

# Image Matting for Sparse User Input by Iterative Refinement

Stephen Tierney, Geoff Bull, Junbin Gao

School of Computing and Mathematics, Charles Sturt University

Bathurst, NSW 2795, Australia

{stierney, gbull, jbgao}@csu.edu.au

**Abstract**—Image matting is the process of extracting the foreground component from an image. Since matting is an under constrained problem most techniques address the case where users supply some dense labelling to indicate known foreground and background regions. In contrast to other techniques our proposed technique is unique in that focuses on achieving satisfactory results with extremely sparse input, e.g. a handful of individual pixel labels. We propose an iterative extension to the class of affinity matting techniques. Analysis of results from affinity matting with sparse labels reveals that the low quality alpha mattes can be processed and re-used for the next iteration. We demonstrate this extension using the recent KNN matting and show that this technique can greatly improve matting results.

**Keywords**—Alpha, Matting, Iterative, Sparse, User, Input.

## I. INTRODUCTION

Image matting is the process of extracting the foreground component from an image. This task is typically performed in image and video editing and computer vision and it has been studied extensively. Most commonly the image capture process is conducted under controlled environments against a constant-coloured background known as “blue or green screen matting”. This controlled capture environment enables very fast and accurate algorithms to be used. In this work we are concerned with the more general case of natural image matting, meaning matting with images that are captured in uncontrolled environments.

The complex nature of the problem has encouraged research and the development of a wide array of techniques and solutions. Most techniques rely on the compositing equation [1]

$$I_i = \alpha_i F_i + (1 - \alpha_i) B_i \quad (1)$$

where  $I_i$ ,  $\alpha_i$ ,  $F_i$  and  $B_i$  are the pixel colour, alpha value, foreground colour and background colour (RGB) respectively at pixel  $i$ . Porter and Duff [1] introduced the alpha channel to control the blending of foreground and background in a compositing scenario. We call the  $\alpha$  values over the entire image the “alpha matte”. The alpha matte values lie in the range  $[0, 1]$ , with the extremes representing background and foreground respectively.

This is a severely under constrained problem, at each pixel the  $\alpha$ ,  $F$  and  $B$  are unknown. Per-pixel there are seven unknown variables that need to be solved from three known (RGB) values. To alleviate this problem user input techniques have been developed to provide extra information. By labelling

regions of the image as foreground or background the computational load is reduced and the extra information can be used to calculate the unknown regions more easily. The two common input methods are the trimap and scribbles. A trimap is an image of the same dimensions as the original that accurately labels known foreground and background regions with only a small unknown strip between each. Similarly scribbles indicate the definitely foreground and definitely background regions by scribbles over the image but on a much coarser scale [2].

The trimap and scribble methods are time consuming for a user to produce, instead we wish to allow extremely sparse input in the form of a small number of single mouse clicks. Recently Chen et al. [3] proposed such a technique by naively labelling a small rectangular area around each click location. This scheme can lead to problems, which we discuss later in section IV. A visual comparison of the trimap, scribble and click input methods are shown in Figure 1.

We believe that the constraints of extremely sparse input are essential to the progress of development in this field and are of great interest to the matting community. If a matting technique is to be widely adopted it must be easy and fast to use, which means that less input should be required of the user. To this end we propose a new technique that accepts sparse input, is not subject to the problems of other existing attempts and produces accurate alpha mattes.

## II. RELATED WORK

Due to image matting being such an under constrained problem it has generated a lot of interest and potential solutions. We refer readers to <http://alphamatting.com> [4] for a comprehensive listing of matting techniques. This website has been created for quantitative evaluation of matting techniques and provides the benchmark dataset in the field.

Most techniques fall into two categories: Colour sampling or Affinity matting. Colour sampling techniques involve statistical sampling or nearby known pixels estimate the likelihood that a pixel is either foreground or background. Colour sampling methods are performed pixel by pixel, propagating into the unknown region from the edges of the labelled foreground and background region. Some popular examples are Bayesian matting [5], Poisson Matting [6]. These methods typically fail with large unknown regions. A group of iterative colour sampling methods have been developed to more robustly estimate the alpha matte. The widely used examples of these methods are [7] and [8].

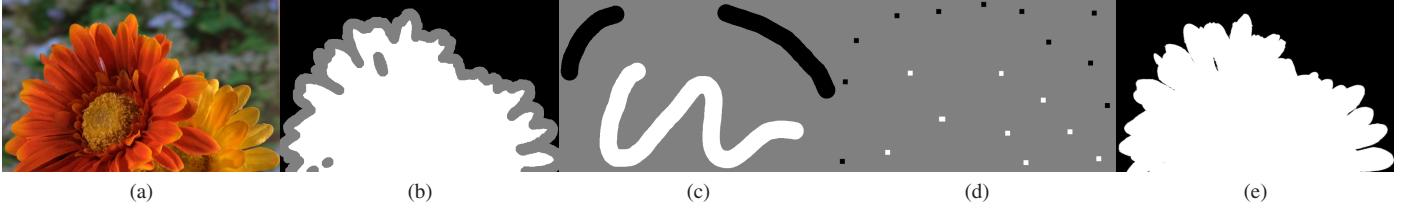


Fig. 1: The matting process: (a) An image to perform matting on, (b) User supplied trimap where black indicates known background, grey the unknown region and white the known foreground, (c) the Scribble alternative to a trimap with white for known foreground and black for known background, (d) mouse clicks (enlarged for visibility) with white for known foreground and black for known background and (e) a resulting alpha matte after matting is performed.

Affinity matting techniques are distinguished by the use of “affinity” or matting laplacian matrices and using this matrix in finding a global minimum of an objective function. The technique that popularised this method is Closed Form matting by Levin et al. [2]. Affinity matting first builds the laplacian matrix which represents a graph connecting or not connecting pixels in the image by some learnt affinity values or weights. Then in one step a global solution is found by solving a system of linear equations. Affinity matting is preferred for its simple and fast implementation. Readers are referred to other examples of affinity matting such as [9], [10], [11], [12], [13], [14]. Recently the KNN matting technique [3] has been shown to produce competitive alpha mattes much faster than existing techniques. We provide more details on affinity matting, particularly KNN matting, in section III.

Our contribution is an iterative extension to the class of affinity matting techniques that aims to improve results when provided with sparse input, a constraint that has seen little relevant work. In contrast to iterative propagation colour sampling techniques our method calculates the entire alpha matte at each step, using the low quality alpha mattes as a basis for input to the next iteration. We use the KNN matting technique to demonstrate our method because of it’s speed and simplicity of implementation. Most affinity matting techniques can be substituted without loss of generality.

### III. AFFINITY MATTING

The affinity matting formulation (developed by Levin et al. [2] and used by others such as KNN matting) can be written as the following quadratic equation

$$\min_{\alpha} f(\alpha) = \alpha^T L \alpha + \lambda (\alpha - m)^T D_m (\alpha - m) \quad (2)$$

where  $\lambda$  is a scalar that indicates how confident the user is with their input and thus how strictly the resulting matte obeys the input,  $m$  is a vector containing the specified alpha values and  $D_m$  is a binary diagonal matrix whose diagonal elements are one for labelled pixels and zero for all others.  $L$  is an affinity laplacian such that

$$L_{ij} = \begin{cases} \sum_{j=1}^n w_{ij} & \text{if } i = j, \\ -w_{ij} & \text{if pixels } i \text{ and } j \text{ are neighbours,} \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

where  $w_{ij}$  is the assigned weight between pixels  $i$  and  $j$ . Since the objective function is quadratic in alpha we find the

global minimum by differentiating w.r.t alpha and setting the derivative to zero as follows

$$(L + \lambda D_m)\alpha = \lambda m \quad (4)$$

The main point of difference among affinity matting techniques is the choice of an appropriate affinity matrix. Closed Form matting calculates affinities in a small local patch around each pixel, under the assumption that only pixels in a local window will share appropriate similarity. In contrast KNN matting removes this assumption and calculates affinities globally over the image, selecting correlated pixels via k-nearest neighbours. In practice the majority of selected neighbours will be in a local area around each pixel but strongly correlated pixels that are further away will also be connected in the affinity matrix. KNN matting calculates the weights between neighbours  $i$  and  $j$  as

$$w_{ij} = 1 - \frac{\|X(i) - X(j)\|}{C} \quad (5)$$

where  $X$  is the matrix of pixel values and  $C$  is the least upper bound of  $\|X(i) - X(j)\|$  to normalise  $w_{ij}$  to the range  $[0, 1]$ .

### IV. MOTIVATION FOR ITERATIVE REFINEMENT

While KNN matting has proven itself to be both fast and competitive with existing algorithms, it employs a trick to allow input in the form of sparse pixel clicks. Each individual user click is interpreted as the centre of an  $w \times w$  window. In effect each click has dimensions  $w \times w$  rather than a single pixel. This propagates the single user clicks into a larger area to increase alpha matte accuracy by providing more labelling information. A comparison of the effect of window sizes on the alpha matte is presented in Figure 2.

Naively labelling surrounding pixels as performed in KNN matting can result in mislabelling. The window size may need to be varied between images of different resolution or even within an image itself. For example using large windows may be useful for working on images with a large resolution but the same sized windows may cross foreground and background borders on smaller images. Additionally if the image has small patches of foreground against a large background (e.g. tree leaves against sky) then the optimum window size is different for foreground and background. Considering these disadvantages we wish to eliminate the window size parameter from the KNN matting process.

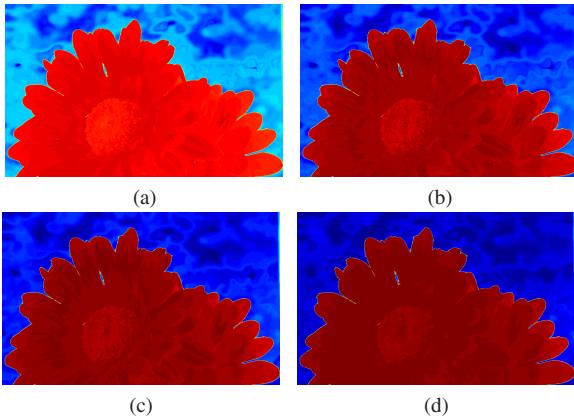


Fig. 2: Comparison of the effect of different window sizes (amount of labelling) in increasing order. (A)  $10 \times 10$ , (B)  $5 \times 5$ , (C)  $3 \times 3$  and (D)  $1 \times 1$ . As the window size decreases the unknown area increases and reduces accuracy. Greyscale image colorised to reveal inconsistencies more clearly.

We propose to remove the reliance on the window size parameter by iteratively performing affinity matting, at each iteration we use the previously calculated alpha matte as the basis for input to the next iteration. This approach is based on analysis of results from the original KNN matting and study of its implementation. We are making a mostly intuitive assumption based on observation and as such we do not provide a mathematical basis or derivation.

To demonstrate our method we will first study the output of KNN matting with sparse input. Figure 3 compares the alpha matte produced by KNN matting with sparse input to the ground truth. While the foreground and background regions are visually separable it can be seen by comparison with the ground truth that we are aiming for a more binary result. Additionally there is large variation in the contiguous regions of foreground and background which should not exist. A histogram of the alpha matte in Figure 3 reveals two distinct foreground and background clusters and a much smaller valley containing unknown values in between. From visual inspection we can see that it is safe to treat the spikes as definitely foreground and background. We now need only drive the alpha matte towards its more binary solution. We propose segmentation of the alpha matte and using the foreground and background clusters as labels and performing the matting process again. This will be repeated until a satisfactory solution is found, we discuss stopping criteria in our implementation discussion.

With this technique we permit the user to label individual pixels, which removes any boundary conflicts with using windows around each click location and removes the window size parameter. A step by step illustration of the technique is provided in Figure 7.

## V. IMPLEMENTATION

After a rough alpha matte has been produced from sparse input the idea is to transform it into an accurate trimap to be used again. We segment the rough alpha matte into three regions: foreground, background and unknown. We propose

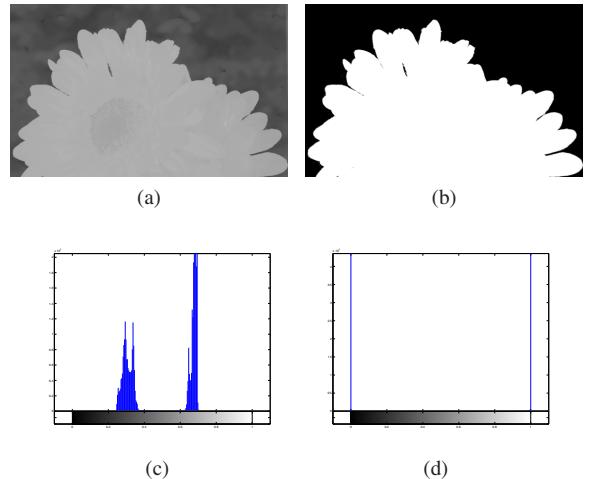


Fig. 3: (a) Alpha matte from KNN matting with sparse input, (b) the ground truth, (c) histogram of the alpha matte from sparse input and (d) the histogram of the ground truth. The foreground and background regions are visually distinct but not as binary as the ground truth which we are aiming for. The histogram (c) reveals numerically distinct foreground and background clusters.

that this task is performed by k-means clustering [15] because of its speed, although other techniques can be substituted. We treat the cluster with the highest mean as foreground values (closest to 1) and likewise the cluster with the lowest mean as the background (closest to 0). We then use the segmented regions as input for the next iteration and repeat the process until a satisfactory alpha matte has been achieved. The algorithm for our iterative refinement technique is shown in Algorithm 1 and a step by step illustration of the technique is provided in Figure 7.

In many cases there can be significant misclassification due to either a numerically inseparable alpha matte after the first iteration or instability with k-means. To alleviate this issue we segment the image, then for each pixel label we apply the label to the entirety of the matching segment. We use the fast segmentation method Statistical Region Merging (SRM) [16] in our implementation. We suggest tuning the segmentation technique so that segments are small and highly localised. This pre-segmentation is not required for all images but can drastically improve results where misclassification occurs.

Our algorithm requires a stopping point to be defined. In our experiments we defined two stopping points: a maximum number of iterations and a percentage pixel coverage of the image. These measures also ensure that the k-means clustering step does not fail to segment the alpha mattes into three clusters.

## VI. RESULTS AND DISCUSSION

Experiments were performed using the 27 low resolution training images from the benchmark dataset <http://alphamatting.com> [4] dataset. Ground truth alpha mattes are provided by the dataset. Parameters were fixed across all experiments with matting regularisation parameter  $\lambda = 1$ , segmentation scale parameter  $Q = 300$  and a maximum

---

**Algorithm 1** Iterative KNN matting

---

**Require:**  $I$  - image,  $\lambda$  - regularisation parameter

- 1: Collect user labels  $m$
- 2: Segment  $I$
- 3: Apply labelling to segments and update  $m$
- 4: Initialise diagonal matrix  $D_m$
- 5: Calculate Laplacian matrix  $L$
- 6: **for**  $j = 1 \rightarrow \text{max iterations}$  **do**
- 7:   **if**  $j > 1$  **then**
- 8:     Segment  $\alpha$  into 3 clusters
- 9:     Map clusters to background and foreground input
- 10:   Update user input  $m$  and  $D_m$
- 11:   **end if**
- 12:   Solve  $(L + \lambda D_m)\alpha = \lambda m$
- 13: **end for**

---

	Max	Min	Mean	Median
Trimap	94.51	60.32	84.31	88.68
Clicks	0.099	0.006	0.027	0.016

TABLE I: Comparison of image covered in percent by the trimaps given from the dataset and our single mouse clicks. We show that even with such small areas of the image manually labelled we can produce competitive alpha mattes.

number of iterations of 5. We provide results for our method with and without pre-segmentation and refer to each as IS-KNN and I-KNN respectively. We compare our method to KNN matting using sparse labelling and trimap and show that our method using sparse labelling is competitive with trimap labelling.

Sparse input for each image was generated by hand. The number of labelled pixels depended on the size and complexity of each individual image and varied considerably. Table I provides a comparison of the image coverage between trimaps and our sparse labelling.

For quantitative comparison we use the MSE (Mean Square Error) and the SAD (Sum of Absolute Differences), which are the standard measurements in the field of image matting [4]. Tables II and III provide some statistics for both MSE and SAD. From the tables we can see that both I-KNN and IS-KNN greatly outperform KNN in most cases when using sparse labelling. We can also see that IS-KNN using sparse labelling is competitive with KNN matting using trimaps. From Figure 4 and 5 we can see that I-KNN outperforms KNN with sparse labelling in 20 out of 27 images when comparing MSE and 26 out of 27 images when comparing SAD. Additionally we see that IS-KNN with sparse labelling always outperforms KNN with and also outperforms KNN using a trimap for image 16.

In Figures 8 and 6 we provide visual comparisons of results. In Figure 8 I-KNN and IS-KNN offer considerable improvement in the quality of the alpha matte. Errors can occur with I-KNN when k-means incorrectly identifies foreground and background components but these errors are reduced or eliminated completely with IS-KNN. Figure 6 demonstrates that IS-KNN can produce superior results when using sparse labelling to KNN using a trimap.

	Sparse Label			Trimap
	KNN	I-KNN	IS-KNN	
Max	0.38	0.42	0.11	0.04
Min	0.11	0.004	0.0004	0.0002
Mean	0.20	0.13	0.014	0.008
Median	0.20	0.09	0.005	0.003

TABLE II: Mean Square Error (MSE) statistics comparing KNN, I-KNN and IS-KNN for sparse labelling and KNN for trimap labelling. IS-KNN is competitive to KNN with trimap labelling.

	Sparse Label			Trimap
	KNN	I-KNN	IS-KNN	
Max	3.03	2.25	0.86	0.47
Min	1.46	0.07	0.01	0.01
Mean	2.06	0.78	0.17	0.13
Median	2.04	0.68	0.09	0.08

TABLE III: Sum of Absolute Differences (SAD) statistics comparing KNN, I-KNN and IS-KNN for sparse labelling and KNN for trimap labelling. All values are coefficients of the order  $10^5$ . IS-KNN is competitive to KNN with trimap labelling.

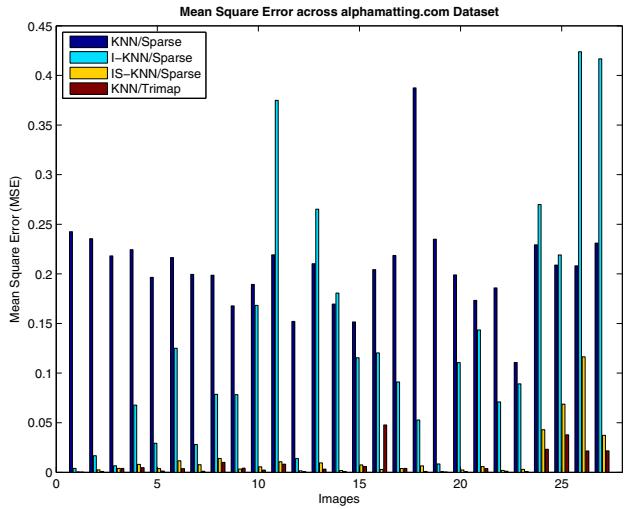


Fig. 4: Mean Square Error (MSE) results. I-KNN outperforms KNN with sparse labelling in 20 out of 27 images and IS-KNN always outperforms KNN with sparse labels. IS-KNN with sparse labelling outperforms KNN with trimap for image 16.

## VII. CONCLUSION

We have presented an iterative extension to affinity matting and shown that it can greatly improve results with sparse input. Additionally we have shown through experimentation that our method when given sparse input can, in some cases, outperform methods using the trimap. This method is general and could be combined with future affinity matting methods.

## VIII. ACKNOWLEDGEMENT

Junbin Gao's work is supported by the Australian Research Council (ARC) through the grant DP130100364. Images used in this paper are from <http://alphamatting.com> [4].

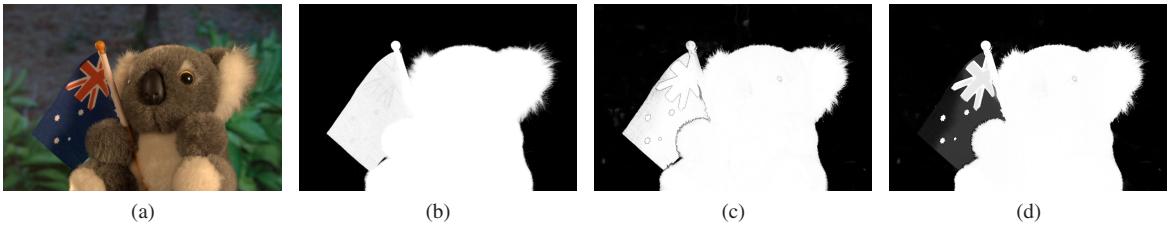


Fig. 6: In some cases IS-KNN using sparse labelling can produce superior results than KNN using a trimap. (a) is the original image, (b) is the corresponding ground truth, (c) IS-KNN results with sparse labels and (d) KNN results with trimap. KNN using the trimap mislabels the transparent flag as mostly background rather than mostly foreground.

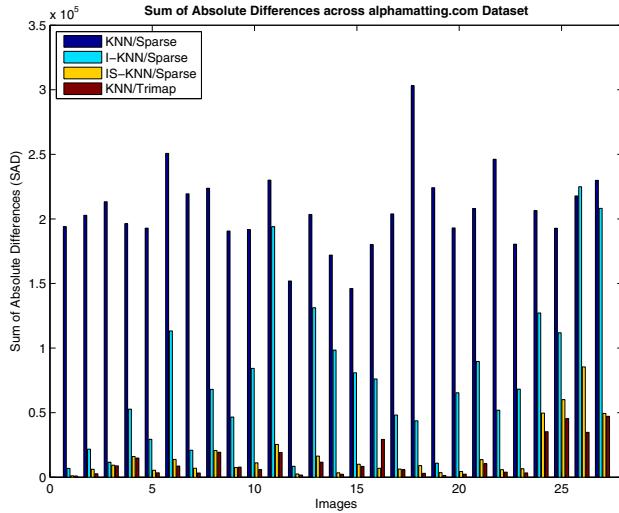


Fig. 5: Sum of Absolute Differences (SAD) results. I-KNN outperforms KNN with sparse labelling in 26 out of 27 images. IS-KNN always outperforms KNN with sparse labels. IS-KNN with sparse labelling outperforms KNN with trimap for image 16.

for unified image segmentation and matting,” in *Proceedings of the Tenth IEEE International Conference on Computer Vision*. 2005, vol. 2, pp. 936–943, IEEE.

- [9] L. Grady, T. Schiwiitz, S. Aharon, and R. Westermann, “Random walks for interactive alpha-matting,” in *Proceedings of VIIP*, 2005, pp. 423–429.
- [10] Junbin Gao, “Image matting via local tangent space alignment,” *International Conference on Digital Image Computing: Techniques and Applications*, pp. 614–619, 2011.
- [11] Kaiming He, Jian Sun, and Xiaou Tang, “Fast matting using large kernel matting Laplacian matrices,” in *Proceedings of CVPR 2010*, 2010, pp. 2165–2172.
- [12] Kwang Hee Won, Soon-Yong Park, and Soon Ki Jung, “Natural image matting based on neighbor embedding,” in *Lecture Notes in Computer Sciences*, A. Gagalowicz and W. Philips, Eds., vol. 4418, pp. 449–460. Springer, 2007.
- [13] P. Lee and Ying Wu, “Nonlocal matting,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 2193 – 2200.
- [14] Junbin Gao, Manoranjan Paul, and Jun Liu, “The image matting method with regularized matte,” in *Proceedings of ICME*, 2012.
- [15] James MacQueen et al., “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. California, USA, 1967, vol. 1, p. 14.
- [16] Richard Nock and Frank Nielsen, “Statistical region merging,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 11, pp. 1452–1458, 2004.

## REFERENCES

- [1] T. Porter and T. Duff, “Compositing digital images,” *ACM SIGGRAPH Computer Graphics*, vol. 18, no. 3, pp. 253–259, July 1984.
- [2] Anat Levin, Dani Lischinski, and Yair Weiss, “A closed-form solution to natural image matting,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 228 – 242, February 2008.
- [3] Q. Chen, D. Li, and C.K. Tang, “Knn matting,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 869–876.
- [4] Christoph Rhemann, Carsten Rother, Jue Wang, Margrit Gelautz, Pushmeet Kohli, and Pamela Rott, “A perceptually motivated online benchmark for image matting,” *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.
- [5] Y.Y. Chuang, B. Curless, D.H. Salesin, and R. Szeliski, “A bayesian approach to digital matting,” in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. IEEE, 2001, vol. 2, pp. II–264.
- [6] Jian Sun, Jiaya Jia, Chi-Keung Tang, and Heung-Yeung Shum, “Poisson matting,” *ACM Transactions on Graphics (TOG)*, vol. 23, pp. 315–321, Aug. 2004.
- [7] H. Chang, Q. Yang, and C. Pan, “An iterative bayesian approach for digital matting,” in *18th International Conference on Pattern Recognition*. IEEE, 2006, vol. 2, pp. 122–125.
- [8] Jue Wang and Michael F. Cohen, “An iterative optimization approach

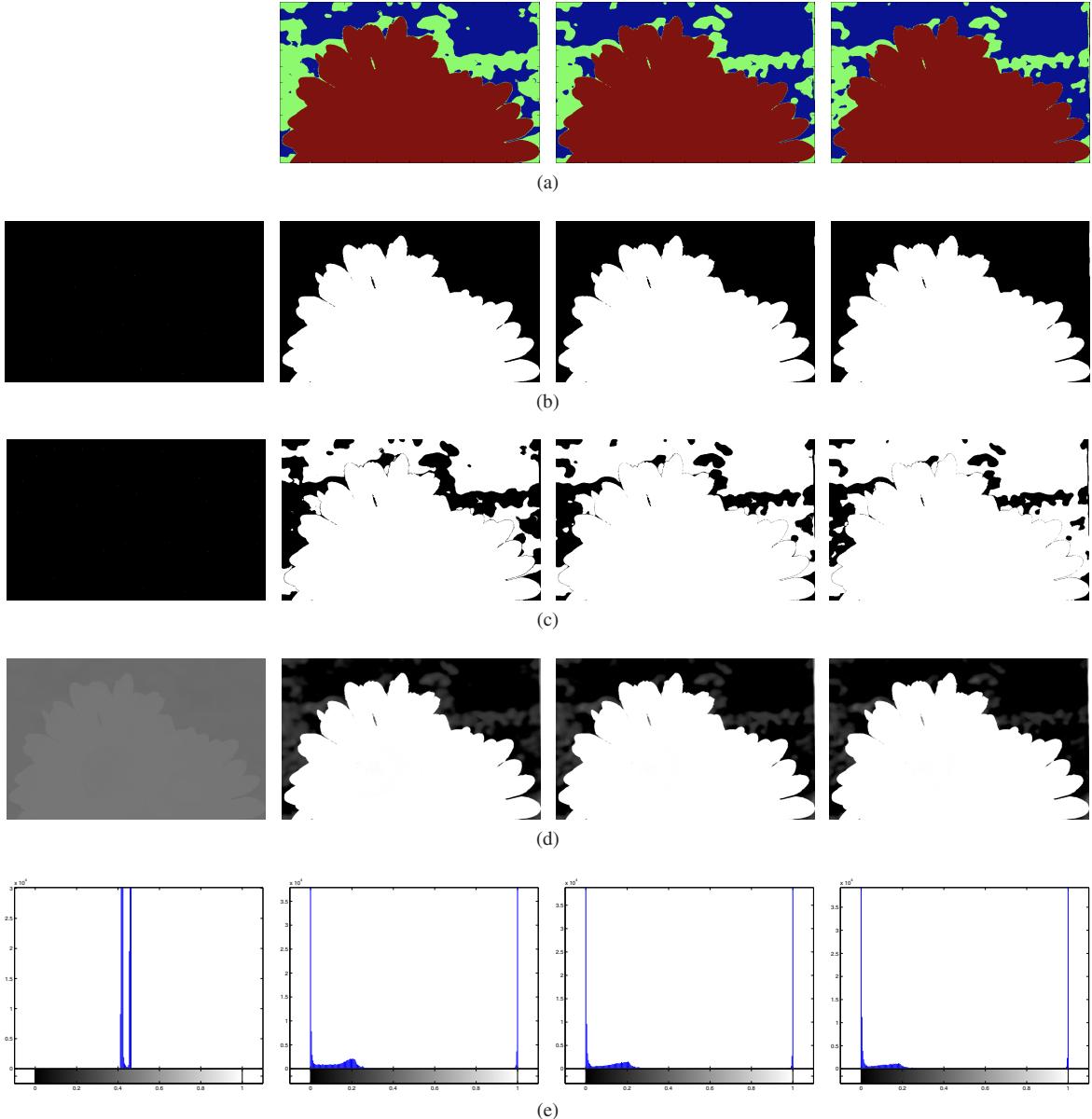


Fig. 7: The iterative matting process for iterations 1 to 4, each column indicates a single iteration. Row (a) shows the segmented alpha matte of the previous iteration. Row (b) shows values treated as the foreground input. Row (c) shows the values treated as total user input. Row (d) shows the alpha value calculated at each iteration. Row (e) shows a histogram of the alpha matte at each iteration. For example the values from the left image of row (d) are segmented as shown in second image of row (a) then transformed into input in rows (b) and (c) which is used to calculate the alpha matte in row(d).



Fig. 8: Visual comparison of results of KNN, I-KNN and IS-KNN for sparse labelling. Column (a) is the original image, (b) is the corresponding ground truth, (c) KNN matting results, (d) I-KNN results and (e) IS-KNN results. I-KNN offers significant improvements when using sparse labelling. Some misclassification can occur when using I-KNN, but these errors are reduced by IS-KNN.