# Clustering Large Image Collections through Pixel Descriptors

Tuan Nhon Dang*
University of Illinois at Chicago

Leland Wilkinson†
Systat Inc.
University of Illinois at Chicago

## Abstract

We introduce a method to cluster large image collections. We first rescale and convert images into gray scales. We then threshold these scales to obtain black pixels and compute descriptors of the configurations of these black pixels. Finally, we cluster images based on their descriptors. In contrast to raster clustering, which uses the entire pixel raster for distance computations, our application, which uses a small set of descriptors, can handle large image collections within reasonable time.

**Index Terms:** I.5.2 [Pattern recognition]: Design Methodology—Pattern analysis

## 1 Introduction

This work is a natural extension of our work on *Scagnostics* [4]. Scagnostics allows us to characterize the "shape" of 2D scatterplots by operating on descriptors of point distributions. Our new image clustering procedure operates on distributions of pixels within images.

Our contributions in this poster are:

- We develop new pixel distribution descriptors for characterizing images.
- We design an interactive environment for visualizing clusters of images. In this environment, each image is attracted by similar images and repelled by dissimilar images. The dissimilarity measure for images is computed based on their descriptors.

## 2 Related Work

In the mid 1980s, John and Paul Tukey developed an exploratory graphical method to describe a collection of 2D scatterplots through a small number of measures of the pattern of points in these plots [2]. We implemented the original Tukey idea through nine Scagnostics defined on planar proximity graphs. Others used analogs of the word to describe feature-based descriptions for parallel coordinates and pixel displays[1, 3].

Although the original motivation for Scagnostics was to locate interesting scatterplots in a large scatterplot matrix, we soon realized the idea had more general implications. In this poster, we extend this work to handle pixels in images and develop new descriptors that are appropriate for images (as opposed to scatterplots).

We now outline our image algorithms.

### 2.1 Transforming images

We begin by rescaling images into 40 by 40 pixel arrays. The choice of rescaling size is constrained by efficiency (too many pixels slow down calculations) and sensitivity (too few pixels obscure features in the images). Then we gray-scale our 40 by 40 pixel images using different thresholds. Black pixels in the gray scale images constitute our data points.

### 2.2 Computing Descriptors

We compute our descriptors based on proximity graphs that are subsets of the Delaunay triangulation. In the formulas below, we use $H$ for the convex hull, $A$ for the alpha hull, and $T$ for the minimum spanning tree.

**Connected** The Connected descriptor is based on the proportion of the total edge length of the minimum spanning tree accounted for by the total length of edges connecting 2 adjacent black pixels (edges length 1).

$$c_{connected} = length(T_1)/length(T) \tag{1}$$

**Dense** Our Density descriptor compares the area of the alpha shape to the area of the whole frame. Low values of this statistic indicate a sparse image. This descriptor addresses the question of how fully the points fill the frame.

$$c_{dense} = area(A)/(40 \times 40) \tag{2}$$

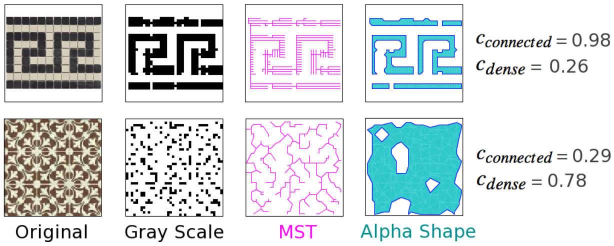

Figure 1: Top image shows high Connected and sparse distribution. Bottom image shows low Connected and dense distribution.

**Convex** Our convexity measure is based on the ratio of the area of the alpha hull and the area of the convex hull. This ratio will be 1 if the nonconvex hull and the convex hull have identical areas.

$$c_{convex} = area(A)/area(H) \tag{3}$$

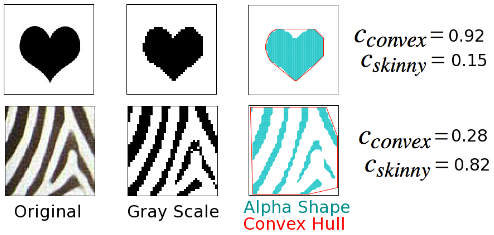

Figure 3: Top image shows high Convex and low Skinny distribution. Bottom image shows low Convex and high Skinny distribution.

**Skinny** The ratio of perimeter to area of a polygon measures, roughly, how skinny it is. We use a corrected and normalized ratio so that a circle yields a value of 0, a square yields 0.12 and a skinny polygon yields a value near one.

$$c_{skinny} = 1 - \sqrt{4\pi area(A)}/perimeter(A) \tag{4}$$

*e-mail: tdang@cs.uic.edu
†e-mail:leland.wilkinson@systat.com

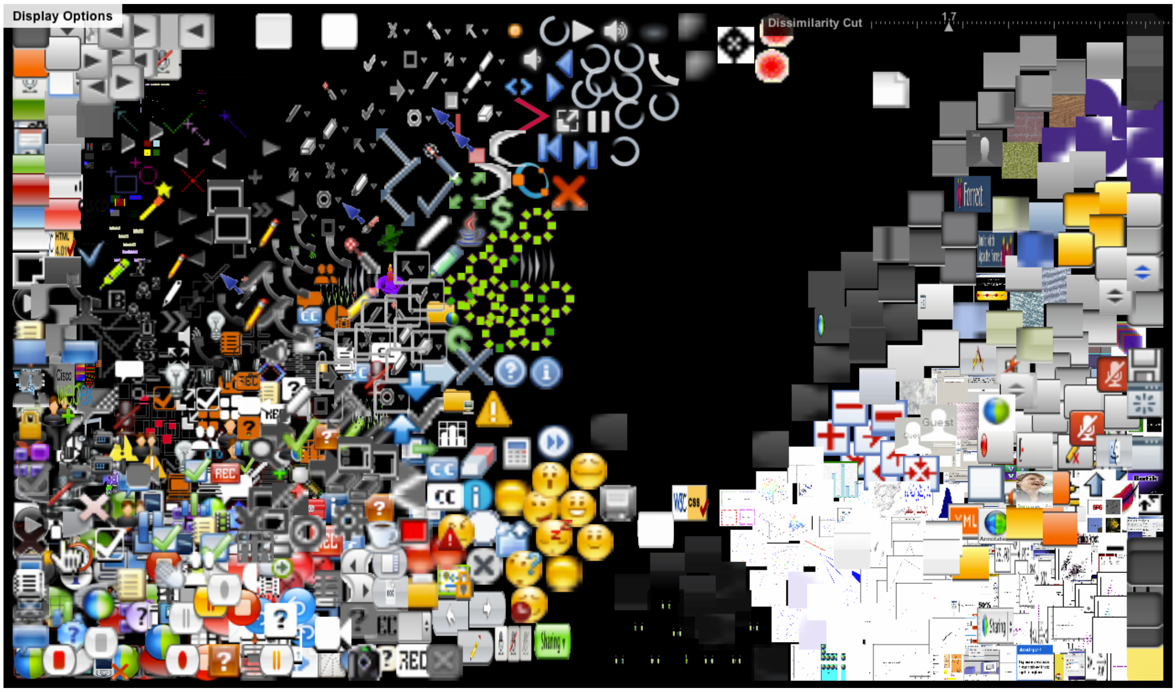

Figure 2: Clusters of 1000 png images on the second author's computer.

## 3 Application

After computing scagnostics of images, we put all images randomly in the output panel. In this environment, each image is attracted by similar images and repelled by dissimilar images. This force-directed clustering has quadratic complexity because it follows the same steps as other force-directed algorithms on complete graphs. Nevertheless, the procedure runs out of space before it runs out of time. That is, we can cluster in practical time (minutes) collections of thousands of images on a typical laptop screen. Clustering a larger corpus runs into display problems that could be ameliorated by pan-and-zoom techniques, although we have not developed these methods at this time. We anticipate additional methods for improving scalability in the future.

The dissimilarity of two images ($S$ and $P$) is computed based by the following equation:

$$Dissimilarity(S,P) = \sqrt{\sum_{i=1}^{4}(S_i - P_i)^2} \quad (5)$$

where $S$ and $P$ are two arrays of four Scagnostics of the two images.

Here is the summary of the algorithm to compute forces applied on images:

1. We get dissimilarity cut $C$ as a user input. We then define $A_{ij} = Dissimilarity(S_i, S_j) - C$.

2. We compute $\overrightarrow{U_{ij}}$ as the unit vector from $S_i$ to $S_j$.

3. If $A_{ij} \leq 0$, $\overrightarrow{F_{ij}}$ is the attraction between $S_i$ and $S_j$:

$$\overrightarrow{F_{ij}} = A_{ij} * \overrightarrow{U_{ij}} \quad (6)$$

4. If $A_{ij} > 0$, $\overrightarrow{F_{ij}}$ is the repulsion of $S_j$ on $S_i$:

$$\overrightarrow{F_{ij}} = \frac{A_{ij} * \overrightarrow{U_{ij}}}{Distance(S_i, S_j)} \quad (7)$$

5. The force applied on $S_i$ is the sum of forces by all images:

$$\overrightarrow{F_i} = \sum_{i=1}^{N} \overrightarrow{F_{ij}} \quad (8)$$

6. Repeat steps 2-5 for all images $S_i$.

Notice in Equation 6, the attraction between $S_i$ on $S_j$ does not depend on their distance. This assures that similar images can find each other no matter where they are in the display.

## 4 Conclusions

In this poster, we propose a novel image clustering technique. We cluster images based on a small set of image descriptors. We have tested this technique on images of several computers and the performance is satisfactory. We are planning to apply this technique in more dynamic environments such as clustering new images posted on Facebook/Twitter, or thumbnails of newly-posted videos on Youtube.

This technique guarantees the same images are in the same clusters. However, due to the fact that we simplify the images to gray levels and obtain shapes based on black pixels, the same image content using different color schemes may end up in different clusters.

## References

[1] A. Dasgupta and R. Kosara. Pargnostics: Screen-space metrics for parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 16:1017–2626, 2010.

[2] T. Hastie and W. Stuetzle. Principal curves. *Journal of the American Statistical Association*, 84:502–516, 1989.

[3] J. Schneidewind, M. Sips, and D. Keim. Pixnostics: Towards measuring the value of visualization. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 199–206, Baltimore, MD, 2006.

[4] L. Wilkinson, A. Anand, and R. Grossman. High-dimensional visual analytics: Interactive exploration guided by pairwise views of point distributions. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1363–1372, 2006.