AutoKey: Human Assisted Key Extraction

*Tomoo Mitsunaga, Taku Yokoyama, Takashi Totsuka**

SONY Corporation

Abstract

Key extraction is an inverse problem of finding the foreground, the background, and the alpha from an image and some hints. Although the chromakey solves this for a limited case (single background color), this is often too restrictive in practical situations. When the extraction from arbitrary background is necessary, this is currently done by a time consuming manual task. In order to reduce the operator load, attempts have been made to assist operators using either color space or image space information. However, existing approaches have their limitations. Especially, they leave too much work to operators. In this paper, we present a key extraction algorithm which for the first time, addresses the problem quantitatively. We first derive a partial differential equation that relates the gradient of an image to the alpha values. We then describe an efficient algorithm that provides the alpha values as the solution of the equation. Along with our accurate motion estimation technique, it produces correct alpha values almost everywhere, leaving little work to operators. We also show that a careful design of the algorithm and the data representation greatly improves human interaction. At every step of the algorithm, human interaction is possible and it is intuitive.

CR Categories: I.3.3 [Computer Graphics]: Picture / Image Generation; I.4.6 [Image Processing]: Segmentation - *Edge and feature detection*; I.4.7 [Image Processing]: Feature Measurement; I.5.2 [Pattern Recognition]: Design Methodology - *Feature evaluation and selection*.

Additional Keywords: Key, alpha value, image composition, object extraction, block matching, edge detection, spline.

1 Introduction

In this paper, we consider the following problem; given a 2D image that consists of a foreground object and a background, determine the alpha value of the foreground object. Since the alpha value is called *key* in the video/movie community, we call this key extraction problem.

* Sony Corporation. 6-7-35 Kitashinagawa, Shinagawa Tokyo 141, Japan e-mail: *mitsunag@av.crl.sony.co.jp, yokoyama@av.crl.sony.co.jp, totsuka@av.crl.sony.co.jp*

Knowing the alpha (key) value is the first step in compositing images [12]. Unless the object shape is already known (e.g., rendered from a 3D model), alpha value is not available at hand. Therefore, usually it must be extracted from images that are taken by a camera or other input devices. Hence, key extraction is essential. Recent wide use of special effects increases the importance of key extraction.

For a limited special case – single constant background color, chromakey is the solution to key extraction. With the chromakey, objects such as actors are placed in front of a screen of a known color which is usually blue or green. By fixing the background color, the key extraction becomes much simpler (see section 2). Despite the limitation due to single background color, the chromakey is widely used since it is the only automated solution at this moment. However, due to the problems listed below, it is often inapplicable to high-end image works.

- Choice of foreground colors is limited. Usually, it must be different from the background color.
- In order to obtain a constant background color, shots are taken in a studio. Thus, for natural compositing, careful adjustment of the lights is necessary to mimic the condition of outside shots. Such adjustment often requires several trials, making indoor shots unattractive.
- Due to light reflection, the foreground color is affected by the background (called contamination). This is particularly noticeable near the object boundary.

When chromakey is insufficient or extraction from an arbitrary background is necessary, key extraction becomes a much harder problem. Although several techniques have been proposed, still much of the tedious pixel-by-pixel cleaning up job is left to human operators. We are told that this manual clean up can consume as much as 70% of operator's work, which is about a half of total production time[1] . This has been a major problem of image compositing works in the real world.

The key extraction from an image with arbitrary background is hard since this requires a high level image understanding that a human would do. However, it is still far beyond current capabilities of a machine vision. As a consequence, collaboration of a computer and an operator is necessary. In order to make such collaborative approach satisfactory for an operator, the following requirements must be fulfilled.

- High accuracy. Even if perfect work is not expected, the algorithm must not leave more that a small acceptable amount of adjustment task to an operator.

[1] The numbers were obtained by our interviews at post-production studios working on TV commercials and movies. The operator work includes rendering, painting, image editing, and compositing.

- Hint directed. It must be easy to give hints to the algorithm. The algorithm should fully exploit the hints and work adaptively.
- Easy to fix problems. That is, the shape of the key must be easily modifiable. This suggests the use of a parameterized representation of the key (i.e., object shape) rather than a discrete (e.g., image) form. Also, such parameterization must be a local one to allow local modifications without changing the entire shape.

In this paper, we present a key extraction algorithm that satisfies these criteria. We first establish the relationship of the alpha value and the first derivative of the image. This relationship provides a quantitative foundation to our new key extraction scheme which consists of the three major components; (1) a color gradient computation algorithm, (2) a motion tracking algorithm, and (3) key (alpha) surface construction algorithm. Our color gradient computation yields significantly better SNR compared to existing edge detectors, contributing to the quality of the key signal. The motion tracking algorithm provides a reliable motion vector while minimizing perceptual artifacts due to errors. It is also robust against background color changes. The key surface construction makes shape modification easier by providing a parametric form of the key.

These techniques are developed by an interdisciplinary research of three related fields; graphics, image processing, and computer vision. Ideas in one field often helped development of a new algorithm in another field. Several examples are shown in our paper.

The rest of the paper is organized as follows. Section 2 reviews related works. Section 3 presents the relationship of the key and the derivative of the image, and section 4 describes the algorithm in detail. Section 5 show results from our implementation and section 6 gives conclusions and possible future directions.

2 Related Works

The color $\boldsymbol{I}$ of an image at a pixel (x, y) is defined as:

$$\boldsymbol{I} = \alpha \boldsymbol{F} + (1 - \alpha) \boldsymbol{B} \tag{1}$$

where $\boldsymbol{F}$ and $\boldsymbol{B}$ are the foreground and the background color, respectively. α is the alpha or key value. The key extraction problem is the inverse problem of equation 1; given $\boldsymbol{I}$ and some hint, compute α, $\boldsymbol{F}$, and $\boldsymbol{B}$. As shown in figure 1(a), in a color space, the key extraction can be viewed as a problem of defining a line segment $\boldsymbol{F}\boldsymbol{B}$ which is divided by point $\boldsymbol{I}$ at the ratio of $(1 - \alpha) : \alpha$. Since the only constraint imposed by equation 1 is that the three points $\boldsymbol{F}$, $\boldsymbol{B}$ and $\boldsymbol{I}$ lie on the same line, for any value for one of $\boldsymbol{F}$, $\boldsymbol{B}$, or α we can find other two that satisfy equation 1. Hence, we need more information in order to determine the three parameters. There are two approaches to add supplementary information; color space approach and image space approach.

Color Space Approach The color space approaches provide some hints about the foreground and background colors. For example, the background color is given with the chromakey. Since both $\boldsymbol{B}$ and $\boldsymbol{I}$ are known, the range of search for $\boldsymbol{F}$ and α degenerates from 3D space to a line defined by $\boldsymbol{B}$ and $\boldsymbol{I}$. α is estimated by the distance of $\boldsymbol{B}$ and $\boldsymbol{I}$ or some other metric.

As a generalization of the chromakey to the arbitrary background case, color distributions of the background, the foreground, or both can be given as a hint. Given a color distribution, α is assigned by some metric to each color (R,G,B) in the color space such that $\alpha = 0$ near the background and $\alpha = 1$ near the foreground. One can control this assignment by specifying one or more iso-surfaces of α

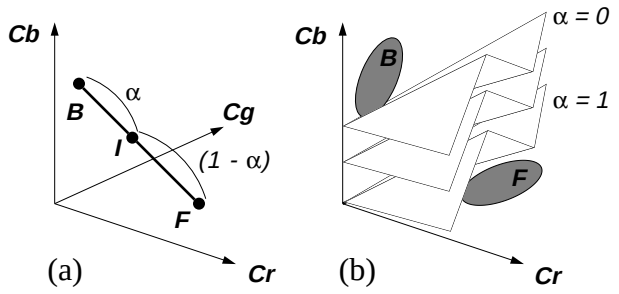

Figure 1: Key extraction in a color space. (a) relationship of the foreground, background, and observed colors, (b) an example of polygonal iso-surfaces of α.

(figure 1(b)). A popular chromakey system uses several triangular patches as the iso-surface [3][17]. A more general polyhedral slice approach was also proposed in[11].

Multivariate statistical analysis is another tool to assign α to RGB space [1]. For example, discriminant analysis[2] can separate colors into two groups and from the variance and distance of the groups, alpha values of the pixels in the intermediate region can be estimated.

These methods, however, do not work when the color distributions of the foreground and the background overlap. Not only due to multiple object colors, but also due to shading and reflection, the color distributions tend to spread, causing overlap of the distributions. Although subdivision of the image into small blocks improves the separation, as the block becomes smaller, the color distribution within such small block is statistically less reliable. We have encountered many examples in which a 16 by 16 block is still too large to separate the two regions. Since the boundary region is sometimes several pixels wide, any smaller block may not capture the entire fore to back transition.

Image Space Approach Image space approaches exploit coherency; the fact that an object is a relatively large chunk, rather than a set of discrete dots. Under this assumption, the problem becomes a boundary detection and tracking problem which has been studied in computer vision.

The snake [9][16] is a popular technique for object tracking. An energy function is defined such that it is minimum when a polyline called snake correctly traces the object boundary. The object motion is tracked as an energy minimization problem. Since the key shape usually is complex and changes quickly, initial placement of the control points may not be appropriate in later frames. Although it is possible to cope with it by providing more control points, increase in the number of points exponentially raises the computational cost of the search. The explosion of the search also occurs as the resolution of the search increases.

Clustering is another approach to tracking. With this approach, pixels are transformed to an appropriate space to find the similarity among them. For example, a five dimensional space (x,y and r,g,b) is used to find groups that represent the foreground object [4]. In the context of the key extraction, the problems of the clustering methods are that they do not provide a soft transition from the foreground to the background and that the clustering does not perform well under low color purity.

A different approach based on edge thinning and widening was also proposed [6] [7]. Edge detection is first performed and then a

[2] Given a set of points, find an axis along which the points are best separated into two groups.

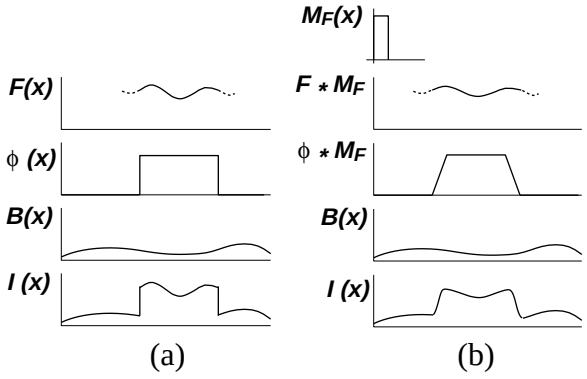

Figure 2: Alpha blending of 1D signals. (a) without motion blur, (b) with motion blur.

thinning process leaves pixels with high edge intensities which are treated as the object boundary. The boundary is then widened again to give a rough estimate of the location of the boundary on the next frame. Since such conventional edge detection responds to any transition of colors (e.g., textures, shadows), high edge intensity does not directly mean that the pixel is on the boundary. Although they mention a way of creating a soft key (smooth transition of the alpha value), it is rather empirical.

3 The Gradient of an Image

In this section, we establish an important relationship of the key (alpha) and the derivative of the image that our algorithm is based on. We also study conditions under which the relationship holds. For simplicity, we consider the key extraction of 1D scalar functions. The argument can be extended naturally to color images (3D vector field defined over a 2D space).

Consider placing a foreground object on top of a background. The foreground object is defined by its color $F(x)$ and the opacity $\phi(x)$. $B(x)$ is the background color. For an opaque object, $\phi(x)$ is a rectangle function which is 1 everywhere foreground object exists and 0 otherwise. $F(x)$ is undefined outside of the foreground region (figure 2(a)). Clearly, the blended result $I(x)$ is given by

$$I(x) \;=\; \phi(x)\, F(x) \;+\; (1-\phi(x))\, B(x) \tag{2}$$

Then the derivative of $I(x)$ is

$$I'(x) \;=\; \phi'(x)\,(F(x)-B(x)) + \phi(x)\,F'(x) + (1-\phi(x))\,B'(x) \tag{3}$$

Near the boundary of the object where $\phi(x)$ changes quickly, we can assume that both F' and B' are relatively small compared to ϕ'. In this case, equation 3 can be approximated by

$$I'(x) \;\approx\; \phi'(x)\,(F(x) - B(x)) \tag{4}$$

Since ϕ is the alpha value we want to determine, we can rewrite as

$$I'(x) \;\approx\; \alpha'(x)\,(F(x) - B(x)) \tag{5}$$

under the condition that

$$\phi'(x) \gg F'(x), \quad \phi'(x) \gg B'(x) \tag{6}$$

That is, the first derivative of the alpha value is proportional to the first derivative of the image. The ratio is given by the difference of the foreground and the background.

We now consider the effect of motion blur. The effect of blurring can be expressed by an operator $M_F(x)$. For example, suppose that the foreground object is moving at a constant velocity V and the shutter speed of the camera is T_s. Then $M_F(x)$ is a rectangle function of width VT_s as in figure 2(b). Using this function, the motion blurred object can be represented by its blurred color $(F * M_F)$ and the opacity $(\phi * M_F)$ where the operator $*$ denotes convolution.

By a similar derivation, we get

$$I'(x) \;\approx\; (\phi(x) * M_F(x))'\,((F(x) * M_F(x)) - B(x)) \tag{7}$$

With the motion blur, the alpha value we want is $(\phi * M_F)$. Hence,

$$I'(x) \;\approx\; \alpha'(x)\,((F(x) * M_F(x)) - B(x)) \tag{8}$$

Again, the derivatives are proportional. One of the condition in equation 6 is replaced by $(\phi(x) * M_F(x))' \gg (F(x) * M_F(x))'$. This condition is no stronger than the previous one, since the derivatives are equally attenuated by the blurring operator. Note that this relationship does not depend on types of blur (shape of M_F). By repeating the similar analysis, we can derive the same relationship, $I' \propto \alpha'$, under motion blur of both the foreground and the background.

In a real situation, we also have to consider the effect of the optical and electrical prefiltering before the sampling process. Since the prefiltering process can be expressed by a low-pass filter H, by replacing the M_F with $M_F * H$[3], we can make the same argument which leads to the same conclusion.

The above argument can be extended to the 2D image case. Since the image is a 2D function, relationship of the gradient vectors is obtained. That is, the gradient of the alpha and the image are parallel:

$$\nabla I(x,y) \;=\; c\,\nabla\alpha(x,y) \tag{9}$$

where c is a scalar constant which represents the difference of the foreground and the background (see equation 5).

The alpha value is obtained by solving this partial differential equation. Given a path $P : \boldsymbol{p}(t)$, α at a point $\boldsymbol{p}(t)$ on P is given by

$$\alpha(\boldsymbol{p}(t)) \;=\; \frac{1}{c}\int_P \nabla I(x,y)\cdot d\boldsymbol{p}(t) \;+\; c_0 \tag{10}$$

Since we have two obvious constraints – $\alpha = 0$ at background and $\alpha = 1$ at the foreground, c_0 can be easily determined.

Note that a color image has more information (i.e., three scalar values per pixel) than what is needed to solve the equation (single scalar value). As we see in section 4.3, this additional information can tolerate the condition that $\boldsymbol{F}$ and $\boldsymbol{B}$ be locally constant.

Using the relationship, we can obtain the alpha value or key from the first derivative of the image. This differential approach has the following advantages:

- Slope of the alpha value is given quantitatively.
- Unlike the chromakey, absolute values of B or F are not necessary. Local difference in the color provides enough information to determine the alpha.
- The requirement that F and B be constant is necessary only around small regions (e.g., a couple of pixels wide) on the boundary where the alpha changes quickly. The color space approaches discussed in section 2 require that the colors be constant over much larger area in order to assure the separation of the color distribution.

[3]The new operator includes the effect of both motion blur and the prefiltering

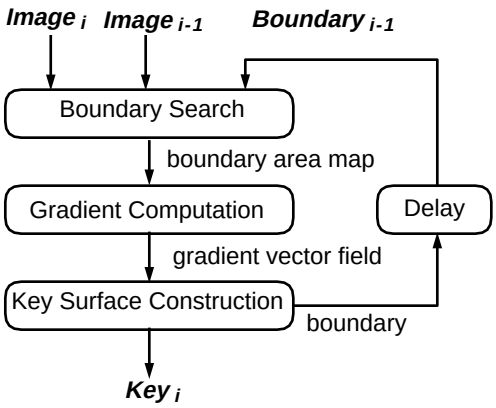

Figure 3: Flow of our key extraction algorithm.

4 Key Extraction

4.1 Processing Cycle

We now describe our key extraction algorithm in detail. The flow or the algorithm is shown in figure 3. For each iteration, the algorithm takes an input frame as well as the previously processed frame which is used as a reference. At the end of the iteration, the key is output as a black and white image. A set of cubic spline segments is also produced which becomes another reference input during the next iteration.

Within the iteration, the following three steps are performed serially.

1. Boundary search. Estimate the boundary of the foreground object in the new frame.
2. Gradient computation. Along the boundary obtained in the previous step, compute the gradient of the image.
3. Key surface construction. Using the relationship described in section 3, compute a parametric surface whose Z coordinate represents the alpha value. The surface is then scan converted into a monochrome image of the alpha values, which is the output of the iteration.

4.2 Boundary Search

The purpose of this step is to provide the following gradient computation step with an accurate estimate of the object boundary. Accurate estimate of the boundary allows the gradient computation to work more precisely on the boundary, spatially eliminating the noise of the gradient vector field.

The estimate is computed by a block matching between the previous and the new frame. For our purpose, the following issues which existing block matching algorithms (e.g. [14]) do not address must be considered: (1) Background color can change drastically as the foreground object moves. Hence, equally evaluating the foreground and the background pixels can cause matching errors. (2) Minimum difference of blocks does not necessarily mean that the two blocks correspond to each other. This is acceptable for image compression [20], but not for object tracking. (3) A poorly chosen block (e.g., a block that only contains a straight boundary) does not always exhibit a sharp peak in the autocorrelation, producing unreliable motion estimate depending on the motion direction. (4) Perceptual error must be minimized.

In consideration to the above issues, our block matching first searches high curvature points on the boundary which becomes the source block of the matching. It is well known in computer vision that human visual system responds strongly to high curvature [2], [5]. Hence, use of such points as the source block is desirable in order to minimize perceptual artifacts. Since such blocks contain curved portion of the boundary, the autocorrelation exhibits a sharp peak on any motion direction, which is also desirable. Figure 4(a) shows an example of source block placement. Yellow dots on the boundary indicates centers of the blocks. In this implementation, blocks are recursively placed till the difference of the orientation is within the predefined threshold.

For each of these blocks, its destination is searched in the new frame. In order to reduce the effect of the background color change, weighted mean square error MSE is computed as follows:

$$MSE = \frac{\sum_{i,j} w[i,j]\,(I_n[i,j] - I_{n-1}[i,j])^2}{\sum_{i,j} w[i,j]} \qquad (11)$$

where $I_n[i,j]$ and $I_{n-1}[i,j]$ are blocks of the new and the previous frames, respectively, and $w[i,j]$ is the weight. In order to emphasize the difference in the foreground region, the alpha value obtained in the previous frame is used as the weight.

Figure 4(b) and (c) show matching results of the blocks in figure 4(a) by conventional block matching (b) and by our method (c), respectively. Figures 4(d)∼(f) repeat the same comparison using a different object. With the conventional method, color changes of the background near the boundary (c) and/or motion of the background (e) degrades the accuracy of the block matching.

Once the block matching is done, the destination blocks are connected to form a continuous boundary.

4.3 Gradient Computation

For pixels within the estimated boundary, gradient of the image is computed. Although the gradient is computed using the Sobel operator, a well known edge detector in image processing [8], its use is quite different in two ways. First, the Sobel operator is not applied to the RGB images. Instead, it is applied to the projection of the color image. Second, selectivity of the edge orientation is controlled by a non-linear post-processing.

As described in section 3, the gradient of the image and the alpha are proportional provided that $\boldsymbol{F}$ and $\boldsymbol{B}$ be constant near the boundary. This is usually true, but there are cases in which this condition no longer holds. Textures, shadows, strong shading, and reflections can sometimes cause non-negligible variation of the color, introducing turbulence to the gradient.

Figure 5(a) shows a typical color distribution of an object. Depending on the orientation of the surface, color changes from a dark color to the fully illuminated object color, and to a brighter color due to the specular reflection. This distribution corresponds to the local illumination model in graphics and it is called the dichromatic distribution in computer vision [10]. When the reflection is not negligible, the global illumination model must be introduced. In this case, the color of the object is shifted since a fraction of the background color, which is the product of the form-factor and the reflectivity, is added to the original color (figure 5(b)).

These observations suggest that we can tolerate the condition about $\boldsymbol{F}$ and $\boldsymbol{B}$ as long as the direction of the color variation described above is different from vector $\boldsymbol{F} - \boldsymbol{B}$, the direction of the color transition across the boundary. This is done by first choosing a color vector C such that it is approximately perpendicular to the

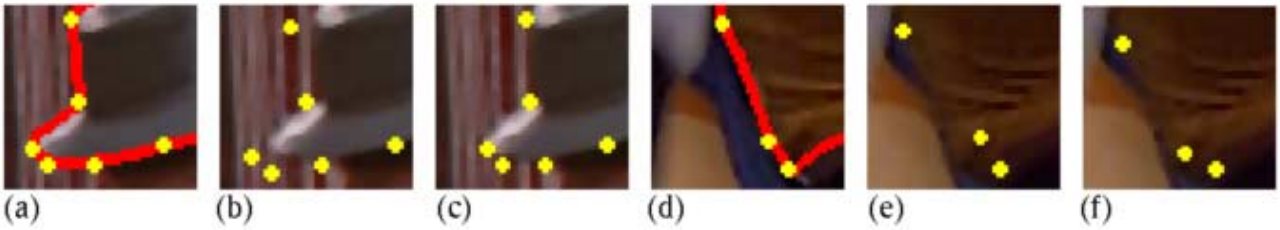

Figure 4: Block matching. (a) and (d): blocks (yellow dot) placed on along the object boundary (red line) in the previous frame, (b) and (e): results of the conventional block matching, and (c) and (f): results of our foreground weighted block matching.

distributions of the foreground and the background, and then projecting color $\boldsymbol{I}$ to the inner product I_p:

$$I_p(x,y) \;=\; (\boldsymbol{I}(x,y)\cdot\boldsymbol{C}) \tag{12}$$

Since $\boldsymbol{C}$ is perpendicular to the main axis of the color distribution, color variations within the region do not cause much difference in I_p. Hence, the SNR of the gradient is considerably improved. Existing color edge detectors fails in this respect, for they try to detect *any* transition of the color [13] [15] [19].

Further improvement of the SNR is possible by weighting the gradient by the difference from the expected orientation which is obtained from the result of the previous frame. Strong textures and shadows can be eliminated as long as their orientation is different. A sharp selectivity which is not available by conventional linear filtering can be achieved by a classical technique in graphics as

$$\boldsymbol{G}(x,y) \;=\; \nabla I_p(x,y)\left(\frac{\nabla I_p(x,y)}{|\nabla I_p(x,y)|}\cdot\boldsymbol{N}\right)^s \tag{13}$$

where $\boldsymbol{G}$ is the weighted gradient, N is the expected orientation (normalized), s is the selectivity index. The Phong shading uses the same technique to obtain a sharp selectivity.

Our implementation of the algorithm first subdivides the given object boundary into shorter regions such that both the color vector $\boldsymbol{F}-\boldsymbol{B}$ and the expected orientation $\boldsymbol{N}$ are similar within the region. Then, for each small region, the best color projection vector C is computed followed by calculations of equations 12 and 13. Finally, since each small region may choose different parameters such as $\boldsymbol{C}$, the output $\boldsymbol{G}$ is rescaled such that the unit length has the same weight elsewhere.

An example is shown in figure 6. The original image (a) is processed by [19] which is claimed to be the best filter based color edge detector, and by our algorithm (c). Note that our algorithm is robust to color variations within the object even with strong shading and reflections.

4.4 Key Surface Construction

The last step generates key image (monochrome image of the alpha value) as well as the parametric form of the key in order to ease modification of the shape when it is necessary. The parametric form is a surface whose Z coordinate represents the alpha value.

Given the gradient vector field from the previous step, the algorithm first generates the contour line of $\alpha = 0.5$ as cubic splines. We denote this curve as $\boldsymbol{C}_M(t)$. The end points of each segment are selected from pixels with high gradient magnitudes. The tangent vectors are set orthogonal to the gradients at these points. Since adjacent segments share the same tangent vector, $\boldsymbol{C}_M(t)$ has G^1 continuity. Finally, the velocities at the end points are chosen to satisfy

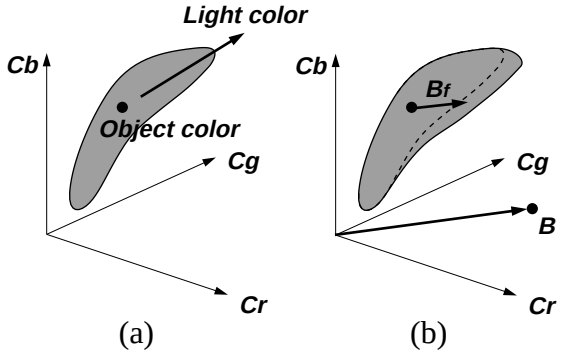

Figure 5: Color distribution. (a) effect of local illumination, (b) effect of reflection (indirect light).

the following two criteria: (1) the orientation is orthogonal to the gradient vector field, (2) the curve passes along peaks of the gradient magnitude (figure 7(a)). This is done by an exhaustive search. Some examples of the curve fitting are shown in figure 8. Both (a) and (c) are resulting cubic spline segments displayed on top of the gradient magnitude images, and (b) and (d) are the same splines but displayed on the original images. These closeups show that the fitting process can follow fine details of the boundary.

Construction of the surface is done as follows. First, examine pixels on both sides of $\boldsymbol{C}_M(t)$ to determine the constant factor c in equation 9. Since the gradient of the image is already known, we get the gradient of the alpha from the equation. By a linear approximation, the endpoints of the contour lines of $\alpha = 0$ ($\boldsymbol{C}_{\boldsymbol{B}}(t)$) and $\alpha = 1$ ($\boldsymbol{C}_{\boldsymbol{F}}(t)$) are obtained. Their tangent vectors are set parallel to $\boldsymbol{C}_M(t)$. The velocities are set such that the three segments share the same center of curvature (figure 7(a)). As a special case, if the distance between the endpoints (e.g., $P0$ and $Q0$) is longer than the curvature radius, the velocity of the inner curve is set to 0. Finally, the key surface $K(s,t)$ is defined by

$$K(s,t) \;=\; s\,\boldsymbol{C}_{\boldsymbol{F}}(t) \;+\; (1-s)\,\boldsymbol{C}_{\boldsymbol{B}}(t) \tag{14}$$

Figure 7(b) shows an example of key surface $K(s,t)$. Clearly, the surface also has G^1 continuity. This surface is scan-converted to generate the key image, which is the other output of the step.

4.5 Human interaction

At each of above steps, human interaction can be easily done. At the boundary search step, human operator can specify additional char-

Figure 6: Edge detection. (a) original image, (b) gradient magnitude computed by a conventional edge detection algorithm, (c) gradient magnitude by our algorithm.

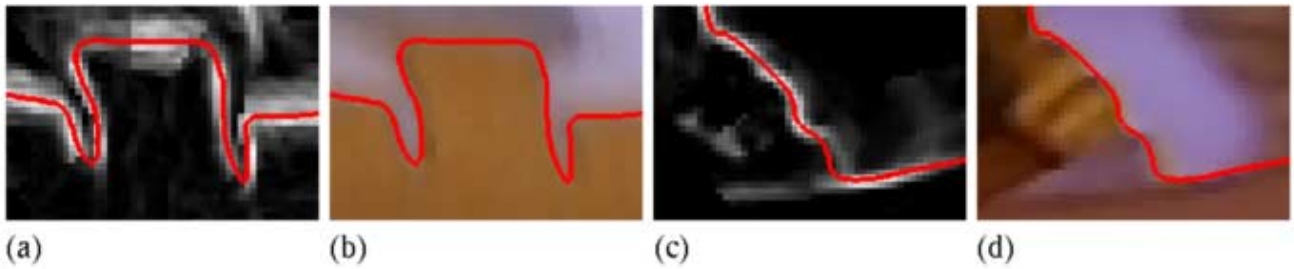

Figure 8: Results of curve fitting. (a) and (c): gradient magnitude and the spline segments, (b) and (d): the same spline segments placed on the original image.

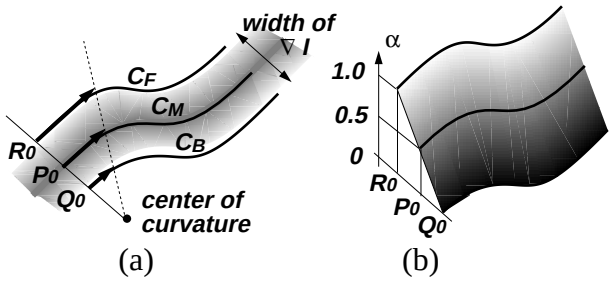

Figure 7: Key surface construction. (a) fitting $C_M(\alpha = 0.5)$ on the gradient vector field and determining C_F ($\alpha = 1$) and C_B ($\alpha = 0$), (b) construction of the cubic spline surface.

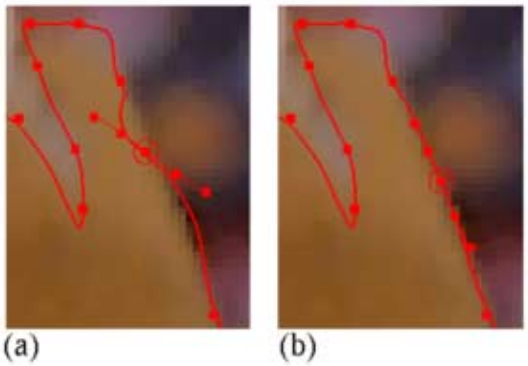

Figure 9: Editing a boundary curve.

acteristic points in order to improve the accuracy of the search.

The color vector $\boldsymbol{C}$ and the expected orientation used in the gradient computation step, can also be given interactively. Especially, the expected or preferred boundary orientation can be intuitively given by a brush movement.

At the key surface construction step, surface shape can be modified by known editors such as the one shown in figure 9.

With our algorithm, human interaction is easy because (1) the parameters are intuitive to a human operator, and (2) the parameters are local, affecting only the local behavior. If direct manipulation of images is necessary, such easy interaction is not possible.

Since our algorithm requires the result of the previous frame as a reference, initial solution must be given manually. We do this by providing a rough drawing of the boundary shape to the gradient computation step. The rough drawing provides an idea about the orientation of the boundary which helps the algorithm to selectively find the boundary.

5 Results

Figure 10 show some results of our key extraction algorithm. In this example, the trouser of the football player is chosen as the foreground object. For the initial frame shown in the first column, a rough drawing of the boundary was given as a hint. Successive frames are processed by the computer with operator's subtle correc-

tion where it is desired. The upper row is the output of the algorithm *before* human correction.

Figures (e), (f), (g), and (h) show that the algorithm can correctly work on such large change of the shape. Fine complex details such as waist-belt and around the left hand are well tracked across frames without human intervention. Despite the big change of the background color near the left knee, the left hand, and the right thigh, the algorithm follows the boundary and produces correct key images.

Figure 11 show some zoom-ups of the same image. (a) and (b) are part of the waist-belt and (c) and (d) are part of the hip. Due to the player's vertical motion, horizontal boundary (a) shows more blur than (c). Figure 11(b) and (d) show the quantitative correctness of our algorithm.

6 Conclusions

The key extraction from images with arbitrary backgrounds is a hard problem which cannot be totally automated. Hence, an algorithm for key extraction must be designed to allow easy human interaction.

By judicious choice of the parameters and the data representation, our algorithm allows human interaction at every step of the processing and it is intuitive to an operator. Because of the parametric form of the key shape, it is easy to incorporate other interactive techniques in graphics (e.g., morphing, key frames) that can further eliminate the need for human interaction.

Based on the relationship between the derivatives of the image and the alpha value, our algorithm provides a quantitative method for key extraction for the first time.

There are many areas that require further development. In the gradient computation, choice of a different color space and a projection should be considered since more compact projection of the color distribution yields yet better SNR.

Better estimate of the foreground color near the boundary is another area of study. Current assumption that the color is constant is not appropriate if there is a strong texture.

To improve the user interface and to improve the accuracy of the boundary estimate, the algorithm should be integrated with the key frame method or other morphing mechanisms.

References

[1] Anderson, T. W., *Introduction to Multivariate Statistical Analysis*, John Wiley & Sons, 1958.

[2] Attneave, F., "Some Informational Aspects of Visual Perception", *Psychological Review*, Vol.61, pp.183-193, 1954.

[3] Dadourian, A., "Compositor for Video Images", patent in U.S.A., US5343252-A, 1994.

[4] Etoh, M. and Shirai, Y., "Automatic Extraction of Complex Objects Using Region Segmentation", *NICOGRAPH'92*, pp.8-17, 1992.

[5] Inui, T., "A Model of Human Visual Memory", *6th Scandinavian Conference on Image Analysis*, pp.325-332, 1989.

[6] Inoue, S., "An Object Extraction Method for Image Synthesis", *IEICE Trans.*, Vol.J74-D-II, No.10, pp.1411-1418, 1991.

[7] Inoue, S. and Koyama, H., "An Extraction and Composing Method for Moving Image Synthesis", *Journal of ITEJ*, Vol.47, No.7, pp999.-1005, 1993.

[8] Jain, A. K., *Fundamentals of Digital Image Processing*, Prentice-Hall, 1989.

[9] Kass, M., Witikin, A. and Terzopoulos, D., "Snakes : Active Contour Models", *International Journal of Computer Vision*, Vol.1, No.3, pp.321-331, 1988.

[10] Klinker, G. J., Shafer, S. A. and Kanade, T., "The Measurement of Highlight in Color Images", *International Journal of Computer Vision*, Vol.2, pp.7-32, 1988.

[11] Mishima, Y., "A Software Chromakeyer Using Polyhedric Slice", *NICOGRAPH'92*, pp.44-52, 1992.

[12] Porter, T. and Duff, T., "Compositing Digital Images", *Computer Graphics*, Vol.18, No.3, pp.253-259, 1984.

[13] Robinson, G. S., "Color Edge Detection", *Optical Engineering*, Vol.16, No.5, pp.479-484, 1977.

[14] Sezan, M. I. and Lagendijk, R. L., *Motion Analysis and Image Sequence Processing*, Kluwer Academic Publishers, 1993.

[15] Trahanias, P. E. and Venetsanopoulos, A. N., "Color Edge Detection Using Vector Order Statistics", *IEEE Trans. Image Processing*, Vol.2, No.2, pp.259-264, 1993.

[16] Ueda, N., Mase, K. and Suenaga, Y., "A Contour Tracking Method Using Elastic Contour Model and Energy Minimization Approach", *IEICE Trans.*, Vol.J75-D-II, No.1, pp.111-120, 1992.

[17] Vlahos, P., "Combining Method for Colour Video Signals", patent in U.S.A., US4625231-A, 1986.

[18] Wong, R. Y. and Hall, E. L., "Sequential Hierarchical Scene Matching", *IEEE Trans. Comput.*, Vol.C-27, No.4, pp.359-366, 1978.

[19] Zenzo, S. D., "A Note on the Gradient of a Multi-Image", *Computer Vision, Graphics, and Image Processing*, Vol.33, pp.116-125, 1986.

[20] MPEG Video Committee, "Information Technology – Generic Coding of Moving Pictures and Associated Audio", ISO/IEC IS 13818-2, Nov. 1994.

Acknowledgements

The authors got comments from many movie and video engineers. We wish to thank all of them for their valuable comments and suggestions. Discussions with Norihisa Shirota and Tatsuo Shinbashi were helpful in developing the algorithm.

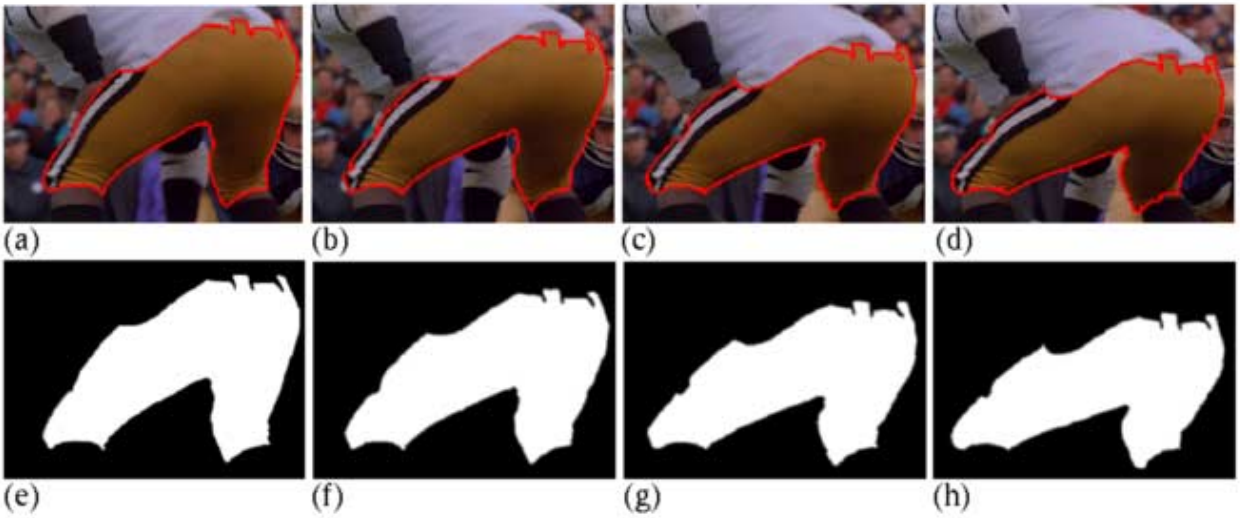

Figure 10: Key image sequence obtained by our algorithm. The four columns correspond to the four successive frames. The upper row shows result of the spline for $\alpha = 0.5$ before correction by an operator, the lower row shows the key image.

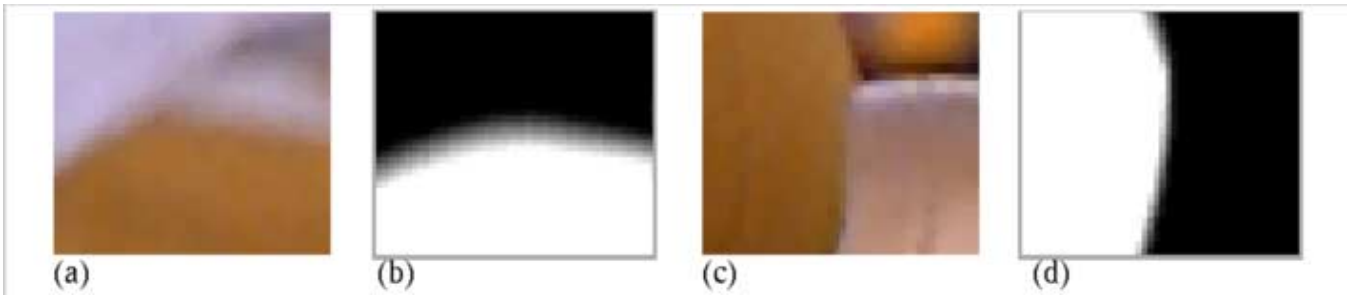

Figure 11: Magnified part of the original and key image by our algorithm. (a) and (b): a horizontal boundary part, (c) and (d): a vertical boundary part.