Multi-Attribute Spaces: Calibration for Attribute Fusion and Similarity Search
Supplemental Psuedocode for Sec. 4

1 Target Attribute Similarity Search

Let us consider the following scenario: a user selects a target image and wants to find images that are similar to it, with respect to a given set of k attributes. The search function should compute distances in the k-dimensional multi-attribute subspace corresponding to the given attributes, but in such a way as to respect the distribution of attribute values in that "neighborhood." The size of the neighborhood considered should be changeable to allow for different levels of similarity, *i.e.*, small neighborhoods result in searching only for "very similar" images, while larger neighborhoods correspond to images which might be only "somewhat similar." Finally, the relative neighborhood size of different attributes should be changeable as well, for emphasizing one attribute compared to another.

Our method for solving this problem applies the EVT normalization to *distances* for each of the target attributes individually, and then sums these for the final similarity score. We first gather images with calibrated attribute values $\alpha_j \leq A_j(I) \leq \beta_j$, where the neighborhood is defined by the range $\alpha_j - \beta_j$ for each attribute (allowing for both different neighbor sizes, and relative weighting of different attributes). For each attribute, we compute the set of L_1 distances between the target attribute value and each of the gathered images. The largest of these distances is assumed to be just outside the "similar" range – the outlier with respect to similarity – and thus the distances immediately smaller (the tail of the extreme values) can be used to fit a Weibull. Intuitively, we are measuring the local distribution of distances for this attribute, close to the target attribute value. As before, we can then use the CDF of the Weibull to estimate the probability that a particular image is "similar" with respect to this attribute and given search range. Finally, we maximize over the L_1 sum of these calibrated probabilities for each attribute (analogously to Eq. 3 in the paper) and return the corresponding images to the user. Since this normalization depends on the particular attribute values of the target image as well as the search ranges, we must perform this calibration at run-time for each query; however, this process is quite fast, typically requiring only a few seconds at most.

Algorithm 1 Calculate distance in normalized score space

Require: A vector of w-scores S'_j for each attribute of interest j, for a total of n attributes
Require: A list of image IDs I_j corresponding to each w-score for each attribute j
Require: Parameters for attribute j: α_j and β_j and a target w-score τ_j; $\alpha_j \leq \tau_j \leq \beta_j$

while $j \leq n$ **do** ▷ Collect similar attribute candidates
  while $i \leq m; m = \| S'_j \|$ **do**
    if $\alpha_j \leq s'_{i,j} \leq \beta_j$ **then**
      D_j.append($|\tau_j - s'_{i,j}|$) ▷ Calc. distance
      C.append($I_{i,j}$) ▷ Add ID to candidate list
    end if
    $i = i + 1$
  end while
  $\hat{D}_j = (1 + \max(D_j)) - D_j$
  Remove the largest value from $\hat{D}_j$
  Fit a Weibull distribution W'_j to $\hat{D}_j$
  $j = j + 1$
end while
while $i \leq m; m = \| C \|$ **do**
  while $j \leq n$ **do** ▷ Combine the norm. distances
    $s''_{C_i} = s''_{C_i} + F(|\tau_j - s'_{C_i,j}|; W'_j)$
    $j = j + 1$
  end while
  $i = i + 1$
end while