Significant Colouring of Image

Design Laboratory Project Report
by
Bhushan Kulkarni (12CS30016)

November 5, 2016

Introduction

Image recolouring is widely studied problem. Identifying significant colours in coloured image and recolour it has various practical applications. It can be used in image compression, since lesser number of colours are required to represent the image. It can also be applied in image segmentation.

Here we present an approach to identify significant colours of image and then apply K-means clustering algorithm to recolour the image. This approach is robust towards the number of colours and segments in the image, adjusts according to image. In the following sections, we discuss the approach in detail.

1 Problem Statement

Given a coloured image, identify its significant colours and recolour it. Approach should not make assumptions about image properties, like number of colours, segments, etc.

2 Solution Approach

We represent the image as a completely connected undirected graph, with the pixels as vertices and RGB colour distance between the pixel colours as weight of edge betweeen corresponding vertices. Now, the problem of identifying significant colours of image can be thought as finding cluster centres in the graph for that image. For that, we use Peak Searching algorithm[1].

Peak Searching Algorithm:

Consider a graph $G = (V, E)$, V is set of vertices and E is set of edges ($|V| = n$). Let W be n x n adjacency matrix corresponding to this graph. The generalized degree of a point V_i is defined as $d_i = \sum_{j=1}^{n} W_{i,j}$.

Now, consider the random walk on graph G with weighted adjacency matrix W. Then, the random walk has a stationary distribution π over all the points. It's reasonable to claim that, the closer to the cluster center a point i is, the higher π_i is. That is to say, π_i is a local maximum of the stationary distribution if i is the closest point to the center. From the theories in stochastic process, we know that, π is the solution of the linear equation:

$$\pi[diag(d_1, d_2, ..., d_n)]^{-1}W = \pi \textit{ subject to } \sum_i \pi_i = 1 \tag{1}$$

Since W is symmetric, we can directly solve π as its proportional to d. So we can use d to indicate the relative value of π. Given the degree of all the vertices, we want to capture one point per area having the highest degree, to locate one cluster center. We call these points the peak points of clusters. To find such points, first we get the point with the highest degree as the first peak point. Note that, peak points should not be close to each other because each of them is near the center of different clusters, therefore, if we have an appropriate neighborhood of the first peak point, then the point of highest degree outside the neighborhood will be the second peak point. Theoretically, we can capture all the rest peak points by cutting off appropriate neighborhoods of the existing peak points. But it is difficult to determine the size of neighborhoods. Here we introduce the concept of persistency of points. As we increase the size of neighborhood, the highest degree outside the neighborhood will decrease. Specifically, we consider a k-nearest neighborhood of the first peak point. And as we increase k, more and more points are included in the neighborhood. The highest degree outside the growing neighborhood is decreasing. But there exist some resistance against such drop tendency. We call such resistance against the drop tendency as the persistency of a point.

Peak points tends to have high persistency than non-peak points. After we get m peak points. We search the $(m+1)$th peak point (if there is any) by the following rules: We cut off the k-nearest neighborhoods of the m current peak points simultaneously and observe the point with the highest d among the rest of the points, then we pick out the most persistent point during the growth of k. Then, the point we pick out is the potential $(m+1)$th peak point. Now we consider the termination of the searching process. One way is to set a lower bound for the persistency, since short persistency does not reflect the stable structure.

In our model, graph is undirected and completely connected. First we select some vertex randomly as the first peak point since all vertices have the same degree. For further cluster centres, we choose a vertex which has maximum persistency, i.e. which is at the maximum distance from already chosen peak points (minimum of the distances from peak points is maximum). We stop the process when the persistence falls below a pre-defined threshold. We set the threshold as maximum RGB distance acceptable for the application. Lower is the threshold, higher is the number of clusters. After choosing cluster means, we apply K-means clustering algorithm, each iteration assigns nearest cluster

Figure 1: Original Image 1

Figure 2: Significant Colours

to each pixel and reassigns colour of the cluster as the mean of colours of pixels assigned to that cluster. Ties are broken arbitrarily.

3 Experiments

We used the image segmentation dataset[2] to test our approach. We generated the output images by varying parameters. The figures show original image, initially identified significant colours, resulting image after running K-means algorithm with number of clusters limited to 5 and similarly for 10.

Initially significant colours are identified and each pixel of image is assigned to the nearest cluster (Figures 2, 6, 10, 14). In these images, we observe the significant colours of image. Running k-means algorithm results in domination of the colour that has majority in image since we are taking the mean of colours in a cluster. For ex. in figure 3, blue colour of the sky dominates. We also observe, as we allow more number of clusters, resulting image resembles the original image more and more. Here, 10 colours show good results. Colours of various objects are displayed properly. The code can be found at `https://github.com/bhushan99/Significant_coloring_of_image`.

In present work, we are not considering spacial distribution of the colours of image. We are treating each edge equally in complete graph. So, we are breaking ties arbitrarily while assigning clusters. This model can be improved upon by taking into account the spacial distribution and other factors like boundary detection of object so that its initial significant colour remains intact in resulting image.

Figure 3: after 5-means clustering

Figure 4: after 10-means clustering

Figure 5: Original Image 2

Figure 6: Significant Colours

Figure 7: after 5-means clustering

Figure 8: after 10-means clustering

Figure 9: Original Image 3

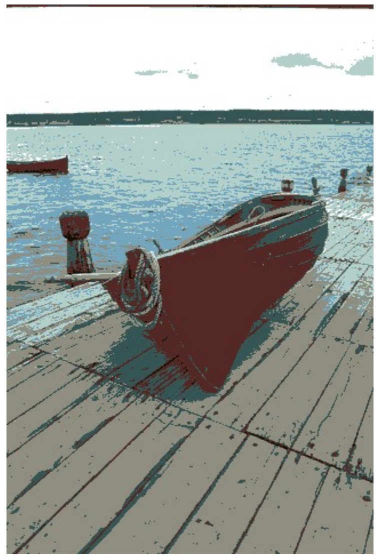

Figure 10: Significant Colours

Figure 11: after 5-means clustering

Figure 12: after 10-means clustering

Figure 13: Original Image 4

Figure 14: Significant Colours

Figure 15: after 5-means clustering

Figure 16: after 10-means clustering

4 Conclusion and Future Work

In the present work, we aimed to identify significant colours of the image and recolour the image by using those colours. We applied peak searching algorithm to identify significant colours. Then K-means algorithm is used to recolour the image. We observed good results by using this approach. The improvements in the approach like considering spacial distribution of colours of pixels remains in the future work.

References

[1] Peng Xu, et. al., Stanford University, CS 229: Machine Learning Final Projects, Autumn 2013, `http://cs229.stanford.edu/proj2013/XuLiu-ClusteringMethodsWithoutGivenNumberOfClusters.pdf`.

[2] UC Berkeley, Computer Vision Group, Contour Detection and Image Segmentation Resources, `https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html`.