Pb Edge Detector:

(Rewritten by Han)

1. Generally introduce Pb Edge detector from Martin et. al. paper.

In our project, the Pb edge detector was used in the mathematical optimization of the superpixel grouping process in the section 3.3, and it was first introduced and implemented by Martin et. al. [7]. It was intended by them to solve the edge detection difficulties, which could not by solved by some traditional algorithms like Canny edge detector [Add another reference of Canny]. Martin et. al. [7] found that the Canny edge detector could not perform well when there are high contrast edges inside a texture area or boundaries among different objects with subtle difference in brightness. Therefore, they [7] brought the idea to combine brightness, color, and texture to calculate a posterior probability of whether it is on a boundary for each pixel. For calculating features, they use local discontinuities for each feature, over multiple ranges and scales. To implement this, they treat brightness by the oriented energy (OE) approach [Add another reference of it]. According to [7], the OE is reproduced as the following:

Here, is the intensity at the pixel; and are functions of even and odd symmetric filters at orientation and scale . According to Martin et. al. [7], the even-symmetric filter is a Gaussian second-derivative, while the corresponding odd one is its Hilbert transform. For more details, please check [7], [ ], and [ ]. In short, this equation, as a standard means, is used to detect and localize composite edges [ ].

Then, the second core in the Pb edge detector implementation is the gradient-based features. They are used to treat color, texture, and also brightness. Martin et. al. [7] used the following formula to calculate the local gradiences for each of these features:

The motivation is to use a circle with two half equal discs and center the circle at each pixel. Then, a diameter at a certain orientation will be the edge between the two discs. The user can then set the number of bins in each disc and calculate the values for each pair of bins from the two discs. The and are the values of the bins from the two discs. The resulting value will summarize the local feature for that pixel. The above equation will then be used to individually calculate for color gradient (CG), texture gradient (TG), and brightness gradient (BG).

To put all the above equations (OE, CG, TG, and BG), Martin el. al. [7] considered that using linear weights to combine them can be enough, and they proposed to use supervised learning approaches to learn the weights. They tested multiple models, including density estimation, classification trees, logistic regression, hierarchical mixtures of experts, and SVM. As a result, they found the weights produced by simple classifiers can perform well. The benefits of using Pb edge detector for detecting edges are to avoid relying too much on the brightness feature by traditional algorithms. Later, Arbelaez et. al. [8], based the work of Martin et. al. [7], proposed an more advanced Pb edge detector named Global Pb edge detector, which also consider the global constrains in the edge computation. Therefore, incorporating the Pb detector in our project can help us prevent treating an edge inside an object as a boundary.