HW6

Yanan Da

March 2021

Algorithm 1: Construct Graph

```
input  : Image I
output: Graph G(V,E)
$grad$ ← calculate gradient of I;
foreach $pixel\ i \in I$ do
    add $i$ to $V$;
    foreach $neighbor\ j\ \ of\ i$ do
        add edge $ij$ to $E$;
        assign edge capacity $c_{ij}$ according to image gradient;
    end
end
```

Algorithm 2: GCBAC

```
input  : initial contour $c_0$, image I, graph G, step_size
$C \leftarrow \emptyset$ ;
$i \leftarrow 0$;
while $true$ do
    $IC_i \leftarrow$ shrink $c_i$ by $step\_size$;
    $OC_i \leftarrow$ dilate $c_i$ by $step\_size$;
    $CN_{c_i} \leftarrow IC_i \cup OC_i$ ;
    $V_s \leftarrow$ pixels corresponding to $IC_i$;
    $G_i, s_i \leftarrow$ node_identification$(G, V_s)$;
    $V_s \leftarrow$ pixels corresponding to $OC_i$;
    $G_i, t_i \leftarrow$ node_identification$(G_i, V_s)$;
    $cut\_value, cut\_sets \leftarrow$ minimum_cut$(G_i, s_i, t_i)$;
    $c_{i+1} \leftarrow argmin_{c \in CN_{C_I} E(c)}$ where $E(c)$ = capacity of cut_set that c
      belongs to ;
    if $c_{i+1} \in C$ then
        break;
    else
        add $c_{i+1}$ to $C$;
        $i \leftarrow i + 1$;;
    end
end
```

```
Algorithm 3: node_identification
input  : $G(V, E), V_s$
output: $G(V', E'), v$
$v \leftarrow$ merge $V_s$ to a new node ;
$E(V_S) \leftarrow \{(u_1, u_2)\}$ where $u_1 \in V_s$ or $u_2 \in V_s$ ;
$E(V) \leftarrow \{(u, v), (v, u)\}$ where $u \in V - V_s$ ;
$c^{'}(v, u) \leftarrow \sum_i c(v_i, u)$;
$c^{'}(u, v) \leftarrow \sum_i c(u, v_i)$;
$V^{'} \leftarrow V - V_s + v$;
$E^{'} \leftarrow E - E(V_S) + E(V)$;
```

Algorithm 1 is used to construct the required graph for algorithm2. In algorithm 2, we can use morphological operations from the opencv library to dilate or shrink the contour [1], and we can use the minimum cut function implemented from the networkx library in python[2]. Algorithm 3 is used to merge all the verticals corresponding to a contour to a single source or sink vertical and construct the new graph that will be used for graph cut algorithm.

[1] `https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_morphological_ops/py_morphological_ops.html`

[2] `https://networkx.org/documentation/stable//reference/algorithms/generated/networkx.algorithms.flow.minimum_cut.html`