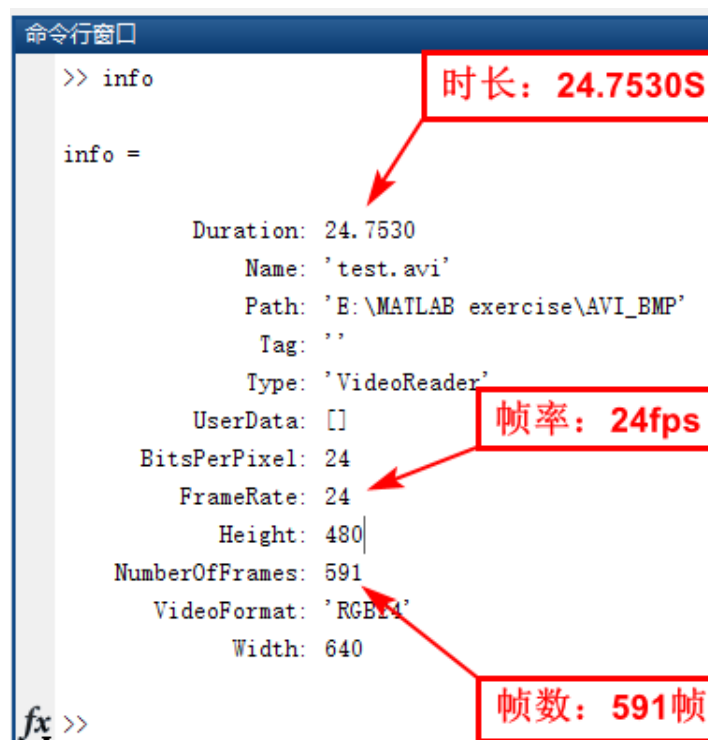


```
编辑器 - E:\MATLAB exercise\AVI_BMP\VAI_info.m
AVI_BMP.m*  VAI_info.m  +
1 -   mov=VideoReader('E:\MATLAB exercise\AVI_BMP\test.avi'); %读取视频
2 -   lastframe=read(mov,inf); %读取视频有关信息
3 -   info=get(avi); %获取
4 -
```

图 2.3 AVI_info 程序源代码

- 2、运行结果：

用来测试的 test.avi 文件的帧数是 591 帧，即验证了问题二的结果。帧率是 24fps，为问题一提供有效依据。



```
命令窗口
>> info

info =

    Duration: 24.7530
      Name: 'test.avi'
     Path: 'E:\MATLAB exercise\AVI_BMP'
       Tag: ''
    Type: 'VideoReader'
   UserData: []
 BitsPerPixel: 24
   FrameRate: 24
      Height: 480
NumberOfFrames: 591
 VideoFormat: 'RGB14'
        Width: 640

fx >>
```

图 2.4 运行 AVI_info 程序获得 AVI 文件相关信息

拓展:

由问题二可知, 将 test.avi 文件每帧进行存储可存储 591 张, 那么将 test.avi 文件每 5 帧存储 1 张可以存储多少张? 每 10 帧呢?

修改程序:

The figure displays three sequential screenshots of the MATLAB editor showing the modification of the AVI_BMP.m script. In each screenshot, the line `frame = read(mov, i);` is circled in red to indicate the change.

Top Screenshot: The original script. The for loop iterates from 1 to the total number of frames. The circled line is `frame = read(mov, i);` with the comment `%每帧读取一次` (read once per frame).

Middle Screenshot: The script is modified to read every 5th frame. The circled line is `frame = read(mov, 5*i);` with the comment `%每5帧读取一次` (read once every 5 frames). The comment for the `name=sprintf` line is updated to `%以整形数命名读取到的图片` (name the read image with an integer).

Bottom Screenshot: The script is modified to read every 10th frame. The circled line is `frame = read(mov, 10*i);` with the comment `%每10帧读取一次` (read once every 10 frames). The comment for the `name=sprintf` line remains `%以整形数命名读取到的图片`.

图 2.5 每 1 帧、每 5 帧、每 10 帧存储一次程序对比

运行结果：

test.avi 文件每 5 帧存储 1 张可以存储 118 张，每 10 帧存储 1 张可以存储 59 张。



图 2.6 每 5 帧存储一次的运行结果（118 张）

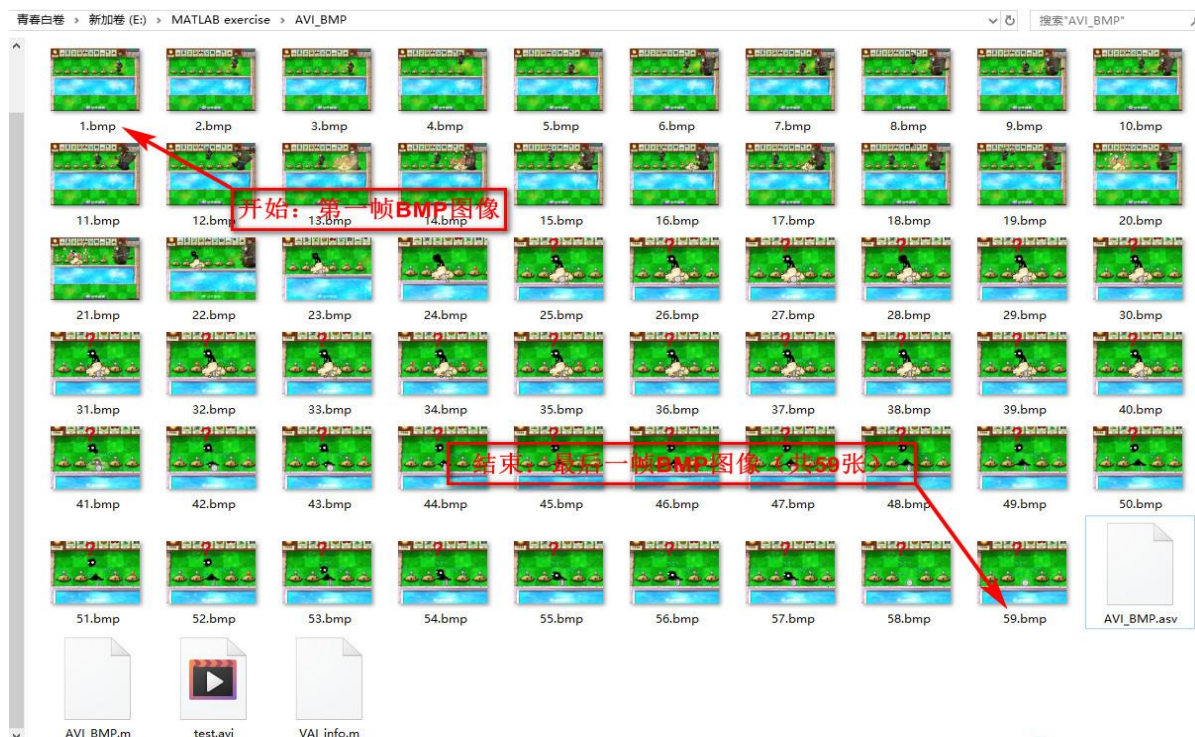


图 2.6 每 10 帧存储一次的运行结果（59 张）