1. Initial segmentation using Quad-tree ($60 \times 60$ pixel blocks, minimum $10 \times 10$ pixel block) all are main segments
2. Initial flow for each segment is $\vec{0}$ vector
3. Iteratively:
   (a) Compute refined OF for each segment each image using the last computed Segmentation This is a matter of establishing mapping $m^{tu}$ between the two frames and seeing the translation of centroids. Iterating over all segments:
      i. Establish $m^{tu}$ by associating segments having similar size, color and proximity. This is done by seeking corresponding segments which maximize:

$$e^{-(\Delta C)^2/\sigma_c^2} T(S_k^t, S_l^u) e^{-(\Delta x - \bar{v}(S_k^t))^2/\sigma_x^2}$$

      $\Delta C$ is the difference in $\mu_k$'s; $\sigma_c$ is the estimated std. deviation in the average colors; $T(S_k^t, S_l^u)$ is the segment size similarity measure taken as $\frac{\min(\#S_k^t, \#S_l^u)}{\max(\#S_k^t, \#S_l^u)}$; $\Delta x$ is the difference between the segment centroids; $\sigma_x$ is the estimated standard deviation of the difference in position of the centroids; $\bar{v}(S_k^t)$ is the weighted average flow (from previous iteration) of the segments under consideration:

$$\bar{v}(S_k^t) = \frac{\sum_l v(S_l^t) e^{-(\Delta C)^2/\sigma_{c_t}^2} e^{-(\Delta x)^2/\sigma_{x_t}^2}}{\sum_l e^{-(\Delta C)^2/\sigma_{c_t}^2} e^{-(\Delta x)^2/\sigma_{x_t}^2}}$$

      $l$ is a segment of the frame $t$; $v(S_l^t)$ is the flow vector of $S_l^t$ from last computed flow; $\Delta C$ is the difference in the average color of segment $k$ and $l$; $\sigma_{c_t}$ is the estimated std. dev. of the difference in avg. colors; $\Delta x$ is the difference in centroid positions of segement $k$ and $l$; $\sigma_{x_t}$ is the estimated std. dev. of the difference in the centroid positions. Essentially, $e^{-(\Delta x - \bar{v}(S_i^t))^2/\sigma_x^2}$ is a regularization factor for keeping the flow consistent among segments.
      ii. Select the segment which has the highest score. To model occlusion, it can be enforced that the highest score is above a certain threshold.
      iii. Compute the flow vector: To avoid computing an errant flow vector due to occlusion: for each surrounding segment, advect all pixels in it by the previously computed flow and find the total color difference. The flow vector given the lowest color difference is chosen as $m_k^{tu}$.
   (b) Compute refined segmentation for each image using the last computed OF: establish the two segment index maps $s_i^1, s_i^2$ by iterating over each pixel:
      i. Get all segment indices $S$ (from the prv. segm) in $5 \times 5$ neighborhood
      ii. Find which in $S$ are main segments. These $C$ are are candidate segments (interior pixels usually has 1 candidate segment)
      iii. If 1 candidate segment. Then $\alpha_i = 1$, and $s_i^1 := C$
      iv. If $> 1$ candidate segment, do the following for every possible pair $P_1, P_2$:
         A. $1^{st}$ score: Count how many pixels in the $5 \times 5$ neighborhood have either of the segments $P$ as the main segment.

B. $2^{nd}$ score: Project using pixel position using $d_i^{tu}$. Using $m^{tu}$ get the mappings for the currently selected candidate segments. Count how many pixels in the $3 \times 3$ neighborhood have either of the segments $m^{tu}(P_1), m^{tu}(P_2)$ as the main segment.

C. $3^{rd}$ score (alpha similarity factor): Sees how close is the pixel color to the alpha blended color. $e^{-R(p,P_1,P_2)^2/\sigma_s^2}$ where $R(p, P_1, P_2)$ is the perpendicular distance of $c_i$ from the line joining $c_{P_1}$ and $c_{P_2}$ (the average color of the segments). If the $c_i$ is ouside the line segment, then only one segment is assigned. $\sigma_s$ is the std. dev. of the colors in the main segment.

D. Multiply the three scores (and optionally divide by the area of the proposed main segment) and the highest overall score gives the winning segments pair. The segment closest to original color is the main segment.