**Mobile Robots and Sensor Fusion**

A thesis submitted in partial fulfillment of the requirement
for the degree of Bachelors of Science in **Physics** from
The College of William and Mary

by

Danielle Dumond

Williamsburg, VA
**May 2007**

# Table of Contents

## Introduction:

Artificially intelligent machines rely on sensors to perceive the world around them. Much like our own five senses, these machines use the information they collect to determine what the object that they are 'looking' at actually is. The machines use specific *features* of objects to come up with a probability that an object is of a certain type. Using features that we have developed, the machines can distinguish one object class from another. In our setup, the machines are using information collected from an infrared camera, but future designs will add more sensors. We are hoping to determine how many features are absolutely necessary to classify objects as well as decide which features are redundant.

Our experiment is now using an infrared camera to create images of everyday objects in hopes of being able to classify these objects. This research will eventually be used to help create autonomous machines that can navigate around unstructured environments such as a war zone. The machine's ability to distinguish a tree from a steel pole will allow it to follow directions based upon such landmarks. Such a machine could be used to pick up injured soldiers or deliver supplies to storm victims. The type of object that the machine comes across could determine what the next course of action for the machine will be. For example, if a machine comes upon a brick wall, it will have to go around it, but if it comes across a hedge, it may be able to go through it.

***Features* and *k-Nearest Neighbor*:**

A *feature* is an attribute of an object used to distinguish one object class from another. Ideally these features are invariant to time of day, orientation, and temperature. For our experiment, we will be attempting to use thermo-physical and texture features. We will identify a set of features with optimal discriminating information. For example, an object could have a curvature of 0.3 and an emissivity of 0.025, this feature vector would then be [0.3, 0.025].

An ideal feature will group objects of one type together with minimum overlap between groups, as displayed in Figure 1. Once we have constructed our feature vectors using our collected data, we can then come up with a probability that an unknown data point belongs to a certain object class using the *k-nearest neighbor* algorithm and Bayes' Theorem. The *k-nearest neighbor* algorithm takes the k nearest data points to the feature vector of an unknown object and calculates a probability that this new data point belongs to a specific object class. The *k-nearest neighbor* approach, shown in Figure 2, uses the Euclidean distance between k points to determine which object class the new object should belong to. [1]

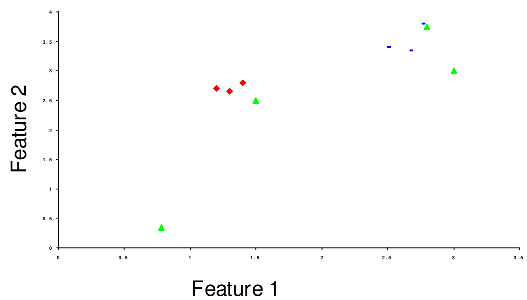



Figure 1: This graph plots the values of different features. We are looking for features that group objects of the same type together without any overlap between object types.

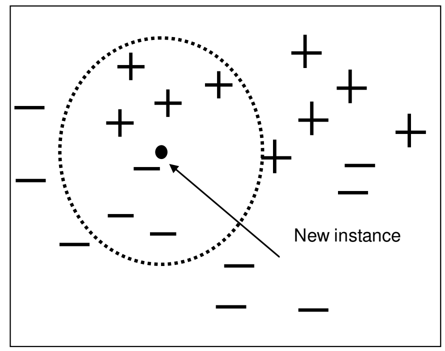



Figure 2: When the new instance is plotted in our feature plot, we must determine if it belongs with the + or the -. (1) Griffith

While the *k-nearest neighbor* algorithm works well when the features are correctly weighted and the object types are sufficiently spread out, it can be troublesome if the features chosen do not group the objects of a class together. Usually the object classes overlap on the feature vectors, making the classification more difficult. The value of k affects the radius of the circle used in Figure 2. A larger k will lead to a larger radius and, in many cases, a better estimate of which object class a new object should belong to. For example, in Figure 2, if we choose k to be one the new data point would belong to -, since the nearest one point is a -. If k was seven, however, the point would be classified as +, since of the seven closest points, there are more +'s than –'s. [2]

Another probability classification that we are using in our data analysis is the Bayesian Theorem. Bayes' Formula says that the probability of T given feature values F, or P(T| F), depends on the probability of the feature values given T as well as the probability of both T and F. Bayes' formula is then,

$$P(T|F) = \frac{P(F|T)P(T)}{P(F)} \qquad (1)$$

These probabilities are calculated using the *k-nearest neighbor* approach on already classified objects. Obviously, we want to come up with features that clearly separate objects from one another. The curvature of a tree would be much higher than that of a brick wall, so those two objects should be fairly distinguishable using a curvature feature.

If a brick wall were rounded, however, this one feature may not be enough to clearly identify the object. If we were to give our machine too many features to help classify objects, however, the computation would become never-ending. In order to make real-time decisions, and thus a practical machine, the number of features analyzed must be limited. Our task is to find the minimum number of features needed to distinguish objects and to have the acquisition of these features be practical. If we were to determine that a certain feature required positioning our machine exactly 30 degrees from normal to that object, the feature would probably not be retained due to its complicated setup.

Our research has been using an infrared camera to take pictures of very specific objects in order to create a data set that we will use to extract pertinent features for classification. We have been collecting this data with hopes of identifying which features are necessary and which are not. We hope to find features that are relatively invariant to temperature, time of day, rotation and orientation. With this information, our goal is to allow for an unstructured environment through which the machine will navigate.

If our features only work when the sun is out, for example, our machine would not be able to classify objects at night, or if it was raining. The objective of this work is to create a machine that can classify objects in an environment in which trees might not always be upright, and where a road may have obstacles. If our features depend on the 'norm' such as a tree's orientation being vertical for classification, then that feature will not work for our application. Finding features that are relatively invariant, therefore, is crucial to our experiment.

**Infrared:**

Every object is radiating some amount of thermal energy. This energy, or blackbody radiation, has a maximum intensity in the infrared region of the electromagnetic spectrum. Any object above absolute zero, even the coldest of cold, is giving off infrared radiation. Unlike visible light, which relies on reflection, infrared radiation can be detected without any light source at all. [3] Infrared cameras display different radiation levels as different shades of gray. An object that was emitting highly, due to either a high emissivity, a high temperature, or both would be assigned a lighter shade of gray than a cooler, and/or less emissive object. [4]

Our camera was initially set to automatically make the image clearer by enhancing the differences between two objects. This enhancing feature is an attempt to make the infrared images easier to interpret visually. Two objects at similar levels of emissivity and equivalent surface temperatures will be assigned similar gray levels, and will thus be difficult to distinguish in the infrared image. With the enhancing feature turned on in the camera, however, the objects with similar emissivity will have drastically different gray levels, making the slight emissivity differences very easy to detect in the infrared image.

So an object at 20 degrees Fahrenheit (F) was nearly black when compared to an object at 60 degrees F. When that same 20 degree F object was placed next to a cooler, less emissive 10 degree F object, however, the first object, which was nearly black a moment ago, became much whiter an example of which can be seen in Figures 3 and 4. In an attempt to keep more consistency with our data, we have turned off this 'enhancing' part of the camera so that a 20 degree F object of a certain emissivity will be assigned approximately the same gray level every time, thus making classification easier since every 20 degree F object of a certain emissivity will be assigned approximately the same gray level.

Figure 3: This brick is much cooler than the cedar stump, and thus looks black.

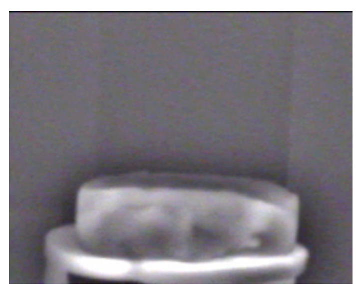


Figure 4: This brick, slightly warmed up now looks white compared to the cedar stump.

**Emissivity:**

The emissivity is defined as the ratio of the energy emitted by an object to that of a blackbody at the same temperature and wavelength. [5] The emissivity of an object is an intrinsic property that is a function of surface temperature, surface texture, shape of the object, and viewing angle. These properties are the reason that we are attempting to use emissivity as a feature in our research. The emissivities of several common objects are included in Figure 5. [6]

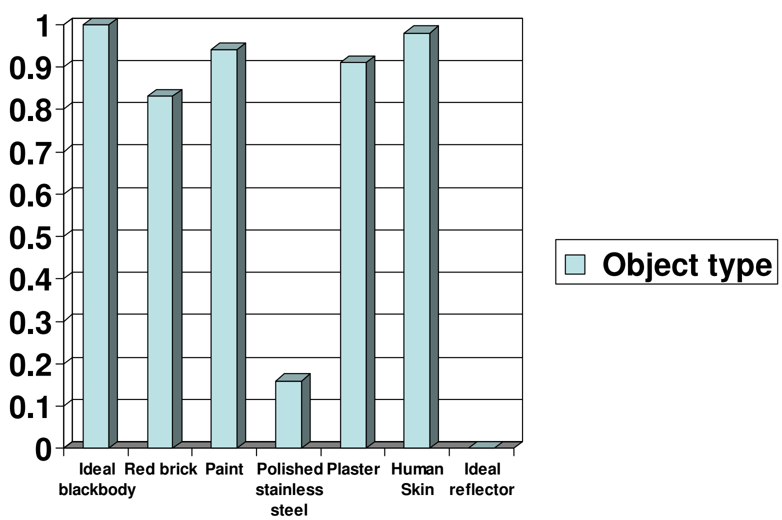



Figure 5

Graph adapted from Holst (2)

Emissivity, however, is not strictly a material property. The surface of the emitting object can affect its emissivity. A rough surface will emit more than a smooth surface since a rough surface has a larger emitting area. The shape of an emitting object also changes the emissivity. A convex object will have a lower emissivity because the radiation is spread out over a larger angle than a flat or concave object. The emissivity of objects also depends on the viewing angle. Some objects emit at a certain amount at a normal angle and a much lower amount at an angle of forty five degrees because these objects do not emit the same at all angles. [7] Usually, a curved object can be identified by its infrared image because they will tend to glow more on the edges in the infrared image than a flat object.

Robert Madding [8] believes that the emissivity of an object can be extracted from a thermal image of that object as long as the background radiation, the ambient temperature and the reference emitter with a known emissivity are all present. Building on Madding's research, we are attempting to use his findings in our analysis of the emissivity feature. Using a thermometer mounted on the robot, we are recording the ambient temperature at the time and location of all of the objects. We are collecting images of a crinkled sheet of aluminum foil placed on the objects at the different angles to get an average reading of the background radiation incident on the object. Also, by means of a piece of black electrical tape attached to all of our objects we hope to create a reference emitter that is constant on all of the objects. Electrical tape has a very high emissivity which makes it a practical blackbody. Using these tools, our initial goal is to recreate Madding's findings and to be able to calculate the emissivities of our objects.

Some of the limitations of our current setup include the uncertainty of the location of the reference emitter, or in our case the black electrical tape. On some objects such as the hedges, the electrical tape is difficult to detect when there is no direct sunlight on the object. Also, the amount of ambient light available on objects is not always uniform. A tree may be glowing when looked at from one angle, and much darker when viewed from another angle. Our ability to detect objects in thermal images collected in the dark depends on the thermal properties of the object and the time history of solar radiation. For example, steel poles in particular have a tendency to heat up more slowly during the daytime solar cycle due to their high specific heat.

Consequently, when a low ambient temperature exists during daylight hours, the steel poles emit minimal thermal radiation after sunset, making them difficult to detect in an infrared image. In these images, the boundaries of the object itself are nearly indistinguishable from the foreground. Also, the electrical tape is difficult to distinguish from the steel pole due to the approximately equivalent emissivity values of the electrical tape and paint on the surface of the steel pole. Whether or not we will be able to use these ‘outliers’ in our data analysis still remains to be seen. These issues imply the need to identify limitations on the use of emissivity as a feature in our classification process.

**Experiment:**

The ‘big picture’ explanation of our experiment is to discover features and objects that are relatively invariant to weather, viewing angle, size and visibility conditions. With these features, we hope to come up with a probability model that will help our robot determine if an object is of a certain class. Since these features need to be relatively constant, we are collecting data at different times of the day, under different climate conditions, and of thermally different types of objects. We hope to show that the thermal properties of a tree are different enough than those of a steel pole so that classification between the two will be straightforward.

Our experiment uses thermal images captured using our robot, rMary, shown in Figures 6-8.

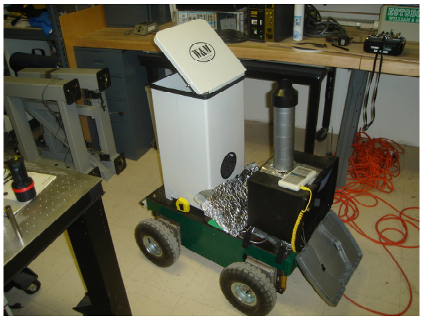


Figure 6: The white steel box on the front acts as a periscope on rMary.

Figure 7: The camera is housed securely in the bottom of the steel box. On the lid of the box is a reflective aluminum sheet.

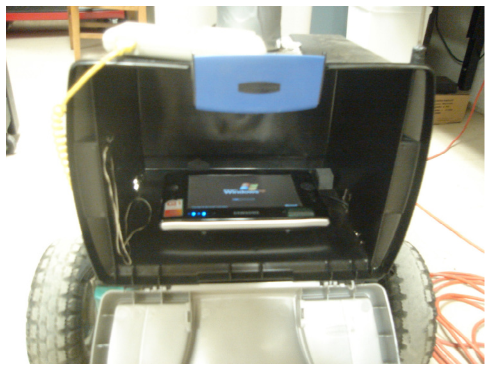


Figure 8: The plastic box in which we hold the tablet PC.

Using Bayes' Theorem along with a feature vector, we hope to come up with probabilities that an 'unknown' object is of a certain type. This theorem is helpful because we could use the knowledge of past data to come up with these probabilities. For example if T is the event that the object is a brick wall and F is the event that the object has emissivity 0.65, then we can use our past knowledge of the probability that a brick wall has emissivity 0.65 times the probability that the object is a brick wall divided by the probability that an object has emissivity of 0.65.

We are attempting to show that using the *k-nearest neighbor* algorithm along with Bayes' Theorem, a machine can be 'taught' how to distinguish one object from another. While our current setup only uses the infrared camera, we hope to eventually add different sensors such as ultrasound and perhaps even GPS to improve the robots chances of navigating around an unstructured environment.

The research that we are performing will eventually be used in real world situations in which a robot will need to make decisions and take action based upon incomplete data in an uncontrolled environment.  Such a situation could be a robot used in a war zone to deliver supplies to a certain location using directions that may be flawed or invalid due to a constantly changing environment.

**Setup:**

Our robot, rMary is operated by a remote control which allows rMary to speed up, turn around, and slow down.  rMary is battery powered and equipped with a thermal camera as well as a tablet PC for running the camera's software.  While large tank-like robots that can move at high rates of speed have their applications, our robot is designed to be much smaller and to move at roughly the speed of a walking human. This way our robots could eventually move about the pedestrian world without getting in the way of our daily lives.  Such a robot could be used to perform tasks alongside soldiers without getting in their way.  Also, a robot that moved at the speed of a truck could potentially cause innocent bystanders harm if the programming failed, whereas a smaller, slower robot in the same situation would be troublesome, but not harmful.

Mounted on the front of the robot is a steel box with a hinged lid.  Inside the lid, we have attached a polished aluminum plate to serve as a reflector.  This plate, along with a hinge that sets the lid at a forty five degree angle, creates a periscope on the robot.  We designed rMary so that the infrared camera that we use to capture all of our thermal images is safely placed in the bottom of the steel box. The camera is placed on top of several layers of cut foam padding so that the bouncing of the robot in motion will not cause harm to the camera.

The camera is aimed so that it looks straight up at the reflective aluminum plate. The angle of the lid can then be adjusted so that the image that the camera collects is either directly in front of the robot, or slightly up. The reason that we decided to use a reflection rather than a direct shot was so that we could protect the camera while still being able to capture images. The back side of the steel box has a hole in order to be able to manually adjust the focus of the camera.

Behind the steel box, we have placed a plastic box turned on its side. This box is used to hold the tablet PC as well as the battery pack while the robot is in motion. We used Velcro to hold these items in place within the plastic box. Mounted on top of the box, we added about two feet of PVC piping capped with a funnel shaped cone. This was added to prevent glare from the sun, which had made it nearly impossible to see the screen of the tablet PC. With this new setup, we are able to keep the tablet PC shaded while still being able to easily see the image. We added an external mouse to the tablet PC in order to be able to manipulate the tablet PC without having to open the box that it sits in.

Our reasoning behind this periscope design was to be able to collect data while the robot was in motion. The tripod that we originally had attached to the front of the robot was not sturdy enough to be sure that the camera would not fall off. With the new periscope design, however, the camera is securely and safely attached to the inside of the steel box.

The problem was that since the lid of the steel box was adjustable, that meant that the reflective plate attached to the lid was not completely stationary. In fact, when we tried to collect data while driving the robot, we found that the terrain was often so bumpy that the lid was bouncing, and the images were shaky and blurry. We hope to fix this problem in the future by securing the lid with a non-adjustable pole that can be taken off when needed. In the mean time, we are left with taking still images.

**Procedure:**

When taking out the robot in order to collect data, we first make sure that the power bank is fully charged. Then we can turn on the robot by flipping the four switches on the back of the robot. Next, we attach the tablet PC to the power bank and turn it on. Once the tablet PC is running, we adjust the display settings so that the software that the camera uses will run correctly. The pixilation of the screen must first be set to 1024x600 by pushing the round button on the left hand side of the tablet PC, as shown in Figure 9, and then clicking on 1024x600.

Button to
switch
pixilation

Figure 9: The tablet PC

After the screen updates itself, we change the settings by clicking on the Start button and then selecting Control Panel. Under Control Panel, we then select Display. There is a tab at the top of the Display window called Settings, which we then select. Under the Settings tab, we then move the Screen Resolution to 1024x768 and click on Apply. Once that is done, we can plug in the camera to the right hand side of the tablet PC and turn it on by flipping the switch on the robot just behind the steel box.

Finally, we start the TurtleBeach program which allows us to preview the thermal images on the tablet PC. Next we attach an external mouse to the tablet PC and place the PC in the plastic box on the back on the robot, making sure that the screen lines up with the viewing tube on the top of the box.

Once we have the camera hooked up and the robot turned on, we can begin collecting data. Before we head outside, we make sure that we have our data sheet to record the ambient temperatures as well as any interesting or troublesome images seen along the way. If we are collecting data in the dark, we must also be sure to bring a flashlight to be able to see the ambient temperature reading as well as the measuring tape. Next we must bring along the crinkled sheet of aluminum foil with duct tape on the corners to be used to collect data about the background radiance. Finally, it is always a good idea to bring along a roll of duct tape as well as a roll of electrical tape in case the electrical tape on the objects has come off or the duct tape on the aluminum foil does not stick.

When we finally get out the door, we first drive the robot up to one of our predetermined objects. We position the robot so that it is normal to the black electrical tape on the object and then use the measuring tape that is attached to the front of the robot to make sure that it is eight feet from the front of the object. Once the robot is lined up with the electrical tape, we open up the lid to the first position, or the forty five degree slot. If the image seen through the tube is off center, we then adjust the robot by pulling or pushing on the rear end until the image is centered.

If the image does not clearly show where the black electrical tape is on the object, we then take a ball of duct tape which is easy to see in the thermal images, and place it to the side of the electrical tape, as shown in Figure 10. We make sure to leave some of the electrical tape uncovered, and then take an image both with the duct tape and without. It is important to not completely cover the electrical tape, because the duct tape tends to warm up the object where it is placed and causes the object to have a faint whitish spot, even after it is removed as shown in Figures 11 and 12.

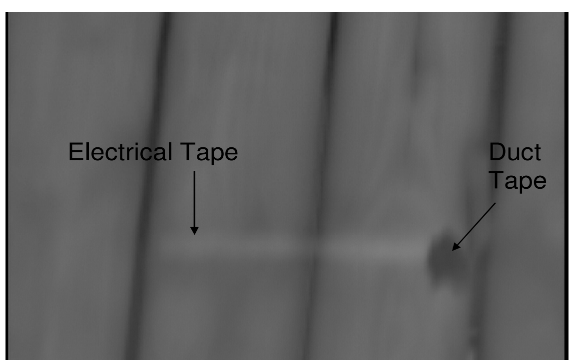



Figure 10: Locating the electrical tape using duct tape

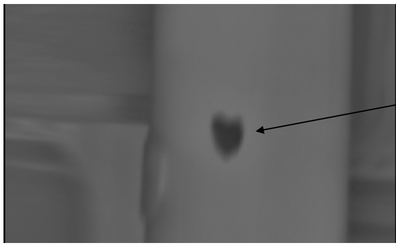



Figure 11: On some objects, the duct tape actually heats up the object and affects the image

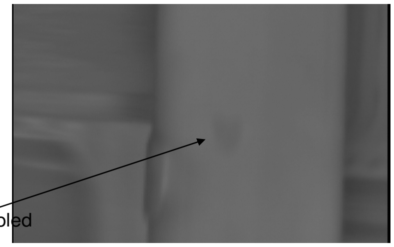



Figure 12: The steel pole was heated up by the duct tape, as can be seen by the darker spot on the image.

After we get the two normal images, we move the robot to a forty five degree angle from normal and line it up with the electrical tape once again. Next we measure the distance from the object and make sure that it is about eight feet. We use the duct tape to identify the electrical tape in the thermal image once again, and then capture the forty five degree image with and without the duct tape.

Next, we take the aluminum foil and tape it to the object while leaving the robot in place at the forty five degree angle. We then capture an image of the aluminum foil, as in Figure 13, making sure that a sizable portion of the image contains the aluminum foil. Next, we move the robot back to a normal position, making sure that it is once again eight feet from the object. If needed, we can adjust the aluminum foil so that the image captures most of the foil. We capture the second aluminum foil image and then record the ambient temperature before moving on to the next object.

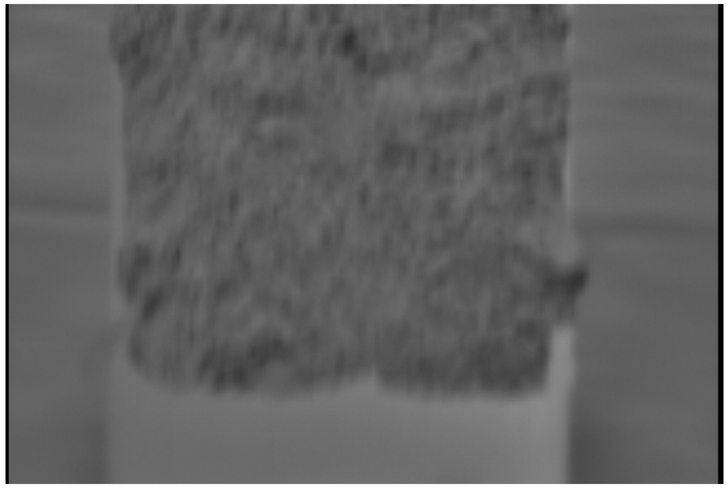


Figure 13: Aluminum foil wrapped around the birch tree

The reason that we put the aluminum foil on last and then move the robot back to the normal position is that the foil heats up or cools down the object, thus changing its thermal properties for the next image. By saving the foil for the end, we do not have to worry about any effect that it may have on the object.

We have been collecting data on ten specific objects common to the William and Mary campus. These objects, shown in Figures 14-23, include a picket fence, a hedge, a gray steel pole (one of the emergency poles), a brick wall, a birch tree, a basswood tree, a wooden wall, an octagon pole, a cedar tree, and a green steel pole.

Figure 14: The picket fence

Figure 15: The hedge

Figure 16: The gray steel pole

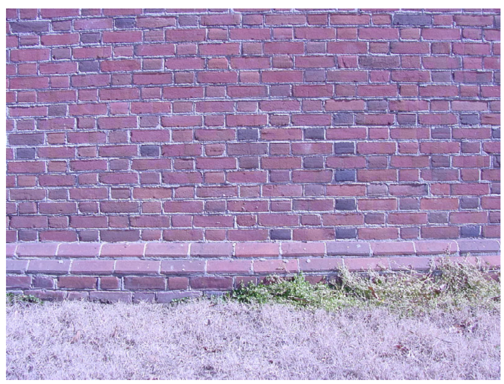


Figure 17: The brick wall

Figure 18: The birch tree

Figure 19: The basswood tree

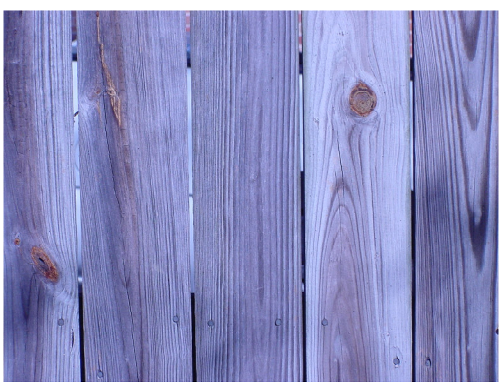


Figure 20: The wood
wall

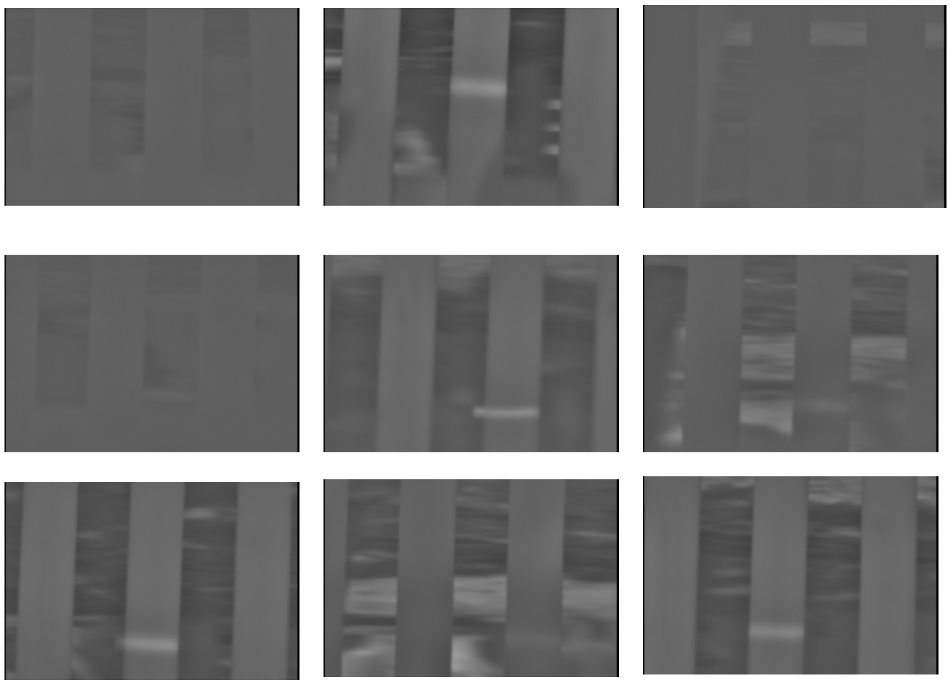
Figure 21: The octagonal pole

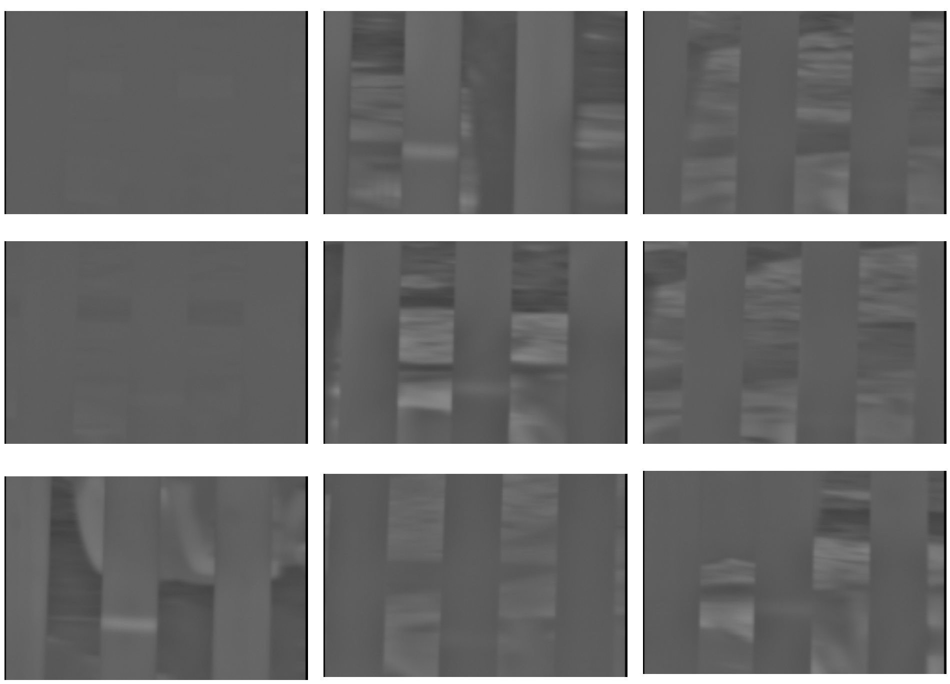
Figure 22: The cedar tree

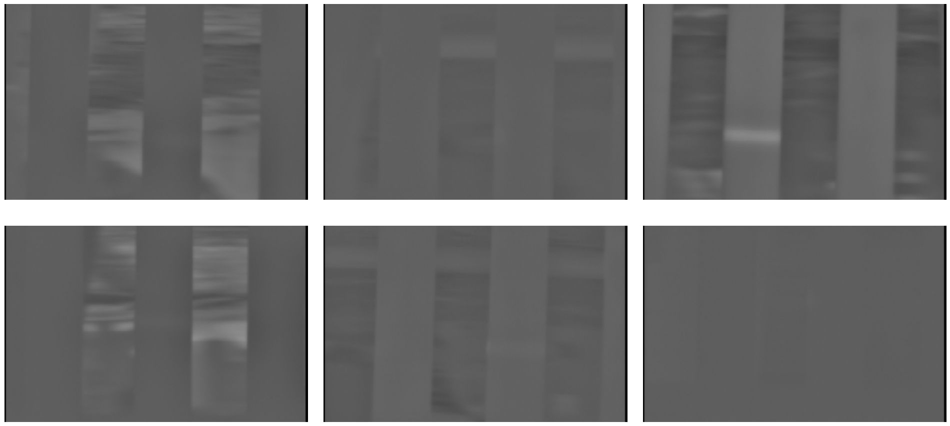
Figure 23: The green steel pole

Included in the Figures below are the normal images of all of the different object classes that we collected during the course of our experiment. The images in which the object is hard to see were taken either before sunrise, or after sunset. The images in which the object, as well as the electrical tape, is very clear were taken during the mid-morning or the afternoon when the objects had had time to collect and then emit the thermal energy

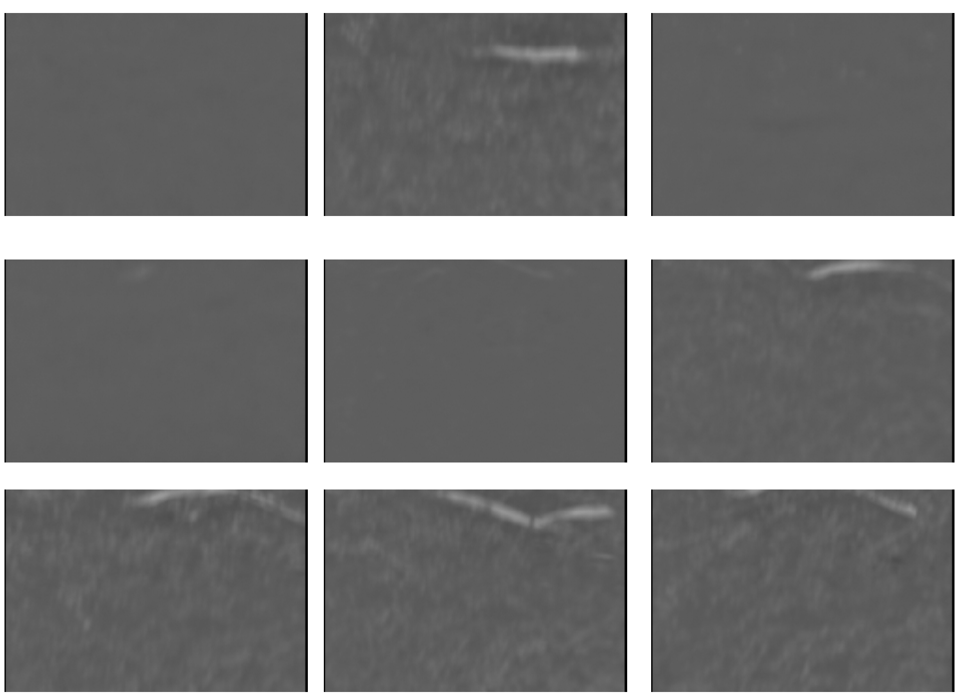
**Picket Fence Images:**

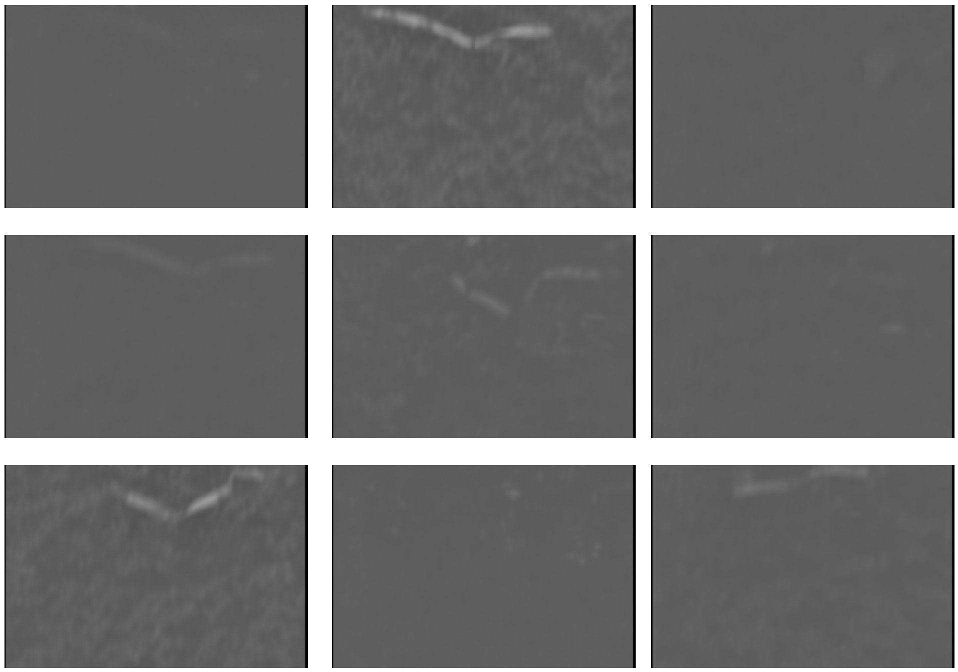


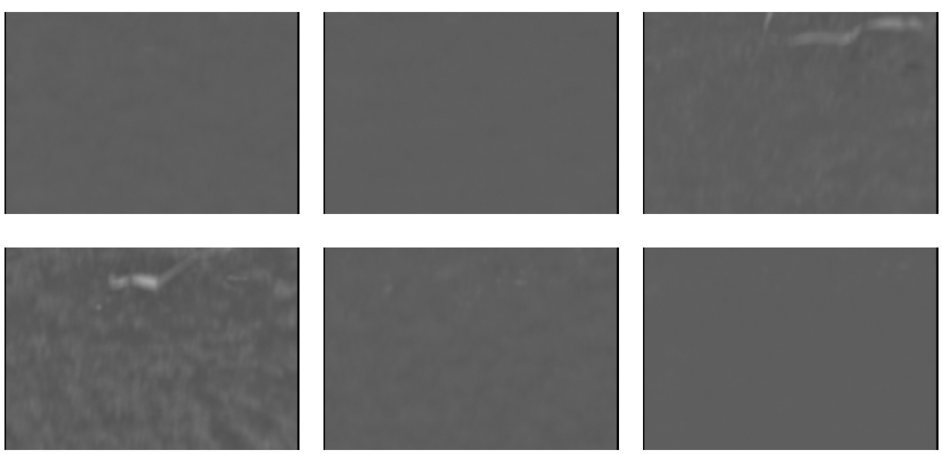




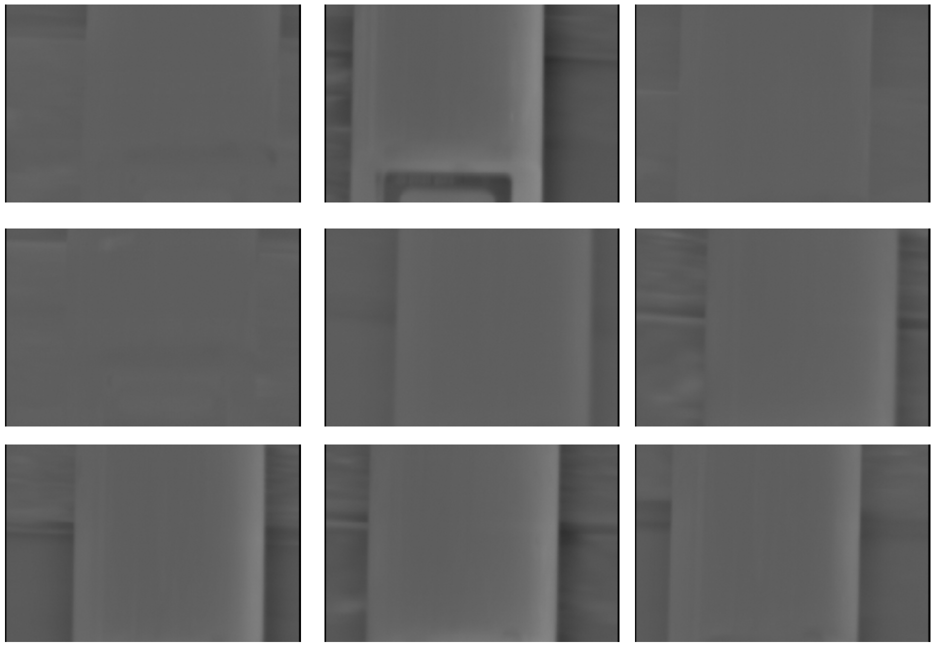
**Hedges:**

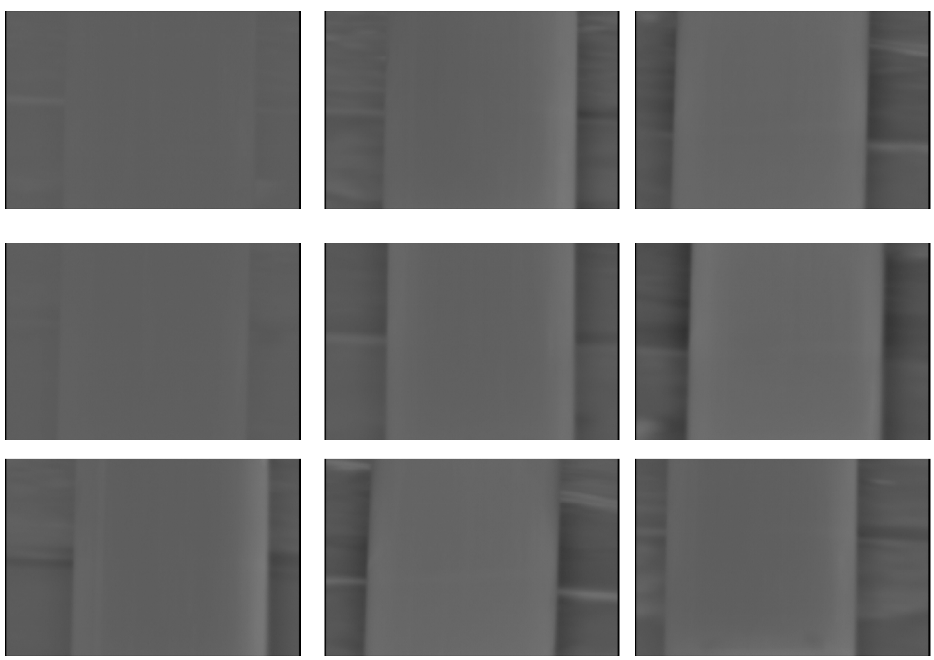


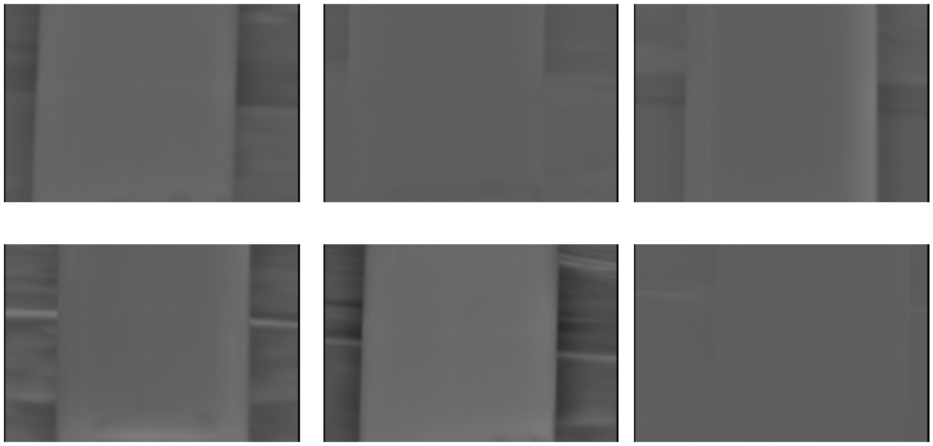




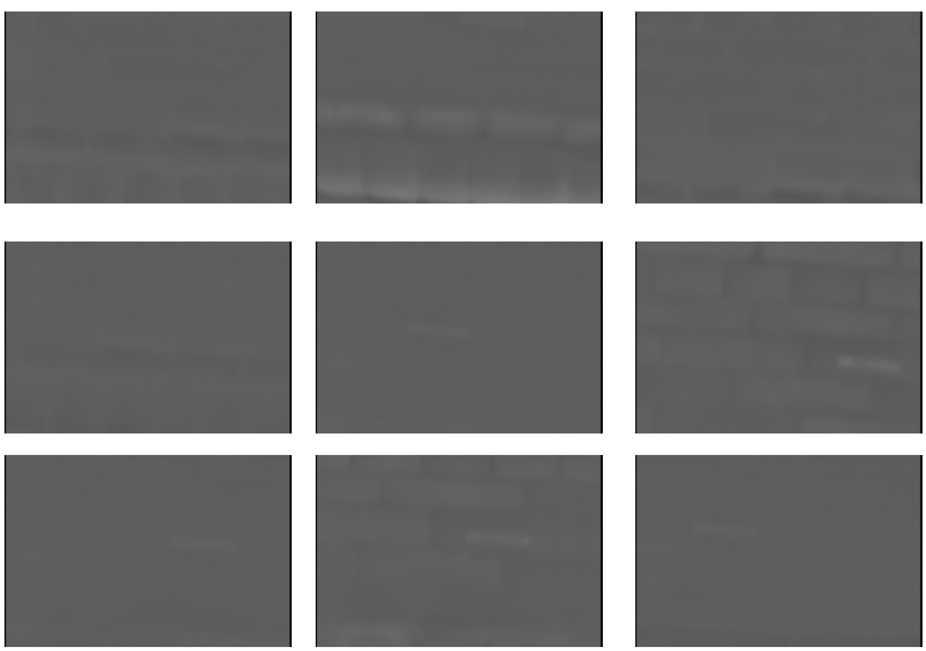
**Gray Steel Pole:**

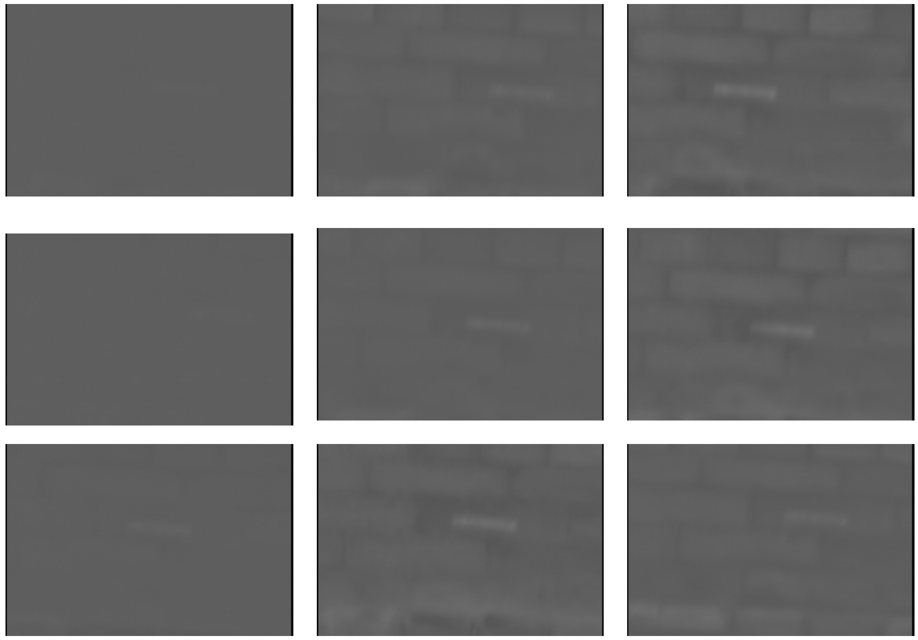


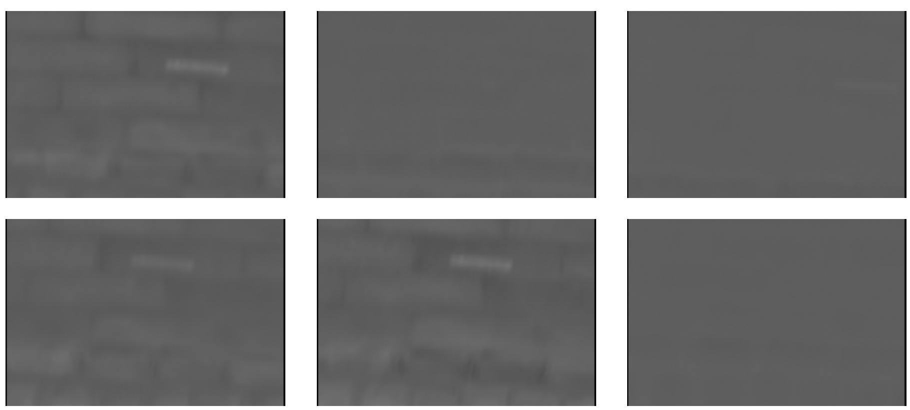




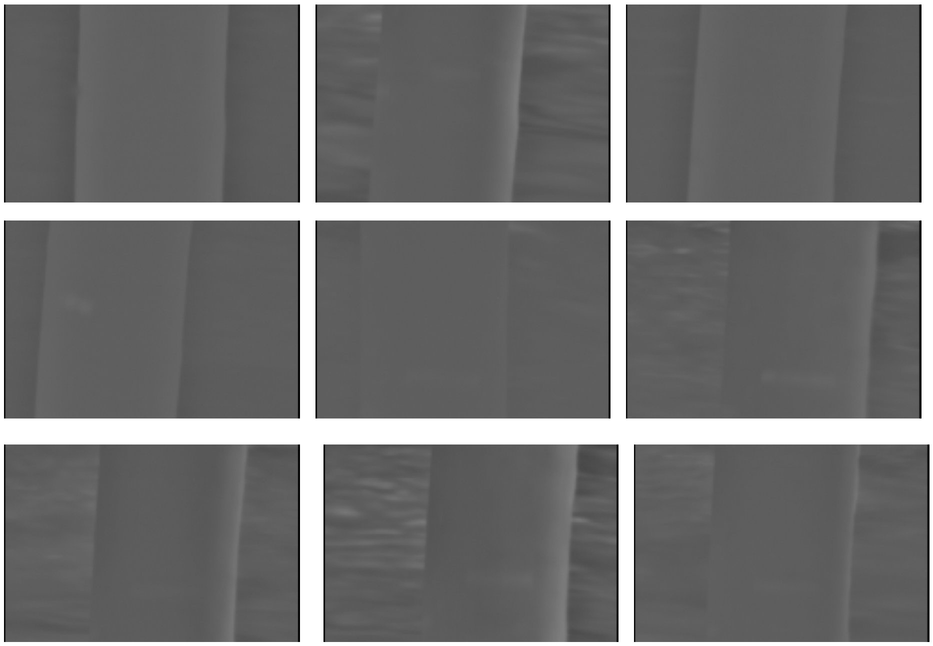
**Brick Wall:**

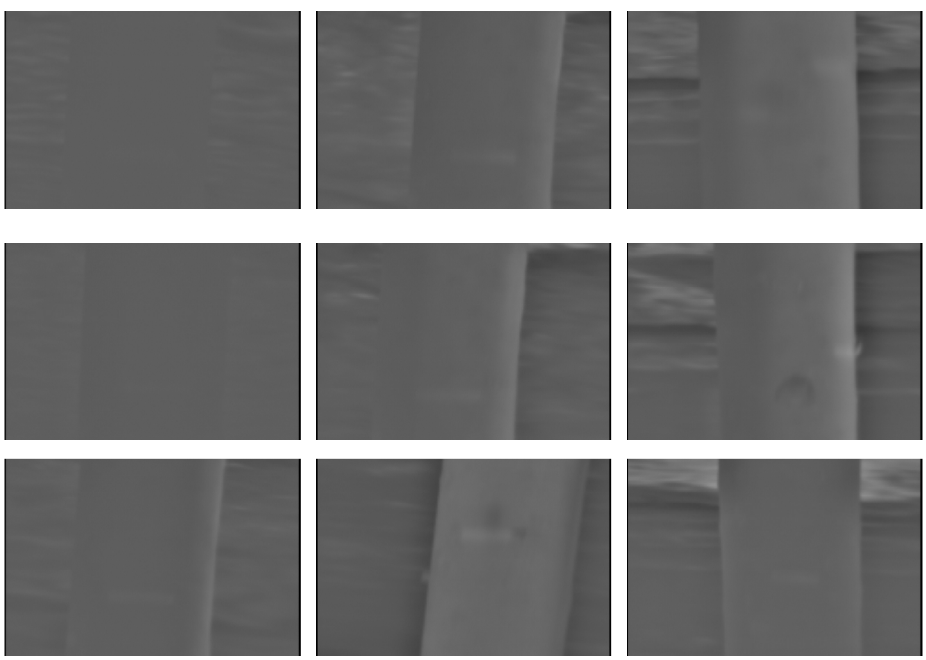


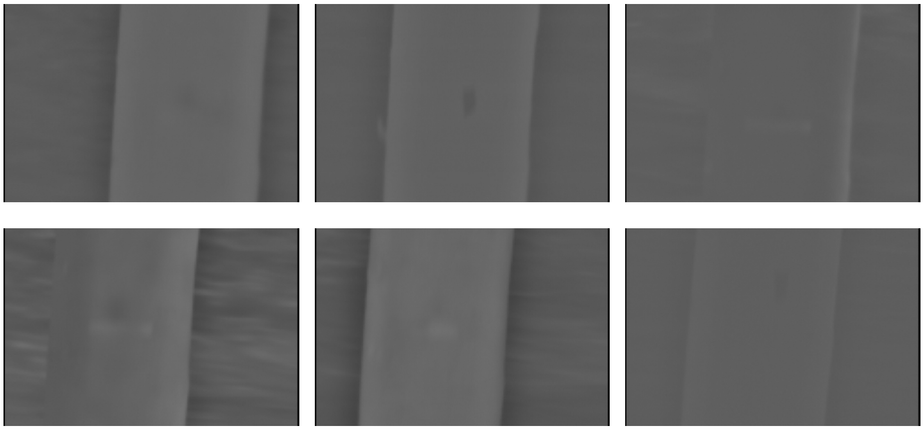




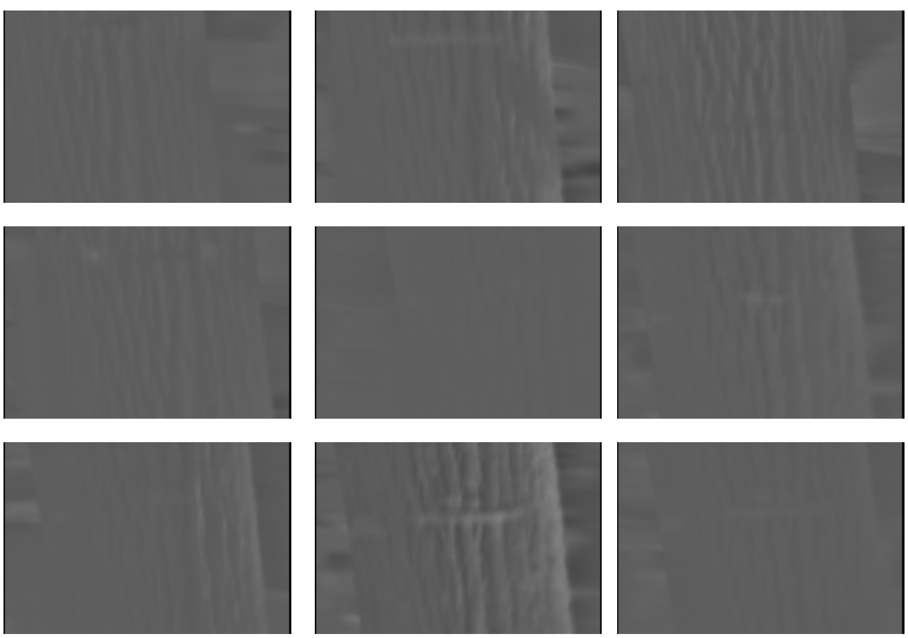
**Birch Tree:**

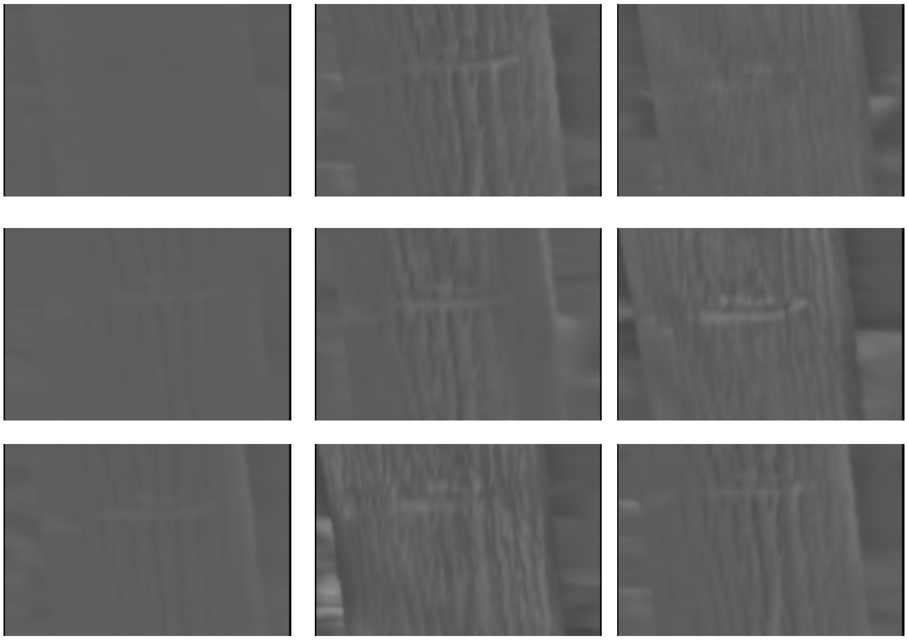




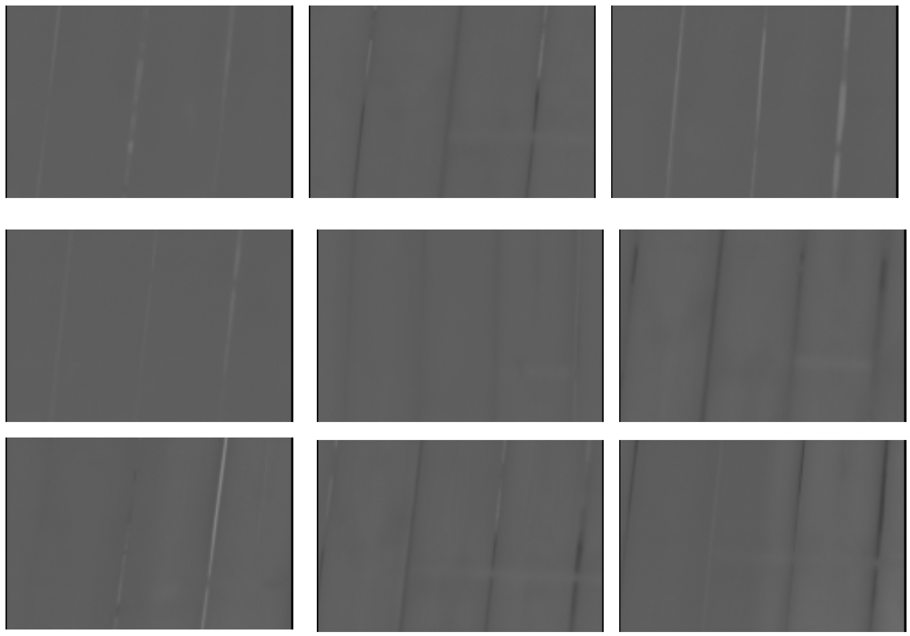


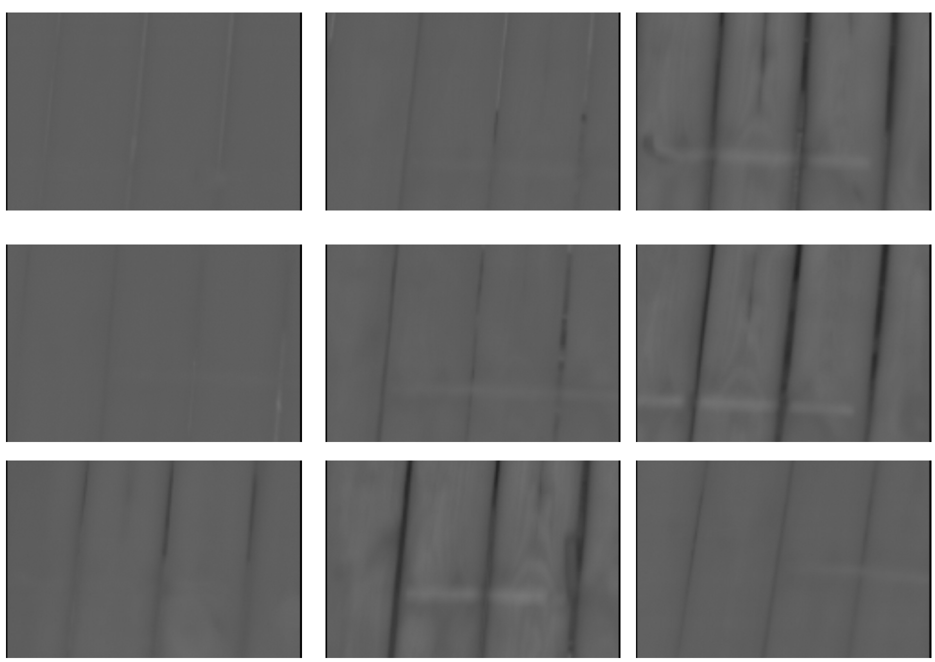
**Basswood Tree:**

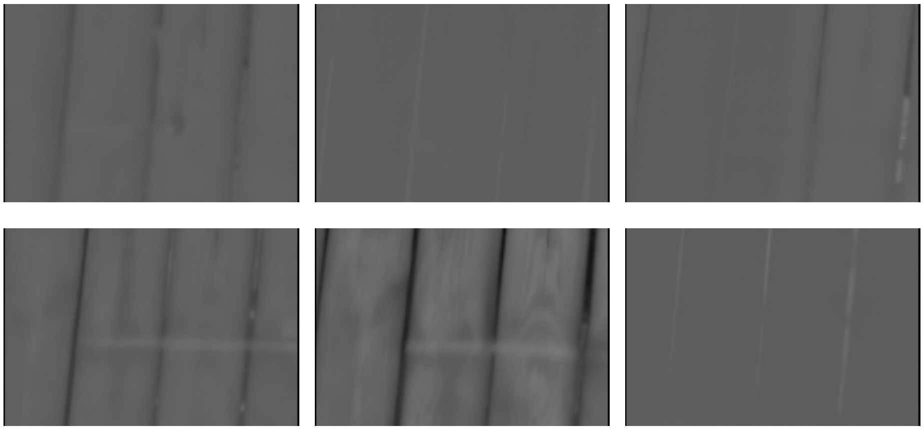




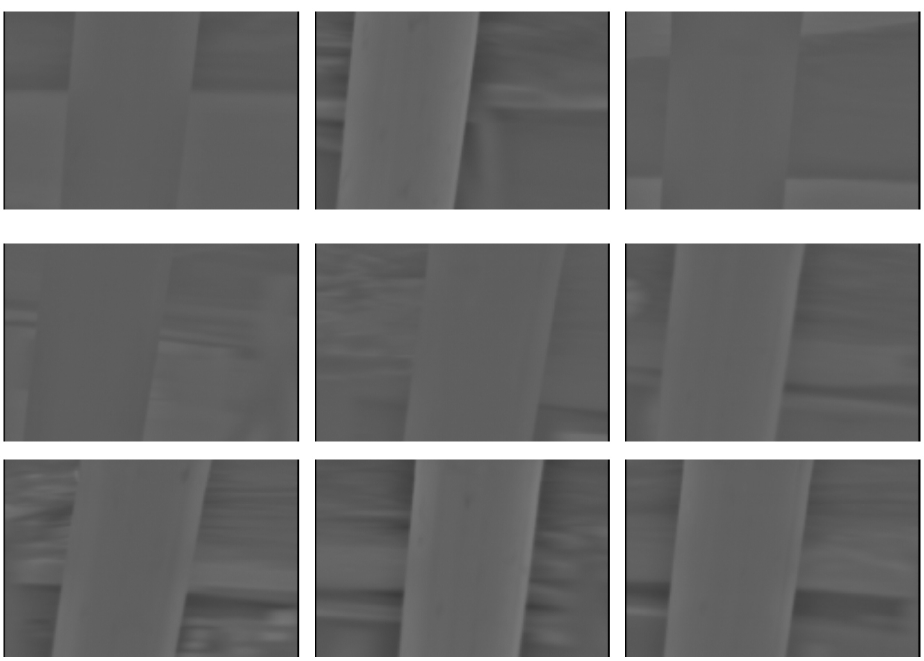
**Wood Wall:**

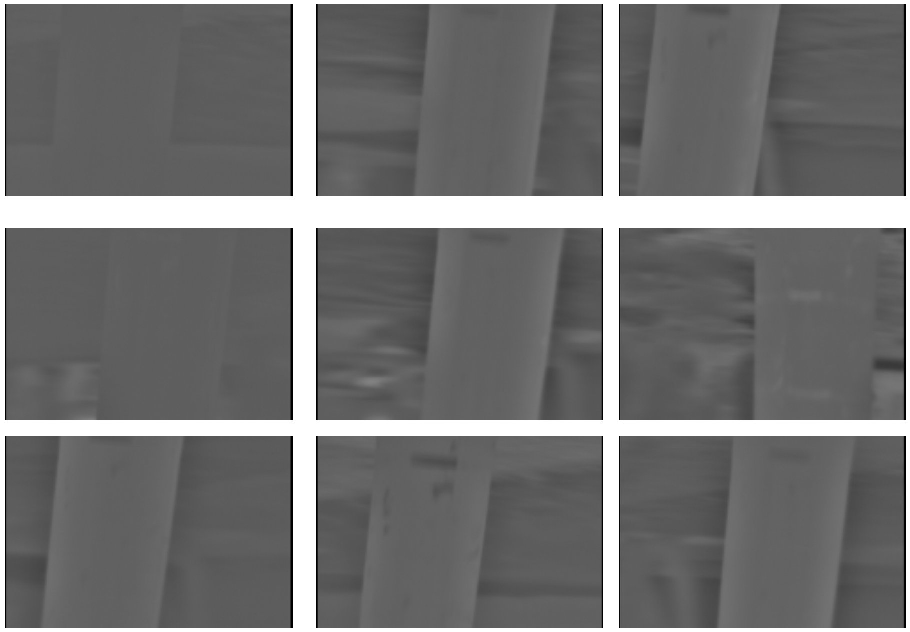


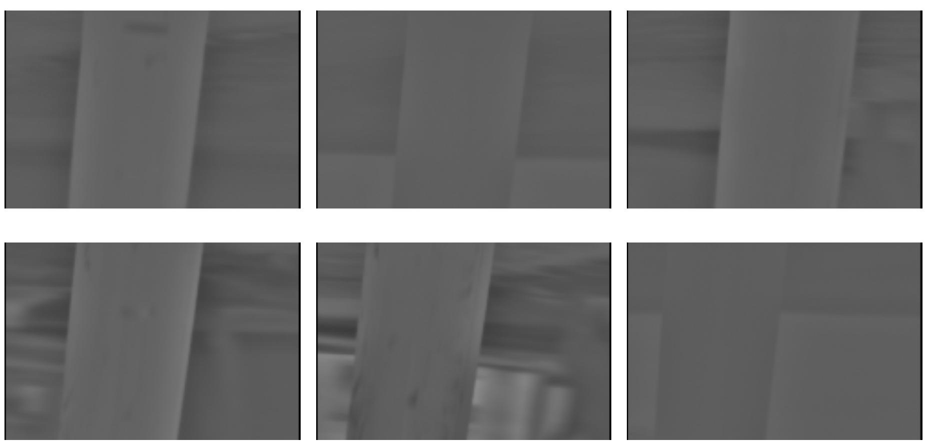




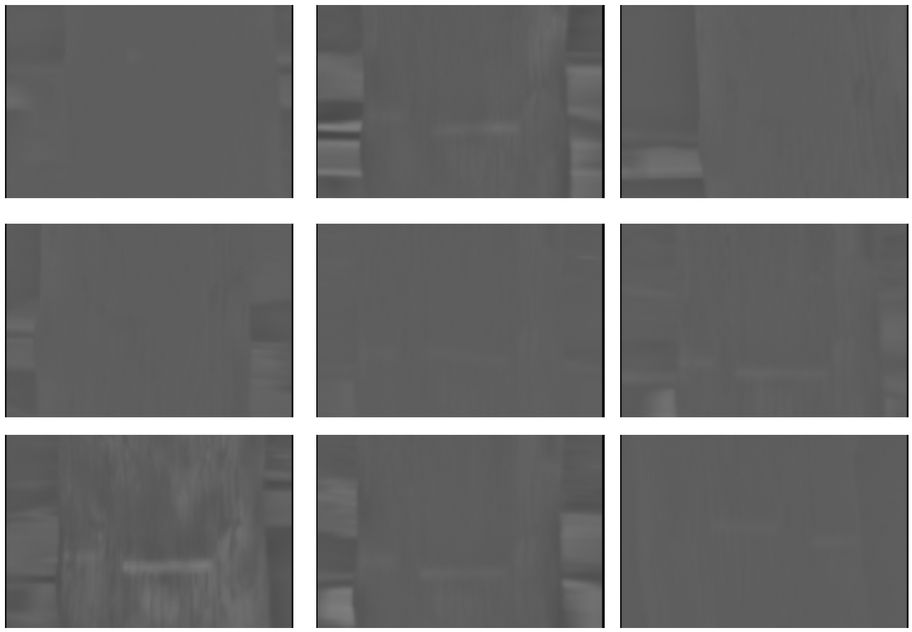
**Octagonal Pole:**

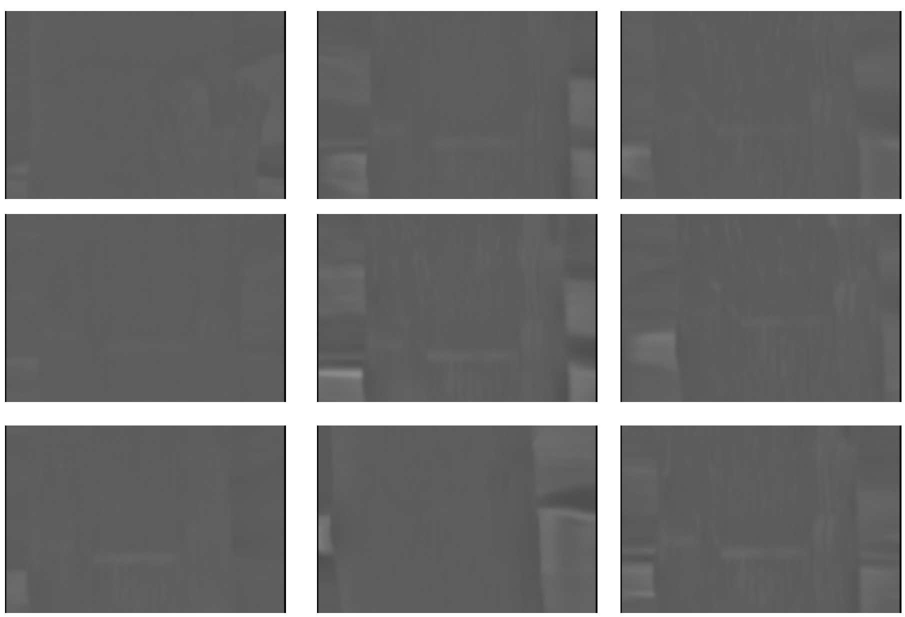


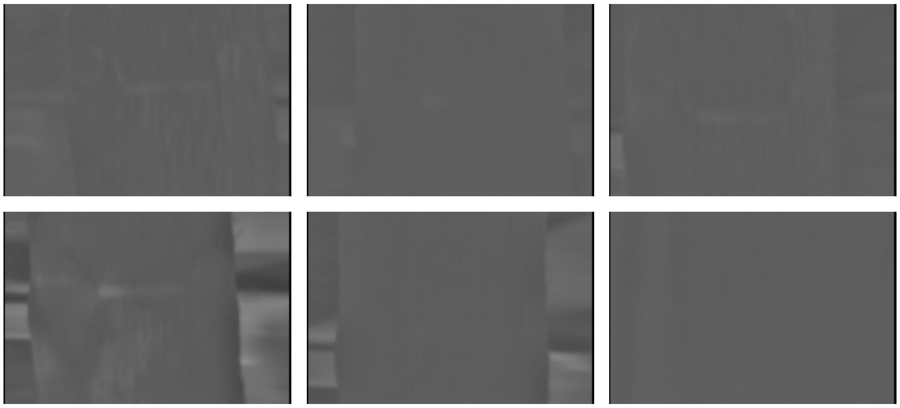




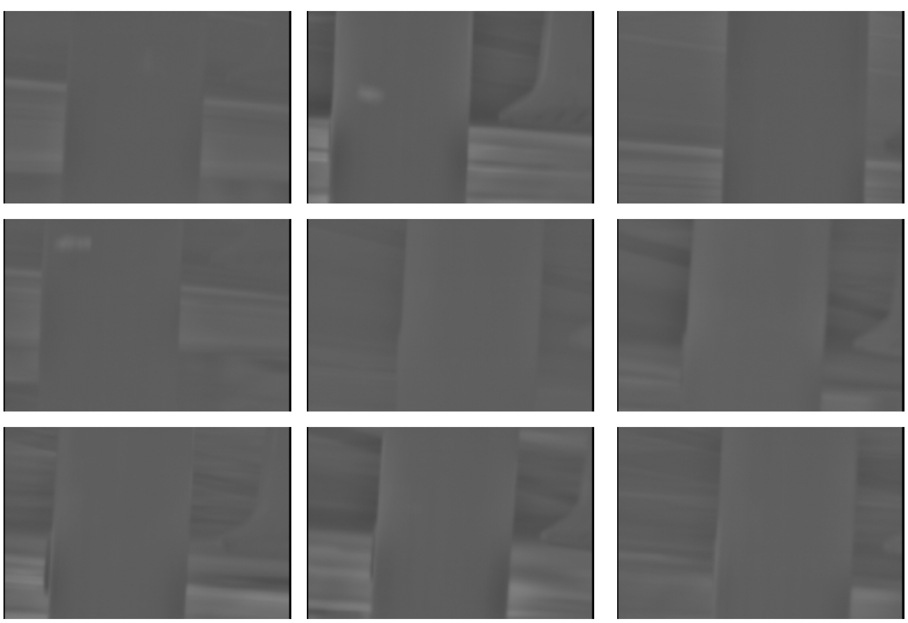
**Cedar Tree:**

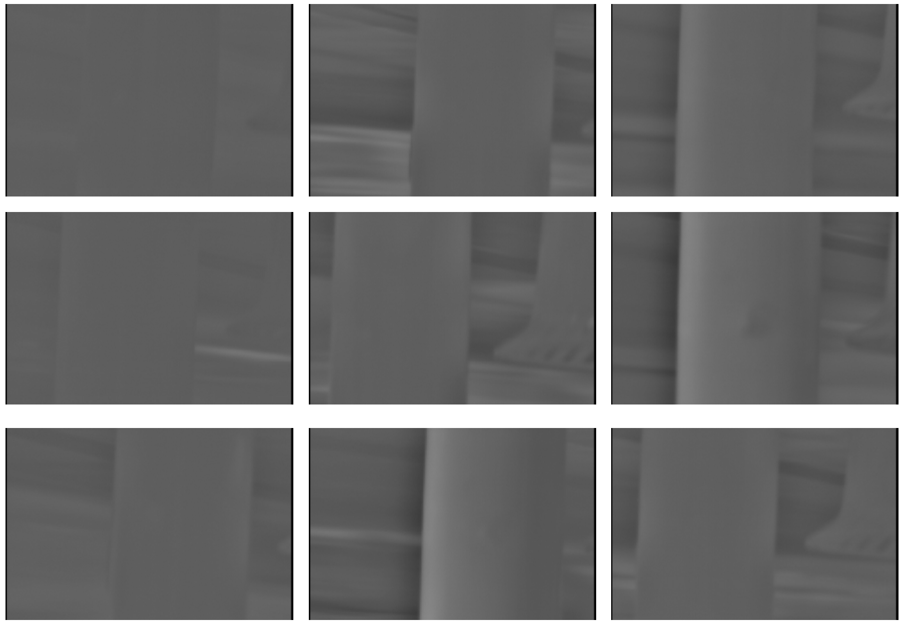


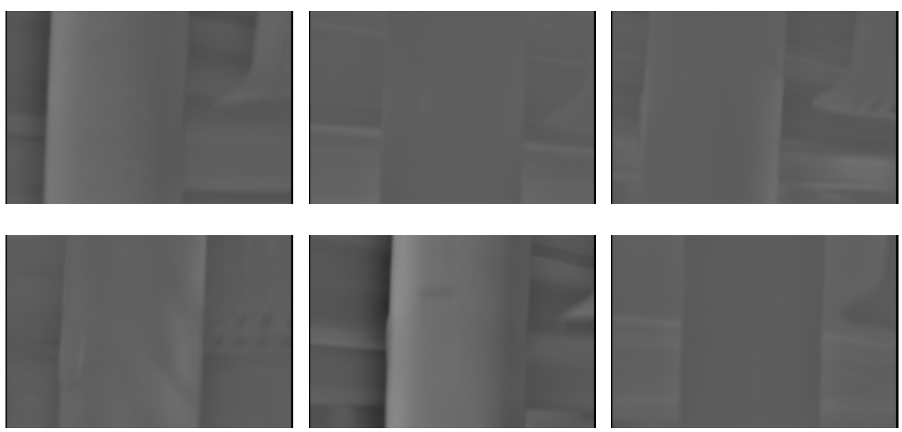




**Green Steel Pole:**







**Analysis:**

Once we collected all of our data with rMary, we began to analyze it to find features which separated the different types of objects from one another. The feature vector that we are analyzing is seventeen dimensional now, but since we cannot render that many dimensions, we must instead look at two or three dimensional cuts of the larger feature vector. For example, we can examine the emissivity feature versus the ambient temperature feature on a scatter plot.

The features that we are examining include the ambient temperature, the temperature rate of change, the mean radiance of the object ($L_o$), the mean radiance of the reference emitter ($L_r$), the quotient of the mean radiance of the object and the mean radiance of the reference emitter ($L_o/L_r$), their difference ($L_o$-$L_r$), and their standard deviation. We also are examining the mean background radiance ($L_b$), the difference between the background radiance and the object radiance ($L_b$-$L_r$), the emissivity, the mean intensity, the average contrast, smoothness, the third moment, uniformity and entropy.

The ambient temperature is the temperature recorded using a thermometer at the time that the image was collected at each object. The rate of change of the temperature was calculated from the KECK lab [9] using the ambient temperatures recorded at the time that the images were collected and thirty minutes before. The mean radiance of the object is calculated by finding the mean gray level in a segmented section of the object. The mean radiance of the reference emitter is the mean gray level of a segmented section of the electrical tape on the objects. The standard deviation is how much spread there is between data points in the two data sets.

The mean background radiance takes a segmented section of the aluminum foil that we placed on all of our objects and then calculates the mean gray level. We calculate the emissivity of the objects by using the equation: [10]

$$E_o = ((L_o - L_b)/(L_r - L_b)) * E_r \qquad (2)$$

where Eo is the estimated emissivity of the object, $L_o$, $L_b$, and $L_r$ are as defined above, and Er is the emissivity of the reference emitter, or the electrical tape.

We then use a Matlab function from Digital Image Processing Using Matlab [11] called "statxture." (Appendix 1) This function looks at several texture features such as smoothness, third moment, or skewness, uniformity and entropy, or randomness. These features are all defined in Figure 24. [12] Using statxture along with our other features, such as ambient temperature, Lo, and Lb, we were able to create a giant matrix which we then plotted using a program called Minitab.

**TABLE 11.2** Some descriptors of texture based on the intensity histogram of a region.

| Moment | Expression | Measure of Texture |
|---|---|---|
| Mean | $m = \sum_{i=0}^{L-1} z_i p(z_i)$ | A measure of average intensity. |
| Standard deviation | $\sigma = \sqrt{\mu_2(z)} = \sqrt{\sigma^2}$ | A measure of average contrast. |
| Smoothness | $R = 1 - 1/(1 + \sigma^2)$ | Measures the relative smoothness of the intensity in a region. $R$ is 0 for a region of constant intensity and approaches 1 for regions with large excursions in the values of its intensity levels. In practice, the variance used in this measure is normalized to the range [0, 1] by dividing it by $(L-1)^2$. |
| Third moment | $\mu_3 = \sum_{i=0}^{L-1} (z_i - m)^3 p(z_i)$ | Measures the skewness of a histogram. This measure is 0 for symmetric histograms, positive by histograms skewed to the right (about the mean) and negative for histograms skewed to the left. Values of this measure are brought into a range of values comparable to the other five measures by dividing $\mu_3$ by $(L-1)^2$ also, which is the same divisor we used to normalize the variance. |
| Uniformity | $U = \sum_{i=0}^{L-1} p^2(z_i)$ | Measures uniformity. This measure is maximum when all gray levels are equal (maximally uniform) and decreases from there. (aka: Energy) |
| Entropy | $e = -\sum_{i=0}^{L-1} p(z_i) \log_2 p(z_i)$ | A measure of randomness. |

Figure 24

With all of these features selected and all of the pertinent images collected we then set out to analyze the data that we collected. Once we removed the outliers from our data set, we then began to analyze the relation between the features. Most of the outliers we detected were collected before sunrise when the objects had already emitted most of the thermal energy that they had collected during the day. We believe that we have these outliers because the ambient temperature and the object temperature were nearly identical, making it very hard to approximate some of the features. This observation may prove to be a limitation of our method since we want to be able to classify objects no matter what time of day it is.

First we used a Matlab function that we wrote called "irfeatgenerator" to input all of our images and to properly calculate the features that we were looking for. (Appendix 2) This function created our multi-dimensional feature matrix that allowed us to graphically represent the data that we were analyzing. With the graphs, we were specifically looking for features that were highly correlated, as well as ones that separated out the data by object type. If any features are highly correlated, such as in Figure 25 which plots entropy vs. uniformity, we can remove one or more of them to decrease redundancy in our feature vector. This decrease in our feature vector size helps decrease the computation time needed for our robot to make a classification.

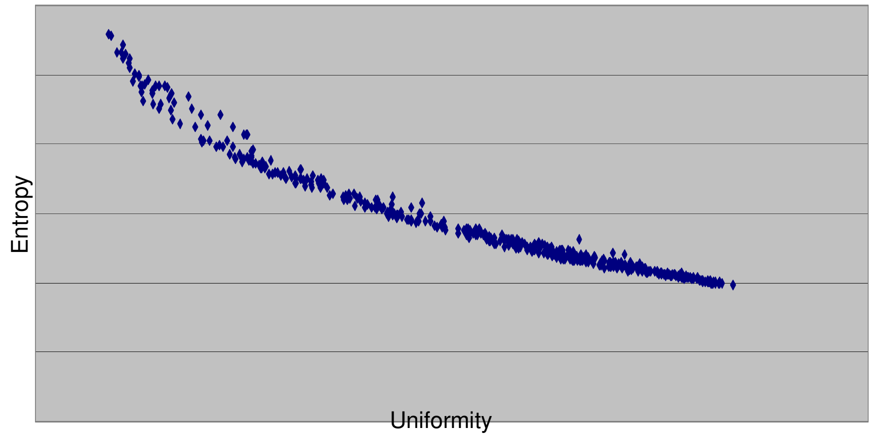



Figure 25: These two *features*, uniformity and entropy are highly correlated, so we can remove one of them to reduce repetition.

Since we are looking for features that group objects of similar types together, while still keeping objects of very different types apart, we are searching for any features that group the different types of trees together, but separate them from the steel poles. Also, we hope to find features that group the trees next to the wooden wall, since they are made out of the same material and thus share thermal properties. Using Minitab, we were able to find some of these features and plot them. In Figure 26, we are plotting the standard deviation of $L_o$ and $L_r$ vs. the average contrast vs. $L_o$-$L_b$. These features plot the data points almost as vectors stemming out from a common origin. The closer that we are to the 'origin', the worse our probability of correct classification currently appears to be. If our data point is further from the origin, however, the probability that we can classify the objects increases since the object classes are further from one another.

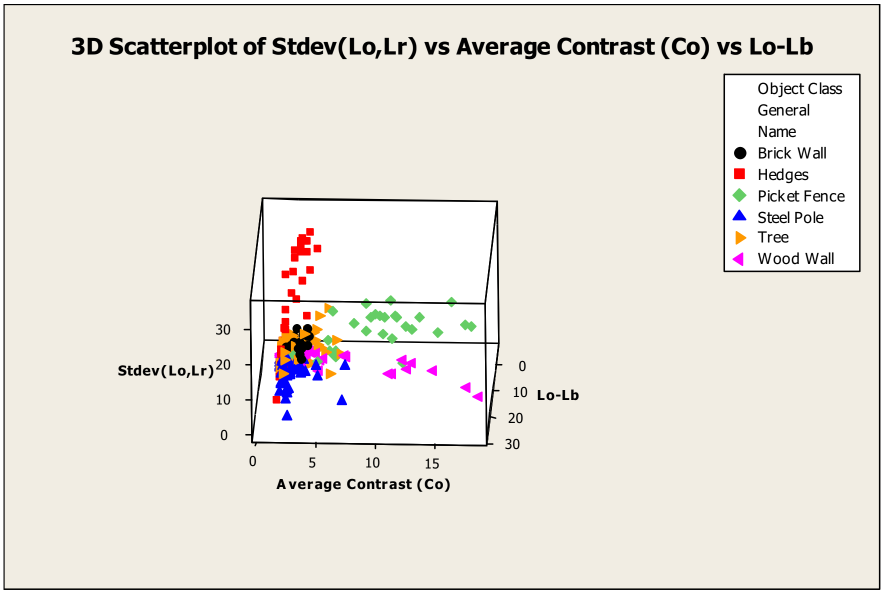



Figure 26: This plot shows how the different object types are separated by some of the *features*.

In Figure 26, we can see that the trees and the wood wall are closely grouped together, while the picket fence and the hedges are clearly separate. While the objects are still somewhat clustered around the origin, we can identify some promising patterns. In Figure 27, the objects are nicely separated with the steel poles on one side, and the trees, hedges, and wooden walls on the other. This plot uses uniformity, mean intensity, and the standard deviation of $L_o$ and $L_{r,}$ all defined above, to create the graph. This separation is again very promising for future research. While we have removed the outliers already in this plot, we believe that the clustering around an origin may be due to images collected after dark or before sunrise, in which the objects do not seem to radiate much at all. This is something that we will need to examine more carefully in the future.

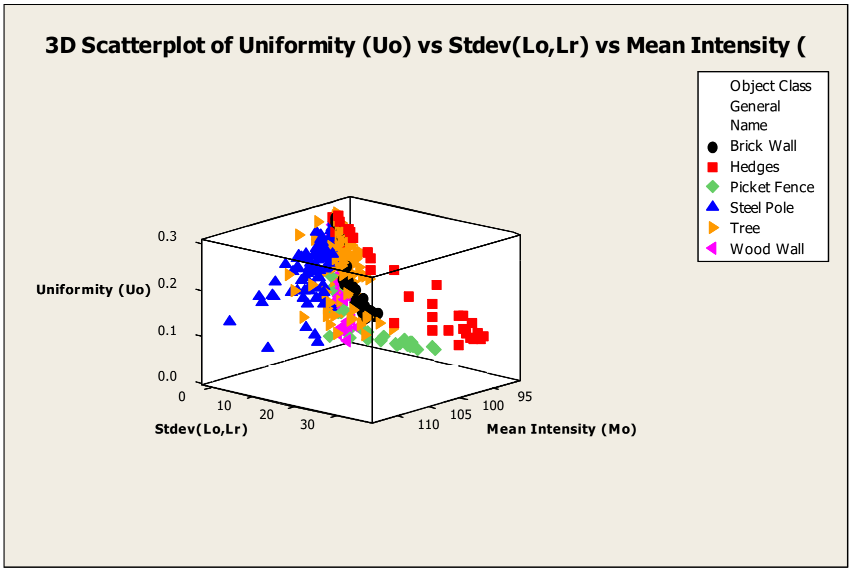



Figure 27: This plot very clearly separates the object types on either side of the origin.

It now seems that the texture features, such as entropy and standard deviation may in fact be more useful for our classification than we had hoped that emissivity would be. As shown by the plots above, entropy, or the randomness of an image, as well as the difference between the object's radiance and that of the background, or $L_o$-$L_b$, seem to be some of the strongest features thus far. Certainly we may have missed many stronger features in our analysis, or we may have simply not discovered them as of yet, but our results at this time look promising.

**Conclusions:**

Our analysis thus far leads us to believe that our hypothesis that we could use thermal imaging to classify objects is correct. We believe that our robot will be able to use the data collected about certain objects to come up with a high probability that the object is of a certain type. Our experiment, however, certainly has its limitations. First of all, our hypothesis does not seem to work very well for images collected before sunrise. Also, while the features that we have chosen separate the objects very well at a distance from the origin, they are still tightly clustered closer to this source, making it very hard to distinguish these objects.

In future research, we plan on analyzing the outlier data points more in order to attempt to find a way in which we can minimize the number of outliers that we collect. If we discover that the images collected before sunset do not give us very useful data points, we may need to use other sensors to collect the data at those times. We are considering such sensors as Ultrasound Sonar [13], which would help us to discover the shape and texture of the objects. Also future plans include an upgrade of rMary to allow us to collect data while the robot is in motion. This data would help us explore the limitations and possibilities of real-time classification decisions that the robot would need to conduct.

Our hypothesis about using emissivity as a feature does not seem to work as well as some of the other features that we are using. We think that this may be due to a limitation to the equation that we are using to calculate the emissivities, since it relies on the background and object radiance, which are very hard to standardize. It seems that the segment size of the aluminum foil used to calculate the background radiance affects the feature value of the emissivity. We plan to explore this observation more by coming up with a standard size for the segmentation.

Certainly our observations thus far lead us to believe that someday soon there will exist an artificially intelligent autonomous machine that will be able to make observations and decisions about its surroundings. While this possibility may sound farfetched and scary to some, we believe that this technology could greatly improve the lives of many. Being able to create a machine that could perform tasks deemed too dangerous or perhaps too unpleasant for mankind could help open up new opportunities for many. We believe that we have proven through this experiment and many others like it, that such a ‘sci-fi’ world is just around the corner.

## Appendix I:

```
function[t] = statxture(f,scale)
%STATXTURE Computes statistical measures of texture in an image.
%T = STATXTURE(F,SCALE) computes six  measures of texture from an
%image (region) F.  Parameter SCALE is a 6-dim row vector whose
%elements multiply the 6 corresponding elements of T for scaling
%purposes.  If SCALE is not provided it defaults to all 1s.  The
% output T is 6-by-1 vector with the following elements:
%   T(1) = Average gray level
%   T(2) = Average contrast
%   T(3) = Measure of smoothness
%   T(4) = Third moment
%   T(5) = Measure of uniformity
%   T(6) = Entropy
%RGB = imread(im);
%f = rgb2gray(RGB);
if nargin == 1
    scale(1:6) = 1;
else % Make sure it's a row vector.
    scale = scale(:)';
end
%Obtain histogram and normalize it.
p = imhist(f);
p = p./numel(f);
L = length(p);
%Compute the three moments.  We need the unnormalized ones
% from function statmoments.  These are in vector mu.
[v,mu] = statmoments(p,3);
%Compute the six texture measures:
%Average gray level.
t(1) = mu(1);
%Standard deviation.
t(2)= mu(2).^0.5;
%Smoothness.
% First normalize the variance to [0 1] by
%dividing it by (L-1)^2.
varn = mu(2)/(L-1)^2;
t(3) = 1-1/(1+varn);
%Third moment (normalized by (L-1)^2 also).
t(4) = mu(3)/(L-1)^2;
%Uniformity.
t(5) = sum(p.^2);
%Entropy.
t(6) = -sum(p.*(log2(p+eps)));
%Scale the values.
t= t.*scale;
```

## Appendix II:

```
function [FM, IF] = irfeatgenerator
%irfeatgenerator: Generates the classfication features from an object's
% infrared image.
%
%Outputs: (1) Feature Matrix (FM) with objects along rows and feature
values
%along columns.  Feature values consist of Object Class Label (L),
Current
%Ambient Temperature (deg F) from probe on rMary (Ta), First Order
Backward
%Difference Quotient of ambient temperature (deg F) based on KECK data
(T1),
%Estimated Emissivity of Object (Eo), and Entropy of Object (En).
%(2) Image File name (IF) of objects in column vector format.
%
% Date last modified: 9 April 2007
%
% [Symbols:]
% N = Number of objects in given class.
% La = Label of Object Class.
% Ta = Ambient Temperature (deg F) from probe on rMary.  Temperature
probe
%       is mobile source to consider local ambient temperature
measurements.
% T1 = First order backward difference quotient of ambient Temperature
%       (deg F) based on KECK data.  KECK weather station is a fixed
source
%       for temperature measurements to consider global rates of change.
% Er = Emissivity of reference emitter (Scotch 700 black electrical
tape).
% im = input RGB thermal image.
% al = imput RGB aluminum foil image.
% Lo = Mean radiance of segment of object.
% Lr = Mean radiance of segment of reference emitter (electrical tape).
% Lb = Mean radiance of segment of aluminum foil (background).
% Eo = Estimated emissivity of object (center segment of cylinders and
%       largest possible segment of flat objects w/o foreground
included).
% Texture Features: The following texture features are computed from
center
%       segment of cylinders and largest possible portion of scene
%       containing flat object w/ foreground included (minus electrical
tape)).
% Mo = Mean gray level of object scene.
% Co = Average Contrast (standard deviation) of gray levels of object
scene.
% So = Smoothness of object scene.
% To = Third moment (normalized by (L-1)^2 also) of object scene.
% Uo = Uniformity of object scene.
% En = Entropy of object scene
% FM = Feature Matrix with objects along rows and feature values along
%       columns.
% IF = Image file names associated with each object in the FM in column
%       vector form.
```

```
%
%[Enter number of objects for given session.]
N = input('Input number of objects for given session:   ')
%
for j = 1:N;
%
%[Enter object class label.]
% 1 = Picket Fence
% 2 = Hedges
% 3 = Brown Steel Pole
% 4 = Brick Wall
% 5 = Birch Tree
% 6 = Basswood Tree
% 7 = Wood Wall
% 8 = Octagon Steel Pole
% 9 = Cedar Tree
% 10 = Green Steel Pole
%
La = menu('Choose object', 'Picket Fence', 'Hedges', 'Brown Steel
Pole', 'Brick Wall', 'Birch Tree', 'Basswood Tree', 'Wood Wall',
'Octagon Steel Pole', 'Cedar Tree', 'Green Steel Pole')
%
%
%[Enter temperature feature values.]
% Ambient temperature (deg F) from probe on robot.
Ta = input('Input ambient temperature (deg F) from probe on rMary:   ')
% First backward difference quotient of ambient temp (deg F) based on
KECK data.
Tk1 = input('Input KECK temperature (deg F) at current time (4 sig
figs):   ')
Tk2 = input('Input KECK temperature (deg F) at current - 30 minute time
(4 sig figs):   ')
T1 = (Tk1 - Tk2)/30;
%
%[Generate image feature values from manually selected crop segments.]
%
%[Emissivty: Estimated emissivity of object (center segment of
cylinders and
%largest possible segment of flat objects w/o foreground included)]
Er = 0.97;
%Enter thermal images, crop and compute mean radiance values.
im = input('Enter thermal image file name as xxx.bmp:   ', 's')
RGB1 = imread(im);
im1 = rgb2gray(RGB1);
'Crop object for Emissivity -- Cylinder Center / w/o Foreground (press
ENTER to continue)', pause
Ico = imcrop(im1);
imshow(Ico);
Lo = mean2(Ico);
%
'Crop electrical tape for Emissivity (press ENTER to continue)', pause
Icr = imcrop(im1);
imshow(Icr);
Lr = mean2(Icr);
%
al = input('Enter aluminum foil image file name as xxx.bmp:   ', 's')
RGB2 = imread(al);
```

```
al1 = rgb2gray(RGB2);
'Crop aluminum foil for Emissivity (press ENTER to continue)', pause
Ica = imcrop(al1);
imshow(Ica);
Lb = mean2(Ica);
%
%Compute estimated emissivity of object.
Eo = ((Lo - Lb)/(Lr - Lb)) * Er;
%
%[Texture Features: Texture features of object (center segment of
cylinders and largest
%possible portion of scene containing flat object w/ foreground
included
%(minus electrical tape))]
'Crop object scene for Texture Features -- Cylinder Center / w/
Foreground (press ENTER to continue)', pause
Ics = imcrop(im1);
imshow(Ics);
%
%Obtain histogram and normalize it.
p = imhist(Ics);
p = p./numel(Ics);
L = length(p);
%Compute the three moments.  We need the unnormalized ones
% from function statmoments.  These are in vector mu from Gonzalez's
% statmoments.m code.
[v,mu] = statmoments(p,3);
%Compute the six texture measures:
%Mean gray level of object.
Mo = mu(1);
%Average Contrast (standard deviation) of gray levels of object.
Co = mu(2).^0.5;
%Smoothness of object.
% First normalize the variance to [0 1] by
%dividing it by (L-1)^2.
varn = mu(2)/(L-1)^2;
So = 1-1/(1+varn);
%Third moment (normalized by (L-1)^2 also) of object.
To = mu(3)/(L-1)^2;
%Uniformity of object.
Uo = sum(p.^2);
%Entropy of object.
En = -sum(p.*(log2(p+eps)));
%
%[Feature Matrix]
FM(j,:) = [La Ta T1 Lo Lr Lb Eo Mo Co So To Uo En];
IF{j,1} = im;
end
```

**Cited Bibliography**:


1. Griffith, Niall. “Lecture 10: Class Notes for Artificial Intelligence CS4816” http://www1.csis.ul.ie/curstudents/modules/2ndsem/cs4816/cs4816_lect10_expert1.doc University of Limerick, Ireland

2. Griffith, Niall

3. Holst, Gerald C. ‘Common Sense Approach to Thermal Imaging’, JCD Publishing, Winter Park, FL,(2000).

4. Holst, Gerald C.

5. Madding, Robert P. ‘Emissivity measurement and temperature correction accuracy considerations’, SPIE Conference on Thermosense XXI, April 1999.

6. Holst, Gerald C.

7. Laverty, Dennis. ‘An Introduction to Emissivity’ http://www.openxtra.co.uk/articles/emissivity_intro.php

8. Madding, Robert P.

9. KeckWeather: Environmental field laboratory at the College of William and Mary http://www.wm.edu/environment/Watershed/Weather/KeckWeather.htm

10. Madding, Robert P.

11. Gonzalez, R.C., Woods, R.E., and Eddins, S.L.‘Digital Image Processing Using Matlab’, Pearson Prentice Hall, London,(2004).

12. Gonzalez et al

13. M. Hinders, W. Gao and W.Fehlman, Sonar Backscatter Interpretation for Outdoor Mobile Robot Navigation, invited chapter in "Mobile Robots, Perception & Navigation” Sascha Kolski, ed. Advanced Robotic Systems, Mammendorf, Germany, (2007).

**Images Bibliography**:

1. Griffith, Niall. 'Lecture 10: Class Notes for Artificial Intelligence CS4816' http://www1.csis.ul.ie/curstudents/modules/2ndsem/cs4816/cs4816_lect10_expert1.doc University of Limerick, Ireland

2. Holst, Gerald C. 'Common Sense Approach to Thermal Imaging', JCD Publishing, Winter Park, FL,(2000).