# AUTOMATED SEPARATION OF REFLECTIONS FROM A SINGLE IMAGE BASED ON EDGE CLASSIFICATION

Kenji Hara, Kohei Inoue and Kiichi Urahama
*Department of Visual Communication Design, Kyushu University, Fukuoka, Japan*

Keywords: Reflection separation, transparency, edge classification, pyramid structure, deterministic annealing.

Abstract: Looking through a window, the object behind the window is often disturbed by a reflection of another object. In the paper, we present a new method for separating reflections from a single image. Most existing techniques require the programmer to create an image database or require the user to manually provide the position and layer information of feature points in the input image, and thus suffer from being extremely laborious. Our method is realized by classifying edges in the input image based on the belonging layer and formalizing the problem of decomposing the single image into two layer images as an optimization problem easier to solve based on this classification, and then solving this optimization with a pyramid structure and deterministic annealing. As a result, we are able to accomplish almost fully automated separation of reflections from a single image.

## 1 INTRODUCTION

One can see that the scene observed through a flat transparent plate such as window glass normally consists of a combination of two images: the reflected image and the transmitted image. In recent years, decomposing the superimposed image into the two images has been a topic of significant interest in the computer vision community because of both its practical importance and its theoretical difficulty.

The great difficulty of the image decomposition problem lies in its high illposedness, since the number of unknowns (twice the number of pixels) is much larger than the number of available constraints (the number of pixels) (Levin et al., 2004a). Thus, much of the image decomposition methods enforce additional constraints such as using as inputs two images taken through a polarizer at different orientations or using an image sequence taken from a video camera (Farid and Adelson, 1999; Irani and Peleg, 1992; Sarel and Irani, 2004b; Sarel and Irani, 2004a; Szeliski et al., 2000; Tsin et al., 2003; Schechner et al., 2000)D

More recently, Levin et al. (Levin et al., 2004a) developed a method for decomposing the input image into two images from a single image. This method is based on a prior that prefers decompositions that minimize a linear combination of the numbers of edges and corners. In this work, they showed that the single image decomposition problem can be solved even without a priori knowledge about content or scene contents. Also at the same period, they proposed a user-guided semi-automatic method, which work well even on complex images that may be difficult for the above method to decompose (Levin et al., 2004b).

These previous works, however, require the programmer to create a database of natural images or require the user to manually provide the position and layer information of feature points in the input image. Hence, they suffer from the problems that the manual data entry tasks are extremely laborious and tedious and that the separation results may depend on the set of images in the database.

In this paper, we present a single image decomposition method that requires less labor, such as image database construction or a considerable amount of user interaction. In order to reduce the ambiguity of the solution, we make the following assumptions within Levin et al's prior framework: (1) both of the layers have edges, (2) the edge of a layer does not

Hara K., Inoue K. and Urahama K. (2007).
AUTOMATED SEPARATION OF REFLECTIONS FROM A SINGLE IMAGE BASED ON EDGE CLASSIFICATION.
In *Proceedings of the Second International Conference on Computer Vision Theory and Applications - IFP/IA*, pages 18-25

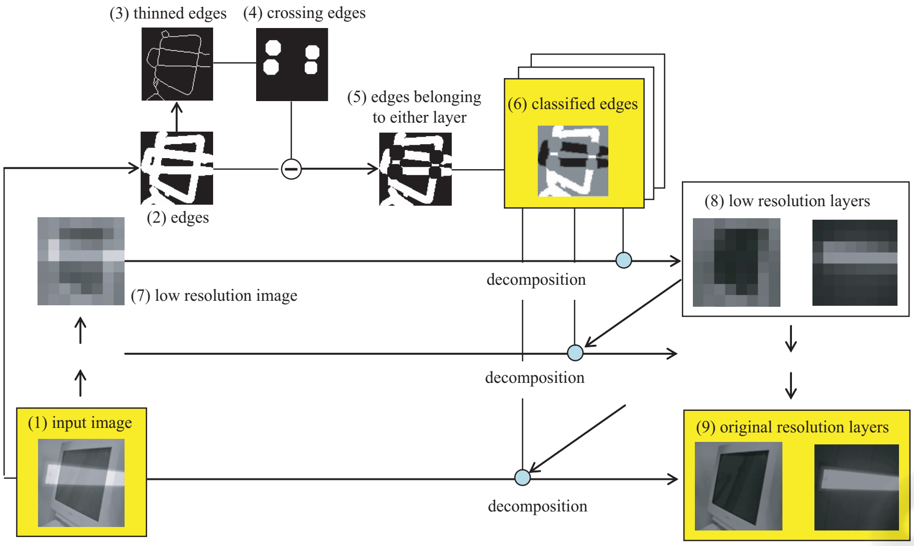


Figure 1: Pipeline of reflection separation.

overlap with any edge of another layer, and (3) the junction edge in the superposition image arises from the crossing between an edge of a layer and an edge of another layer.

We first identify each edge pixel in the input image to which layer it belongs using only low level image features. We then formulate the problem of decomposing the single image into two layer images as an optimization problem easier to solve based on this classification. We show that this optimization can be approximately solved using a pyramid structure and deterministic annealing (Geiger and Girosi, 1991; Urahama and Nagao, 1995; Ueda and Nakano, 1998).

## 2 PROBLEM FORMULATION

Recently, Levin et al. proposed an optimization framework for separating reflections from a single image (Levin et al., 2004a). They formulated the problem of decomposing the single superimposed image $I = [I(x,y)] \in \mathbb{R}^{\mathbb{D}}$ into two layers $I_1 = [I_1(x,y)] \in \mathbb{R}^{\mathbb{D}}$ and $I_2 = [I_2(x,y)] = [I(x,y) - I_1(x,y)] \in \mathbb{R}^{\mathbb{D}}$, where $\mathbb{D}$ is the input image domain, as minimizing the cost function:

$$cost_1(I_1, I_2) = cost_1(I_1) + cost_1(I_2), \tag{1}$$

where $cost_1(I_1)$ is the total amount of edges and corners of $I_1$:

$$cost_1(I_1) = \sum_{(x,y)\in\mathbb{D}} |\nabla I_1(x,y)|^{\alpha} + \eta\, c(x,y;I_1)^{\beta}, \tag{2}$$

where $\nabla$ is the gradient edge operator and $c(\cdot)$ is the Harris-like corner operator. Also, $\alpha$, $\beta$, and $\eta$ are set to 0.7, 0.25, and 15, respectively, which are obtained from the histgrams of the operators in natural images (Levin et al., 2002). They provided solutions for the problem by dividing the image into overlapping patches and then selecting the optimal set of patches from a database of natural image patches.

In the approach mentioned above, the problem of separating reflections using a single image without such training data or user intervention have not been tackled. In this section, to address this difficulty, we will provide an optimization framework.

### 2.1 Automatic Classification of Edges

In our work, we first estimate to which layer each edge pixel in the input superimposed image belongs according to the following procedure.

1. **Detect edges**
   We detect edges in the input image with the SUSAN edge detector (Smith and Brady, 1997) (see also Figure 1 (1))

$$n(x_0,y_0;I,t) = \sum_{(x,y)\in\mathbb{D}} c(x,y,x_0,y_0;I,t)\ , \tag{3}$$

$$c(x,y,x_0,y_0;I,t) = 1 - \exp\left[-\left(\frac{I(x,y) - I(x_0,y_0)}{t}\right)^6\right] \tag{4}$$

   where $I(x_0,y_0)$ is the intensity value at the pixel position $(x_0,y_0)$ of the input image $I$ and $t$ is the edge difference threshold parameter. We hereafter denote by $E$ the set of edge pixels.

2. **Extract edges belonging to either layer**
   We obtain the set of connected regions, $C = \{C_i | 1 \leq i \leq m\}$, by first reducing edges to unit thickness while maintaining its topology (Figure 1(2)) and then performing morphological dilation of each pixel lying on the intersection of the two lines with a circular structuring element of appropriate size (Figure 1(3)). A region which is $C$ subtracted from the region $E$ can be regarded as belonging to either layer and can be segmented into regions $\omega = \{\omega_j | 1 \leq j \leq n\}$, where each $\omega_j$ is a connected region and $\omega_j \cap \omega_k = \phi$ for $j \neq k$ and $n$ is the number of $\omega_j$ (Figure 1(4) and Figure 2(a)).

3. **Identify for each edge to which layer it belongs**
   While tracking around the pixels in the outer boundary of each $C_i$, we sequentially assign a label to each pixel indicating to which $\omega_j$ it belongs (no label is assigned to the pixel belonging to none of $\omega_j$) (Figure 2(b)). In this process, four kinds of labels $j_1, j_2, j_3, j_4$ will appear (Figure 2(c)). Then, we can impose the constraint that two regions $\omega_{j_1}$ and $\omega_{j_3}$ (also, $\omega_{j_2}$ and $\omega_{j_4}$) belong to the same layer. Hence, we can classify each edge region to which layer it belongs by solving a set of the constraints for all $C_i \in C$ (Figure 1(5)). We denote by $\Omega_1$, $\Omega_2$, and $\Omega_0$ the edge region of the first layer, the edge region of the second layer, and the other region, respectively (Figure 2(d)).

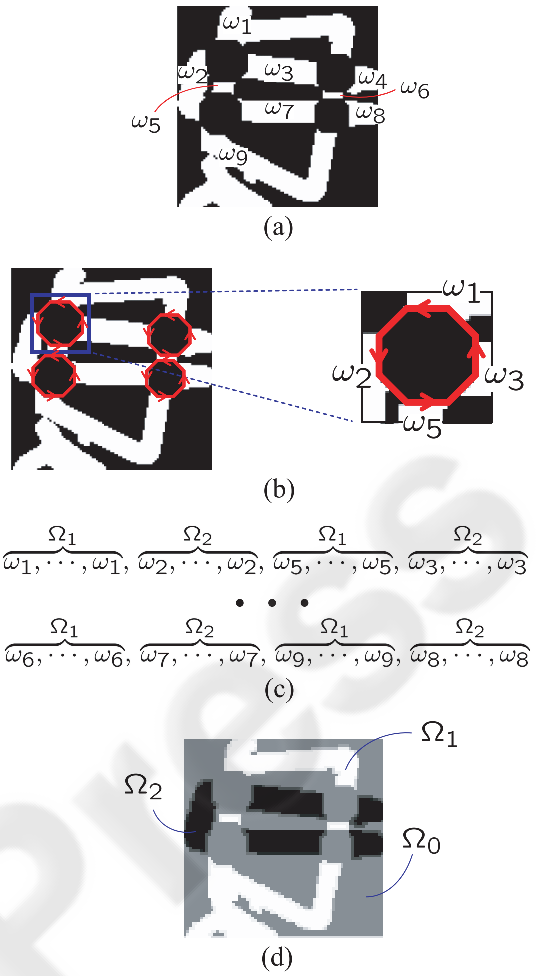
## 2.2 Definition of Objective Function

We formulate the problem of separating reflections from a single image. To make the presentation easy to read without loss of generality, we will describe $I_1$ and $I_2$ as $I_1 = [I_1(x,y)] = [\alpha(x,y)I(x,y)] \in \mathbb{R}^{\mathbb{D}}$ and $I_2 = [I_2(x,y)] = [(1-\alpha(x,y))I(x,y)] \in \mathbb{R}^{\mathbb{D}}$, respectively, by introducing the set of unknown coefficients, $\alpha = [\alpha(x,y)] \in [0,1]^{\mathbb{D}}$. Then, the task of separating two transparent layers becomes solving the minimization problem as follows.

$$\min_{\alpha} f(\alpha; I, t) \equiv \min_{\alpha} \sum_{(x,y)\in\mathbb{D}} w_1(x,y)\, n(x,y;\alpha * I, t) + w_2(x,y)\, n(x,y;(E-\alpha) * I, t), \quad (5)$$

where $\alpha * I = [\alpha(x,y)I(x,y)] \in \mathbb{R}^{\mathbb{D}}$, $(E-\alpha) * I = [(1-\alpha(x,y))I(x,y)] \in \mathbb{R}^{\mathbb{D}}$, $n(x,y;I,t)$ is the function defined by Equation (4), $w_1 = [w_1(x,y)] \in \mathbb{R}^{\mathbb{D}}$ and $w_2 = [w_2(x,y)] \in \mathbb{R}^{\mathbb{D}}$ are the variable weighting functions as

$$w_1(x,y) = \begin{cases} w_{normal} & (x,y) \in \Omega_0 \\ w_{small} & (x,y) \in \Omega_1 \\ w_{large} & (x,y) \in \Omega_2 \,, \end{cases} \quad (6)$$



$\omega_1$
$\omega_2$
$\omega_3$
$\omega_4$
$\omega_6$
$\omega_5$
$\omega_7$
$\omega_8$
$\omega_9$
(a)
$\omega_1$
$\omega_2$
$\omega_3$
$\omega_5$
(b)
$\underbrace{\omega_1,\cdots,\omega_1}_{\Omega_1}, \underbrace{\omega_2,\cdots,\omega_2}_{\Omega_2}, \underbrace{\omega_5,\cdots,\omega_5}_{\Omega_1}, \underbrace{\omega_3,\cdots,\omega_3}_{\Omega_2}$
$\cdots$
$\underbrace{\omega_6,\cdots,\omega_6}_{\Omega_1}, \underbrace{\omega_7,\cdots,\omega_7}_{\Omega_2}, \underbrace{\omega_9,\cdots,\omega_9}_{\Omega_1}, \underbrace{\omega_8,\cdots,\omega_8}_{\Omega_2}$
(c)
$\Omega_1$
$\Omega_2$
$\Omega_0$
(d)

Figure 2: Classifying each edge according to the layer it belongs to.

$$w_2(x,y) = \begin{cases} w_{normal} & (x,y) \in \Omega_0 \\ w_{large} & (x,y) \in \Omega_1 \\ w_{small} & (x,y) \in \Omega_2 \,, \end{cases} \quad (7)$$

where $w_{normal} = 1.0$, $w_{large} = 10.0Cw_{small} = 0.1$. For a relatively small region of $\Omega_1 \cup \Omega_2$ (for a relatively large region of $\Omega_0$), the cost function in Equation (5) becomes multimodal and there are local minima.

## 2.3 Heuristic Minimization

Next, we consider to employ a simple local search strategy with a steepest descent heuristic to solve the nonlinear problem of solving Equation (5). This consists of setting the initial state of $\alpha$, selecting a pixel with coordinates $(x,y)$ at random, and then replacing the value of $\alpha(x,y)$ at the current time step by the neighboring solution that minimizes

Equation (5) within the neighborhood $\{\alpha(x,y) - \delta,\ \alpha(x,y),\ \alpha(x,y) + \delta\}$ ; if a solution with no improving neighbor has been reached, the above heuristic stops, and otherwise it is iterated.

Although the objective function in Equation (5) appears to be easier to minimize than the objective function in Equation (1), Equation (5) also still has high multimodality and high dimensionality, and thus the above local search often leads to local minima in the search space. Examples of the results obtained using this approach are shown in Figure 3 on a very simple synthetic image. Figure 3(a) and Figure 3(b) show the input superimposed image and the ground truth layers, respectively. Figure 3(c) shows the result obtained using the local search described in this section. Figure 3(d) shows another result where the edge classification described in Section 2.1 was not utilized. These results show that even though the edge classification can improve the performance, this complex nonlinear optimization problem in Equation (5) cannot be solved in any way directly using the simple heuristic described above. To solve Equation (5), we can apply powerful deterministic annealing techniques as we will see in the next section.

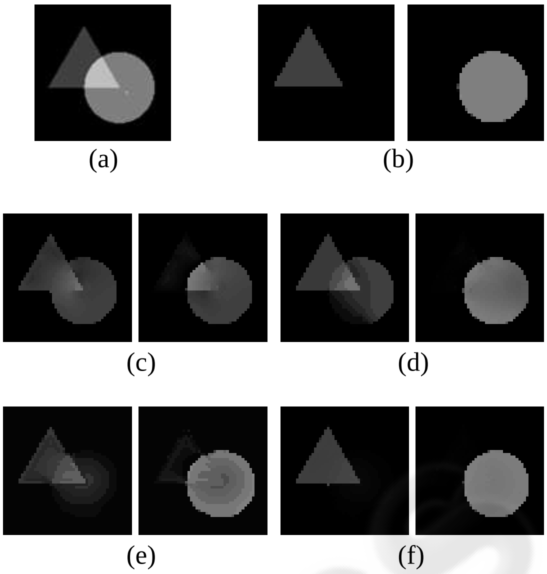

(a) (b) (c) (d) (e) (f)

Figure 3: Importance of edge classification and annealing. (a) Input image. (b) Ground truth images. (c) Without edge classification and without annealing. (d) With edge classification and without annealing. (e) Without edge classification and with annealing. (f) With edge classification and with annealing.

# 3 IMAGE DECOMPOSITION

The output of the SUSAN edge detector ranges from 0 (non-edge) to 1 (edge). In Figure 4, each horizontal bar in the graph represents the values of the SUSAN edge detector, $n(x,y;I,t)$, for a fixed $t$, with respect to variation of the gradient magnitude of pixel at position $(x,y)$. One can see that large $t$ makes the SUSAN edge detector less sensitive and thus makes $n(x,y;\alpha * I,t)$ and $n(x,y;(E-\alpha) * I,t)$ in Equation (5) more uniform over the *xy*-plane. Also, the lower the space frequency of the input image $I$ is, the more uniform $n(x,y;\alpha * I,t)$ and $n(x,y;(E-\alpha) * I,t)$ will become. In this way, the number of local minima of the objective function (5) decreases monotonically with increasing $t$ and smoothed image $I$.

Based on the above observation, in our framework, we solve Equation (5) using a pyramid structure and deterministic annealing (Geiger and Girosi, 1991; Urahama and Nagao, 1995; Ueda and Nakano, 1998). After building a multiresolution image pyramid from the input image, we start decomposing the low resolution image into two images. Until the highest resolution (original image), the solution is propagated to the next higher resolution where it is used as the initial estimate. At each resolution level, the deterministic annealing is performed by initially setting $t$ to a sufficient large value and then gradually decreasing it after each iteration, as shown in Figure 5. The whole procedure of our algorithm is described in detail below.

1. Extract edges from the input image ($I$) and classify each edge to which layer it belongs (Section 2.1).
2. Build a pyramid representation of the input image. In this case, the multiresolution image pyramid has multiple layers with the original image at the bottom and compressed (lower spatial resolution) images at the upper layers. A layer pixel has the value averaged over the corresponding next lower (higher spatial resolution) layer four pixels. In this paper, we choose the number of pyramid layers as 2.
3. Set the current layer to the top most layer and initialize the separation coefficients ($\alpha$) such that $\alpha(x,y) = 0.5$ for all $(x,y) \in \mathbb{D}$.
4. Decompose the current layer image into two images. In this case, if the current layer is the top most layer, the initial values of $\alpha$ are set as specified above, otherwise are set such that the estimated value of $\alpha(x,y)$ of the next upper layer is mapped to the corresponding four pixels of the current layer, for all $(x,y)$. Then, decompose the current layer image according to the following annealing procedure (steps (a) to (c)).

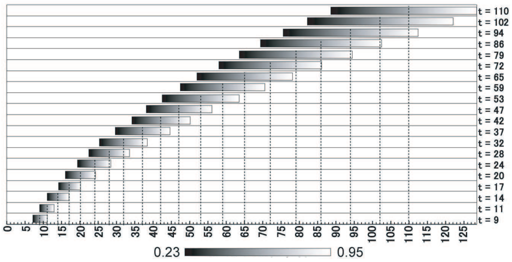


Figure 4: Edge detection sensibility with respect to variations of $t$. The horizontal and vertical axes represent the gradient magnitude and $t$, respectively.

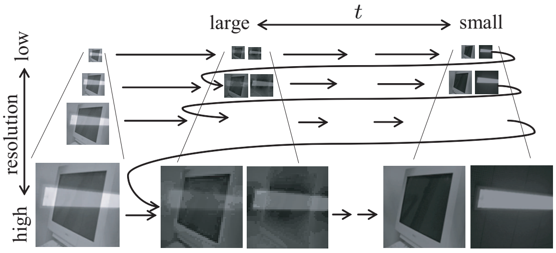


Figure 5: Reflection separation with a pyramid structure and deterministic annealing.

(a) Initially set the edge difference threshold parameter ($t$) of the SUSAN edge detector to a large value.

(b) Compute $\alpha$ under the current layer and $t$ value using the heuristic minimization (Section 2.3).

(c) Iterate between decreasing $t$ and the above step (b) until $t$ reaches its destination.

5. Iterate between moving to the next lower layer and finding the best composition under this layer (step. 4) until the bottom layer (the original image) is reached.

6. If multiple edge classifications are possible (Section 2.1), perform the image decomposition (steps 1 to 5) for each classification and then select the decomposition result which solves Equation (5) in the case where $\Omega_1 \cup \Omega_2 = \phi$, i.e., $w_1(x,y) = w_2(x,y) = w_{normal}$ for all pixel $(x,y) \in \mathbb{D}$.

Figures 3(e), (f) show the decomposition results obtained using deterministic annealing with and without edge classification, respectively. One can see that good decomposition results cannot be obtained without using edge classification. Although deterministic annealing offers no theoretical guarantee of finding the global optimum, it is well known that it can avoid many local minima (Geiger and Girosi, 1991; Urahama and Nagao, 1995; Ueda and Nakano, 1998).

## 4 RESULTS

To demonstrate and evaluate our method, we used real and synthetic images as inputs. First, we photographed a doll behind a glass window which was reflected by the glass. Figure 6 shows the input image (100x100 pixels). We extracted edges using the SUSAN edge detector (Figure 7(a)), thinned the edges (Figure 7(b)), eliminated the crossing edges (Figure 7(c)) and classified each edge to which layer it belongs (Figure 7(d)), as described in Section 2.1. Using the estimated classification of edges, we constructed an image pyramid from the input image and decomposed the top most layer, as shown in Figure 8. As shown in Figure 9, using this decomposition result as an initial guess, we decomposed the original image into two images. This appears to be a qualitatively good result. In this case, we processed each RGB channel separately.

Next, in order to evaluate the results of our method quantitatively against ground truth, we used as input the generated image by summing two already captured or synthetic images. Figures 10– 14 show the input images, corresponding ground truth decompositions, and resulting decompositions. Figures 10 and 11 show examples of using as inputs the generated images by summing two relatively simple images. One can see that quite good results were obtained. Figures 12 and 13 show examples of using as inputs the addition of simple and complex images. Despite some undesired ghost images such as the baby's face in Figure 12(c) and highlight in Figure 13(c), fairly good results seem to be obtained. Figure 14 shows an example of using as an input the generated image by summing two relatively complex images. In Figure 14(c) one can see several ghost images. The decomposition algorithm roughly took about 15 minutes on a single Pentium IV 2.4 GHz processor.

Again, for the purpose of verifying the effectiveness of the edge classification described in Section 2.1, we separated the input images in Figures 11(a), 13(a) without classifying edges, as attempted in Figure 3(e). In both of these examples, the image decomposition failed due to convergence at local minima, as shown in Figure 15(a), (b). Those examples demonstrate the significance of classifying edges for separating.

## 5 CONCLUSIONS

We proposed a novel method for decomposing a single superimposed image into two images. Despite using neither image database nor user intervention, we showed that the almost fully automatic layer extraction can be achieved using only a single input image. One of the research problems to be solved in our method is to provide robustness to errors in the edge detection and classification. We are currently studying to adapt the different error recovery schemes.

## REFERENCES

Farid, H. and Adelson, E. H. (1999). Separating reflections from images by use of independent components analysis. JOSA, vol.16, no.9, pp.2136–2145.

Geiger, D. and Girosi, F. (1991). Parallel and deterministic algorithms from mrf's: Surface reconstruction. IEEE PAMI, vol.13, no.5, pp.401–412.

Irani, M. and Peleg, S. (1992). Image sequence enhancement using multiple motions analysis. Proc. IEEE CVPR, pp.216–221.

Levin, A., Zomet, A., and Weiss, Y. (2002). Learning to perceive transparency from the statistics of natural scenes. IEEE NIPS.

Levin, A., Zomet, A., and Weiss, Y. (2004a). Separating reflections from a single image using local features. Proc. IEEE CVPR, pp.306–313.

Levin, A., Zomet, A., and Weiss, Y. (2004b). User assisted separation of reflections from a single image using a sparsity prior. Proc. ECCV, pp.602–613.

Sarel, B. and Irani, M. (2004a). Separating transparent layers of repetitive dynamic behaviors. Proc. IEEE ICCV, pp.26–32.

Sarel, B. and Irani, M. (2004b). Separating transparent layers through layer information exchange. Proc. ECCV, pp.328–341.

Schechner, Y., Shamir, J., and Kiryati, N. (2000). Blind recovery of transparent and semireflected scenes. Proc. IEEE CVPR, pp.1038–1043.

Smith, S. M. and Brady, M. (1997). Susan -a new approach to low level image processing. IJCV, vol.23, no.1, pp.45–78.

Szeliski, R., Avidan, S., and Anandan, P. (2000). Layer extraction from multiple images containing reflections and transparency. Proc. IEEE CVPR.

Tsin, Y., Kang, S. B., and Szeliski, R. (2003). Stereo matching with reflections and translucency. Proc. IEEE CVPR, pp.702–709.

Ueda, N. and Nakano, R. (1998). Deterministic annealing em algorithm. Neural Networks, vol.11, no.2, pp.272–282.

Urahama, K. and Nagao, T. (1995). Direct analog rank filtering. IEEE Trans. Circuits Syst. I, vol.42, pp.385–388.

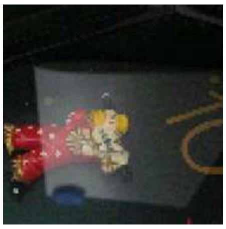


Figure 6: Input image.

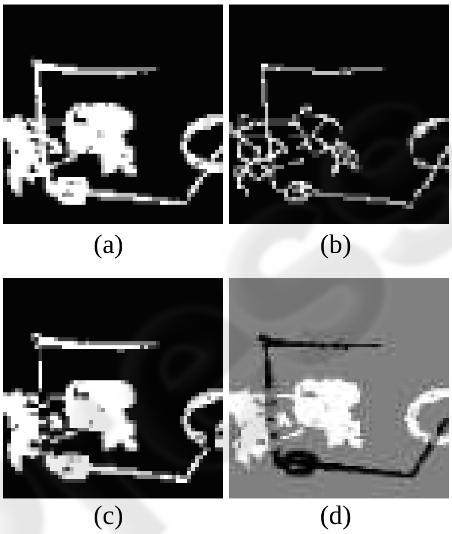


Figure 7: Edge detection and classification.

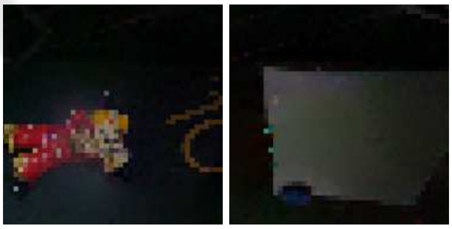


Figure 8: Intermediate result (decomposition on low resolution image).

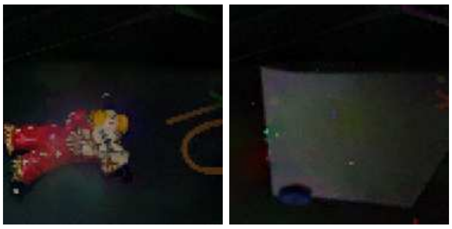


Figure 9: Decomposition.

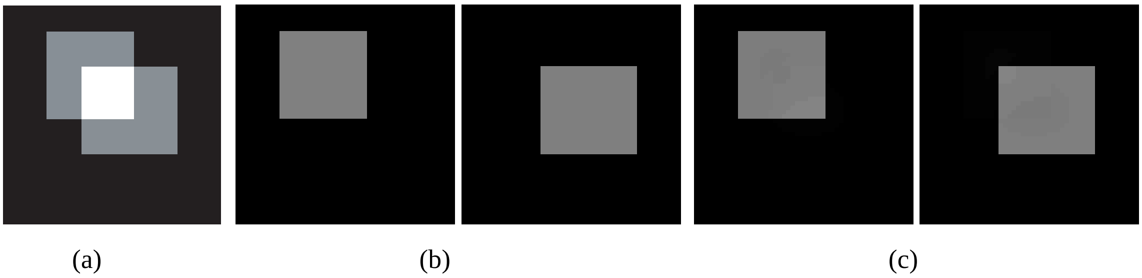

(a) (b) (c)

Figure 10: Result. (a) Input image (synthesized by summing the simple images in b). (b) Ground truth images. (c) Decomposition.

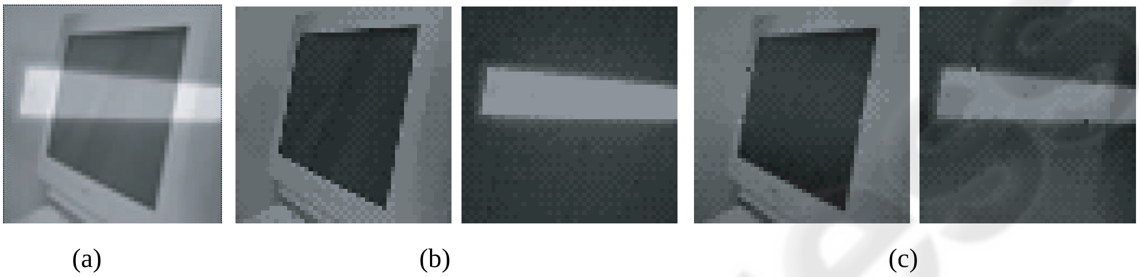

(a) (b) (c)

Figure 11: Result. (a) Input image (synthesized by summing the simple images in b). (b) Ground truth images. (c) Decomposition.

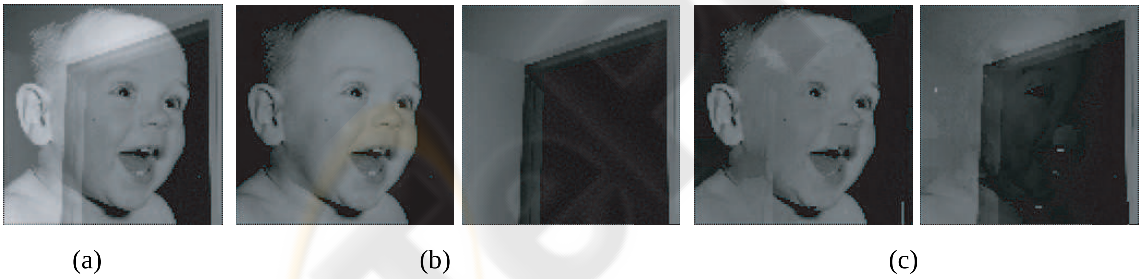

(a) (b) (c)

Figure 12: Result. (a) Input image (synthesized by summing the complex and simple images in b). (b) Ground truth images. (c) Decomposition.

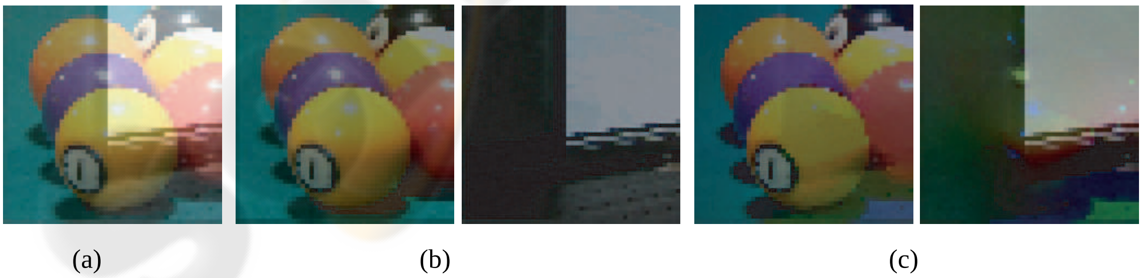

(a) (b) (c)

Figure 13: Result. (a) Input image (synthesized by summing the complex and simple images in b). (b) Ground truth images. (c) Decomposition.

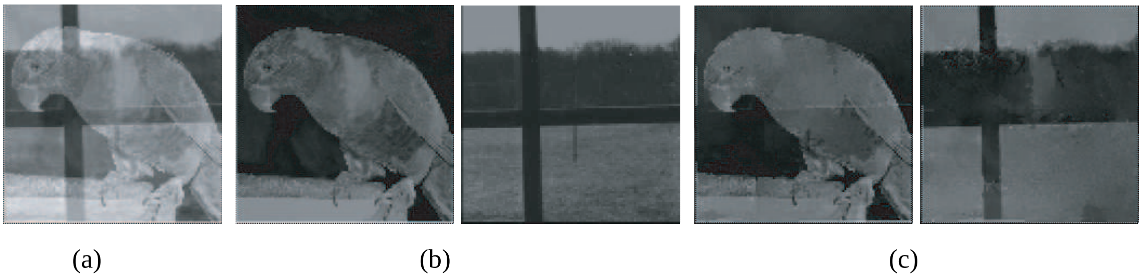


Figure 14: Result. (a) Input image (synthesized by summing the complex images in b). (b) Ground truth images. (c) Decomposition.

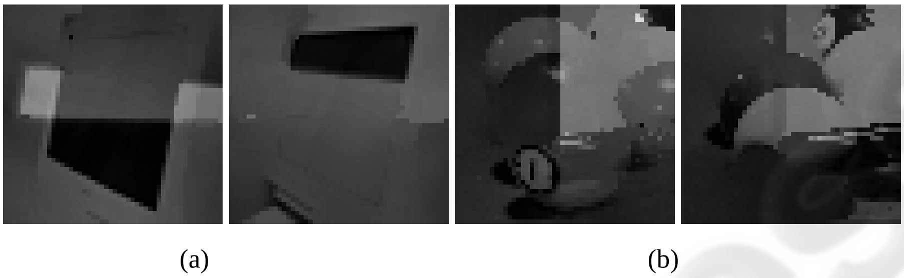


Figure 15: Results without classifying edges. (a) Decomposition when the image of Fig. 11(a) is taken as an input. (b) Decomposition when the separated green channel image of the image of Fig. 13(a) is taken as an input.