Free HAND-Drawn Sketch segmentation

zhebang Sun , Changhu Wang Liquing zhang and Lei Zhang

March 19, 2015

seminar report

Supervised by Dr. Abhiram Ranade.

1 Introduction

1.1 Some Definition

Segment: Any of the parts into which something can be divided .
Segmentation It is process of dividing things into multiple segments(or parts/sets) having similar properties.
Image segmentation: It is process of dividing an Image into Distinct region having similar looking pixels.
Stroke: It is the collection of points in the interval from pen down to pen up while drawing.
Sketch Segmentation: It is the grouping of strokes that constitute the same object .

1.2 Problem Description/Objective

Humans can easily recognize a sketch as an object, but it is very difficult for a computer to do recognize it.for example consider a image show in Figure 6 , It can be easily recognized as wheel by a human being but, for a computer to do is still a great challenge.
The main Objective is to study how to segment a freehand sketch at a object

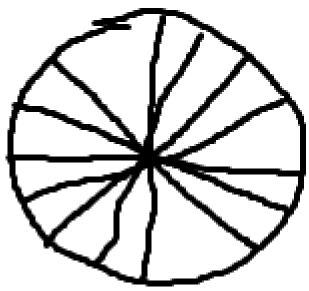

Figure 1: Wheel Example

level .If there is a sketch of many overlapping objects then each individual object must be correctly recognized, It is a hard problem even for Human beings.

1.3 solution proposed

Basically two level of factors should be considered in sketch segmentation:

1. low-level perception
2. hig-level knowledge

Figure 2: (a)

Low-level aspects such as *proximity* is the basic factor for segmenting things. For example if we are asked to segment the sketch in figure 2 , we will easily segment the different things the figure without any proior knowledge about them only on the factor of their proximity as things are certain distance apart from each other.

high-level knowledge also has an importance in the process of sketch segmentation . In figure

2 Understanding Object-Level Sketch Segmentation

Two approaches were introduce:

1. proximity-based segmentation
2. semantic-based segmentation

2.1 proximity-based segmentation

factors like *proximity* and *similarity* are used as these are most important factors to influence the natural groupings of human being: because it is the tendency of person to group similar appearing things together.

Q: How sketch segmentation is different from image segmentation ?

A: because in image segmentation information from the intensities and color values can be used but the sketch generally lack these kind information.

Figure 3: parrot Image

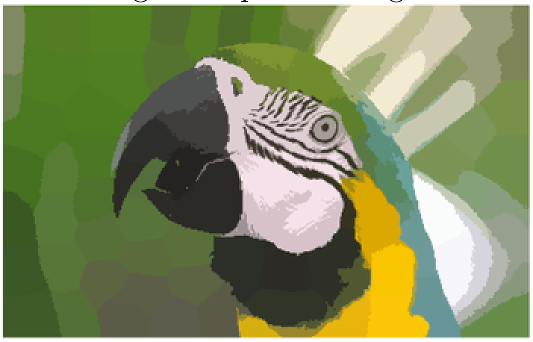

Figure 4: Parrot image after segmentation

first, we will study the graph based approach for image segmentation then a modified graph based approach is introduced for the sketch segmentation let us first understand the graph based approach for image segmentation.

2.1.1 Efficient Graph-based Image Segmentation (EGIS)

To represent an image an undirected graph is used, Let $G = (V, E)$ denote the graph . $\forall$ vertices $v \in V$ denotes the pixel in the image. and there is weighted edge set $E\{(v_i, v_j)|\forall v_i, v_j \in V$ and $v_i \neq v_j\}$ where non-negative weight $w(v_i, v_j)$ indicate the dissimilarity between the pixels connected by the edge(e.g., the difference in the intensity, color, location or some other local attribute).

Segmentation It is to partition set V into set of segments$S$,in which each Segment S (i.e. a region which is collection of pixels with similar properties) is like a subgraph which is defined as:

$$G' = \{S, E'\}$$

where $E' \subset E$. Here the *internal difference* of a Segment S is defined as the largest possible weight in the minimum spanning tree $MST(S, E')$ constructed of a Segment(which is a subgraph $G' = \{S, E'\}$) i.e,

$$Int(S) = max_{e \in MST(S,E')} w(e) \tag{1}$$

and the *minimal internal difference* of two segments S_1, S_2 is also defined same as min of *Internal difference* $Inst(S_i)$ plus some tolerance $T(S_i)$ of two segments,i.e,

$$MInt(S_1, S_2) = min(Int(S_1) + T(S_1), Int(S_2) + T(S_2)) \tag{2}$$

in which $T(S) = K/|S|$,where $|S|$ is size of S and k is a parameter. The *segment Distance* between segments S_1 and S_2 is given as a the minimum edge between two segment:

$$Dist(S_1, S_2) = min_{v_i \in S_1, v_j \in S_2, (v_i, v_j) \in E} w(v_i, v_j) \tag{3}$$

merging criterion for image segmentation is i.e.
if $Dist(S_1, S_2) \leq MInt(S_1, S_2)$ then two segments S_1 and S_2 should be merged to single segment.

Algorithm 1 . Graph Based Image segementation

Input A Image
Output A set of segments of Image
1: Initially each individual pixel is considered as a segment and add it to Set of Segments SS.
2: For each segment $s \in SS$
3: Repeat the following step(4-12) until no merging is possible
4: calculate the internal difference $Int(s)$ of s using eq 1
5: for every other segment $s' \in SS$ such that $s' \neq s$
6: calculate the internal difference $Int(s')$ of s' using eq 1
7: calculate the minimal internal difference $MInt(s, s')$ of s and s' using eq 2.
8: calculate the minimum edge D between two segments s and s'
9: If $D \leq MInt(s, s')$
10: remove s and s' from SS
11: merge s and s' to get segment $snew$
12: add $snew$ to SS. GOTO step 4
13: set of Segment SS is returned as output.

2.1.2 Modified Efficient Graph-based Image Segmentation for sketch segmentation

In this approach,
vertices $v \in V$ denotes the strokes of drawing (instead of every pixel in the image segmentation)
and there is weighted edge set $E\{(v_i, v_j)|\forall v_i, v_j \in V$ and $v_i \neq v_j\}$ where weight $w(v_i, v_j)$ is the minimum Euclidean between two vertices(strokes).

Here the *internal difference* of a Segment S is defined in the similar as in Image Segmentation i.e.

$$Int(S) = max_{e \in MST(S,E')} w(e) \tag{4}$$

and the *minimal internal difference* of two segments S_1, S_2 is also defined same as :

$$MInt(S_1, S_2) = min(Int(S_1) + T(S_1), Int(S_2) + T(S_2)) \tag{5}$$

in which $T(S) = K/|S|$,where $|S|$ is total length of the stroke divided by maximal side of the panel. The *segment Distance* between segments S_1 and S_2 is given as:

$$Dist(S_1, S_2) = min_{v_i \in S_1, v_j \in S_2, (v_i, v_j) \in E} w(v_i, v_j) \tag{6}$$

merging criterion was same as it was for image segmentation i.e.
if $Dist(S_1, S_2) \leq MInt(S_1, S_2)$ then merge two segments

effects of modification to graph Segmentation criterion and Motivation for proximity based approach
Q : why above graph segmentation fail ??
A : because of following problems

problem 1 **proximity** : above algorithm tends to separates the small strokes from the object. e.g. a small tick near to sun (main object) is considered a separate object.
reason : this problem arise due to equally treating the two segments while calculating the *minimal internal difference.*
motivation : a short stroke may be the part of the large stroke regardless of some distance between them.

problem 2 **enclosure**: it fails when some segment is enclosed by some another segment
reason : The second problem arises from considering the strokes as vertices

of the graph rather than pixel of the image .
Motivation : algorithm should merge two segment component together if one lies “on“ or completely enclosed by another.

2.1.3 Proximity Based sketch Segmentation

after analyzing the two problems discuss above solution to these problems is given as:

for the first problem, we will change our *minimal internal difference* definition to :

$$MInt(S_1, S_2) = min(Int(S_1), Int(S_2)) + max(T(S_1), T(S_2)) \tag{7}$$

the term $max(T(S_1), T(S_2))$ relaxes the merging threshold as: T(S) will be large for a very small stroke as $|S|$ will be large for small stroke this this takes care of proximity problem.

To fix the second problem *enclosure correlation* between two segment is taken in to consideration which is calculated as :

$$Corr(S_1, S_2) = \frac{|C_{S_1} \cap C_{S_2}|}{min\{|C_{S_1}|, |C_{S_2}|\}}, \tag{8}$$

where C_s is the convex hull of segment S and $|C_s|$ is its area. similarly, $|C_{S_1} \cap C_{S_2}|$ is overlap convex hull region between C_1 and C_2

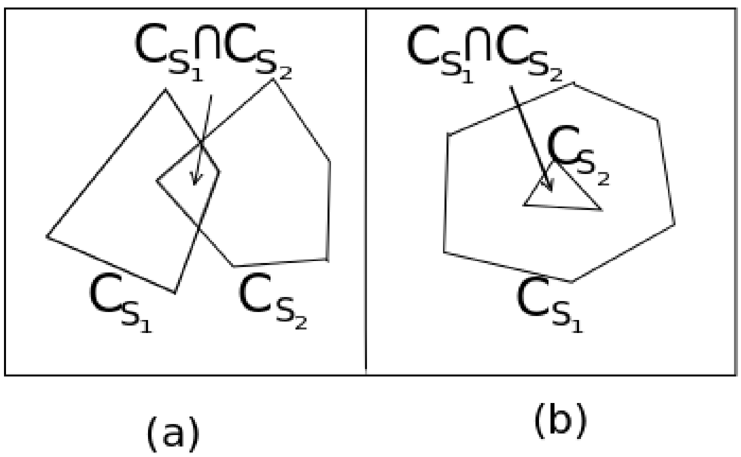

Figure 5: Convex-hull

In the figure 5 (a) Here area of Intersection of convex hull of two segment $|C_{S_1} \cap C_{S_2}|$ is less than the area of Convex hull of individual segment i.e.

min($|C_{S_1}|, |C_{S_2}|$) thus correlation between two segment will be low, hence they are likely to be separated. In the figure 5 (b) Here area of Intersection of convex hull of two segment $|C_{S_1} \cap C_{S_2}|$ is $|C_{S_2}|$ itself(since segment S_2 lie inside segment S_1 i.e. $|C_{S_2}| = min(|C_{S_1}|, |C_{S_2}|)$ thus correlation between two segment will be high hence they are likely to be merged as single segment . Now this term is added to the Modified *minimal internal difference* to ensure that, overlapped segments must be merged together.

$$MInt(S_1, S_2) = min(Int(S_1), Int(S_2)) + max(T(S_1), T(S_2)) + \lambda Corr(S_1, S_2). \tag{9}$$

---

correlation : It is a statistical measures of how two set of Data are related to each other. It is defined as .
$corr(X, Y) = \frac{cov(X,Y)}{\sigma_x \sigma_y}$
Convex Hull : Convex hull or convex envelop of a set X of points in the Euclidean plane or space is smallest convex set that contain X . It can be realized as a rubber band stretched around the of points, to give minimum convex shape polygon, which enclose all the points of the set.

Algorithm 2 . Proximity Based sketch Segmentation

Input A squence of Strokes
Output A set of segments of strokes
1: Initially each individual stroke is considered as a segment and add it to Set of Strokes SS.
2: For each segment $s \in SS$
3: Repeat the following step(4-12) until no merging is possible
4: calculate the internal difference of segment $Int(s)$ of s using eq 4
5: for every other segment $s' \in SS$ such that $s' \neq s$
6: calculate the internal difference $Int(s')$ of s' using eq 4
7: calculate the minimal internal difference $MInt(s, s')$ of s and s' using eq 9
8: calculate the minimum distance D between two segments s and s'
9: If $D \leq MInt(s, s')$
10: remove s and s' from SS
11: merge s and s' to get segment $snew$
12: add $snew$ to SS. GOTO step 4
13: set of Segment SS is returned as output.

This proximity based Sketch Segmentation us first step to segment separable sketches.This approach did not able to segment the overlapped parts into meaningful object, for this we need a High-level information approach rather than a low-level perception .

2.2 Semantic Based Sketch Segmentation

we will make use of the high-level knowledge for distinguishing the object from complex background.To do this we need a knowledge base, we will look at two approaches for this :
1. Robust Entropy Descent Merging
2. Greedy Backward Segmentation

Entropy : It measures the unpredictability of the information content. It is given as

$$H(x) = E[-log(P(X))] = \sum_{i=1}^{n} P(x_i)logP(x_i)$$

where X is the random variable can take values $x1, ..., xn$ and P(x) is probability mass function.
note: Entropy of a meaningful object/data set is usually lower than that of non object/data set.

The knowledge base is the collection of 1 million clip art image from the web with textual information associated with them such as image title and some text for its description

2.2.1 Robust Entropy Descent Merging

The intution here is that Entropy of textual information for meaningful image will be less than that of meaningless image .
Here search engine [1] was used to get the textual information. This search engine takes a sketch as input and returns the closed matched images with given sketch from the its 1 million image corpus and also the textual/word information associated with the images .

Sketch Entropy for a sketch $'Sk'$, if there are N images returned by the search engine,and Let $W = w_1, w_2, .., w_m$ be set of M unique words in the textual description of the N images, then the Entropy for this sketch Sk is given as :

$$Pr(w) = \frac{Score(w|Sk)}{\sum_{w_i \in W} Score(w_i|Sk)} \tag{10}$$

where score between word w and Sketch is given by:

$$Score(w|Sk) = \sum_{n=1}^{N} \delta(w, I_n) \times Sim(Sk, I_n), \tag{11}$$

Where function $\delta(w, I_n)$ is defined as:

$$\text{saf } \delta(w, I_n) = \begin{cases} 1: & \text{if w appear in description of} I_n \\ 0: & \text{otherwise} \end{cases}$$

Where $Sim(Sk, I_n)$ is the similarity between the image I_n and the Sketch Sk which is determined from the search Engine.

Sketch Entropy then is given by :

$$H = \sum_{w \in W} -Pr(w) log Pr(w). \tag{12}$$

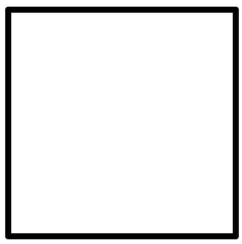

Figure 6: Wheel Example

Consider an example from the figure (4) here a square image is given to search engine following words appear in the textual information of this image:

words	**occurences**
square	4
rectangle	2
window	1

Table 1: word infromation for square image

Thus its entropy is calculated as:
pr(square) = 4/7

pr(rectangle) = 2/7
pr(window) = 1/7
H = -[pr(square)log(pr(square))
+pr(rectangle)log(pr(rectangle))
+ pr(window)log(pr(window))]
H = 0.9554

We will merge two segment S_1 and S_2 if their individual Entropy is greater than their merged segment,i.e. $H(S_1 \cup S_2) < min\{H(S_1), H(S_2)\}$. In practice adding a small stroke might cause small fluctuation in the Entropy this gives us intuition that the small stroke might belong to same segment, if the stroke causes a large entropy change then this might be a separate segment.But, we should also be take difference word distributions information into account,for this KL-divergence is used between the distribution of two sketches which is given as:

$$D_{KL}(Sketch_1, Sketch_2) = \sum_{w \in W_1} Pr_1(2) log \frac{Pr_1(w)}{Pr_2(w)} \tag{13}$$

Where $Sketch_1$ is the old segment and $Sketch_2$ is the new segment after adding a stroke to $Sketch_1$.

KL-Divergence(Kullback Leiber divergence It is the measure of difference between probability distribution P and Q . KL-Divergence of Q from P or $D_{KL}(P||Q)$ is a measure of information lost when Q is used to approximate P. given as,

$$D_{KL}(P, Q) = sum_i P(i) log \tfrac{P(i)}{Q(i)}$$

defined if $Q(i) = 0 \Rightarrow P(i) = 0$

---

merging criteria: Two segments S_1 and S_2 will be merged together if,

$$H(S_1 \cap S_2) < min\{H(S_1 + \tfrac{\beta}{D_{KL}(S_1, S_1 \cup S_2)}), H(S_2 + \tfrac{\beta}{D_{KL}(S_2, S_1 \cup S_2)})\}$$

Here β was choosen 0.3 empirically. **Calculation of K-L Divergence**

In the above figure 7, S_1 is the segment already there in the sketch before, then segment S_2 is added to it so the sketch becomes $S_1 \cap S_2$ Suupose the following words information is obtain after firing query for the sketch S_1 and $S_1 \cup S_2$:

Given above data the following is calculated :
Pr(Sun|S_1) = 2/6 = 0.333
Pr(wheel|S_1) = 3/6 = 0.5

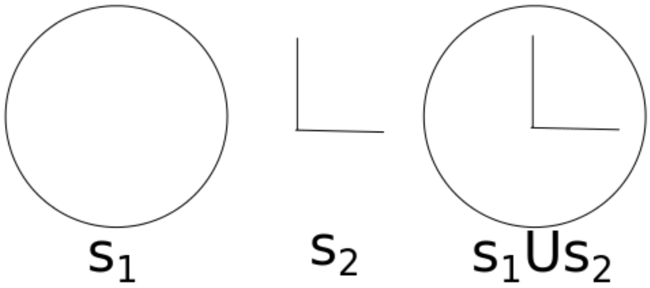

Figure 7: calculation of K-L divergence

words	**occurences**
sun	2
wheel	3
clock	1

Table 2: Result for query sketch S_1

Pr(clock$|S_1$) $= 1/6 = 0.167$
also,
Pr(Sun$|S_1 \cup S_2$) $= 0/6 = 0$
Pr(wheel$|S_1 \cup S_2$) $= 1/6 = 0.167$
Pr(clock$|S_1 \cup S_2$) $= 5/6 = 0.833$
Thus,
KL-Divergence is calculated as :

$$
\begin{aligned}
D_{KL}(S_1, S_1 \cup S_2) = {} & Pr(Sun|S_1)log(\frac{Pr(Sun|S_1)}{Pr(Sun|S_1 \cup S_2)} \\
& + Pr(wheel|S_1)log(\frac{Pr(wheel|S_1)}{Pr(wheel|S_1 \cup S_2)} \\
& + Pr(clock|S_1)log(\frac{Pr(clock|S_1)}{Pr(clock|S_1 \cup S_2)} \\
& = 0.33log(\frac{0.33}{0}) + 0.5log(\frac{0.5}{0.167}) + 0.167log(\frac{0.167}{0.833}) \\
& = 0.2381 - .11655 \\
& = 0.1215719
\end{aligned}
$$

if word information changes too much then D_{KL} will increase thus the term $\frac{\beta}{D_{KL}}$ will decrease so they are likely to be not merged . if word information does not change too much(due to a small stroke) it D_{KL} will be less and hence the term $\frac{\beta}{D_{KL}}$ will be high and greater chances of getting merged .

words	occurences
sun	0
wheel	1
clock	5

Table 3: Result for query sketch $S_1 \cup S_2$

2.2.2 Greedy Backward Segmentation algorithm

In this order of drawing is leveraged.The main idea is to check for each newly drawn stroke whether it could be merged with the all the existing segments to be one segment . Here the order of the drawing of strokes is taken into consideration.

Algorithm 1 . Greedy Backward Segmentation algorithm

Input: The set of strokes ordered by drawing sequence,$s_i, i = 1, 2, ..., n$.
Ouput: set of Segments ,$SS = S_j, j = 1, ..m$.
Process
Set $SS = \phi, m = 0$.
For each stroke $s_i, i = 1, ...n$
    $SS \leftarrow SS \cup s_i, m \leftarrow m + 1$.
    Let SS_j is the set of last j segments i.e. $SS_j = S_{m-j+1}, ..., S_m$ and C_j denotes the all the segment with all the strokes in SS_j .
    For j=m,...1
        For each segment $S_k \in SS_j$.
            if $H(C_j) < H(S_k) + \frac{\lambda}{D_{KL}(S_k, C_j)}$,
                then $SS \leftarrow S_1, S_2, ...S_m - j \cup C_j, m = m - j + 1$,Go to 14.
            End
        End
    End
End
Return SS.

Explaination In the above Greedy Backward segementation Algorithm we defined our input as a set of strokes ordered by drawing sequence(time when the drawn) as s_i. Let total number of strokes be n. And output of the algorithm is a set of segments SS , that contains m segments $S_j, j = 1, ..m$. where each segement S_j is set of sequence of strokes. thus m¡n.

Initially SS is empty and $m = 0$,then for each stroke s_i we assign it to SS as a individual segment i.e $SS \leftarrow SS \cup s_i$ where s_i is a segment itself and also increment the counter m by 1 . for the last j segments a set SS_j is defined as $SS_j = S_{m-j+1}, ..., S_m$, and C_j is the new segment that is formed by union of all the segments in SS_j i.e. all previous j segments are combined to form a new segment SS_j . Initially whole SS is taken as segment to compare that's why j starts from m. for each C_j merging condition is check if the merging condition is satisfied for any value of j i.e. entropy of all the individual segment S_k in SS_j plus KL-divergence of $S_k to C_j$ is less than the entropy of c_j, then C_j is added as new segment after removing all the previous j segments from the SS and changing value of m to $m - j + 1$ process stops for a stroke s_i, then same process is repeated for next stroke . Finally SS is Returned as desired output.

2.3 Classification Based On Intitutive Clues

1: similarity
2: Continuity of Direction
3: symmetry
4: closure
5: Compactness
6: Change of Stroke Length, convex hull, bounding box

To study more : Random forest Ression , chamfer distance , Noramlized cut .

References

Query-Adaptive Shape Topic Mining for Hand-Drawn Sketch Recognition. ACM Conference on Multimedia, 2012.