Pixel Statistics & Template Matching

Dr. Daniel B. Rowe
Professor of Computational Statistics
Department of Mathematical and Statistical Sciences
Marquette University

Outline

Introduction

Pixel Statistics

Template Matching

Discussion

Homework

Introduction

We previously explored the statistical implications of performing image convolution via a convolution kernel.

We used a constant image to examine the statistical effects of performing linear processing operations on a noisy image.

In practice, images are not constant and we may not have homogeneous regions to examine the statistical properties of individual pixels.

When we have heterogeneous regions we generally need repeated images for individual pixel statistics. When we don’t have repeated images, we can use convolution to compute approximate pixel statistics.

Introduction

Imagine that we have the heterogeneous car scene image.

Original

How do we calculate the mean value of a pixel if we do not have repeated images?

How do we compute a pixel's variance or standard deviation?

How do we calculate the correlation between pixels in this heterogeneous image?

Introduction

Imagine that we have the heterogeneous car scene image.

Original

We can use a convolution kernel to calculate local statistics at each pixel.

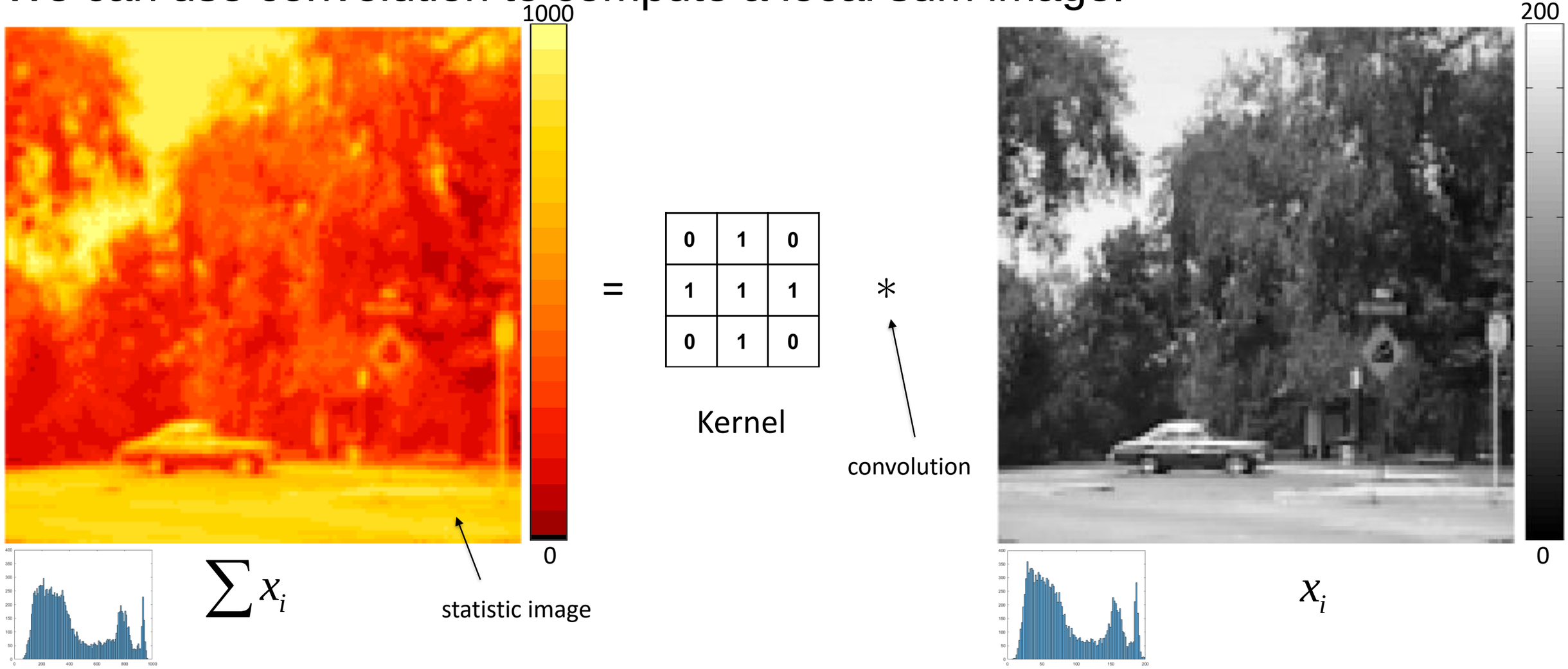
0	1	0
1	1	1
0	1	0

Average value of 5 pixels for center pixel's mean value?

Variance of 5 pixels for variance of center pixel?

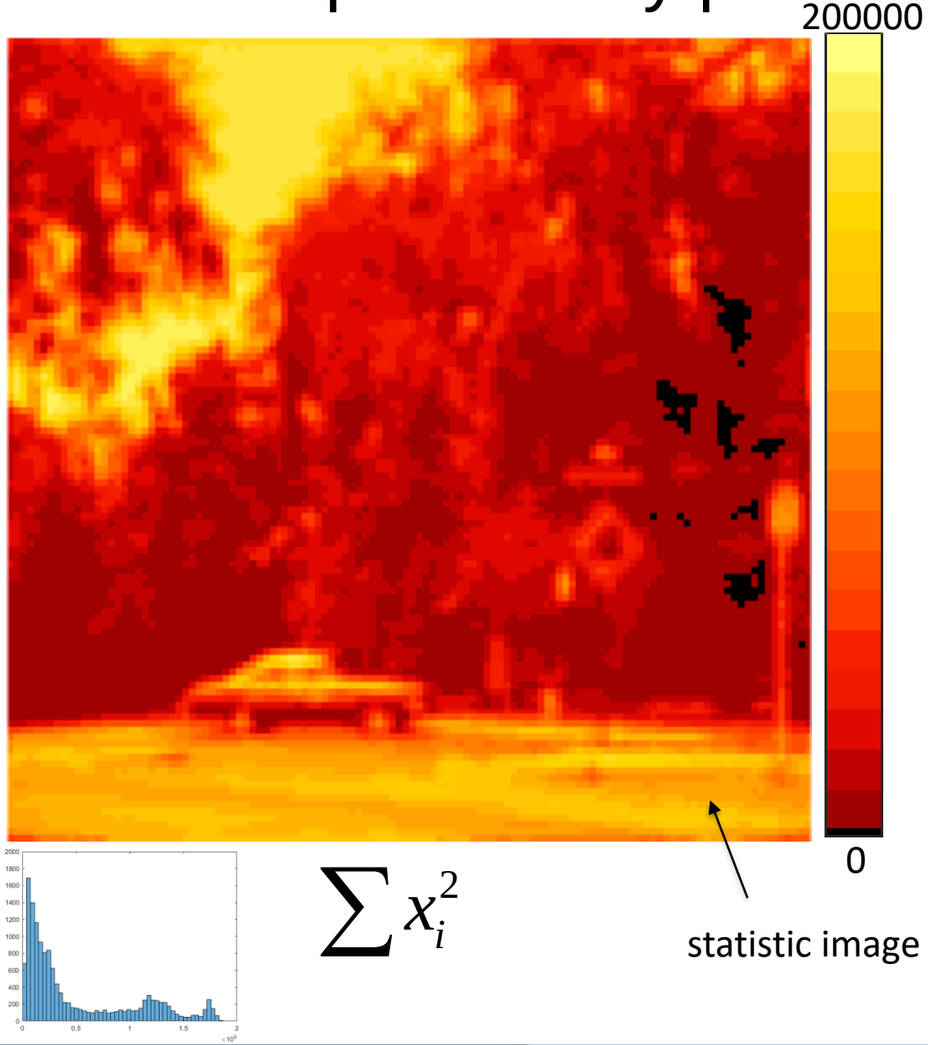
Pixel Statistics

We can use convolution to compute a local sum image.



Pixel Statistics

We can square every pixel value and compute a sum of squares.



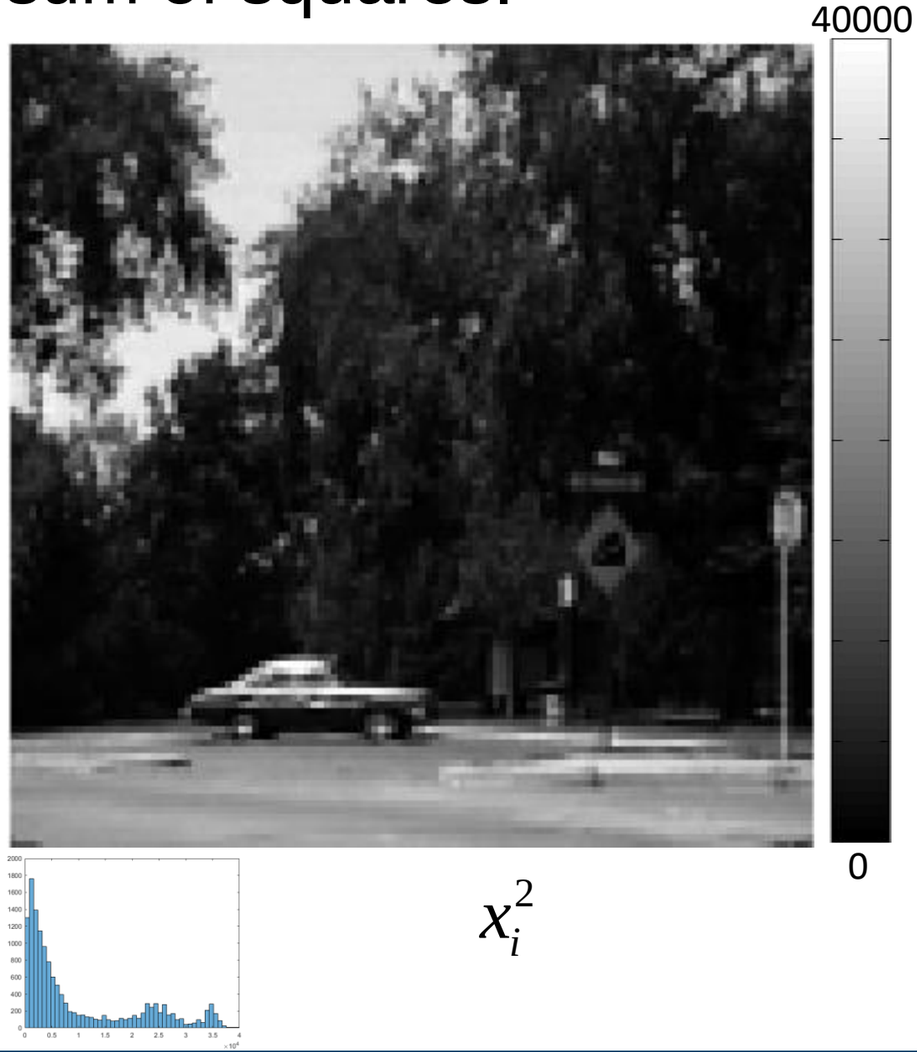
=

0	1	0
1	1	1
0	1	0

Kernel

$*$

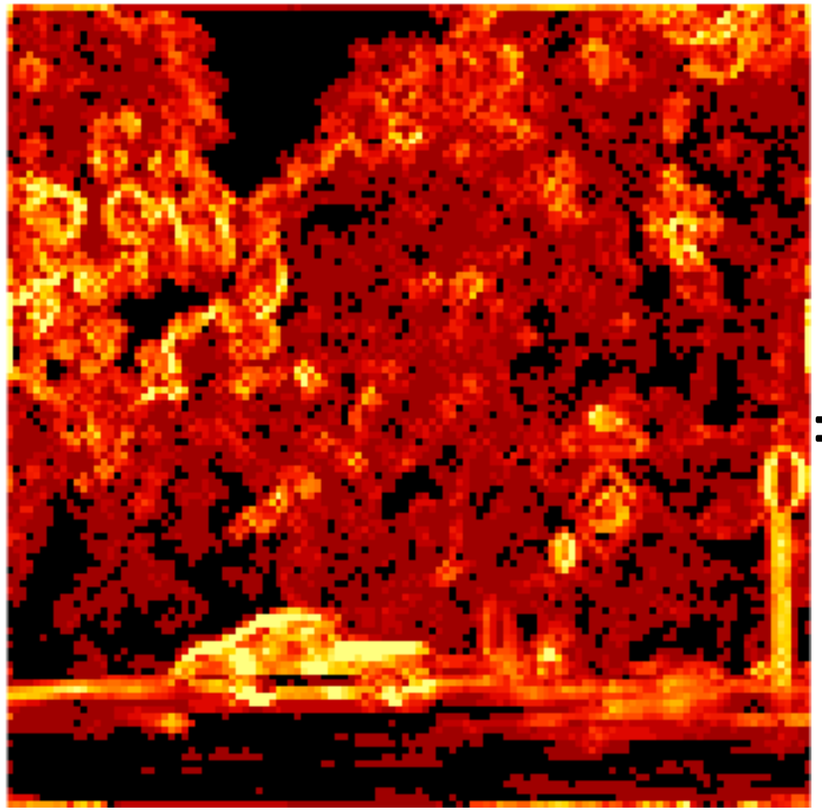
convolution



Pixel Statistics

$$s^2 = \frac{\sum x_i^2 - \left(\sum x_i\right)^2 / n}{n-1}$$

We can use sum and sum of squares to compute usual the local variance.



=

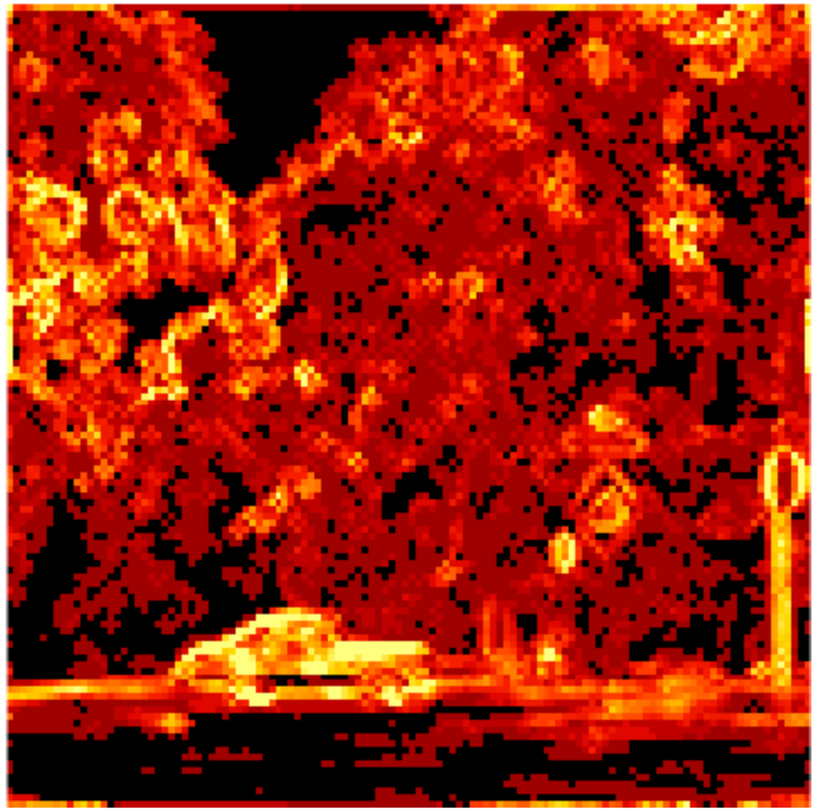
-

^{2} / n

n-1

Pixel Statistics

The variance is low in homogeneous and high in heterogeneous areas.



Variance

Original

Mean

Perhaps we could smooth more in low variance areas?

Pixel Statistics

```
load myposmapblk.txt

limMin=0; limMaxI=200; yhist=400; yhist2=2000;

kernel=[0,1,0;...
        1,1,1;...
        0,1,0];
ksum=sum(kernel(:));

load cardata.txt
[n,p]=size(cardata);
nx=sqrt(n); ny=nx;
I=double(reshape(cardata,[ny,nx])');

Ibar=mean(I(:)), s2I=var(I(:))

figure;
imagesc(I,[limMin,limMaxI])
colormap(gray), axis image, axis off
figure;
histogram(I(:),(limMin:2:limMaxI))
xlim([limMin,limMaxI]), ylim([0,yhist])
```

```
I2=I.^2;
I2bar=mean(I2(:)), s2I2=var(I2(:))
figure;
imagesc(I2,[limMin,limMaxI^2])
colormap(gray), axis image, axis off

figure;
histogram(I2(:),50)
xlim([limMin,limMaxI^2]), ylim([0,yhist2])

% calculate local sum statistics
O=MyConv(I,kernel);
Obar=mean(O(:)), s2O=var(O(:))
figure;
imagesc(O,[limMin,ksum*limMaxI])
colormap(myposmapblk), axis image, axis off

figure;
histogram(O(:),(limMin:8:5*limMaxI))
xlim([limMin,5*limMaxI]), ylim([0,yhist])
```

Pixel Statistics

```
% compute local sum of image square
O2=MyConv(I2,kernel);
O2bar=mean(O2(:)), s2O2=var(O2(:))
figure;
imagesc(O2,[limMin,ksum*limMaxI^2])
colormap(myposmapblk), axis image, axis off

figure;
histogram(O2(:),50)
xlim([limMin,ksum*limMaxI^2]),
ylim([0,yhist2])

% compute local variance
S2=(O2-(O.^2)/ksum)/(ksum-1);

figure;
imagesc(S2,[0,2000])
colormap(myposmapblk), axis image, axis off
```

Template Matching

In Machine Vision, we often want to find a specific item in an image.

For example, if I have a “template image” (kernel) for an object, and want to look in an image for the object. How do I do this?

What if I had a picture of somebody I wanted to find, and wanted to look through a large number of images for that person?

This problem is called template matching.

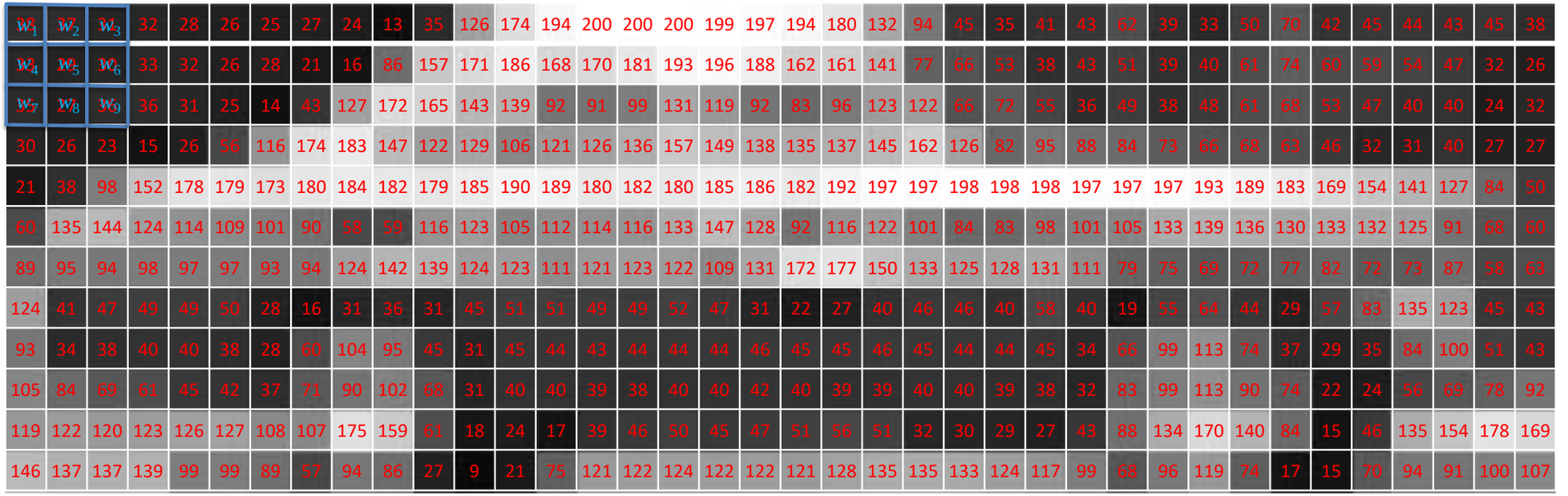
Template Matching

We just demonstrated that we could compute $\sum p_i$ and $\sum p_i^2$ to compute s^2 for a pixel within a neighborhood. We used "weights" that were ones.

$$q = 1(78) + 1(154) + 1(178) + 1(169) + 1(100) \equiv 679$$

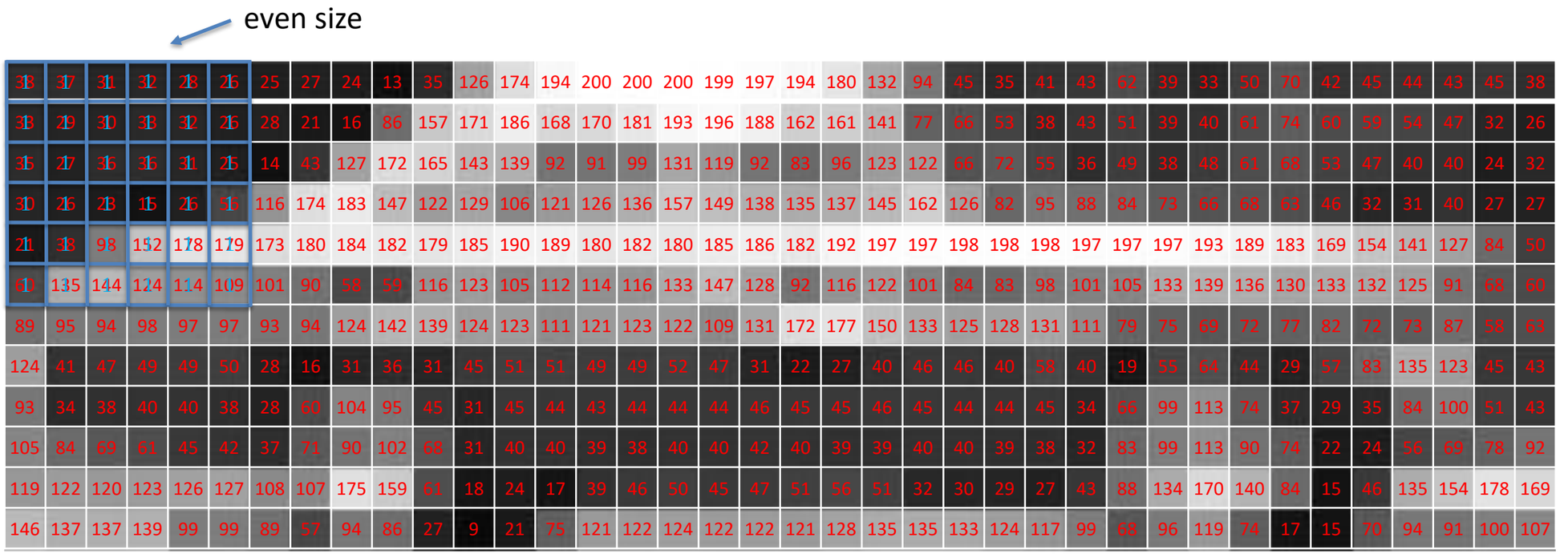
$$q = 1(78)^2 + 1(154)^2 + 1(178)^2 + 1(169)^2 + 1(100)^2 \equiv 100045$$

0	1	0
1	1	1
0	1	0



Template Matching

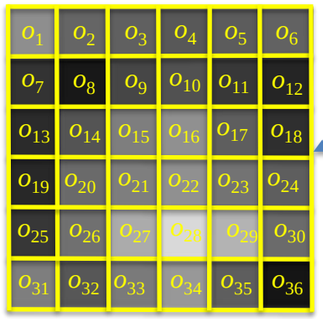
We could use a larger template to compute $\sum p_i$ and $\sum p_i^2$ to compute s^2 for a pixel within a neighborhood.



Template Matching

But I like car wheels, I have my favorite image of a wheel (object) and can compute $\sum o_i$ and $\sum o_i^2$.

car wheel



car wheel

Template Matching

I now have $\sum o_i$ and $\sum o_i^2$ for my object of interest.

o_1	o_2	o_3	o_4	o_5	o_6
o_7	o_8	o_9	o_{10}	o_{11}	o_{12}
o_{13}	o_{14}	o_{15}	o_{16}	o_{17}	o_{18}
o_{19}	o_{20}	o_{21}	o_{22}	o_{23}	o_{24}
o_{25}	o_{26}	o_{27}	o_{28}	o_{29}	o_{30}
o_{31}	o_{32}	o_{33}	o_{34}	o_{35}	o_{36}

car wheel

Template Matching

I now have $\sum o_i$ and $\sum o_i^2$ for my object of interest along with $\sum p_i$ and $\sum p_i^2$. for every neighborhood the same size as my object or "template" in the larger scene image.

o_1	o_2	o_3	o_4	o_5	o_6
o_7	o_8	o_9	o_{10}	o_{11}	o_{12}
o_{13}	o_{14}	o_{15}	o_{16}	o_{17}	o_{18}
o_{19}	o_{20}	o_{21}	o_{22}	o_{23}	o_{24}
o_{25}	o_{26}	o_{27}	o_{28}	o_{29}	o_{30}
o_{31}	o_{32}	o_{33}	o_{34}	o_{35}	o_{36}

car wheel

Template Matching

I now have $\sum o_i$ and $\sum o_i^2$ for my object of interest along with $\sum p_i$ and $\sum p_i^2$.
for every neighborhood the same size as my object image or "template" in the larger scene image.
Now all I need to do is compute $\sum o_i p_i$.

o_1	o_2	o_3	o_4	o_5	o_6
o_7	o_8	o_9	o_{10}	o_{11}	o_{12}
o_{13}	o_{14}	o_{15}	o_{16}	o_{17}	o_{18}
o_{19}	o_{20}	o_{21}	o_{22}	o_{23}	o_{24}
o_{25}	o_{26}	o_{27}	o_{28}	o_{29}	o_{30}
o_{31}	o_{32}	o_{33}	o_{34}	o_{35}	o_{36}

car wheel

38	37	31	32	28	26	25	27	24	13	35	126	174	194	200	200	200	199	197	194	180	132	94	45	35	41	43	62	39	33	50	70	42	45	44	43	45	38
33	29	30	33	32	26	28	21	16	86	157	171	186	168	170	181	193	196	188	162	161	141	77	66	53	38	43	51	39	40	61	74	60	59	54	47	32	26
35	27	36	36	31	25	14	43	127	172	165	143	139	92	91	99	131	119	92	83	96	123	122	66	72	55	36	49	38	48	61	68	53	47	40	40	24	32
30	26	23	15	26	56	116	174	183	147	122	129	106	121	126	136	157	149	138	135	137	145	162	126	82	95	88	84	73	66	68	63	46	32	31	40	27	27
21	38	98	152	178	179	173	180	184	182	179	185	190	189	180	182	180	185	186	182	192	197	197	198	198	198	197	197	197	193	189	183	169	154	141	127	84	50
60	135	144	124	114	109	101	90	58	59	116	123	105	112	114	116	133	147	128	92	116	122	101	84	83	98	101	105	133	139	136	130	133	132	125	91	68	60
89	95	94	98	97	97	93	94	124	142	139	124	123	111	121	123	122	109	131	172	177	150	133	125	128	131	111	79	75	69	72	77	82	72	73	87	58	63
124	41	47	49	49	50	28	16	31	36	31	45	51	51	49	49	52	47	31	22	27	40	46	46	40	58	40	19	55	64	44	29	57	83	135	123	45	43
93	34	38	40	40	38	28	60	104	95	45	31	45	44	43	44	44	44	46	45	45	46	45	44	44	45	34	66	99	113	74	37	29	35	84	100	51	43
105	84	69	61	45	42	37	71	90	102	68	31	40	40	39	38	40	40	42	40	39	39	40	40	39	38	32	83	99	113	90	74	22	24	56	69	78	92
119	122	120	123	126	127	108	107	175	159	61	18	24	17	39	46	50	45	47	51	56	51	32	30	29	27	43	88	134	170	140	84	15	46	135	154	178	169
146	137	137	139	99	99	89	57	94	86	27	9	21	75	121	122	124	122	122	121	128	135	135	133	124	117	99	68	96	119	74	17	15	70	94	91	100	107

Template Matching

Which means that for every location I have

$$\sum p_i \quad \sum p_i^2 \quad \sum o_i \quad \sum o_i^2 \quad \sum o_i p_i$$

precomputed

correlation

$$r = \frac{\sum p_i o_i - \frac{1}{n}\left(\sum p_i\right)\left(\sum o_i\right) / n}{\sqrt{\sum p_i^2 - \frac{1}{n}\left(\sum p_i\right)^2}\sqrt{\sum o_i^2 - \frac{1}{n}\left(\sum o_i\right)^2}}$$

o_1	o_2	o_3	o_4	o_5	o_6
o_7	o_8	o_9	o_{10}	o_{11}	o_{12}
o_{13}	o_{14}	o_{15}	o_{16}	o_{17}	o_{18}
o_{19}	o_{20}	o_{21}	o_{22}	o_{23}	o_{24}
o_{25}	o_{26}	o_{27}	o_{28}	o_{29}	o_{30}
o_{31}	o_{32}	o_{33}	o_{34}	o_{35}	o_{36}

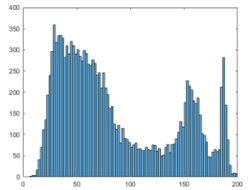
38	37	31	32	28	26	25	27	24	13	35	126	174	194	200	200	200	199	197	194	180	132	94	45	35	41	43	62	39	33	50	70	42	45	44	43	45	38
33	29	30	33	32	26	28	21	16	86	157	171	186	168	170	181	193	196	188	162	161	141	77	66	53	38	43	51	39	40	61	74	60	59	54	47	32	26
35	27	36	36	31	25	14	43	127	172	165	143	139	92	91	99	131	119	92	83	96	123	122	66	72	55	36	49	38	48	61	68	53	47	40	40	24	32
30	26	23	15	26	56	116	174	183	147	122	129	106	121	126	136	157	149	138	135	137	145	162	126	82	95	88	84	73	66	68	63	46	32	31	40	27	27
21	38	98	152	178	179	173	180	184	182	179	185	190	189	180	182	180	185	186	182	192	197	197	198	198	198	197	197	197	193	189	183	169	154	141	127	84	50
60	135	144	124	114	109	101	90	58	59	116	123	105	112	114	116	133	147	128	92	116	122	101	84	83	98	101	105	133	139	136	130	133	132	125	91	68	60
89	95	94	98	97	97	93	94	124	142	139	124	123	111	121	123	122	109	131	172	177	150	133	125	128	131	111	79	75	69	72	77	82	72	73	87	58	63
124	41	47	49	49	50	28	16	31	36	31	45	51	51	49	49	52	47	31	22	27	40	46	46	40	58	40	19	55	64	44	29	57	83	135	123	45	43
93	34	38	40	40	38	28	60	104	95	45	31	45	44	43	44	44	44	46	45	45	46	45	44	44	45	34	66	99	113	74	37	29	35	84	100	51	43
105	84	69	61	45	42	37	71	90	102	68	31	40	40	39	38	40	40	42	40	39	39	40	40	39	38	32	83	99	113	90	74	22	24	56	69	78	92
119	122	120	123	126	127	108	107	175	159	61	18	24	17	39	46	50	45	47	51	56	51	32	30	29	27	43	88	134	170	140	84	15	46	135	154	178	169
146	137	137	139	99	99	89	57	94	86	27	9	21	75	121	122	124	122	122	121	128	135	135	133	124	117	99	68	96	119	74	17	15	70	94	91	100	107

Template Matching

We can use this idea and the entire car as an object “template!”

$\star$

correlation



$$= \quad r = \frac{\sum p_i o_i - \frac{1}{n}\left(\sum p_i\right)\left(\sum o_i\right) / n}{\sqrt{\sum p_i^2 - \frac{1}{n}\left(\sum p_i\right)^2}\sqrt{\sum o_i^2 - \frac{1}{n}\left(\sum o_i\right)^2}}$$

Template Matching

We want to perform computations as fast as possible.

For our object template we can pre-compute

$\sum o_i$ $\sum o_i^2$

Then as we loop through our image, compute

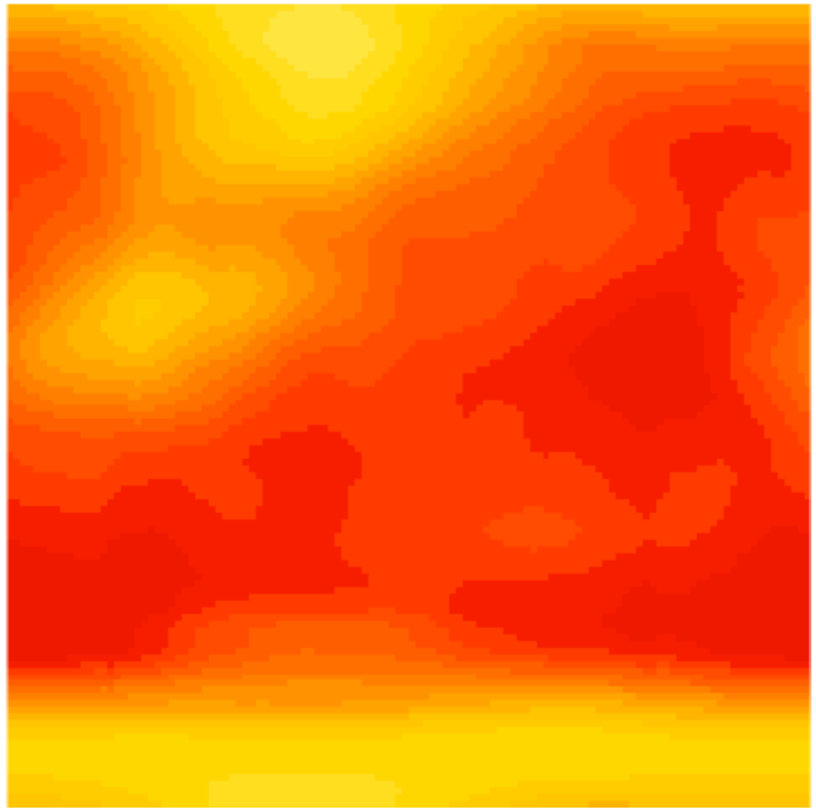
$\sum p_i$ $\sum p_i^2$ $\sum o_i p_i$

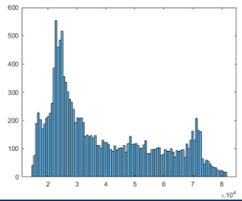
$$r = \frac{\sum p_i o_i - \frac{1}{n}\left(\sum p_i\right)\left(\sum o_i\right)}{\sqrt{\sum p_i^2 - \frac{1}{n}\left(\sum p_i\right)^2}\sqrt{\sum o_i^2 - \frac{1}{n}\left(\sum o_i\right)^2}}$$

Template Matching

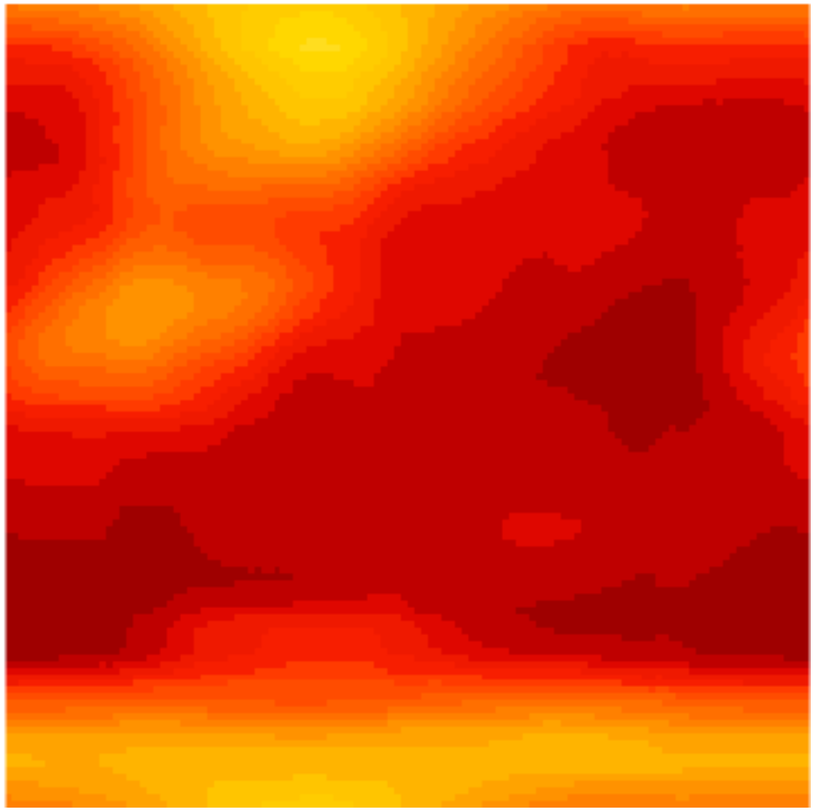
$\sum o_i = 40120$ $\sum o_i^2 = 4781080$

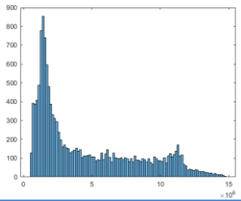
Here are our statistic images.



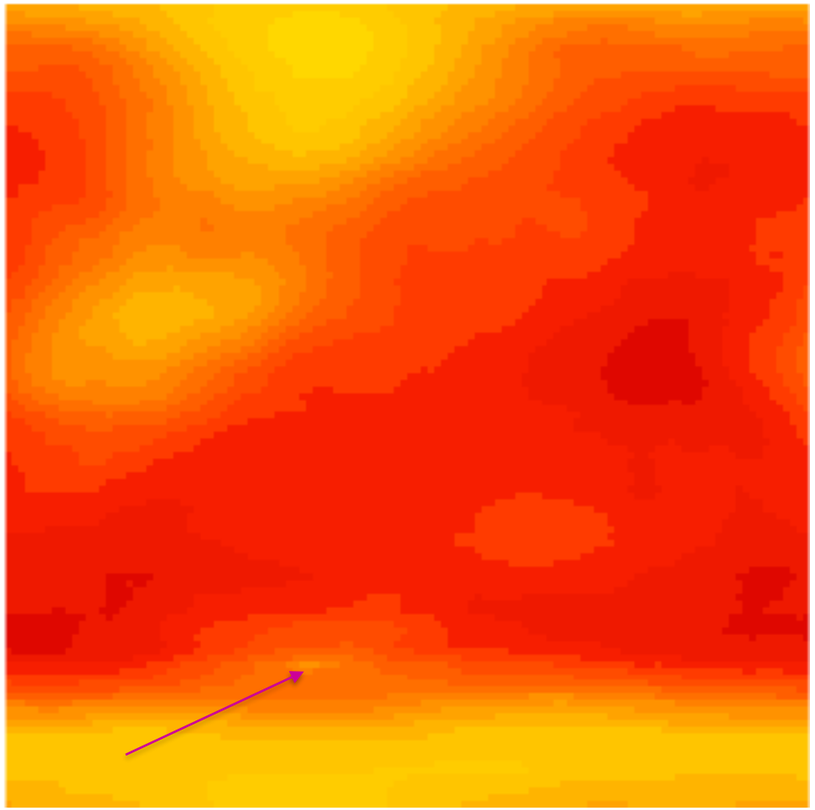


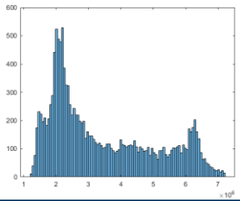
$\sum p_i$





$\sum p_i^2$

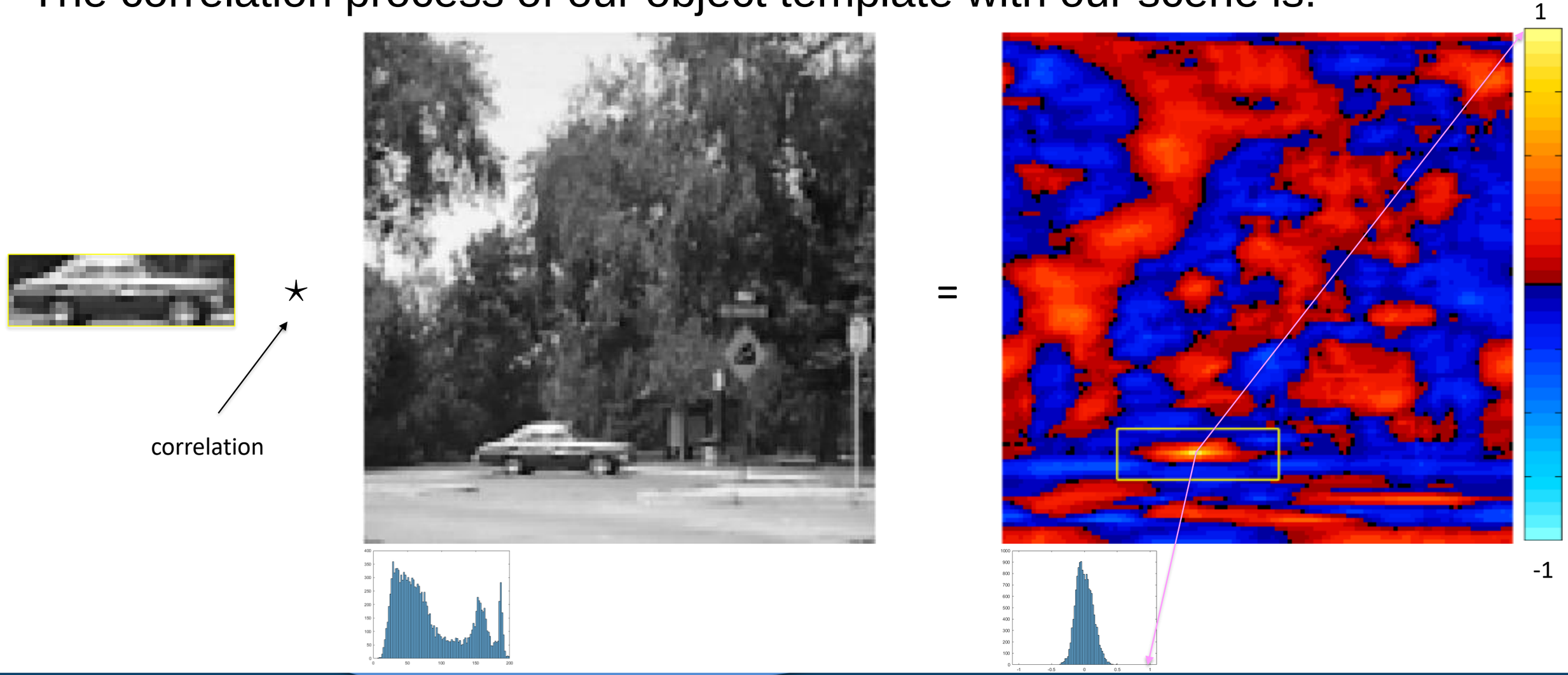




$\sum o_i p_i$

Template Matching

The correlation process of our object template with our scene is:



Template Matching

```
% template matching code
load myposnegmapblk.txt
load myposmapblk.txt

% load oiginal car scene image
load cardata.txt
[n,p]=size(cardata);
nx=sqrt(n); ny=nx;
I=reshape(double(cardata),[ny,nx])';

limMin=0; limMax=200;

figure;
imagesc(I,[limMin,limMax])
axis image, colormap(gray), axis off

car=I(93:104,27:64);
% extract car portion
figure;
imagesc(car,[limMin,limMax])
axis image, colormap(gray), axis off
```

```
% template match for car
template=car;
[a,b]=size(template);
tsum=sum(template(:)); t2sum=sum(template(:).^2);
[O,psum,p2sum,ptsum]=MyCorr(I,template);
Omax=max(max(O));
[indy,indx]=find(O==Omax)
indy0=indy; indx0=indx;
% display pixel sums
figure;
imagesc(psum,[0,a*b*limMax])
axis image, colormap(myposmapblk), axis off
figure;
histogram(psum(:),100)
figure;
imagesc(p2sum,[0,a*b*limMax^2])
axis image, colormap(myposmapblk), axis off
figure;
histogram(p2sum(:),100)
figure;
imagesc(ptsum,[0,a*b*limMax^2/2])
axis image, colormap(myposmapblk), axis off
figure;
histogram(ptsum,100)
```

Template Matching

```
figure;
imagesc(O,[-1,1])
axis image, colormap(myposnegmapblk), axis off
line([0.5+indx-b/2,0.5+indx-b/2],[0.5+indy-a/2,0.5+indy+a/2],'Color',[1,1,0],'LineWidth',1.0)
line([0.5+indx+b/2,0.5+indx+b/2],[0.5+indy-a/2,0.5+indy+a/2],'Color',[1,1,0],'LineWidth',1.0)
line([0.5+indx-b/2,0.5+indx+b/2],[0.5+indy-a/2,0.5+indy-a/2],'Color',[1,1,0],'LineWidth',1.0)
line([0.5+indx-b/2,0.5+indx+b/2],[0.5+indy+a/2,0.5+indy+a/2],'Color',[1,1,0],'LineWidth',1.0)

figure;
histogram(O(:),[-1:.02:1])
```

Template Matching

```
function [O,psum,p2sum,ptsum]=MyCorr(I,template);

[a,b]=size(template);
[n,m]=size(I);
tvec=reshape(template',[a*b,1]);
O=zeros(n,m); psum=zeros(n,m); p2sum=zeros(n,m); ptsum=zeros(n,m);
is_a_even=~mod(a,2); is_b_even=~mod(b,2);
if (is_a_even==0)&&(is_b_even==0) % appends border pixels for wrap-around
    IW=[I(n-(a-1)/2:n,m-(b-1)/2:m),I(n-(a-1)/2+1:n,1:m),I(n-(a-1)/2+1:n,1:(b-1)/2);...
        I(1:n          ,m-(b-1)/2+1:m),I(1:n          ,1:m),I(1:n          ,1:(b-1)/2);...
        I(1:(a-1)/2    ,m-(b-1)/2+1:m),I(1:(a-1)/2    ,1:m),I(1:(a-1)/2    ,1:(b-1)/2)];
    for j=1:n % perform correlation
        for i=1:m
            patch =reshape(IW(j:j+a-1,i:i+b-1)',[a*b,1]);
            psum(j,i)=sum(patch);
            p2sum(j,i)=sum(patch.^2);
            ptsum(j,i)=sum(patch.*tvec);
            tempcor=corrcoef(patch,tvec);
            O(j,i)=tempcor(1,2);
        end
    end
```

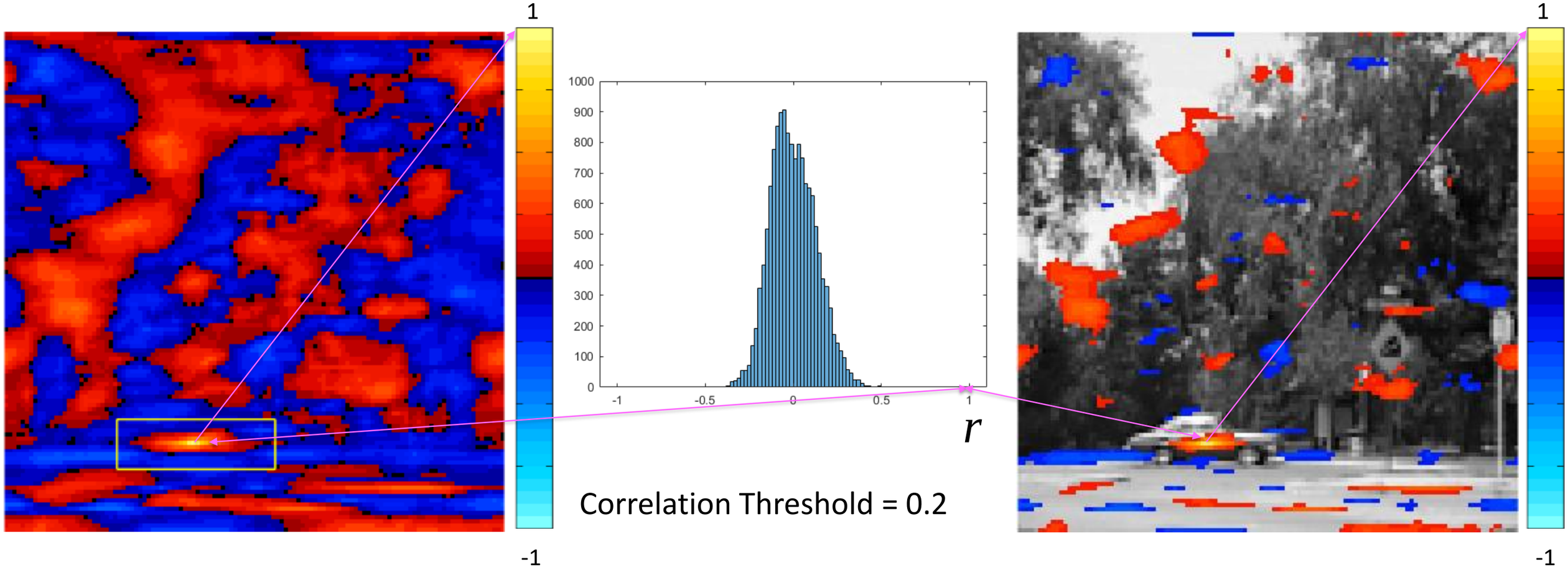
Template Matching

```
elseif(is_a_even==1)&&(is_b_even==1) % appends border pixels for wrap-around
    IW=[I(n-a/2+1:n,m-b/2+1:m),I(n-a/2+1:n     ,1:m),I(n-a/2+1:n,1:b/2);...
        I(1:n      ,m-b/2+1:m),I(1:n           ,1:m),I(1:n      ,1:b/2);...
        I(1:a/2    ,m-b/2+1:m),I(1:a/2         ,1:m),I(1:a/2    ,1:b/2)];
    for j=1:n % perform correlation
        for i=1:m
            patch =reshape(IW(j:j+a-1,i:i+b-1)',[a*b,1]);
            psum(j,i)=sum(patch);
            p2sum(j,i)=sum(patch.^2);
            ptsum(j,i)=sum(patch.*tvec);
            tempcor=corrcoef(patch,tvec);
            O(j,i)=tempcor(1,2);
        end
    end
end

end
```

Template Matching

Here are our statistic images.

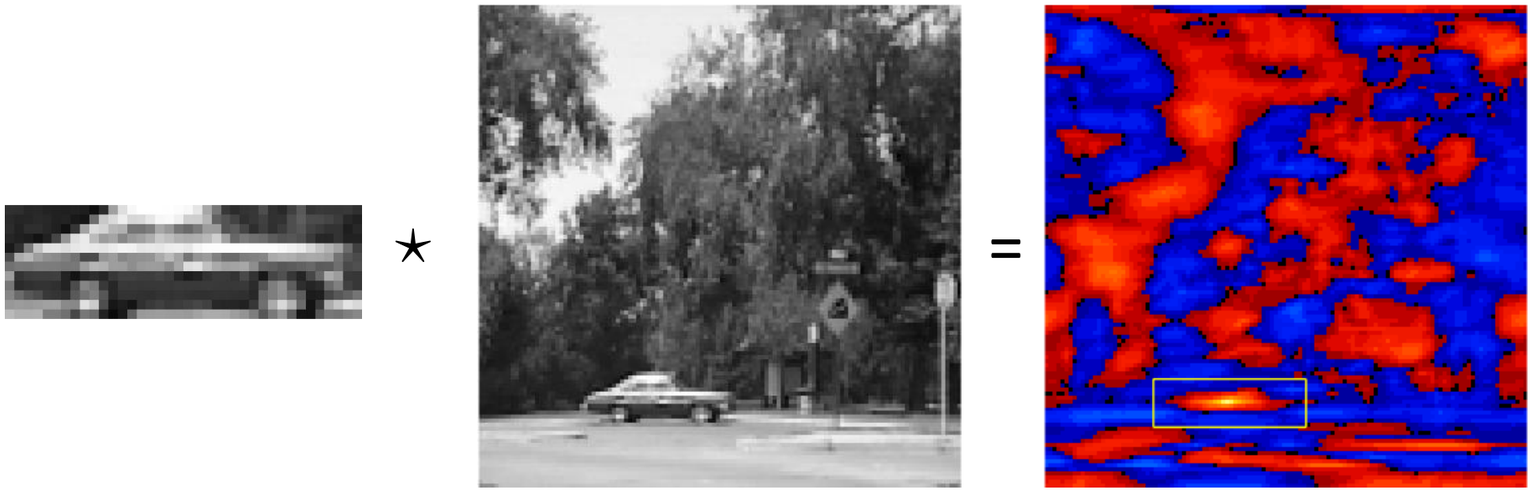


Correlation Threshold = 0.2

Discussion

We can compute local image statistics such as $\bar{x}$ and s^2.

If we precompute sum $\sum o_i$ and sum of squares $\sum o_i^2$ of a template, then when computing local image statistics $\sum p_i$ and $\sum p_i^2$, also compute $\sum o_i p_i$, then we can calculate the correlation between the object template and the region of pixels, hence finding the likely location of an object in our image.



Discussion

Questions?

Homework 5

1. Modify the code in the lecture to compute the local variance in an image of yours. Present sum, sum of squares, and the variance images. You may choose to have a larger kernel of ones.

2*.For your image, use the local variance to choose the size of a smoothing kernel. Smooth more in homogeneous areas and less in heterogeneous areas. i.e. use larger σ in kernel or larger kernel in homogeneous areas.

3. Extract a small template image of an object in your image. Compute a correlation image. Find the pixel with maximum correlation and hence find your object. Present pixel sum, sum of squares, and sum of template*pixel images.

*For students in MSSC 5770.

Homework 5

4. Use a picture with the same object at two different orientations.
   Your second object can be the first object rotated and “placed”
   in the image. Use two templates to find the objects and their orientations.

5*. Obtain an image with multiple objects. Obtain multiple templates.
   Use correlation to find all the objects in the image.

Submit one document with all your results (no Matlab code) and
some discussion of thoughts or commentary.
Separately also submit executable Matlab code and any needed files.
Do not use a built in Matlab convolution function.

*For students in MSSC 5770.