Abstract

In this paper I assert a relatively efficient and effective means of segmenting man-made objects in digital images. The algorithm relies on determining the relative uniformity of the regions in a given image and assumes that man-made or artificial images will be more uniform than natural patterns in the background of the image, such as grass. Once these areas are found, they are tested for straight, parallel lines in order to reduce the number of false positive detections. Due to the process involved in the algorithm, the scope of images that can hope to be segmented by it is relatively limited. The system is most likely to work when there are not a great deal of artificial objects and the background of the image is either natural or non-uniform. A number of variations with respect to color spaces, resolution, and uniformity measurement are examined with varying degrees of efficacy.

1. Previous Work

Much work has been devoted to the recognition of objects in images within the field of computer vision. As a first step toward recognizing an object, it may be useful to narrow the search space of potential identities for the object by first determining whether it is artificial or natural. The problem of detecting artificial objects in aerial and satellite images has been thoroughly investigated [5], but doing so in the realm of images at large has not been covered as deeply. [3] identifies three broad categories that cover the most widely-researched techniques for object detection: model-based, example-based learning, and use of the patterns
in the image data, also known as feature-based detection or photometry.

When one attempts to detect objects using a model-based algorithm, he or she will give the system a set of shapes, a *model*, which defines the object that is to be detected. The algorithm then searches for instances of the model within the given image, and if a portion of the image is similar enough to the model, the object has been found. There have been attempts, as in [2], toward detecting artificial objects in this manner using models of basic shapes, such as squares and circles, with the assumption that man-made objects will tend to have these forms. Despite the promising results of this work, the processor-intensive operations of transforming and skewing the vector representations of these basic shapes, then searching an entire image for these variations appear to be prohibitive at the moment.

Example-based learning algorithms are those which utilize artificial intelligence techniques in order to learn what the desired object will look like by examining example images of the object, as in [3]. While it is perhaps conceivable that such a system could be developed that attempts to segment artificial objects in images, it may be that the magnitude of the number of artificial objects, which is certainly growing and amorphous, would be rather overwhelming as far as gathering and analyzing examples for training the system.

As these first two categories appear to be unsuitable for the problem at hand, at least for the moment, we are left with the final category for object detection: using the patterns in the image's data.

2. Introduction

*Uniformity*

Above, the search spaces for detecting man-made objects in images via model-based and
example-based systems were decided to be prohibitively large. Now we turn toward using the data from the image to directly decipher whether an artificial object is stored within it or not. The key insight in this type of detection is that the intensity levels for an unnatural object will tend to be more uniform than those where a natural object appears. [1] uses this insight and a two-dimensional gradient, or derivative, in order to determine which regions of the image contain the most uniform intensity levels. Much of the following work is based upon the experiment detailed in [1].

A similar approach, in [4], makes use of Zipf's law regarding the frequency of elements in a set of symbols. Here, an \( H \times H \) pattern is considered to be a symbol, and in artificial objects these symbols will not be nearly as randomly distributed as they are in a natural object, so we could potentially use this measure of uniformity in place of the gradient. This algorithm is mentioned because it is one example of another way a system might determine an image's uniformity. Additionally, it was implemented in the process of coming to the forthcoming algorithm, though I abandoned it because it was very slow – indeed, it took a significant amount of time to finish with even very low-resolution input – and the results were not terribly promising (Figures 1, 2). It is also worth noting that fractal patterns have been used in much the same way, though most commonly in aerial images, as in [6] and [7].

**HSV Color Space**

While the traditional RGB color space is useful and widely used, it is far more natural for humans to think of color in terms of hue, saturation, and value (HSV). In addition, this alternative representation of color is occasionally useful for segmenting images or determining the uniformity of a given region. For instance, when using a gradient such as the Sobel...
operator on an RGB image, the image is first converted to grayscale. This necessarily
eliminates some information which is potentially useful since the operator is now left with the
average intensities of red, green, and blue in the image, but it does not know how they relate.
[8] discusses at least a partial solution to this problem: a gradient in the HSV space.

Carron and Lambert introduce two forms of the gradient [8]. The simpler is essentially
an average of the Sobel operator used on both dimensions and all planes. The latter form does
the same, but it only counts the hue in proportion to the amount of saturation for the given
pixel since saturation is a measure of the purity of the hue. Both forms of the gradient deal
with the problem of representing the angular measure of hue in the linear form of values
between some maximum and minimum values. The problem with this is that, though the
traditional difference between a hue value near the maximum and one near the minimum will
be large, the two values are actually rather close in the angular representation of hue. In order
to provide a reasonable gradient, both forms take this into account and adjust it properly. The
result tends to be a very accurate gradient that is well suited for determining the uniformity of
regions in an image. This is one of the uniformity operators that my algorithm employs.

*Straight Lines*

It is not enough to simply find uniform regions in an image and call them artificial
objects since there are a few naturally occurring images, such as the sky and snow, that are
very uniform. There must be some way of determining whether, or how likely it is that, a
uniform region contains an artificial object. Here we use the Hough transform, which, given a
binary image representing the edges in an image, can locate straight lines in the image. With
this output we can then find out if there are any parallel or perpendicular lines, either of which
is a relatively certain indicator that the detected object is indeed artificial.

3. Algorithm

Uniformity

The first step in the algorithm is finding the uniform parts of the given image. We follow Caron's method, outlined in [1], for this. The image is given to some gradient function without a threshold. The result is a grayscale image with higher intensities in the areas with the most change – edges. We then attempt to get the 20% of the image which sees the least amount of change, and this is done in the following manner:

1. Find the histogram of the result from the gradient function,
2. Use this to find the probability distribution function,
3. Then find the cumulative distribution function (CDF), and
4. The CDF may be used in a straightforward manner to find the desired ratio of the pixels.

As mentioned in the introduction, there are numerous ways to find the gradient of an image. Our implementation gives the user the choice of Sobel in grayscale or HSV space (see Figures 8 and 9 for a comparison).

Finding the Regions of Interest

Now that we have a binary image with clumps in the least uniform areas, the algorithm must separate these into potential objects. Here are the steps the algorithm takes in doing so:

1. Label the connected components and find their bounding boxes, and
2. Merge bounding boxes that intersect into larger boxes.

The second step is included in order to combine separate components which may belong to the
same object. Now we know that if there is the possibility of anything interesting in a larger box, it may be present in any of the components whose boxes combined to make the larger box. If we did not make this step, there is the chance that a component of an object would be missed unnecessarily.

Parallel Check and Segmentation

At this stage the algorithm has the regions in which there may or may not be artificial objects. All that is known about them regarding their artificiality is that they scored in the 80\textsuperscript{th} percentile in some uniformity test. Now we need a way of weeding out those regions which likely do not contain objects of interest. As was mentioned in the introduction, this algorithm uses the Hough transform to aid in determining artificiality. Here is a more detailed account of the process:

\textit{For each region bounded by a box:}

1. Perform the Hough transform on a low-resolution image of the region. If the resulting lines contain either
   - Two or more perpendicular lines or
   - Two or more parallel lines
   Then we deem the region to contain an artificial object. If neither condition is met, go on to the next region.
2. Create a mask by adding all of the components associated with this region.
3. Now, find the output of the Hough transform on the normal-resolution region and add points to the mask using some of the lines close to the center of the region.
4. Use this mask in a color segmentation or region growing algorithm in order to segment the object from the image.

This marks the last step in the algorithm. As we will discuss, the main points of potential improvement lie in finding the uniformity of an image and finding the edges which determine the artificiality of a given region. With a proper implementation, it is not difficult to
4. Results

While this algorithm is fairly simple, it is effective within a limited scope. The first major potential point of failure is in the uniformity test. If something is naturally uniform, such as snow or sky, appears in the image, this technique will not necessarily mistake it for being man-made, but it will be distracted from other, potentially man-made, objects. For this reason, it was particularly useful that the HSV edge detection [8] was found. While this particular gradient does not eliminate the obstacle of naturally uniform objects, it does tend to do a better job of isolating uniform regions in general, thus making it easier to concentrate on deciding which regions plausibly contain an interesting object.

The artificiality test does tend to reduce the number of false positives without significantly reducing the number of correct detections. I found that using a resolution of about half the original resolution for a region would force stronger lines and thus effectively eliminate a number of regions which do not contain objects but might be mistaken to at their full resolution. Below this resolution, the algorithm would fail to find good lines in regions which do indeed contain objects of interest. Unfortunately, as is apparent in the table of results below, the test does not eliminate all regions which do not contain man-made objects, especially when the region contains the sky and some tree branches which enable it to form parallel or perpendicular lines between them (Figure 7).

Finally, the best segmentation method tends to depend on the particular image. However, I tended to prefer the output of the region-growing algorithm because it often produced a more cohesive segmentation of the object it found. See Figures 3 – 6.
Here are the results of running the algorithm on each image in the test database:

<table>
<thead>
<tr>
<th>Image</th>
<th>Objects Found</th>
<th>Non-Objects Found</th>
<th>Objects Not Found</th>
</tr>
</thead>
<tbody>
<tr>
<td>Man1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Man2</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Man3</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Man4</td>
<td>1, partial</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Man5</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Man6</td>
<td>0</td>
<td>1</td>
<td>2+</td>
</tr>
<tr>
<td>Man7</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Man8</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Man9</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Man10</td>
<td>1, partial</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**Conclusion**

It is worth noting that performing this type of analysis – where the only data which is used is contained in the image being analyzed – has natural limits. Surely, there is much more that can be done with it than has been done within this paper, but it seems that doing significantly better might require a new insight into the manner of artificial objects in digital images. Two key insights were exploited during this research: that man-made objects will tend to be more uniform than natural ones [1], and that man-made objects will often have parallel or perpendicular lines. The degree to which we can use these criteria is certainly limited, though, as we can only get so much better at testing either of them, and there are many interesting man-made objects that simply will not obey either criterion. For these reasons, I feel that model- and learning-based approaches to this problem will be more useful in the long-run. Perhaps one could pair models with the type of algorithm here described by searching for the models within the regions that have been flagged as being interesting rather
than searching the entire image.

5. References


6. Figures

Figures 1 and 2: Zipf's law pattern filtering on a low-resolution image.

Figures 3 and 4: A comparison of the Euclidean (3) and region growing (4) segmentations.
Figures 5 and 6: A comparison of Mahalanobis (5) and region growing (6) segmentations.
Figure 7: A false positive, though the portion of sky in the upper-left was eliminated as a potential object.

Figures 8 and 9: A comparison of the grayscale Sobel gradient (8) and the HSV gradient (9).