# Lab: Vixels, Veillance Flux, and the Veillance Vector Field, for intelligent image processing

version 1.0 - 2014.01 - ECE516

Ryan Janzen
Department of Electrical and Computer Engineering, University of Toronto

***Abstract*—This lab will introduce you to new research in image processing, and help you develop an intuitive hands-on understanding of some basic programming and image capture techniques.**

**To do well in this course, this will help you "wrap your head around" physical, mathematical and computational concepts at the same time, that combine both hardware with software.**

**This lab introduces new physical and mathematical concepts: the *veillon*, the *vixel*, and the *veillance vector field*, to measure optical (visual) veillance.**

## I. Preparation: Getting Octave or Matlab running

Get Octave or Matlab running on your computer. Octave is open-source, and can be installed for free on Linux, Windows, or Mac OS computers. (http://octave.sourceforge.net/)

Join together in a group. One or more members of your group need to bring a:

- Laptop, with either Octave or Matlab running;
- Camera (any cellphone or pocket camera that can shoot video), with a USB cable or memory card that can transfer your video to the laptop. (check its ports!)

Then, make sure you can import a video from your camera into Octave/Matlab. You should try this out in advance. Some helpful hints are in the following section.

If it is absolutely not possible for your group to get Octave or Matlab running on your own laptop, and bring the laptop to the lab, there are a **very** limited number of Windows "seats" in BA3135 for remote-login to run Matlab through ECF-Windows. However, the difficulty is that you will need to figure out how to transfer your video files off your camera to your ECF-Windows remote account, to be able to process it in the remote login session.

It is easier, and you will learn more, if you can get the code up and running yourself!

Finally, read this handout, plus the paper, "Understanding, Quantifying, and Measuring Veillance, Surveillance and Sousveillance".

## II. Importing video from your camera into Octave/Matlab

You can read in videos into Octave/Matlab using different methods, but you need to make sure you can do at least one of them.

Our sample code, `read_in_video.m` uses the `VideoReader` command in Matlab (with Image Processing toolbox). Try running `read_in_video.m` to see if your installation has that capability.

If you don't have this, you will need to use some other method to read in videos.

For example, break the video into a sequence of images, and then import each frame image into Octave/Matlab using the '`imread`' command. To break a video into frame image files, you can use ffmpeg. ffmpeg can be installed in Linux or Windows (free, open-source). It runs by command-line. To do this, install ffmpeg and run:

```
ffmpeg -i INPUT_MOVIE.MOV -r FRAME_RATE    ...
               -f image2 frame-%07d.png
```

Then, in Octave/Matlab, write a 'for' loop that imports each frame file separately, with the `imread` command.

## III. Testing your camera

Time to have fun with your camera! Inspect your camera for AGC (automatic gain control). You can easily see this by watching the display's brightness while recording a video.

| | |
|---|---|
| Camera uses AGC<br>**before** video shooting (yes/no) | |
| Camera uses AGC<br>**during** video shooting (yes/no) | |
| AGC can be turned off (yes/no) | |
| AGC has discrete gain levels,<br>or is too continuous to notice? | |

**Question about AGC**: If a camera's AGC can't be disabled, how would you adjust the scene that you are filming to hold the gain more constant? (a constant gain even when laser is being turned on/off)

## IV. Veillance flux density

We will measure the density of effective vixels impinging the surface of an object in a scene.

For each surface, $S_n$, we could individually illuminate and test every single point on the surface in 2 dimensions (a very long process!) However, if we have an isotropic cross-dependency of vixels, we can scan along two orthogonal tracks ($T_1$ and $T_2$). The number of significant PCA components (significant eigenvalues), $\Omega\{T_1\}$ and $\Omega\{T_2\}$ is found separately for each track. We can then estimate the extrapolated number of significant PCA components (significant eigenvalues) for the entire surface as:

$$\tilde{\Omega}\{S_1\} = \Omega\{T_1\} \cdot \Omega\{T_2\} \tag{1}$$

$\tilde{\Omega}\{S_1\}$ gives the estimated number of effective vixels impinging the surface — that is, the effective veillance flux ($\Phi_{VE}$).

For the average effective veillance flux density (veillance intensity), in [vixels/m$^2$], we divide by the surface area:

$$\overline{V}_E = \frac{\Phi_{VE}}{S_1} = \frac{\tilde{\Omega}\{S_1\}}{S_1} \tag{2}$$

The veillance rate (effective) for the object's surface [bits/s], simply uses the bit depth of the camera, $B$ (number of bits for each pixel), and frame rate, $r_F$:

$$\overline{r}_{VE} = r_F \cdot B \cdot \Phi_{VE} = r_F B \tilde{\Omega}\{S_1\} \tag{3}$$

We will measure $\tilde{\Omega}\{S_n\}$ using the process of "Laser scanning vixel principal component density analysis" which requires:

- **Sufficient number of images/frames**: Sufficient images (or video frames) are needed in the experiment to independently test each hypothesized vixel. Otherwise PCA components will max out at the number of video frames. (Enough images/video frames are needed to give each potential effective vixel the opportunity to be expressed in a linearly independent vector of pixels). A good rule of thumb: Have more than enough frames of video, rather than not enough. *Diagnostic*: If the PCA coefficients do not have a sudden dropoff on the graph, a possible reason is that the dropoff is actually beyond the end of the graph, and there are not enough frames of video to express them. Video length is a tradeoff, since excessive frames will slow down computation time.
- **Slow motion**: The illumination test point must move slowly enough (relative to the frame rate) that it can be expressed in each vixel along the track axis. That is, we need high vixel resolution along the track axis.
- **Small test point**: The illumination test point is sufficiently small to spatially access individual vixels. e.g. For a given size of illuminated point, the object is suffiently far away from the camera so that the "dot" is smaller than one pixel in the image. Otherwise the SNR (signal-to-noise ratio) of the measurement will be lower and the resulting PCA components less definitive to distinguish.
- **Uniform reflection**, to emit light approximating a light source positioned at the point being tested. For example, when testing the veillance impinging the surface of a mirror, then an easy solution is to place a piece of paper in front of the mirror, during testing, since the paper's rough surface gives a more uniform reflection than the mirror.

## V. Putting it into practice

Go ahead and take a video to look for vixels!

Make sure the camera is set on a solid surface and *does not move* during the entire video. (Otherwise, PCA will think that many more pixels have independently changed) Also, make sure AGC *does not change the gain* on the video partway through. (Otherwise, the background subtraction will need to be reprogrammed to adapt.)

Record video while tracing the laser point along each track.

Once you have imported the video into Octave/Matlab, you can segment it into the block of frames belonging to Track 1, process those, and then afterward you can process the Track 2 frames.

```
video_filename = 'IMG_0617.MOV';
read_in_video;    % sample code
downsample = 1;   % 1:1, no downsampling
analyze_frames;   % sample code

% Only use certain frames, for this track.
videoFramesGreyUnwoundMatrix =
    videoFramesGreyUnwoundMatrix(:,50:90);
analyze_pca;      % sample code
```

If computation is too slow, you can increase 'downsample' to 2: that means 1/4 as many spatial pixels will be used. How should this affect your result?

## VI. More complex laser tracks

If you want to find the veillance flux on an object with multiple faces (such as a cylinder), then each surface can be analyzed separately, with more tracks:

$$\begin{aligned} \tilde{\Omega}\{S_{TOT}\} &= \tilde{\Omega}\{S_1\} + \tilde{\Omega}\{S_2\} \\ &= \Omega\{T_1\} \cdot \Omega\{T_2\} + \Omega\{T_3\} \cdot \Omega\{T_4\} \end{aligned} \tag{4}$$

**Question**: How can we estimate the veillance flux on the back *and* front (i.e. all the way around the curved face) of a cylindrical object? Hint: The camera doesn't need to be moved, but the test tracks can be changed. What is the simplest change to the design of laser scan tracks, and how should the calculations be changed? (Draw a diagram, change Equation 1, and explain why it works.)

## VII. Measurements

You will be given a sample scene in the lab, and will be asked to measure veillance flux on one of the objects. **Calculate, show your reasoning, and explain:**

- Effective veillance flux, falling on the object [vixels]
- Effective veillance flux density, falling on the object [vixels/m$^2$]
- Number of real, geometric vixels falling on the object, directly from the camera [vixels]
- Effective veillance rate, falling on the object [bits/s]

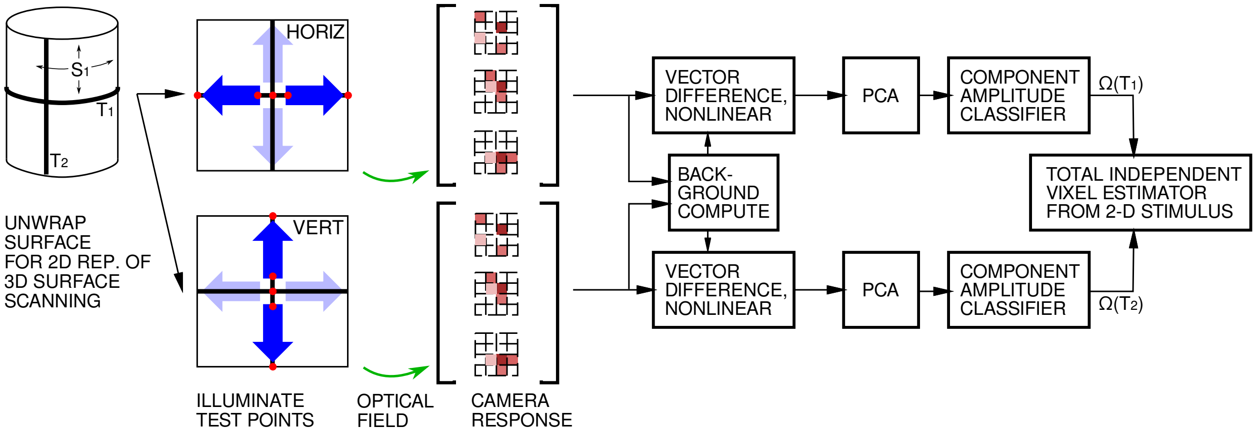
S1
T1
T2
UNWRAP SURFACE FOR 2D REP. OF 3D SURFACE SCANNING
HORIZ
VERT
ILLUMINATE TEST POINTS
OPTICAL FIELD
CAMERA RESPONSE
VECTOR DIFFERENCE, NONLINEAR
BACK-GROUND COMPUTE
VECTOR DIFFERENCE, NONLINEAR
PCA
PCA
COMPONENT AMPLITUDE CLASSIFIER
COMPONENT AMPLITUDE CLASSIFIER
Ω(T1)
Ω(T2)
TOTAL INDEPENDENT VIXEL ESTIMATOR FROM 2-D STIMULUS

Fig. 1. Laser scanning vixel principal component density analysis.

With your lab submission, also include copies of your PCA coefficient graphs, marked with your interpretation. Also include any code you modified.

**Question**: What happens if you re-trace the laser dot back and forth, following along the exact same track? Explain your prediction and show results.

## VIII. Extensions

Bonus marks and possible project ideas could come out of the following, if you want to get something working, or collaborate with us on a new project:

Human eye: what is the veillance rate?

Design a system to let you see unseen objects, which are only reflected off an abrasive surface with a limited number of vixels.

Measuring veillance in everyday situations: Measuring surveillance in department stores, universities. Measuring sousveillance from wearable computers on yourself or other people. Other types of tests?

Veillance for sound, touch, smell? (i.e. following the sensitivity to vibrations or chemical concentrations as they propagate through matter or space.)

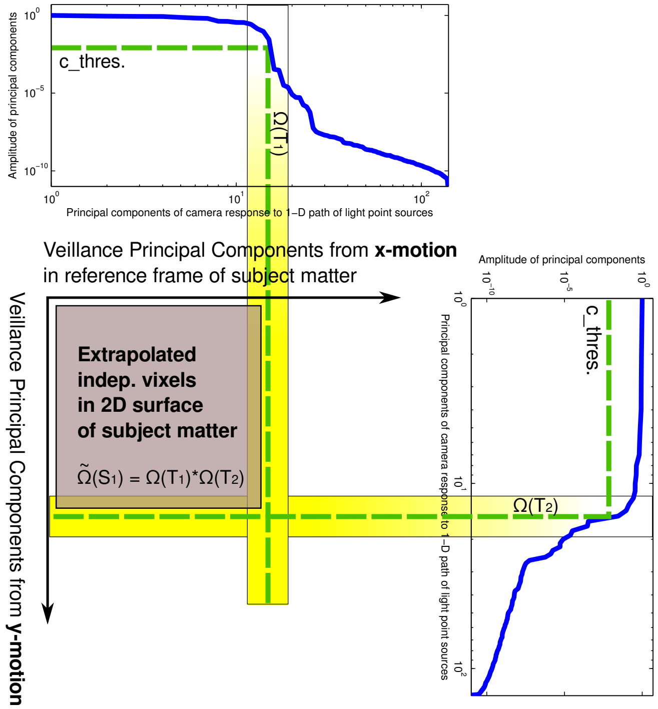

Fig. 2. PCA output as two data sets (for horizontal and vertical scanning with pointwise illumination), to form a metric to estimate the total number of independent vixels, if the entire object's surface area had been tested point-by-point. This employs the symmetric degeneracy assumption (where we have "fairly" illuminated regions of the object, as opposed to being biased for or against areas close to mirrors, etc.) giving an estimate of vixel independence for the object's full surface area.