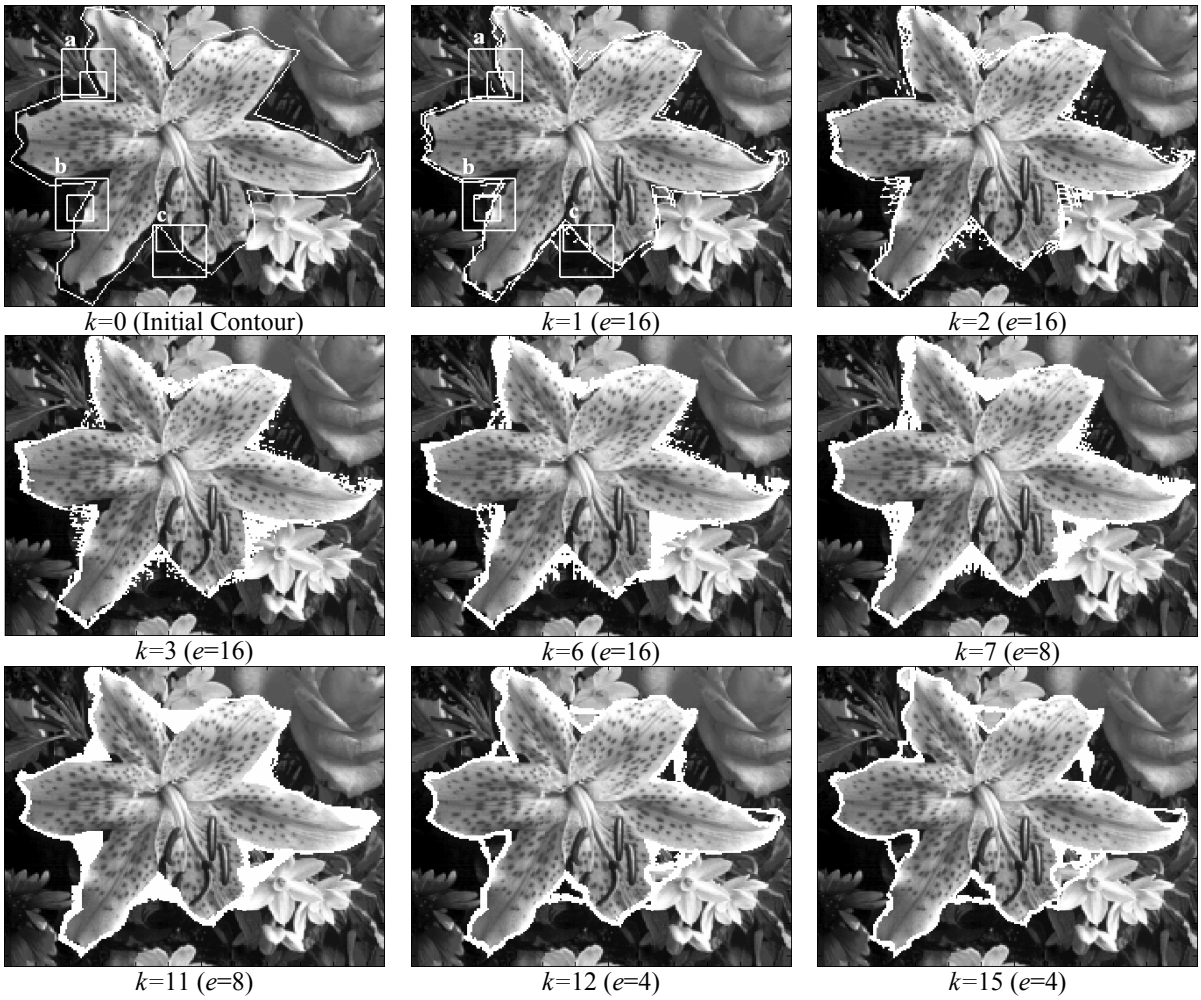
Enhanced Ida's Algorithm

In this supporting document, we state the enhanced Ida's algorithm in more details. We discarded these details in the original manuscript for summarization.

In Ida's algorithm, the alpha mask which simply includes the mapped pixels does not provide any condition for the contour topology. In other word, using such alpha mask to define the contour, the algorithm can not avoid the abnormally deformations made by the fractal behavior as illustrated in Fig. 1.



k=0 (Initial Contour) *k*=1 (*e*=16) *k*=2 (*e*=16)

k=3 (*e*=16) *k*=6 (*e*=16) *k*=7 (*e*=8)

k=11 (*e*=8) *k*=12 (*e*=4) *k*=15 (*e*=4)

Fig. 1. Deformations of the alpha mask foreground during running Ida's algorithm for the Flower image with e_{max}=16 and e_{min}=4.

However, if the contour is defined as a parametric contour according to (1), the topology of the contour will remain unchanged against the fractal behavior [14].

$$\boldsymbol{x}_e^k\left(\frac{n}{L_e}\right)=\left[x_e^k\left(\frac{n}{L_e}\right),y_e^k\left(\frac{n}{L_e}\right)\right],\qquad n=0,1,...,L_e-1 \tag{1}$$

where L_e indicates the number of contour points for the domain size e and k is an iteration counter. Since each contour point may be included by several M_i, its new position (in each mapping step) is obtained by computing the mean of all possible mapped points as follows:

$$\boldsymbol{x}_e^{k+1}(\frac{n}{L_e})=\frac{1}{|P|}\sum_{i\in P}m_i\left(\boldsymbol{x}_e^k(\frac{n}{L_e})\right),\qquad P=\left\{i\,\middle|\,\boldsymbol{x}_e^k(\frac{n}{L_e})\in M_i\right\} \tag{2}$$

where the operator |.| returns the cardinality of a set.

The larger domains can increase the capture range of the algorithm while smaller domains are suitable to take account object boundary details. Therefore, the computational cost of the algorithm can be reduced by primarily choosing an enough small $L_{e_{\max}}$ and increasing L_e by decreasing e as follows:

$$L_e=e\times\left(\frac{L_{e_{\max}}}{e_{\max}}\right) \tag{3}$$

Furthermore, once the algorithm converges with one domain size, the initial contour for the new half domain size is given by:

$$\boldsymbol{x}_{\frac{e}{2}}^0(\frac{n}{L_{\frac{e}{2}}})=\frac{1}{2}\left(\boldsymbol{x}_e^{\eta_e}(\frac{1}{L_e}\left\lfloor\frac{n-1}{2}\right\rfloor)+\boldsymbol{x}_e^{\eta_e}(\frac{1}{L_e}\left\lceil\frac{n+1}{2}\right\rceil)\right),\qquad n=0,1,...,L_{\frac{e}{2}}-1 \tag{4}$$

where the operators $\lfloor\cdot\rfloor$ and $\lceil\cdot\rceil$ give the next larger and preceding smaller integers, respectively.

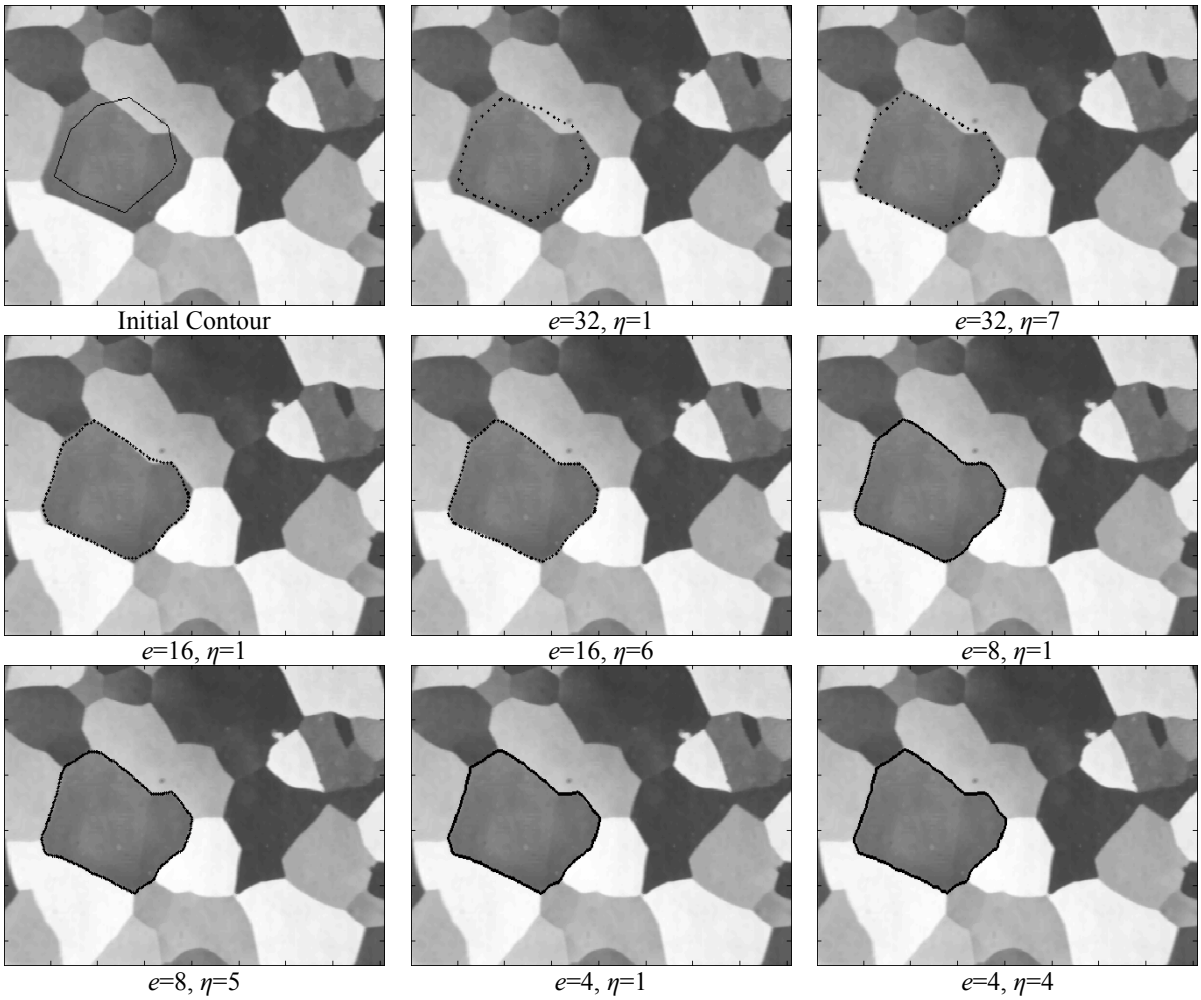


Fig. 2. The contour deformations of the enhanced Ida's algorithm for the Tile image with e_{max}=32, e_{min}=4, and $L_{e_{max}} = 60$.

Figs. 2 and 3 illustrate the contour deformations of the enhanced Ida's algorithm for the Tile and Flower images, respectively. As shown in Fig. 3, the number of mapping points during running the algorithm was significantly decreased compared to the results given in Fig. 1. Furthermore, the enhanced algorithm, in contrast to the original one, successfully fitted the contour to the Flower boundary and avoided the fractal behavior. However, it still suffers from two important drawbacks: short capture range and high computational complexity.

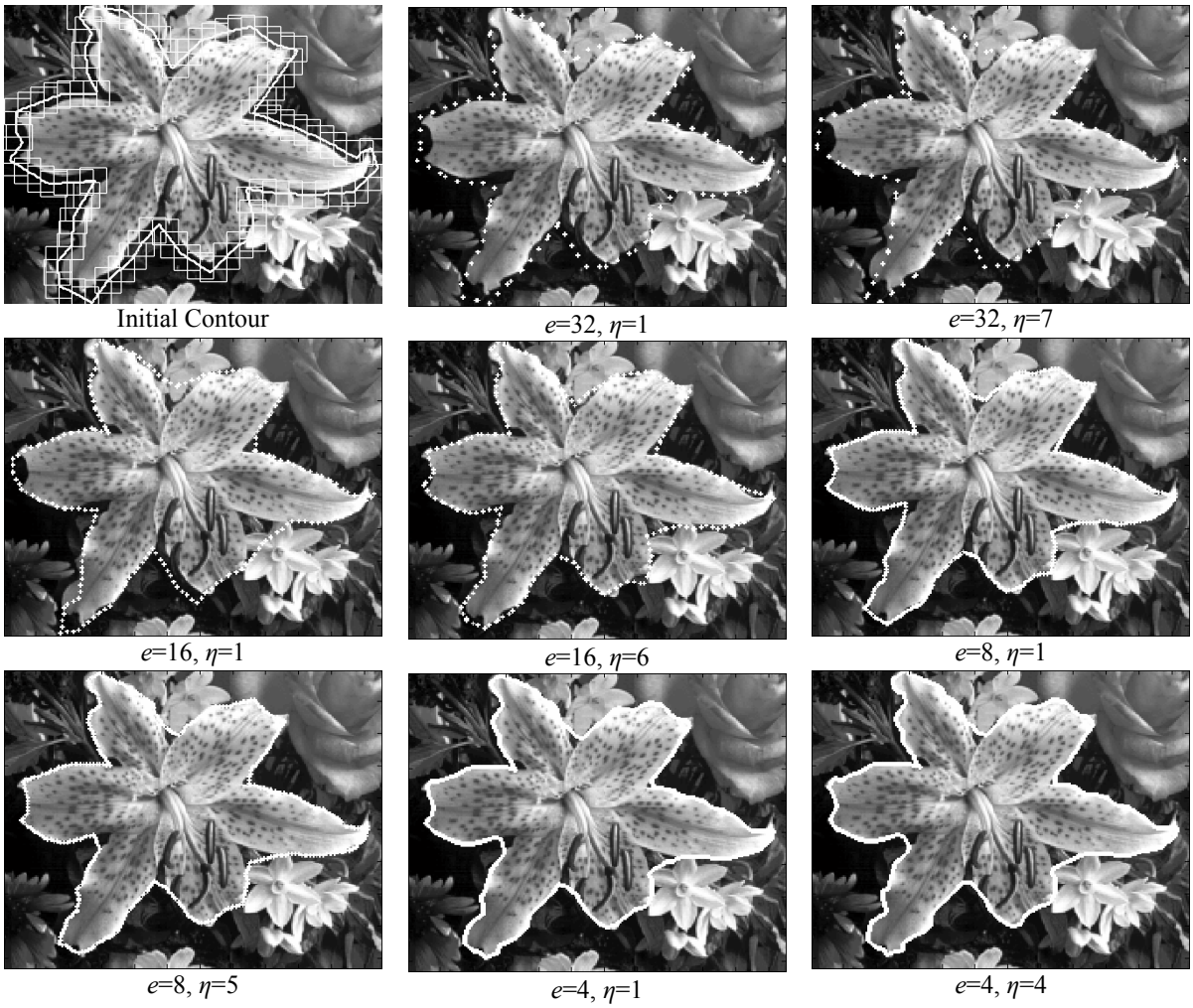


Fig. 3. The contour deformations of the enhanced Ida's algorithm for the Flower image with $e_{\max}$=32, $e_{\min}$=4 and $L_{e_{\max}} = 110$.