如何对表示信号和图像的整数数据执行算术运算。

1装入整数信号数据

负载测量数据集，包括使用8位和16位A到D的四种仪器的信号，导致数据保存为int8、int16和uint16。时间存储为uint16。

程序1

load integersignal

% Look at variables

whos Signal1 Signal2 Signal3 Signal4 Time1

2 图数据

首先，我们将绘制两个信号来查看信号范围。

程序2

plot(Time1, Signal1, Time1, Signal2);

grid;

legend('Signal1','Signal2');

这里我们看到int8的值。很可能需要对这些值进行缩放，以计算信号所代表的实际物理值，例如伏特。

处理数据

我们可以对整数(如+、-、\*和/)执行标准算术。假设我们想求出Signal1和Signal2的和。

SumSig = Signal1 + Signal2; 这里我们对整数信号求和。

现在我们画出信号的和，看看它在哪里饱和。

程序3

cla;

plot(Time1, SumSig);

hold on;

Saturated = (SumSig == intmin('int8')) | (SumSig == intmax('int8')); % Find where it has saturated

plot(Time1(Saturated),SumSig(Saturated),'rd');grid;

hold off;

标记显示信号已饱和的位置。

加载整数图像数据

接下来我们将研究一些图像数据的算法

street1=imread('street1.jpg'); % Load image data

street2=imread('street2.jpg');

whos street1 street2

这里我们看到的图像是24位彩色的，存储为uint8数据的三个平面。

首先显示图像。

cla;

image(street1); % Display image

axis equal; axis off

显示第二个图片

image(street2); % Display image

axis equal; axis off

规模一个图像

我们可用双精度常数对图像进行缩放，但要将图像存储为整数。例如,

程序1

duller = 0.5 \* street2; % 使用双常数缩放图像，但创建一个整数

whos duller

程序2

subplot(1,2,1);

image(street2);

axis off equal tight

title('Original'); % Display image

subplot(1,2,2);

image(duller);

axis off equal tight

title('Duller'); % Display image

添加图片

我们可以把这两个街道图像加在一起，画出可怕的结果。

combined = street1 + duller; % Add |uint8| images

subplot(1,1,1)

cla;

image(combined); % Display image

title('Combined');

axis equal; axis off