Object Similarity Evaluation

M. Fisher[1], V. Agarwal[2] and T. Beri[2]

[1]Adobe Research, USA
[2]Adobe Inc., India

Gestalt rules of organization [DMM03, DMM07] highlight the importance of proximity, similarity, continuity, and connectedness for perceptual grouping and their conjoins [Pal77, NSX*11, LZH*17] have provided a formal way for machines to produce human-like clustering. [LZH*17] shows that shape, size, structure, location and context based descriptors are important to compute an accurate co-relation among graphical objects in images. We model vector graphic objects on same lines and compute their co-relation on these established principles.

We evaluate every pair of objects in a *containment group* on five similarity descriptors - *Color*, *Stroke*, *Size*, *Shape*, and *Spatial Placement*. One way to evaluate scores of these descriptors is to reduce these vector objects into rasters and use one of the several techniques from literature. For example, [SO95] presents a robust method to compute a similarity measure of colored images, [LL00] compares parts of an image to arrive at an aggregate shape similarity score, and [BE98] compares entire spatial scenes to produce a similarity estimate. However, we find that this causes a loss in precision and more accurate results can be produced by comparing objects in their native vector forms, where shape boundaries (and other characteristics) are precisely known and there is no need to make approximations in image domain. The rest of this document presents our evaluation methods for each of these descriptors. Note that we also normalize all descriptor scores to the range $[0-1]$ in order to compute a grand score, which is a weighted average of normalized scores.

1. **Color**: This descriptor refers to how the objects are filled. Vector graphics allow a wide variety of fills including flat-colors, gradients, patterns, and images. We first describe our treatment to flat-colors and then to the rest. A flat-color may be defined in any color space like CMYK, RGB, GrayScale, etc. To compare two flat colors, we first reduce them to the LAB [Luo14] color space. This color space, being device-independent, is more robust and suitable for color comparison. To compute difference between two LAB flat-colors, we use CIEDE2000 [Luo16] formula for $\triangle E^*_{ab}$ color difference metric. This metric is used to obtain color similarity score between two objects, o_i and o_j, at a time. The following formula shows the computation -

$$Score = 1 - \frac{\triangle E^*_{c_i c_j}}{\triangle E^*_{c_{min} c_{max}}}$$

where, c_i and c_j respectively refer to the LAB flat-colors of objects o_i and o_j. c_{min} and c_{max} are the extreme LAB colors $(0,-128,-128)$ and $(100,127,127)$, respectively. While this method can be used to compare two objects with flat-colors, objects with other types of fills are first scaled (with aspect ratio maintained) to a fixed size (say $100*100$) and then reduced to an array of flat-colors (with each flat-color taken from the corresponding pixel in the scaled object). Three histograms (one per color channel) are created for both objects and each is populated by counting the corresponding color channel values from the array of flat-colors. A similarity score is computed for each of the three pairs of histograms (pairing is done for same color channels). We set the final color similarity score equal to the average score across all histogram pairs. The following formula depicts the overall computation for objects o_i and o_j -

$$Score = \frac{1}{3} * \sum_{l=0}^{3} \frac{\sum_{k=c_{min}[l]}^{c_{max}[l]} \min(H[l]_{k_{o_i}}, H[l]_{k_{o_j}})}{\sum_{k=c_{min}[l]}^{c_{max}[l]} \max(H[l]_{k_{o_i}}, H[l]_{k_{o_j}})}$$

where, $H[l]_{k_{o_i}}$ refers to the value corresponding to entry k in the histogram of object o_i for color channel l. $c_{min}[l]$ and $c_{max}[l]$ are the minimum and maximum permitted color values for these channels, respectively.

2. **Stroke**: Stroke is defined by several parameters including *stroke-width*, *line-cap*, *line-join*, *dash-array* and *stroke-color*. We treat all of them equivalently and, as described below, an arithmetic average of similarity scores for all stroke parameters is used to obtain the final stroke similarity score -

   a. stroke-width (*SW*): $SW = 1 - \frac{|sw_{o_i} - sw_{o_j}|}{\max(sw_{o_i}, sw_{o_j})}$
   where, sw_{o_i} and sw_{o_j} represent the stroke-widths of objects o_i and o_j respectively.

   b. line-cap (*LC*): $LC = \begin{cases} 1, & \text{if } \mathrm{lc}_{o_i} = \mathrm{lc}_{o_j} \\ 0, & \text{otherwise} \end{cases}$
   where, lc_{o_i} and lc_{o_j} represent the *line-cap* styles (*butt cap*, *round cap*, etc.) of objects o_i and o_j respectively.

   c. line-join (*LJ*): $LJ = \begin{cases} 1, & \text{if } \mathrm{lj}_{o_i} = \mathrm{lj}_{o_j} \\ 0, & \text{otherwise} \end{cases}$
   where, lj_{o_i} and lj_{o_j} represent the *line-join* styles (*bevel join*, *miter join*, etc.) of objects o_i and o_j respectively.

   d. dash-array (*DA*): We find the lengths of dash-arrays of both objects o_i and o_j. Both arrays are traversed up to their minimum common length and the matching entries are counted. The count is divided by the greater of the two lengths. The following formula depicts the computation -

$$DA = \frac{\sum_{n=1}^{\min(\|da_{o_i}\|,\|da_{o_j}\|)} Z_n}{\max(\|da_{o_i}\|,\|da_{o_j}\|)}$$

$$Z_n = \begin{cases} 1, & \text{if } \mathrm{da}_{o_i}[\mathrm{n}] = \mathrm{da}_{o_j}[\mathrm{n}] \\ 0, & \text{otherwise} \end{cases}$$

where, da_{o_i} and da_{o_j} represent the *dash-arrays* of objects o_i and o_j respectively. n indexes into the arrays and $\|da_{o_i}\|$ & $\|da_{o_j}\|$ represent their lengths. Note that if none of the objects have a *dash-array*, DA is set to 1.

e. stroke-color (SC): Similarity score of stroke color of two objects is computed in the same way as object color (see 1.).

Overall, stroke similarity score is given by -

$$Score = \frac{SW + LC + LJ + DA + SC}{5}$$

3. **Size**: Size is the area of an object. It can be computed by finding area under the vector curves or by subdividing vector object into primitives like triangles. The object area can then be computed as sum total of the area of these triangles. The size similarity score of objects o_i and o_j (with areas A_{o_i} and A_{o_j}) is given by -

$$Score = 1 - \frac{|A_{o_i} - A_{o_j}|}{\max(A_{o_i}, A_{o_j})}$$

where, numerator is the absolute difference and denominator is the maximum of the two input areas. For non-vector objects like images, their area is same as that of their rectangular bounding box and size similarity score is computed like vector objects.

4. **Shape**: We reduce every vector object to a histogram representation and find shape similarity score among two objects by finding the degree of similarity in their respective histograms. Each histogram records peripheral distances to Bezier curves (from center of the object) and thus is a fingerprint of the vector object it represents. To build it, we scale the vector object to a fixed size $s * s$ (maintaining aspect ratio) and its Bezier paths are linearized (i.e. approximated to small line segments). From the center $(\frac{s}{2}, \frac{s}{2})$ of the object, cartesian distance (rounded off to the nearest whole number) is computed to every path point found by linearization. The histogram records the number of occurrences of every distance (from center to linearized point) in the object. The minimum distance possible in this setup is 0, while the maximum is $\frac{s}{\sqrt{2}}$. To compare two objects, we scan the $1 + \frac{s}{\sqrt{2}}$ entries in their respective histograms and find the minimum and maximum distance count recorded in each entry. The shape similarity score between objects o_i and o_j is computed as -

$$Score = \frac{\sum_{k=0}^{\frac{s}{\sqrt{2}}} \min(H_{k_{o_i}}, H_{k_{o_j}})}{\sum_{n=0}^{\frac{s}{\sqrt{2}}} \max(H_{k_{o_i}}, H_{k_{o_j}})}$$

where, $H_{k_{o_i}}$ refers to the value corresponding to distance k in the histogram of object o_i.

5. **Spatial placement**: This refers to the shortest distance between two objects. Since vector objects are made up of several cubic Bezier curves, geometrically computing shortest distance between them is computationally heavy. Thus, we do an approximation. We find the centers of the two vector shapes and compute the center (C) of the imaginary line joining these two shape centers. From C, we project imaginary lines all around it, each line 15 degrees apart (making a total of $\frac{360}{15} = 24$ lines). Intersection of each of these lines is found with both the vector shapes and the nearest intersection point (from C) for both the shapes is recorded. The distance between these two intersections points serves as a good approximation of the distance between both objects. The spatial similarity score is computed as follows -

$$Score = 1 - \frac{D_{o_i,o_j}}{\max_{\forall i,j} D_{o_i,o_j}}$$

where, D_{o_i,o_j} is the computed approximate distance between objects o_i and o_j. Note that overlapping objects are a special case and their score is directly set to 1, bypassing this evaluation.

References

[BE98] BRUNS H., EGENHOFER M.: Similarity of spatial scenes. *Seventh International Symposium on Spatial Data Handling* (02 1998). 1

[DMM03] DESOLNEUX A., MOISAN L., MORE J.-M.: A grouping principle and four applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence 25*, 4 (2003), 508–513. 1

[DMM07] DESOLNEUX A., MOISAN L., MOREL J.-M.: *From gestalt theory to image analysis: a probabilistic approach*, vol. 34. Springer Science & Business Media, 2007. 1

[LL00] LATECKI L. J., LAKAMPER R.: Shape similarity measure based on correspondence of visual parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 10 (2000), 1185–1190. doi:10.1109/34.879802. 1

[Luo14] LUO M. R.: *CIELAB*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014, pp. 1–7. URL: https://doi.org/10.1007/978-3-642-27851-8_11-1, doi:10.1007/978-3-642-27851-8_11-1. 1

[Luo16] LUO M. R.: *CIEDE2000, History, Use, and Performance*. Springer New York, New York, NY, 2016, pp. 202–207. URL: https://doi.org/10.1007/978-1-4419-8071-7_7, doi:10.1007/978-1-4419-8071-7_7. 1

[LZH*17] LUN Z., ZOU C., HUANG H., KALOGERAKIS E., TAN P., CANI M.-P., ZHANG H.: Learning to group discrete graphical patterns. URL: https://doi.org/10.1145/3130800.3130841. 1

[NSX*11] NAN L., SHARF A., XIE K., WONG T.-T., DEUSSEN O., COHEN-OR D., CHEN B.: Conjoining gestalt rules for abstraction of architectural drawings. *ACM Trans. Graph. 30*, 6 (Dec. 2011), 1–10. URL: https://doi.org/10.1145/2070781.2024219, doi:10.1145/2070781.2024219. 1

[Pal77] PALMER S. E.: Hierarchical structure in perceptual representation. *Cognitive Psychology 9*, 4 (1977), 441–474. URL: https://www.sciencedirect.com/science/article/pii/0010028577900160, doi:https://doi.org/10.1016/0010-0285(77)90016-0. 1

[SO95] STRICKER M. A., ORENGO M.: Similarity of color images. In *Storage and Retrieval for Image and Video Databases III* (1995), Niblack W., Jain R. C., (Eds.), vol. 2420, International Society for Optics and Photonics, SPIE, pp. 381 – 392. URL: https://doi.org/10.1117/12.205308, doi:10.1117/12.205308. 1