General-purpose localization of textured image regions

Ruth Rosenholtz*

Xerox Palo Alto Research Center, 3333 Coyote Hill Rd.
Palo Alto, CA 94304

ABSTRACT

In computer vision and image processing, we often perform different processing on “objects” than on “texture.” In order to do this, we must have a way of localizing textured regions of an image. For this purpose, we suggest a working definition of texture: Texture is a substance that is more compactly represented by its statistics than by specifying the configuration of its parts. Texture, by this definition, is stuff that seems to belong to the local statistics. Outliers, on the other hand, seem to deviate from the local statistics, and tend to draw our attention, or “pop out”[1, 2]. This definition suggests that to find texture we first extract certain basic features and compute their local statistics. Then we compute a measure of *saliency*, or degree to which each portion of the image seems to be an outlier to the local feature distribution, and label as texture the regions with low saliency. We present a method, based upon this idea, for labeling points in natural scenes as belonging to texture regions. This method is based upon recent psychophysics results on processing of texture and popout.

Keywords: texture, segmentation, attention, region of interest, saliency, computer vision, outlier, statistics

1. WHAT IS TEXTURE, AND WHY DO WE WANT TO FIND IT?

In a number of problems in computer vision and image processing, one must distinguish between image regions that correspond to objects and those which correspond to texture, and perform different processing depending upon the type of region. We do object recognition on objects, and texture classification on texture. We determine object shape and pose differently from shape from texture. We might compress textured image regions differently from non-textured. Current computer vision algorithms assume one magically knows this region labeling. But what is texture? We have the notion that texture involves a pattern that is somehow homogeneous, or in which signal changes are “too complex” to describe, so that aggregate properties must be used instead[3]. There is by no means a firm division between texture and objects[3]. The characterization often depends upon the scale of interest, so that a leaf is an object, but in the company of other leaves it becomes part of a leafy texture, which in turn is part of a tree object, in a forest texture, and so on.

Ideally the definition of texture should probably depend upon the application (e.g. texture might be that which can be coded with high fidelity using a particular texture synthesis algorithm). With a number of possible applications in mind, we investigate a definition that we believe will be of fairly general utility: Texture involves a substance that is more compactly described by its statistics than by the configuration of its parts. Texture, by this definition, is stuff that seems to belong to the local statistics. We propose extracting several texture features, at several different scales. We label as texture those regions whose feature values are likely to have come from the local distribution.

Outliers to the local statistics, on the other hand, tend to draw our attention[1, 2]. Items in a scene which have, for example, a significantly different color, motion, depth, or orientation from neighboring parts of the scene are often important ecologically, and should and do draw our attention. The unusual item is said to “pop out” at the observer, and the phenomenon is often referred to as “popout.” Thus, in the process of labeling as texture the regions that are homogeneous in the local statistics, we can simultaneously highlight salient regions that are outliers to the local statistics. The latter might be faults in a material (a crack in a wall, a hole in fabric), or possible locations to search for a particular object.

In Section 2, we briefly discuss human perception of popout. In Section 3, we discuss previous work in finding texture and regions of interest in an image. In Section 4, we describe our method. We present results on a number of real images in Section 5, and end, in the final section, with discussion.

[*] Email: rruth@parc.xerox.com.

2. HUMAN VISION AND POPOUT

See Wolfe[4] for a review of the visual search literature. Popout is typically studied using displays like that in Figure 1. The experimental subject searches for the unusual, *target* item, among the other, *distractor* items. One typically attempts to judge the "saliency," or degree to which the target pops out, by studying the efficiency of search for that item. The reasoning is that if an item draws our attention, search for that item should proceed more efficiently than search for an item that does not draw our attention. Typically popout is modeled by a relatively low-level operator, which operates independently on a number of basic features of the image, including orientation, contrast/color, depth, and motion. In this paper, we look only at the features of contrast and orientation.

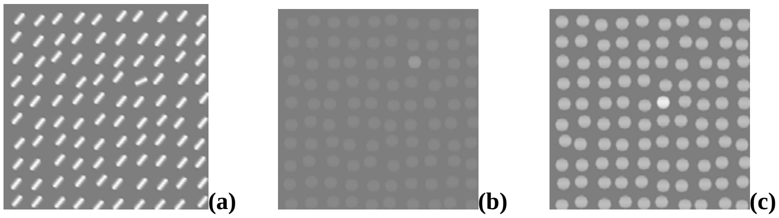

Figure 1: Three Popout Examples (Orientation and Contrast Features). (a) line at 64° among lines at 40°; (b) L=158 among L=138; (c) L=252 among L=192. (L=gray value.)

Both texture and popout processing require the computation of statistics. For this we need to know the *integration region* over which to collect the statistics. Evidence from human vision research has shown that in texture processing, the size of this region is independent of viewing distance and dependent only upon the support of the filters used to calculate the feature values[5].

Subjects can *distinguish* lines differing by only 1-2 degrees in orientation, but require a 20-40 degree difference for easy and efficient visual search. We account for this by adding uncertainty in the feature measurements, by using Parzen-window density estimation with a Gaussian window blurring the distribution. The window width may not be shift-invariant over the range of possible feature values. Figure 1 shows why: all three images have roughly the same degree of popout, yet the low contrast example requires less contrast difference than the high contrast example, implying that a wider Parzen window is required for higher-contrast distractors.

3. PREVIOUS WORK

Within the image-processing field, much of the work that has been done in finding texture has defined as texture any region with a high luminance variance, e.g. Vaisey & Gersho[6]. Unfortunately, the luminance variance in a region containing an edge can be as high as that in a textured region. Won & Park first use model fitting to detect image blocks containing an edge, and then label blocks with high variance as containing texture[7]. This requires only a weak homogeneity for texture, but it may be sufficient for the simple block transform based coding schemes that they study.

What little previous work there has been in the computer vision field in finding texture was generated by recent interest in image database search, where many systems do search based upon texture similarity, yet still require human intervention to mark texture regions as candidates for the comparison. Leung & Malik found regions of completely deterministic texture, but we do not want to require that textures be deterministic[8]. Other researchers have used the definition that if the luminance goes up and then down again (or vice versa) it's texture, and if it only goes up it's an edge[9]. However, in texture coding, one doesn't want to code lines the same way as texture, yet their method will label them the same. They also have no notion of similarity within a texture, and thus would mark a "fault" in a texture as belonging to that texture. This would be unacceptable for a texture synthesis application, in which a routine that tried to synthesize such a texture would most likely fail to reproduce the (highly visible) fault. Furthermore, they make no mention of processing scale. More recently, Shi and Malik have presented a method for segmenting images based upon texture and other features[10]. This method performs extremely well at the segmentation task, dividing an image into regions with internal similarity that is high compared to the similarity across regions. However, it is difficult to compare their results with algorithms designed to label texture regions, since they do not explicitly determine which of the resulting regions are texture and which are not. Furthermore, this method may also tend to mark a "fault" in a texture as belonging to that texture. This might happen both because it is biased against separating out small regions, and because the grouping of a patch with one region depends as much upon the difference between that patch and other regions as it does upon the similarity between the patch and the given region.

There has been very little computer vision work done on attentional cues. Ruggero Milanese et al found salient image regions using both top-down information and a bottom-up "conspicuity" operator, which marks a local region as more salient the greater the difference between a local feature value and the mean feature value in the surrounding region[11]. However, researchers have shown that, for the same difference in means, a local region is less salient when there is a greater variance in the feature values in the surrounding region[12, 1]. We use as our saliency measure a test for outliers to the local distribution. This captures, in many cases, the dependence of saliency on difference between a given feature value and the local mean relative to the local standard deviation. We will discuss our saliency measure in greater detail in the following section.

4. FINDING TEXTURE AND REGIONS OF INTEREST

We compute multiresolution feature maps for orientation and contrast, and then look for outliers in the local orientation and contrast statistics. We do this by first creating a Gaussian pyramid representation of the image, and filtering with both oriented and unoriented filterbanks, respectively. For the examples in this paper, we used a 3-level pyramid. To extract contrast, we filter the pyramid with a difference of circularly symmetric Gaussians. This can be thought of as a true contrast measure rather than merely a luminance difference if one imagines dividing all of the filter outputs by the mean luminance of the image.

The response of these filters will oscillate, even in a region with constant-contrast texture (e.g. a sinewave pattern). We approximate a computation of the maximum response of these filters over a small region by first squaring the filter responses, and then filtering the contrast energy with an appropriate Gaussian. Finally, we threshold the contrast to eliminate low-contrast regions ("flat" texture). These thresholds (one for each scale) were set by examining the visibility of sinewave patterns with various spatial frequencies and contrasts, and the same thresholds were used throughout our examples.

We compute orientation in a simple and biologically plausible way, using Bergen & Landy's "back pocket model" for low-level computations, as shown in Figure 2[13]:

1) Filter the image with horizontal, vertical, and ±45° oriented Gaussian second derivative filters at a number of different scales.

2) Compute opponent energy by squaring the filter outputs, pooling them over a region 4 times the scale of the second derivative filters, and subtracting the vertical from the horizontal response and the +45° from the -45° response.

3) Normalize responses at each scale by the total energy in the 4 orientation energy bands at that scale.

The result is two images at each scale of the pyramid. To a good approximation, in regions which are strongly oriented, these images represent $k\cos(2\theta)$ and $k\sin(2\theta)$, where θ is the local orientation at that scale, and k is a value between 0 and 1 which is related to the local orientation specificity. From this we solve for the local orientation and orientation specificity.

One should threshold out, and ignore, orientation values from points with low specificity, as they tend to be very noisy. This threshold has to do with reliability of the orientation estimates rather than with visibility of patterns. In images of white noise, 80% of the estimates of k fall below 0.5, therefore we are 80% confident that an orientedness value of $k>0.5$ did not occur due to chance -- i.e. that the pattern is actually oriented at that point -- and take this value as our threshold.

We compute a non-parametric measure of saliency by first estimating the local feature distribution. For each feature and each scale, we find a histogram of feature values over a local integration region. Work on human vision has suggested that a disk of radius roughly $5S$ is a reasonable approximation of the integration region in the human visual system, where S is the support of the Gaussian second derivative filters[5, 14].

We use the method of Parzen windows to estimate the local feature distribution, with a Gaussian window. The Parzen window width was chosen based upon popout examples such as those in Figure 1. We chose the Parzen window such that our algorithm correctly declared that the "target" elements in those examples would pop out (see below).

Bottom-up attentional cues tend to be outliers to the local distribution, while textured regions tend to consist of features that belong to the local distribution. To assess the extent to which a feature is an outlier to the local distribution (and thus seems not to belong to the local distribution), we next compute a measure of saliency.

The distribution estimates give us an estimate of the likelihood that a given feature value, v, was observed, given the local distribution of feature values. We compute as our saliency measure:

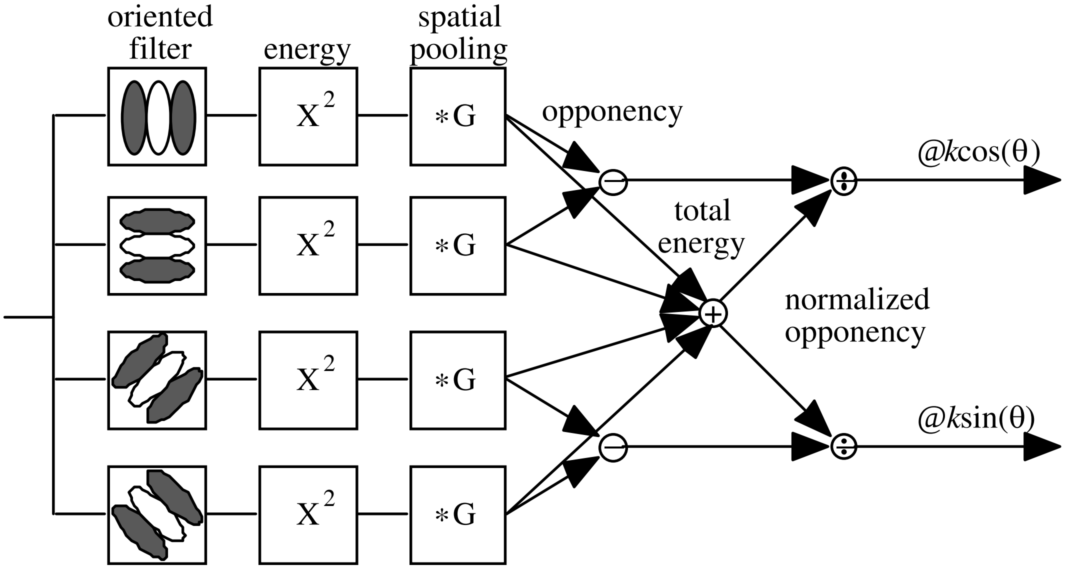

Figure 2: Filterbank Computation of Orientation and Orientedness.

$$\text{saliency} = -\log\left(\frac{P(v \mid D)}{\max_x P(x \mid D)}\right) \tag{1}$$

where *D* is the estimated local distribution. Scaling by the peak of the distribution ensures that the saliency of the most-likely feature value is 0. To aid one's intuition about this saliency measure, note that if *D* is normally distributed, $N(\mu,\sigma^2)$, this simplifies to

$$\frac{(x-\mu)^2}{2\sigma^2} \tag{2}$$

which should be compared to the standard parametric test for outliers, which uses the measure $(x-\mu)/\sigma$. Our saliency measure is essentially a more general, non-parametric form of this measure (i.e. it does not assume a Gaussian distribution), but for a squaring and a factor of 2. We have also implemented a parametric test for outliers, and there is little difference in the results for many of the images we have tested.

Points with saliency less than 0.5 are labeled as candidate texture points. To understand this threshold, for a Gaussian distribution this would correspond to points within one standard deviation of the mean. Points with saliency greater than 3.1 are labeled as candidates for bottom-up attentional cues. For a Gaussian distribution this would correspond to points farther than 2.5 standard deviations from the mean, a standard parametric test for outliers. Both the texture images and the region of interest images are median-filtered to remove extraneous points. One could, of course, keep the raw saliency values, as a measure of the likelihood that a region contained texture, rather than setting a hard threshold. We use a hard threshold in our examples to better display the results.

5. EXPERIMENTAL RESULTS

Figure 3 shows several example images. Figure 4 shows texture found at the finest scale of processing. The striped pattern in the images in the top row indicates oriented texture identified by the algorithm. The checkered pattern in the images in the bottom row indicates texture with homogeneous contrast. Figures 5 and 6 are similar to Figure 4, but show texture at the medium and coarse scales. The absence of an image in any of these figures means that no texture of the given type was found in that image at the given scale. Note that we perform no segmentation of one texture from another.

We see that for the building image, the bricks were identified as texture at the fine scale, while at a coarser scale the algorithm thought that the edges making up the windows and shutters constituted texture. The leopard skin was correctly labeled as texture, as well as the low frequency stripes in the lower right corner of the leopard image. In the desk image, the fake wood texture was correctly identified. In the hotel image, parts of the building, with its regular pattern of windows, were marked. In the house image, we see oriented texture on the wood siding of the house, and contrast-based texture on the siding, trees, and part of the grass (much of the grass was labeled as too low contrast to be anything but "flat" texture). One of the bushes is correctly identified as having coarser texture than the other has. Finally, in the lighthouse image, the house (but not its window) and part of the fence are oriented, relatively high-frequency texture, the tower has high-frequency unoriented texture, and the clouds have some low frequency oriented texture.

Figure 7 shows the regions of interest that were found (the striped and plaid patterns here have no meaning but were chosen for maximum visibility). Most complex natural scenes had few interesting low-level attentional areas. This suggests that what "pops out" in natural scenes may not be well predicted by the purely low-level features that describe popout in the typical simplistic psychophysical stimuli used to study visual attention. In the lighthouse image, the life preserver in the lower left is marked. In the hotel, curved or unusual angular windows are identified as attentional cues, as well as the top of the building. Both of these results are in agreement with psychophysical results showing that observers quickly identify curved or bent lines among straight lines (reviewed in Wolfe[4]). The simpler desk scene yields much nicer results, with each of the 3 objects labeled, as well as the phone cord.

6. DISCUSSION

We have suggested that bottom-up attentional cues are outliers to the local distribution of features, and texture is the absence of such outliers. We presented a method for finding outliers to contrast and orientation distributions, and results both on localizing texture and on finding popout in natural images. For the simple desk image, the algorithm highlights salient regions that correspond to our notions of the important objects in the scene. On complicated natural scenes, its results are less intuitive; suggesting that search in natural scenes makes use of higher-level processing such as grouping into objects. This result should not be terribly surprising, but serves as a useful check on simple low-level models of visual attention. The algorithm does a good job of identifying textured regions at a number of different scales.

ACKNOWLEDGMENTS

This work was partially supported by an NRC postdoctoral award at NASA Ames. Many thanks to David Marimont and Eric Saund for useful discussions.

REFERENCES

1. R. Rosenholtz, "Basic signal detection theory model does not explain search among heterogeneous distractors." *Invest. Ophthal. and Vis. Sci. (Suppl.)* **38**, 4, p. 687, 1997.
2. R. Rosenholtz, "A simple saliency model explains a number of motion popout phenomena." *Invest. Ophthal. and Vis. Sci. (Suppl.)* **39**, 4, p. 629, 1998.
3. E. Saund, "Scale and the Shape/Texture Continuum," Xerox Internal Technical Memorandum, 1998.
4. J. M. Wolfe, "Visual search: a review," *Attention*, H. Pashler (ed.), pp. 13-74, Psychology Press Ltd., Hove, East Sussex, UK, 1998.
5. F. A. A. Kingdom and D. Keeble, "The mechanism for scale invariance in orientation-defined textures." *Investigative Ophthalmology and Visual Science (Suppl.)* **38**, 4, p. 636, 1997.
6. J. Vaisey and A. Gersho, "Image compression with variable block size segmentation." *IEEE Trans. Signal Processing* **40**, 8, pp. 2040-2060, 1992.
7. C. S. Won and D. K. Park, "Image block classification and variable block size segmentation using a model-fitting criterion," *Opt. Eng.* **36**, 8, pp. 2204-2209, 1997.
8. T. K. Leung and J. Malik, "Detecting, localizing, and grouping repeated scene elements from an image," *Proc. 4th European Conference On Computer Vision*, **1064**, 1, pp. 546-555, Springer-Verlag, Cambridge, 1996.
9. D. Forsyth, J. Malik, M. Fleck, H. Greenspan, T. Leung, S. Belongie, C. Carson, and C. Bregler, "Finding pictures of objects in collections of images," *ECCV Workshop on Object Representation*, Cambridge, 1996.
10. J. Shi and J. Malik, "Self Inducing Relational Distance and its Application to Image Segmentation," *Proc. 5th European Conference on Computer Vision*, Burkhardt and Neumann (eds.), **1406**, 1, pp. 528-543, Springer, Freiburg, 1998.
11. R. Milanese, H. Wechsler, S. Gil, J. -M. Bost, and T. Pun, "Integration of bottom-up and top-down cues for visual attention using non-linear relaxation," *Proc. IEEE CVPR*, pp. 781-785, IEEE Computer Society Press, Seattle, 1993.
12. J. Duncan and G. Humphreys, "Visual search and stimulus similarity," *Psychological Review* **96**, pp. 433-458, 1989.

13. J. R. Bergen and M. S. Landy, “Computational modeling of visual texture segmentation,” *Computational Models of Visual Processing*, Landy and Movshon (eds.), pp. 252-271, MIT Press, Cambridge, MA, 1991.
14. F. A. A. Kingdom, D. Keeble, D., and B. Moulden, “Sensitivity to orientation modulation in micropattern-based textures,” *Vision Research* **35**, 1, pp. 79-91, 1995.

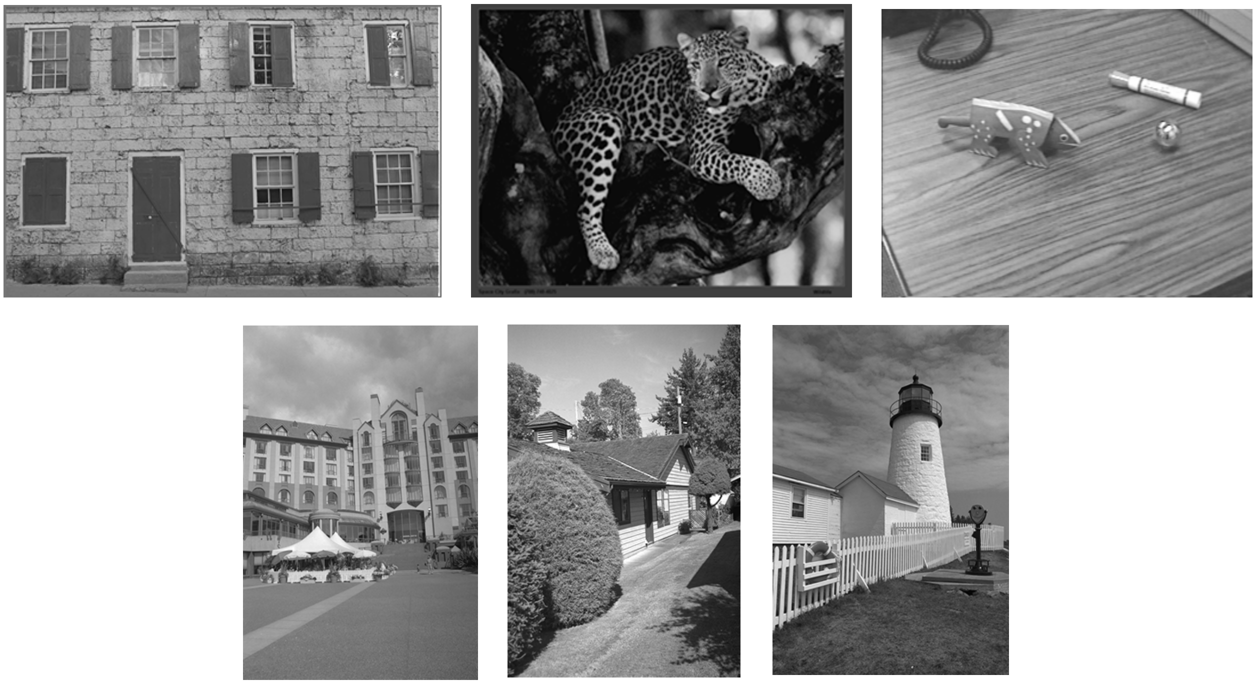

Figure 3: Original Images

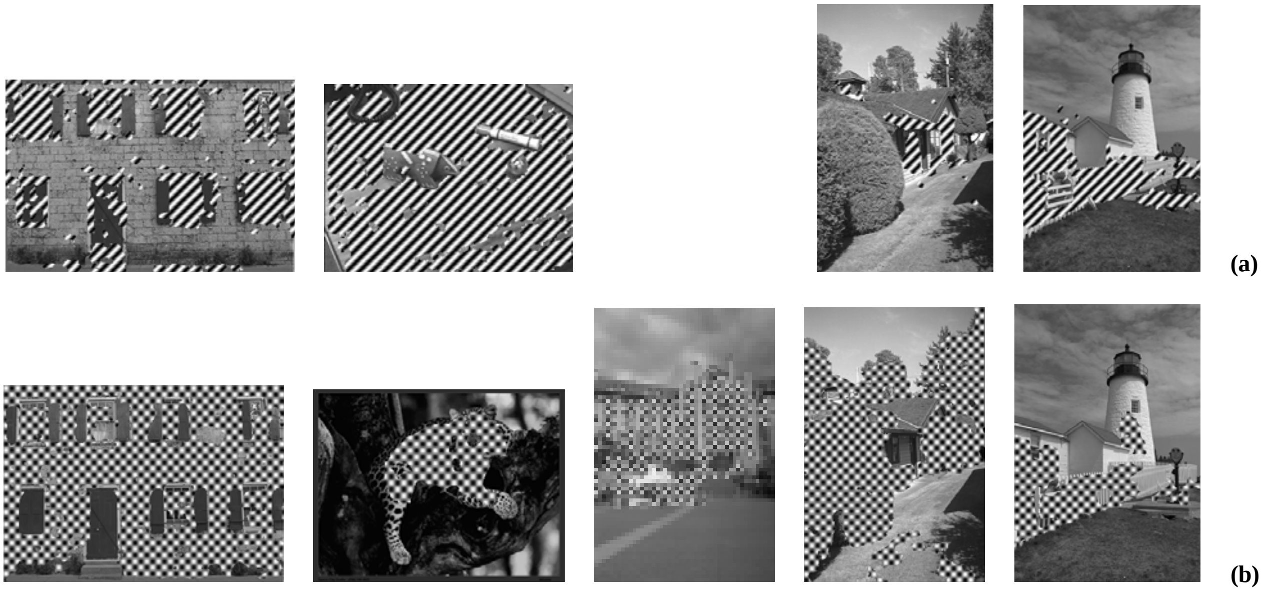

Figure 4: Fine-Scale Texture. (a) oriented texture, (b) homogeneous contrast texture.

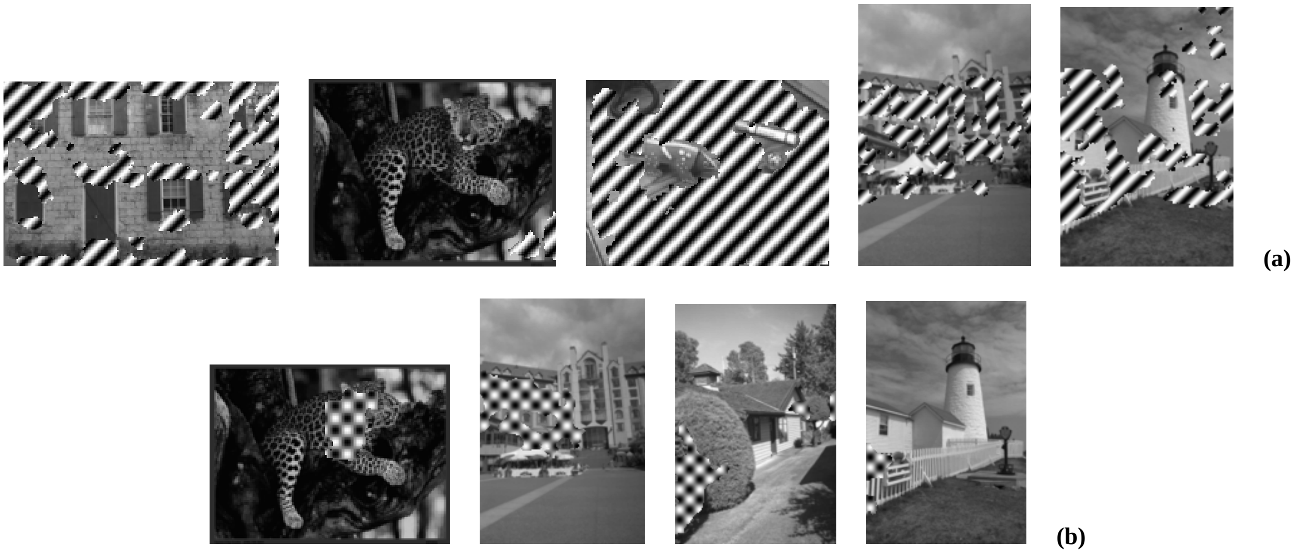

Figure 5: Medium-Scale Texture. (a) oriented texture, (b) homogeneous contrast texture.

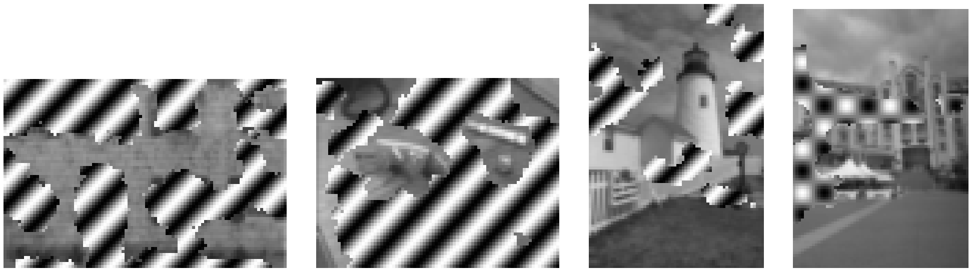

Figure 6: Coarse-Scale Texture. (a) oriented texture, (b) homogeneous contrast texture.

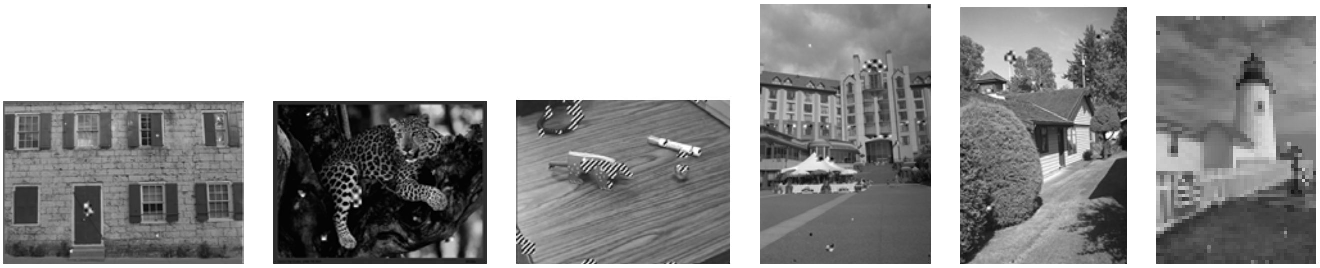

Figure 7: Regions of Interest