Enumerating birds in a video

Justin Pearson
2014-10-25

Abstract

We explore computational techniques for enumerating birds in a video file. The video was filmed in November 2011 in Simi Valley, California, and features thousands of crows on their daily evening commute from the city to a large green open space. We determine that the 8-minute video captures roughly 1200 crows.

In[1]:=
```
Clear["Global`*"]
SetDirectory[NotebookDirectory[]];
```

In[3]:=
```
Import["crows.gif", {"ImageList", 1}]
```

Out[3]=

Since the crows are exiting the frame through the top and left sides, the main idea is to count the crows by counting when they leave the top or left side of the frame.

Footage

Here's our footage:

In[4]:=
```
fname = "short_clip.avi";
```

In[5]:=
```
show[f_] := Grid[{{"File name", f}, {"File Size (bytes)", FileByteCount[f]}} ~
    Join ~ Table[{e, Import[f, e]},
     {e, {"BitDepth", "ColorSpace", "Duration", "FrameCount",
       "FrameRate", "ImageSize", "VideoEncoding", {"ImageList", 100}}}
    ],
   Frame → All]
```

In[6]:=
```
show[fname]
```

Out[6]=

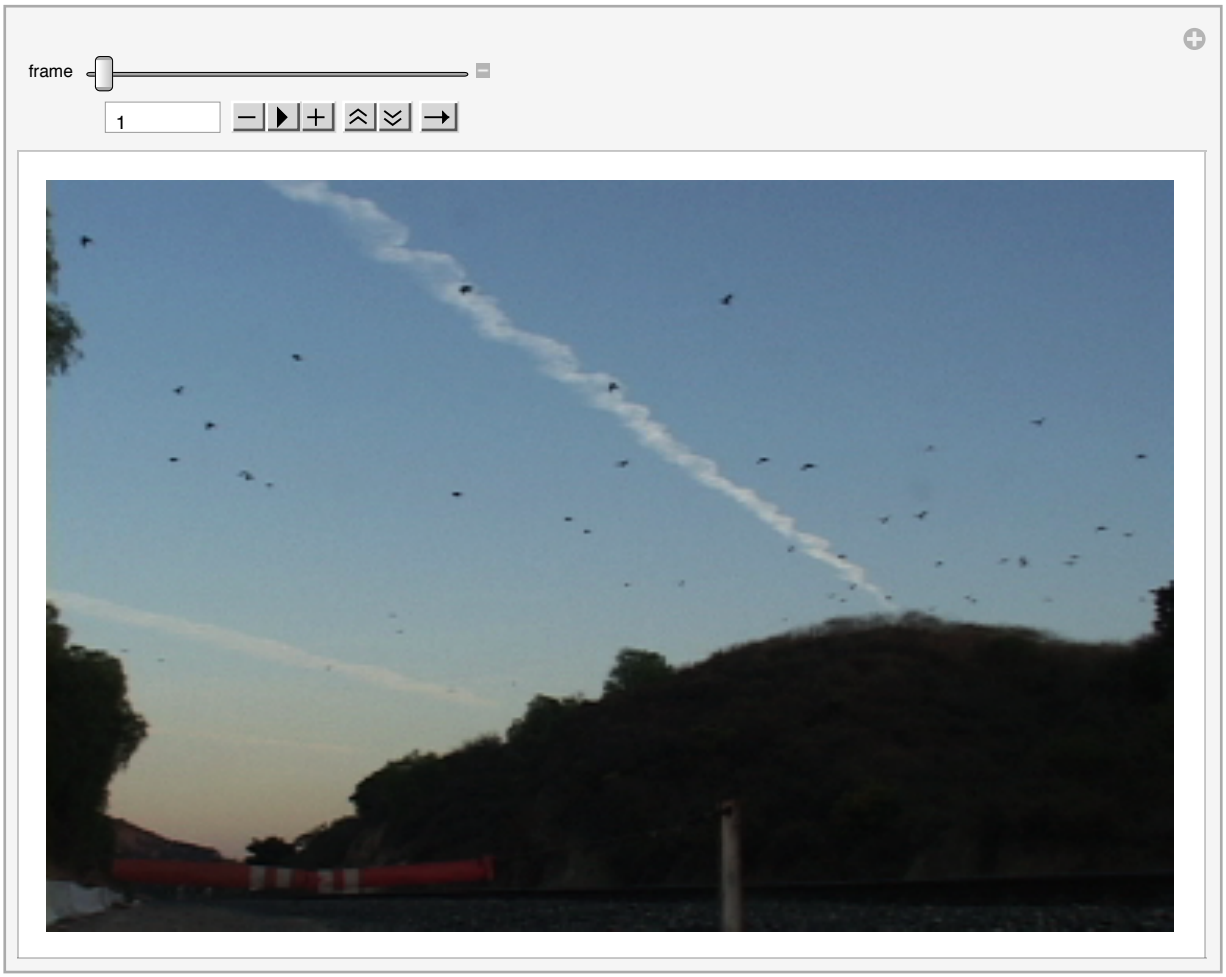
File name	short_clip.avi
File Size (bytes)	12 939 120
BitDepth	8
ColorSpace	RGB
Duration	3.46667
FrameCount	104
FrameRate	30.
ImageSize	{720, 480}
VideoEncoding	DV/DVCPRO - NTSC
{ImageList, 100}	

In[7]:=
```
{time, frames} = AbsoluteTiming@ Import[fname, "ImageList"];
time
```

Out[8]=
```
1.24085
```

In[9]:=
```
Manipulate[frames[[i]], {{i, 1, "frame"}, 1, Length@frames, 1, Appearance → "Open"}]
```

Out[9]=



Idea 1: View video as 3D Image

If we mask out the crows and stack the frames up to form a 3D image, each crow will trace out a long tube-like shape. Counting the tubes tells us the number of crows in the whole video.

In[10]:=
```
mask = (Import[FileNameJoin[{NotebookDirectory[], "mask.jpg"}]] // Binarize);
Thumbnail[mask]
```

Out[11]=

In[12]:=
```
Manipulate[Module[{f, maskedim, num},
  f = frames[[i]];
  maskedim = ImageMultiply[mask, ColorNegate@Binarize[f, t]];
  num = ComponentMeasurements[maskedim, "Count"] // Length;
  Column[{
    ToString[num] <> " Crows:",
    Show[HighlightImage[f, Dilation[maskedim, r]], ImageSize → 800]
   }, Alignment → Center]
 ],
 {{r, 5, "highlight"}, 1, 10, 1, Appearance → "Open"},
 {{i, 1, "frame"}, 1, Length@frames, 1, Appearance → "Open"},
 {{t, .41, "masking threshold"}, 0, 1, Appearance → "Open"}
]
```

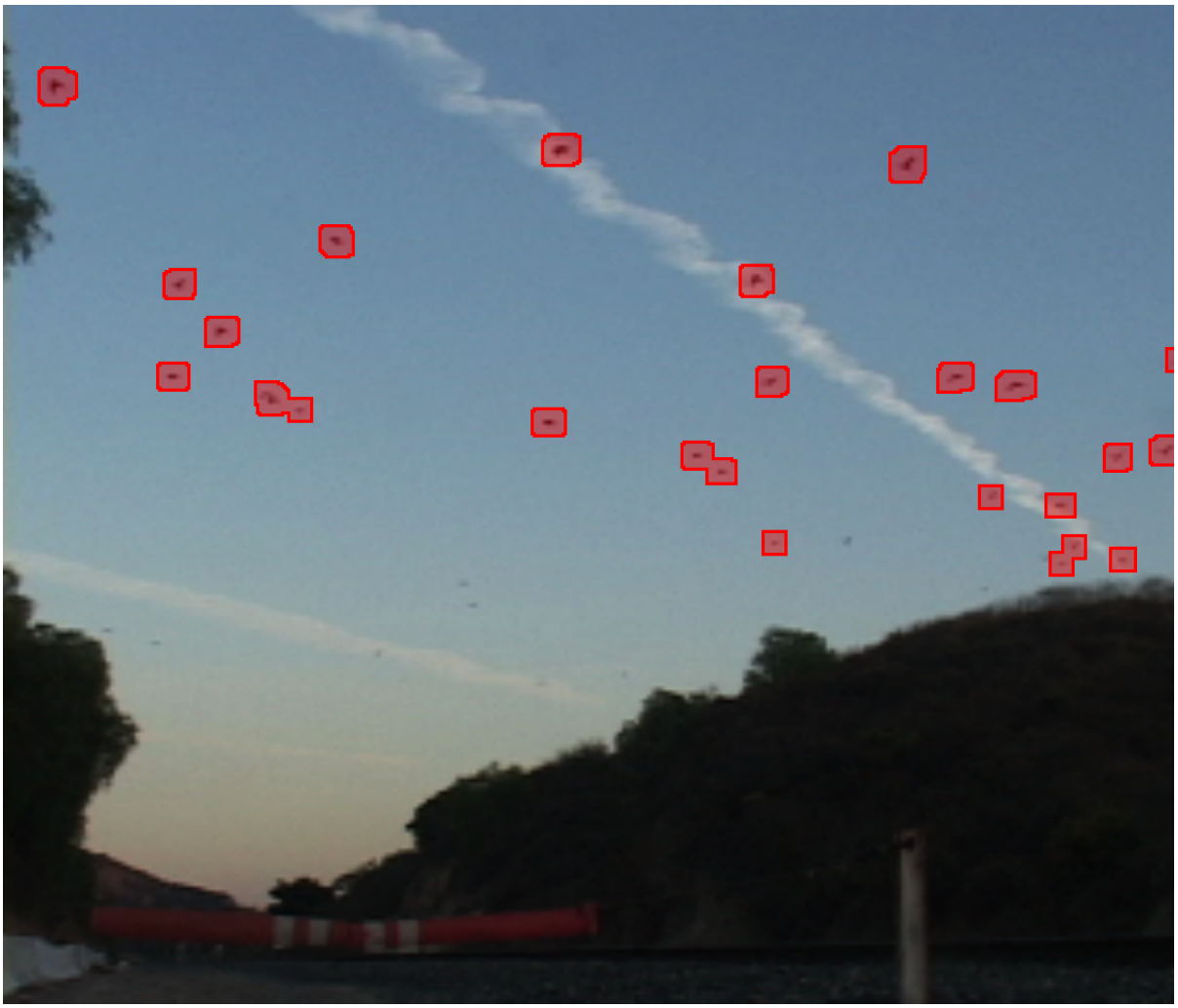
highlight 5

frame 1

masking threshold 0.41

Out[12]=

37 Crows:



Here's a good threshold value:

In[13]:=
```
threshold = .41;
```

Make & Export 3D image

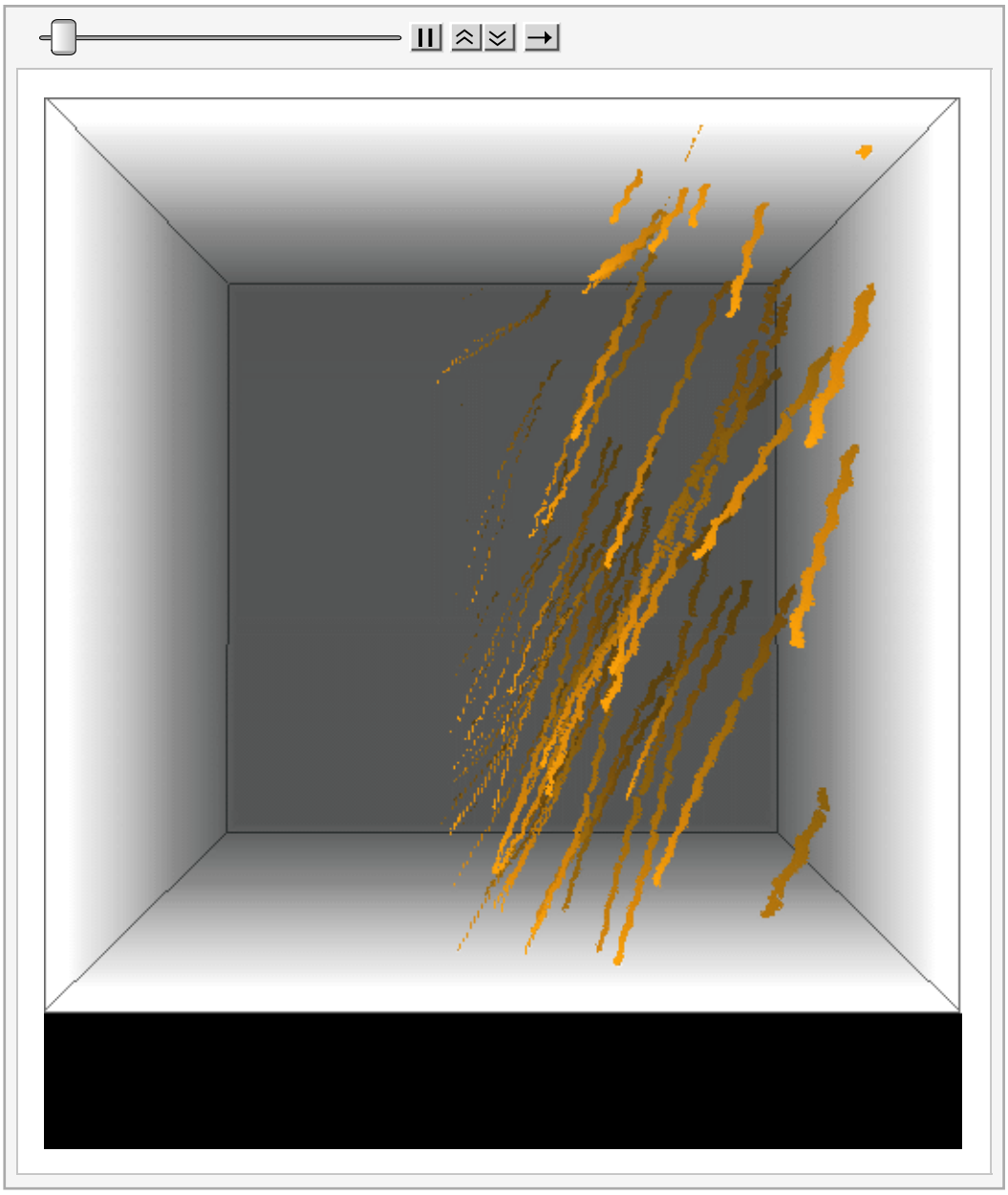
This takes a long time, so I locked these cells.

```
im3d = (ImageMultiply[mask, ColorNegate@Binarize[#, threshold]] & /@
      shortframes[[1 ;; 100]] // Image3D);

{w, h, d} = ImageDimensions@im3d

im3d2 = ImageResize[im3d, {w / 4, h / 4, d}];

tab = ParallelTable[
    Show[im3d, ImageSize → 500, (*AxesLabel→{"x","y","frame"},
     Axes→True,*)BoxRatios → {1, 1, 1}, ViewPoint → {r, 0, 2},
     PreserveImageOptions → False], {r, .01, 2 π, .1}];
Length@tab
Export["3d.gif", tab]
```

Show 3D image

In[14]:=
```
ListAnimate[Import["3d.gif"]]
```

Out[14]=



Idea 2: Count when crows leave the frame

Import the frames as raw pixel values instead of Image[] objects.

In[15]:=
```
{time, data} = AbsoluteTiming[Import["short_clip.avi", "Data"]];
Print[time, " seconds"]
```

```
1.3534 seconds
```

In[17]:=
```
Dimensions@data
```
Out[17]=
```
{104, 480, 720, 3}
```

In[18]:=
```
Short[data[[100]]]
```
Out[18]//Short=
```
{ <<1>> }
```

In[19]:=
```
im = Image[data[[50]] / 255] (* don't forget to normalize *)
```
Out[19]=

The fact that the lefthand side has some dark trees will make things difficult; let's just crop them out.

In[20]:=
```
Dimensions@data
```
Out[20]=
```
{104, 480, 720, 3}
```

In[21]:=
```
data = data[[All, ;; -50, 120 ;;, All]];
```

In[22]:=
```
im = Image[data[[50]] / 255] (* don't forget to normalize *)
```

Out[22]=

Take vertical column of pixels from each frame

In[23]:=
```
vert = data[[All, All, 1, All]];
{nf, h, nc} = Dimensions@vert
```

Out[24]=
```
{104, 431, 3}
```

In[25]:=
```
imv = (Image[vert / 255] // ImageRotate[#, -90 Degree] &)
```

Out[25]=

Take horizontal slice of pixels from each frame

In[26]:=
```
horiz = data[[All, 1, All, All]];
{nf, w, nc} = Dimensions@horiz
```

Out[27]=
```
{104, 601, 3}
```

In[28]:=
```
imh = Image[horiz / 255]
```

Out[28]=

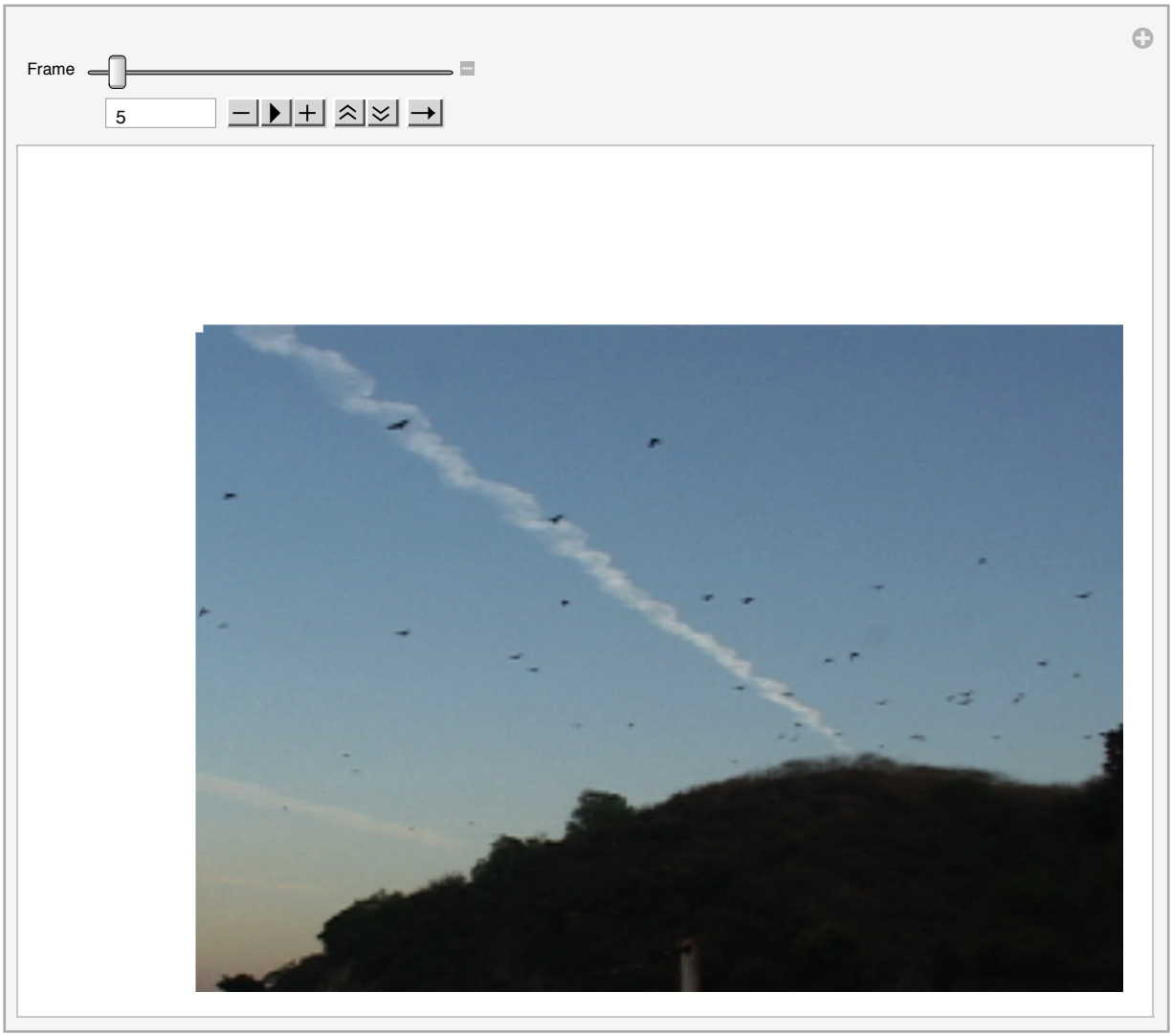
Quick check: Show a video of crows leaving the frame

You can see the crows getting “stuck” when they hit the upper and left edges.

In[29]:=
```
images = Import["short_clip.avi", "ImageList"];
images2 = ImageCrop[#, {w, h}, {Left, Bottom}] & /@ images;
```

In[31]:=
```
Manipulate[Module[{imhcrop, imvcrop, im1, im2, im3, im4, imass, nframes},
  nframes = Length@images;
  imhcrop = ImageCrop[imh, {Full, i}, Bottom];
  imvcrop = ImageCrop[imv, {i, Full}, Left] // ImageReflect[#, Left] &;
  im1 = Image[ConstantArray[1, {i, i}]];
  im2 = imhcrop;
  im3 = imvcrop;
  im4 = images2[[i]];
  (*ImageDimensions/@{im1,im2,im3,im4}*)
  imass = ImageAssemble[ { {im1, im2}, {im3, im4}}];
  ImagePad[imass, {{nframes - i, 0}, {0, nframes - i}}, White]
 ]
 , {{i, 5, "Frame"}, 1, Length@images, 1, Appearance → "Open"}]
```

Out[31]=



Looks like ~15 leave the RHS and 6 leave the top.

The idea will be to have MMA count the number of crows that leave the left & top sides.

Crows exiting top of frame

Here's the top few rows of a 15-minute video of crows. The whole file was too big to import.

In[32]:=
```
fname = "horiz_720x24_quicktime_h264.mov";
show@fname
ims = Import[fname, {"ImageList", Range[5000, 5300]}];
```

Out[33]=

File name	horiz_720x24_quicktime_h264.mov
File Size (bytes)	29 436 035
BitDepth	8
ColorSpace	RGB
Duration	479.479
FrameCount	14 370
FrameRate	29.97
ImageSize	{720, 24}
VideoEncoding	H.264
{ImageList, 100}	

In[35]:=
```
Manipulate[ims[[i]], {{i, 1, "frame"}, 1, Length@ims, 1, Appearance → "Open"}]
```

Out[35]=

Import video in chunks of 300 frames; extract the top row of pixels from each frame; parallelize this for speed.

In[36]:=
```
nframes = Import[fname, "FrameCount"]
```

Out[36]=
```
14 370
```

In[37]:=
```
partition[list_List, n_Integer] := Partition[list, n, n, {1, 1}, {}]
```

In[38]:=
```
partition[Range[25], 10]
```

Out[38]=
```
{{1, 2, 3, 4, 5, 6, 7, 8, 9, 10},
 {11, 12, 13, 14, 15, 16, 17, 18, 19, 20}, {21, 22, 23, 24, 25}}
```

In[39]:=
```
chunksize = 300;
```

In[40]:=
```
chunkinds = partition[Range@nframes, chunksize];
```

In[41]:=
```
Length /@ chunkinds
```

Out[41]=
```
{300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300,
 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300,
 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 270}
```

In[42]:=
```
Length@chunkinds
```

Out[42]=
```
48
```

In[43]:=
```
assembleFrames[frameInds_] :=
  (Import[fname, {"ImageList", frameInds}] // Map[ImageData, #] & //
    Part[#, All, 1, All, All] &)
```

In[44]:=
```
{time, m} = AbsoluteTiming@
    (ParallelTable[assembleFrames[i], {i, chunkinds}] // Flatten[#, {1, 2}] &);
time
```

Out[45]=
```
31.3078
```

In[46]:=
```
Dimensions@m
```

Out[46]=
```
{14 370, 720, 3}
```

In[47]:=
```
imm = ImageRotate[Image[m], Right];
```

In[48]:=
```
Thumbnail[imm, 600]
```

Out[48]=

In[49]:=
```
(*Export[fname<>".tiff",imm]*)
```

Detection via GradientFilter

In[50]:=
```
imgrad = ImageAdjust@GradientFilter[imm, {{0, 2}}];
```

In[51]:=
```
ImageDimensions[imgrad]
```

Out[51]=
```
{14 370, 720}
```

In[52]:=
```
Thumbnail[imgrad, 600]
```

Out[52]=

It's hard to see in this pic, but if you zoom in, black is background and white is crows.

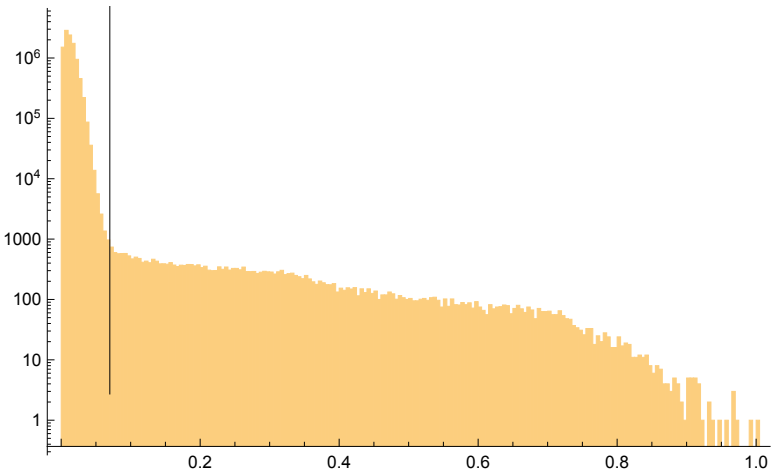
In[53]:=
```
(*Export[fname<>".grad.tiff",imgrad]*)
```

Find a good threshold pixel value to differentiate between "crow" and "background".

In[54]:=
```
imgdat = Flatten@ImageData[imgrad];
```

In[55]:=
```
TimeConstrained[Histogram[imgdat, 255, ScalingFunctions → "Log",
  Epilog → Line[{{0.07, 1}, {0.07, 1000}}], PlotRange → {{0, 1}, Automatic}], 45]
```

Out[55]=



So it looks like a good dividing line between black background and white crows is around 0.07.

In[56]:=
```
imthres = (Binarize[imgrad, 0.075] // DeleteSmallComponents[#, 1] &);
```

In[57]:=
```
(*Export[fname<>".grad.thres.tiff",imthres]*)
```

Looks pretty good. Let's highlight them:

In[58]:=
```
imhigh = HighlightImage[imm, imthres // Dilation[#, 2] &, Method → {"Compose", 0.1}];
```

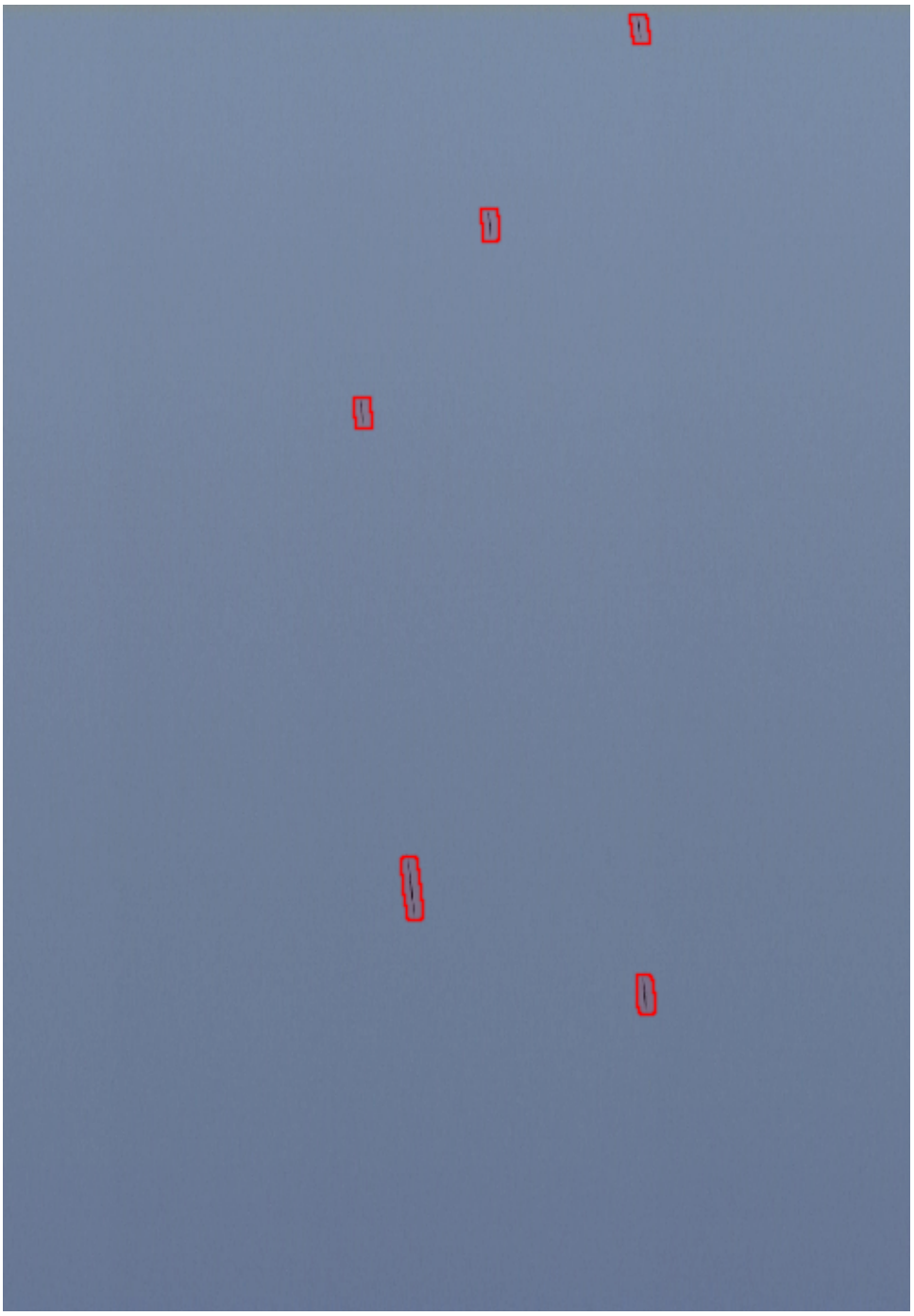
In[59]:=
```
Thumbnail[imhigh, 500]
```

Out[59]=

Here's a zoom-in of the left side of the image:

In[60]:=
```
ImageTake[imhigh, All, {1, 500}]
```

Out[60]=



Hard to see, but the crows are highlighted slightly.

In[61]:=
```
(*Export[fname<>".grad.thres.highlight.tiff",imhigh]*)
```

So, how many crows left the top of the frame during this 8-minute video?

In[62]:=
```
outTop = imthres // ComponentMeasurements[#, "Count"] & // Length
```

Out[62]= 592

Crows exiting left side of frame

Do essentially the same thing, but for the video that's the left side of the frame.

A long file. Import in pieces.

In[63]:=
```
fname = "vert_24x350_quicktime_h264.mov";
show@fname
```

Out[64]=

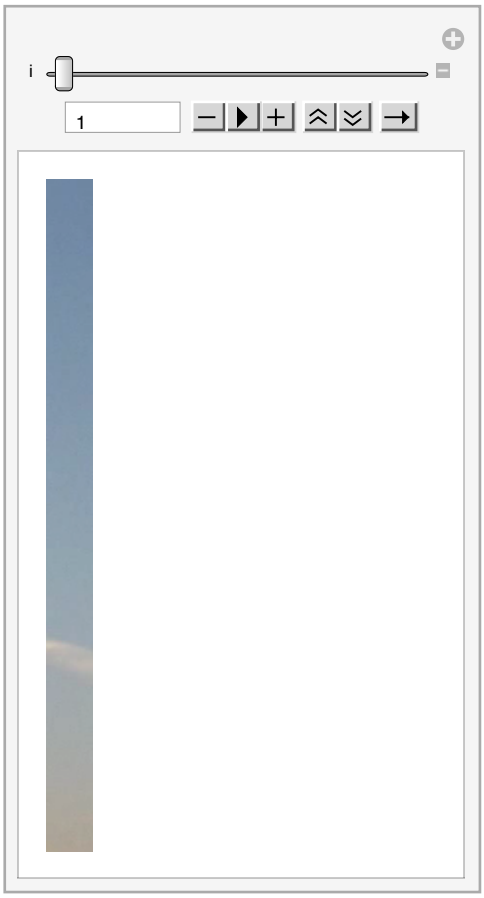
File name	vert_24x350_quicktime_h264.mov
File Size (bytes)	18 630 731
BitDepth	8
ColorSpace	RGB
Duration	479.479
FrameCount	14 370
FrameRate	29.97
ImageSize	{24, 350}
VideoEncoding	H.264
{ImageList, 100}	

Preview

In[65]:=
```
ims2 = Import[fname, {"ImageList", Range[5000, 5300]}];
```

In[66]:=
```
Manipulate[ims2[[i]], {{i, 1}, 1, Length@ims2, 1, Appearance → "Open"}]
```

Out[66]=



Aside: Consider just assembling these skinny frames instead of only their leftmost cols.

The easiest thing to do would be to just squish all the frames together.

In[67]:=
```
imass = ImageAssemble[{ims}];
ImageTake[imass, All, {1, 1000}]
```

Out[68]=

Too dotty.

Import in pieces.

In[69]:=
```
nframes = Import[fname, "FrameCount"]
```

Out[69]=
```
14 370
```

In[70]:=
```
partition[list_List, n_Integer] := Partition[list, n, n, {1, 1}, {}]
```

In[71]:=
```
partition[Range[25], 10]
```

Out[71]=
```
{{1, 2, 3, 4, 5, 6, 7, 8, 9, 10},
 {11, 12, 13, 14, 15, 16, 17, 18, 19, 20}, {21, 22, 23, 24, 25}}
```

In[72]:=
```
chunksize = 300;
```

In[73]:=
```
chunkinds = partition[Range@nframes, chunksize];
```

In[74]:=
```
Length /@ chunkinds
```

Out[74]=
```
{300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300,
 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300,
 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 270}
```

In[75]:=
```
Length@chunkinds
```

Out[75]= 48

In[76]:=
```
Clear[assembleFrames];
assembleFrames[frameInds_] :=
 (Import[fname, {"ImageList", frameInds}] // Map[ImageData, #] & //
   Part[#, All, All, 1, All] &)
```

In[78]:=
```
m = (ParallelTable[assembleFrames[i], {i, chunkinds}] // Flatten[#, {1, 2}] &);
```

In[79]:=
```
Dimensions@m
```

Out[79]= {14 370, 350, 3}

In[80]:=
```
imm = ImageRotate[Image[m], Right];
```

A very long image:

In[81]:=
```
ImageDimensions[imm]
```

Out[81]= {14 370, 350}

In[82]:=
```
Thumbnail[imm, 500]
```

Out[82]=

Left side of the image:

In[83]:=
```
ImageTake[imm, All, {1, 600}]
```

Out[83]=

In[84]:=
```
(*Export[fname<>".tiff",imm]*)
```

Detection via GradientFilter

In[85]:=
```
imgrad = ImageAdjust@GradientFilter[imm, {{0, 2}}];
```

In[86]:=
```
ImageTake[imgrad, All, {1, 600}]
```

Out[86]=

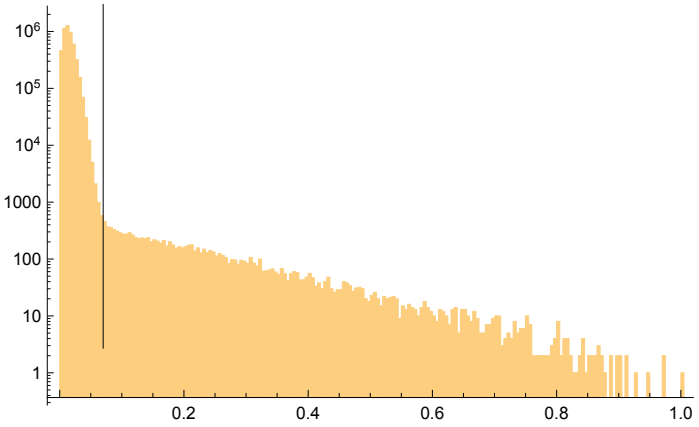
Black is background and white is crows.

In[87]:=
```
(*Export[fname<>".grad.tiff",imgrad]*)
```

In[88]:=
```
imgdat = Flatten@ImageData[imgrad];
```

In[89]:=
```
TimeConstrained[Histogram[imgdat, 255, ScalingFunctions → "Log",
  Epilog → Line[{{0.07, 1}, {0.07, 1000}}], PlotRange → {{0, 1}, Automatic}], 45]
```

Out[89]=



So it looks like a good dividing line between black background and white crows is around 0.07.

In[90]:=
```
imthres = (Binarize[imgrad, 0.075] // DeleteSmallComponents[#, 1] &);
```

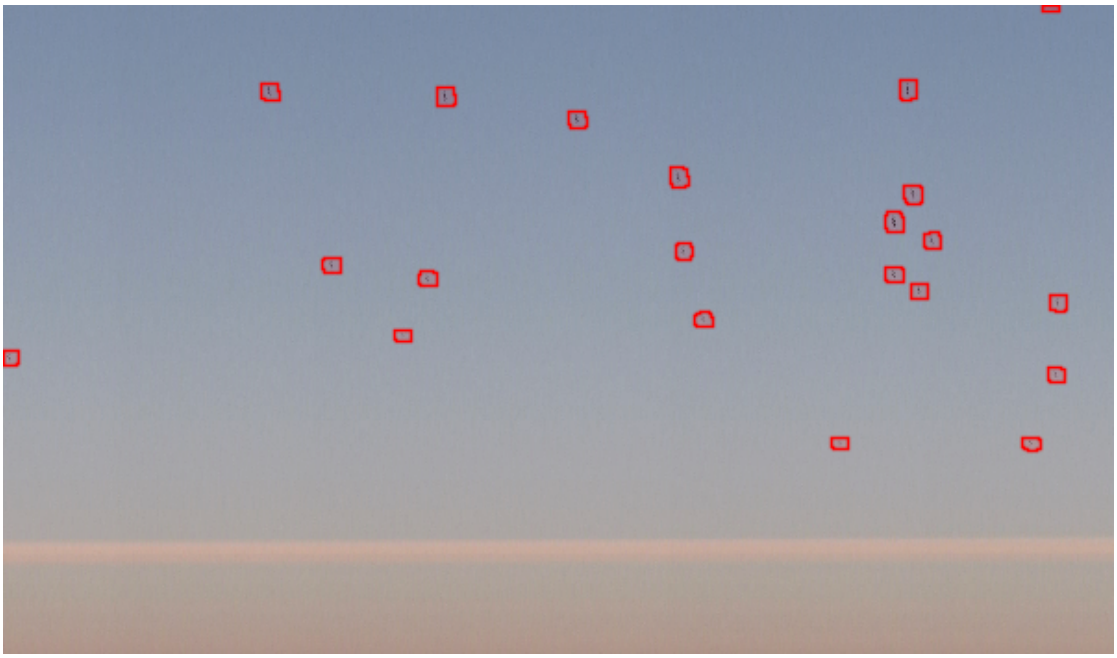
In[91]:=
```
(*Export[fname<>".grad.thres.tiff",imthres]*)
```

Looks pretty good. Let's highlight them:

In[92]:=
```
imhigh = HighlightImage[imm, imthres // Dilation[#, 2] &, Method → {"Compose", 0.1}];
```

In[93]:=
```
ImageTake[imhigh, All, {1, 600}]
```

Out[93]=



In[94]:=
```
(*Export[fname<>".grad.thres.highlight.tiff",imhigh]*)
```

So, how many crows departed out the left side of the frame during the 8-minute video?

In[95]:=
```
outLeft = imthres // ComponentMeasurements[#, "Count"] & // Length
```

Out[95]= 677

Conclusion

In[96]:=
```
outLeft + outTop
```

Out[96]= 1269

About 1400 crows flew through the video over the course of 8 minutes: about 3 crows a second. Wow!