TZi Technologie-Zentrum Informatik

Technical Report 40

Digital Keying Methods

Christopher Schultz and Thorsten Hermes

TZI-Bericht Nr. 40
2006

Universität Bremen

TZI-Berichte

Herausgeber:
Technologie-Zentrum Informatik
Universität Bremen
Am Fallturm 1
28359 Bremen
Telefon: +49-421-218-7272
Fax: +49-421-218-7820
E-Mail: info@tzi.de
http://www.tzi.de

ISSN 1613-3773

Abstract

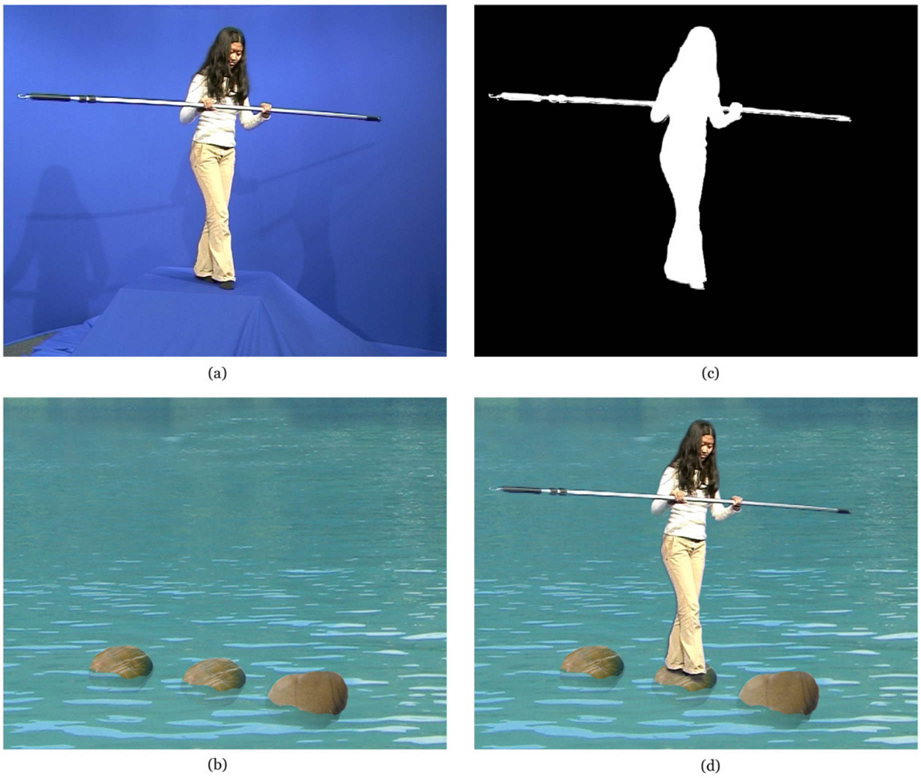
Keying has become a common image processing operation in TV and film production to seperate elements from a background. With the introduction of digital image processing more control is offered to the keying process. Like this it is possible to plot image pixel into a virtual 3D space and mask the background pixel according to their coordinates. This concept is called “3D keying”. In this report a detailed overview to common keying approaches is given and a self-developed 3D keying application is presented. Results of an evaluation against commercial software show that the software performs fastest of all.

Contents

1 Introduction

The digital combination of several image elements to a new composited image has become a common technology in nowadays television and film production. Like this it is for example possible to place an actor or object seamless into a new scene. However, composition requires the extraction of the wanted object from another image. Separating an object from the rest of the image is a basic operation to create new compositions. This process is called *keying* or *matting*.

Due to the importance of object extraction as a basic operation in visual effects, keying has become a standard technique in movie production. Regarding the top 20 international movies of all time, all of them applied keying technology to produce visual effects [Dat06]. They all used a blue or green background to key out objects, like illustrated in figure 1.


(a)

(c)

(b)

(d)

Figure 1: The four elements of a keying operation: (a) the foreground image, (b) the background image, (c) the foreground matte and (d) the final composition.

In the early days of movie production, objects were cut out and animated by hand, other methods used photochemical processes to separate the wanted object [Sey05]. With the change of movie technology from analogue film editing to digital post-production new keying methods were invented and implemented in software. Special visual effect applications enable the artist to realize keying and compositing in a fraction of the time the film pioneers needed. Beside the temporal aspect, the digital post-production allows the keying process to be highly controlled: a multitude of options and tools help to create precise keying results even for elements like fine hair, semi-transparent clothes or reflecting water.

This report will give an overview to state-of-the-art keying methods and present the results from a self-developed keying software. In section 2 the general keying problem is explained, whereas section 3 presents a summary of current digital keying approaches to solve this problem. A self-developed keying application is described in section 4, section 5 illustrates its keying results compared to commercial keying software. Finally, a conclusion is presented in section 6.

Additional bluescreen footage and evaluation result video material can be found at `http://www.tzi.de/tzikeyer/`.

2 The General Keying Problem

Keying is the most popular technique in visual effects for extracting objects from an image. It can be defined as the process of separating an object from the rest of an image by creating a *matte* which represents transparency information about the original image. Originally, a matte was a strip of monochrome film that was overlayed with the color film strip so that only parts of the movie were visible [SB96, p.259]. In computer graphics, a matte is "a one-channel image used to define the transparent regions of the foreground and background elements of a composite [Wri02, p.307]. This transparency matte equals an "alpha channel" [PD84].

The purpose of keying is to produce a matte that sets only the unwanted background fully transparent and the wanted foreground object fully visible (compare figure 1c). Because of this, the process of keying is also referred as *matting* ([SB96], [CAC$^+$02], [Chu04]). However, a simple binary matte concept does mostly not offer satisfying results for complex scenes. Fine details like hair, smoke or shadows can only be preserved if the matte supports semi-transparency – values between fully invisible (like 0.0) and fully visible (like 1.0).

The process of keying can be understood as the opposite of a composition process. According to [PD84], the general composition equation calculates a composition pixel value C from a foreground color F, a background color B and the opacity α.

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

Since digital pictures of photography, film, and video are mostly recorded, processed, and stored as three channel images (here referred as X, Y, and Z) we can refine the general composition equation to a three-channel-composition equation with 10 variables.

$$\begin{pmatrix} C_X \\ C_Y \\ C_Z \end{pmatrix} = \alpha \begin{pmatrix} F_X \\ F_Y \\ F_Z \end{pmatrix} + (1 - \alpha) \begin{pmatrix} B_X \\ B_Y \\ B_Z \end{pmatrix} \tag{2}$$

While C is the resulting value for a composition process, it is known for a keying process (C is the image to be keyed). Hence, keying is a problem of seven unknown variables [Chu04, p.6]. To solve this problem, most keying techniques use one of the following requirements:

- **A controlled background**: If either the background color is solid, or an image of the scene without the wanted object is available, B can be considered as known. Like this, the problem equation is limited to four unknown variables, but still under-constrained. Another disadvantage is that this method requires very precise studio conditions. Especially the backdrops have to be even lighted to obtain a solid background color.

- **Multiple images**: If the wanted object is shot against several backgrounds, it is possible to triangulate F [SB96]. However, this approach requires even more controlled studio conditions as is only applicable for stills.

- **A priori knowledge about the foreground and background area**: If the user marks the foreground and the background area, the left over space – the borders of an foreground object – is considered as unknown area. Figure 2 presents such a map, also called *trimap* [CAC$^+$02]. There exist several approaches that use this information to calculate transparency values for the unknown area, like [CCSS01].

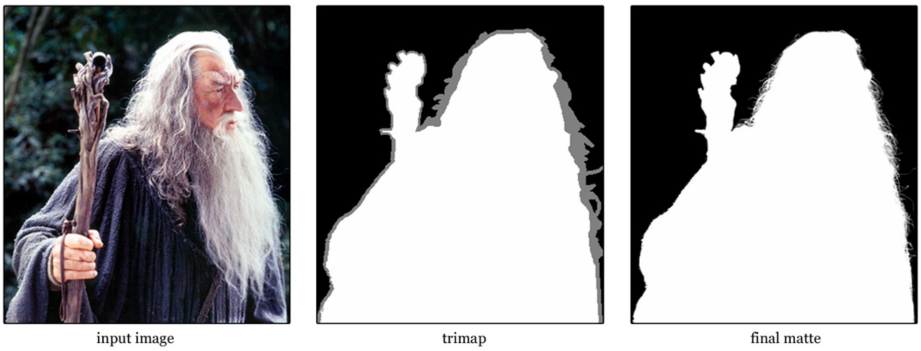


Figure 2: An example of a trimap for an image: the background marked by the user are represented as black areas, the marked foreground as white, the left over areas (the unknown area) are represented as grey fields (source: [Chu06], edited).

3 Related Work and Approaches

There exists a wide variety of keying approaches, each providing advantages and disadvantages for certain keying situations. An excellent classification of different keying and compositing methods was listed by [Chu04, p.13, table 1.1]. In this section the most common keying techniques will be presented. Since the film industry is mainly using blue- and greenscreen footage, we will mainly focus on keying approaches for a controlled backgrounds.

3.1 Luma Keying

One of the most simple concepts is called luma keying [Bri99, p.81]. Since most video signals (like PAL, NTSC, SECAM [Wik06] or DV Y'CbCr411 [Con95]) provide brightness information (luminance) independent from color information (chrominance) it is relatively easy to generate a matte based on the luminance data only. If the image is represented in RGB color space it is helpful to convert it to HLS representation first to access luminance (L) directly. By setting a simple threshold it is possible to create a binary matte setting the pixel fully visible or invisible like figure 3 illustrates.

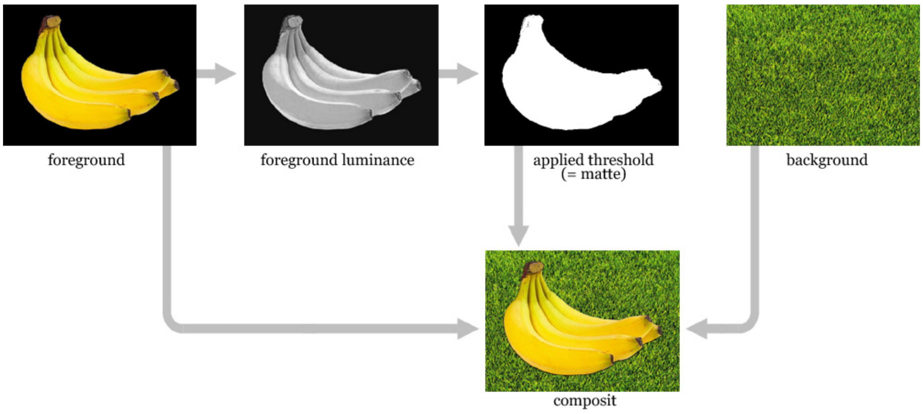


Figure 3: An example of a luma key.

More sophisticated keying software allows to define a softness and tolerance value, like shown in figure 4. Softness describes the gradual transition from fully visible to invisible (which is important to key semi-transparent objects like glass or water and to keep details like hairs partly visible). Tolerance describes the value range around a reference value that represent the visible maximum. The use of softness and tolerance ranges are not limited to luma keying but used by the following keyinh approaches as well.

Even if due to its color blindness a luma keyer is not useful for the majority of keying jobs, it may be helpful for special keying jobs like text extraction (black text on white background).

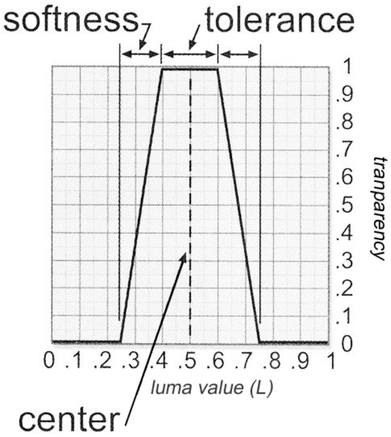


Figure 4: Tolerance and softness range for a luma key (source: [Wri02, p.18]).

3.2 Difference Keying

A difference matte describes "the matte generated by taking the absolute value of the difference between two images, one with the item of interest present and an identical one without the item of interest" [Wri02, p.302]. Difference keying is not limited to solid color backgrounds only but allows to shot the item of interest against any background. Figure 5 shows an example of a difference key: the absolute difference between a foreground and a background image pixel determines the matte pixel value.

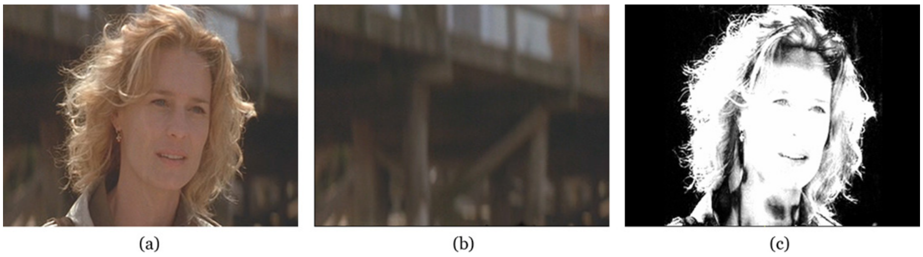


Figure 5: An example of the difference keying process: (a) the foreground image, (b) the background image and (c) the resulting difference matte (source: [Chu06], edited).

Flash keying is a special difference keying method for still sceneries. An object is recorded twice – first normally and then using a flash light. Due to the range limitation of the flash light, it lightens only the foreground object but not the background. Hence, it is possible to calculate the difference between the two images and create a mask out of it.

For high quality results the subtraction process of difference keying requires the very same background pixel values. However, most film, photography or video shots do not provide the exact identical image twice due to film grain, image compression, light changes or camera perspective changes [Bri99, p.82].

3.3 Chroma Keying

In the true sense of the term "chroma keying" we understand a matting process based on the color information of an image (H), while luminance (L) saturation (S) values are ignored. Since blue screen videos offer a more or less solid background color it is possible to process a key only based on the color information.

However, even professional blue screen studios cannot produce the exact same background color all over the image (due to natural limitations in lightning technique). Hence, the matte M is produced by setting a tolerance value T_h to the picked H value H_{key}. If the H value of an image pixel H_{pixel} is within this tolerance range, the matte value of this pixel M_{pixel} is set to 1, otherwise to 0.

$$M_{pixel} = \begin{cases} 1 & \text{if } (H_{key} - T_h) < H_{pixel} < (H_{key} + T_h) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Hence, a chroma key can be performed on "any arbitrary color" [Wri02, p.17]. However, the quality of chroma keys is usually very limited, especially when compressed footage is used. The popular consumer DV format for example compresses images in Y'CbCr 4:1:1[1] which leads to pixel artifacts in the color channel, like shown in figure 6. A chroma key based on this compresses H information generates a blocky matte.

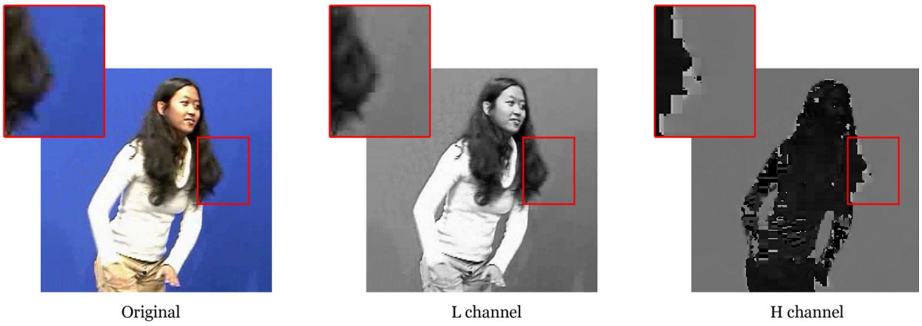


Figure 6: An extract from typical DV blue screen footage: due to the compression the color channel looks blocky.

3.3.1 HLS Keyer

A simple chroma key approach can be improved by including saturation and luminance information, as well with a defined tolerance value (T). A HLS keyer processes hue (H), lumninance (L) and saturation (S) values of an image and mixes the results to the final matte value (M). As described in section 3.1, this approach can be improved by applying an additional softness range to H_{matte}, L_{matte} and S_{matte}.

[1] 4:1:1 indicates that for each 4 samples of luminance there is only 1 sample each for R-Y and R-Y signals. 4:1:1 is the DV compression for the American (NTSC) market, in Europe (PAL) DV is compressed in 4:2:0.

$$H_{matte} = \begin{cases} 1 & \text{if } (H_{key} - T_h) < H_{pixel} < (H_{key} + T_h) \\ 0 & \text{otherwise} \end{cases}$$

$$L_{matte} = \begin{cases} 1 & \text{if } (L_{key} - T_l) < L_{pixel} < (L_{key} + T_l) \\ 0 & \text{otherwise} \end{cases}$$

$$S_{matte} = \begin{cases} 1 & \text{if } (S_{key} - T_s) < S_{pixel} < (S_{key} + T_s) \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{aligned} M_{pixel} &= \alpha H_{matte} + \beta S_{matte} + \gamma L_{matte} \\ &\text{where} \quad \alpha + \beta + \gamma = 1 \end{aligned} \tag{4}$$

Even if [Wri02] points out the limits of the HLS keying approach ("hard edges that require blurring, eroding, and other edge processing" [Wri02, p.17]) there are several commercial effect software packages using HLS keyers. The software company for video and computer graphics *discreet/autodesk*[2] has included two HLS keyers into their video effect software *Combustion*[3]: the *Discreet Keyer* and the *Diamond Keyer*.

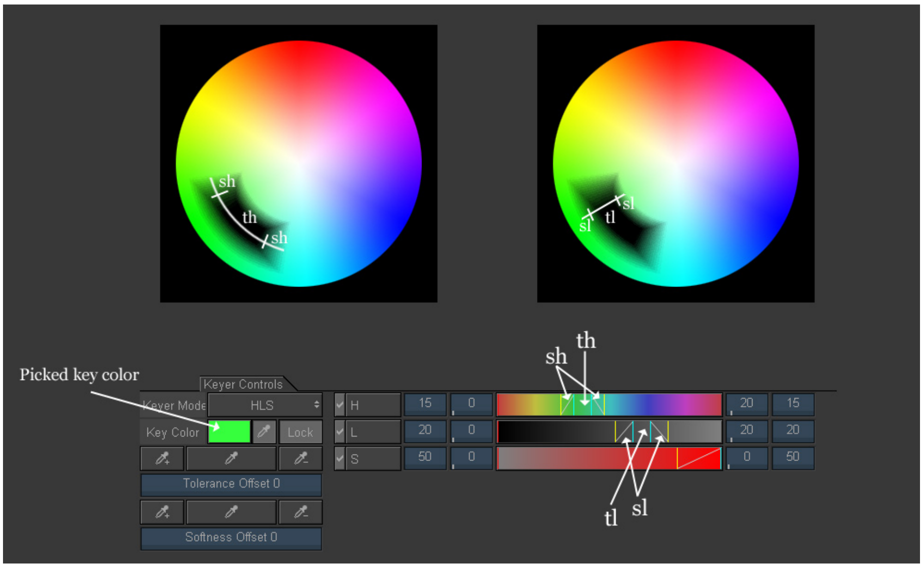
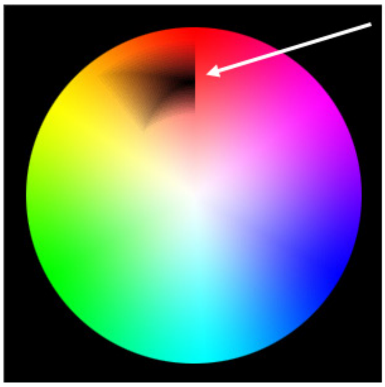


Figure 7: The Discreet Keyer performing a HLS key on two color cycle images.

The Discreet Keyer offers keying in several color spaces, beside HLS, YUV as well RGB and RGBCYML. As shown in figure 7, the HLS option presents a simple user interface with three color bars (H, L and S) with sliders for tolerance and softness. While testing the demo version of this software it turned out the discreet keyer module does not consider the cyclical H range. As figure 8 proofs, this "bug" affects the keying result significantly when a red background should be keyed out.

[2] `http://www.autodesk.com` , accessed July 2006

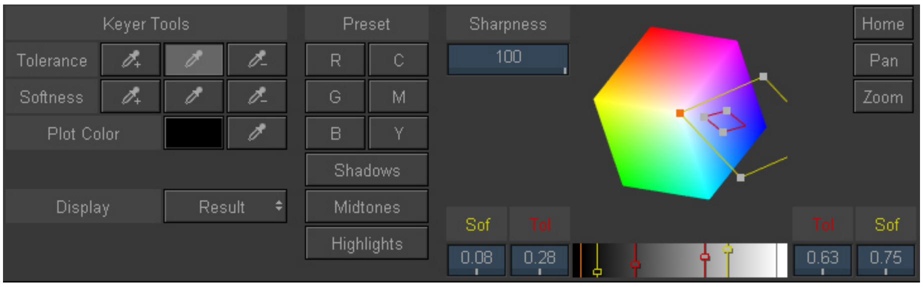
[3] Trial version: `http://www.autodesk.com/combustion-trial` , accessed July 2006



A key on red color is cut off at 360 °

Figure 8: The Discreet keyer performing a key for red color (black area): the software does not consider that color information is cyclical which results in a cut of the keyed red area.

The core of the Diamond Keyer user interface is a vectorscope displaying *chrominance* – the combination of color (H) and saturation (S) information. By selecting a preset or sampling a background pixel the Diamond Keyer creates an tolerance and softness area within the chrominance space in the shape of two diamonds (compare figure 9). In a second step the wanted luminance tolerance and softness range can be changed using bar sliders. The color cycle bug of the Discreet Keyer does not occur using the Diamond Keyer.



Sharpness 100

Display Result

Sof 0.08

Tol 0.28

Tol 0.63

Sof 0.75

Figure 9: The Diamond Keyer user interface.

3.3.2 Term Definition and Differentiation

The term “chroma key” is nowadays used for a multitude of approaches that are actually not pure chroma keys. The reason for this unclear use of the term may be the digital representation of an image in the most popular RGB color space. Keying algorithms that process an image internally in RGB work with the three color channels - and not with luminance or saturation information. Although these keyers are called “chroma keyers” by the producer or company their approach differs clearly from the described chroma key methods. Because of this reason the following approaches are not listed as Chroma Keyer but as independent techniques.

3.4 Color Difference Keying

The color difference method was invented by Petro Vlahos in 1964 [Sey05]. His method was originally developed for chemical and optical color processing of film material. The core idea of the color difference algorithm for an image F is the assumption to determine the transparency value based on the difference between the color channels R, G and B (see figure 10). For blue screen footage, it is determined by the difference of the blue channel value (F_B) to either the red (F_R) or green channel value (F_G)[4]:

$$\alpha = F_B - MAX(F_R, F_G) \tag{5}$$

Originally invented for creating mattes using film material (and not digital images), this approach was later implemented in hardware for analogue video signals and since 1995 also in software distributed by Vlahos founded company Ultimatte[5]. [SB96] shows in detail how Vlahos core idea was improved over time. Hence, the Ultimatte company holds a multitude of patents concerning the improvements of the color difference approach ([Vla82], [Vla83], [Vla86]). This may be the reason why Ultimatte is the only company that offers professional keying solutions based on the color difference approach.

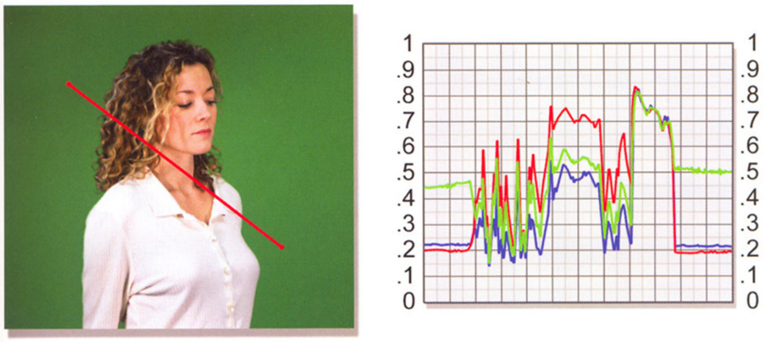


Figure 10: A color channel slice graph through an image (source: [Wri02]). The color difference method calculates the matte based on the difference of R, G, and B value.

3.5 3D Keying

3D Keying describes matting processes based on the image representation in a three dimensional space like RGB or HLS. Figure 11 shows how an image can be represented into the 3D space by plotting each image pixel as a point into the RGB space. The image background is represented as a (blue) compact pixel cloud, whereas the foreground object shows a longer pixel area (ranging from dark red over brown to light yellow).

[4] As [Wri02] points out, the resulting matte usually has to be scaled up to fit the value range from 0.0 - 1.0 (or 0 - 255 respectively).

[5] http://www.Ultimatte.com , accessed July 2006

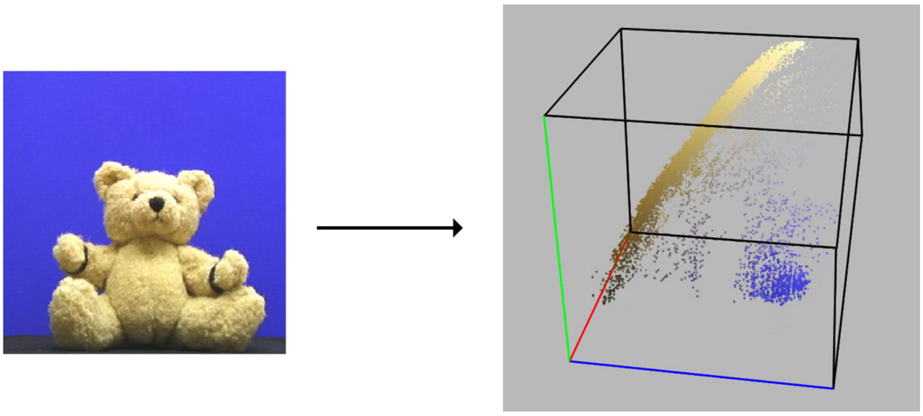


Figure 11: An RGB image plotted into 3D space (source: [Chu06], edited).

The main idea of 3D keying is to use surfaces and 3D shapes to separate the background pixel cloud from the rest of the image. In [SB96] these approaches were named "separating surface models" [SB96, p.266]. Usually, two separation shapes are required: shape S_1 defines the tolerance space, covered by shape S_2 defining the softness area. Like explained in [Pho05], the most simple approach is to place a sphere (S_1) in this space covering the background pixel cloud, representing the tolerance area like figure 12 illustrates. All pixels within this sphere are set fully invisible. The coordinate of the sphere can be created by averaging background pixel values the user samples.

To gain a soft matte for semi-transparent pixels a second, bigger sphere (S_2) can be located at the same reference point, representing the softness area. As figure 12 shows, pixels located within S_1 are set fully transparent, pixels outside S_2 are fully visible, and pixels within S_2 but outside S_1 are semi-transparent. Semi-transparent values are calculated based on the spacial distance of a pixels coordinate and either the reference point or the surface of S_1 and S_2.

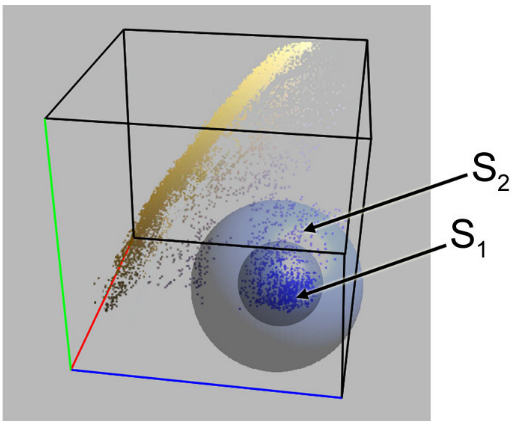


Figure 12: The background pixel cloud covered by the inner tolerance sphere S_1 and the outer softness sphere S_2 (source: [Chu06], edited).

Generally, any 3D shape can be used for this separation, but mostly convex hull objects are used to create S_1 and S_2. Therefore, user interaction is required: several background pixels are marked so that a convex hull is modified to cover them. By marking foreground pixels the sampled pixel values are extracted from the convex hull (see figure 13 and 14).

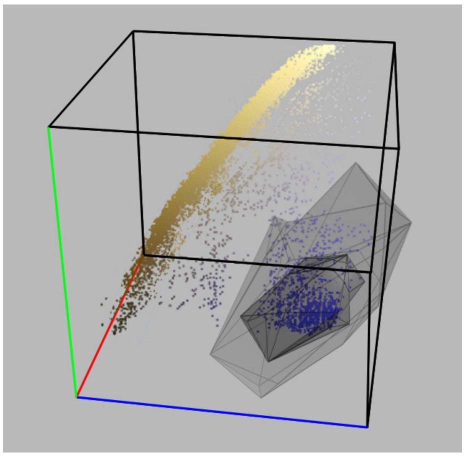


Figure 13: The background pixels covered by convex hull objects.

Currently, there are two professional applications on the market performing 3D keying with separation surfaces: the *Primatte Keyer* by Imagica[6] and the *Modular Keyer* by discreet/autodesk. They differ in 3D shape handling and user interaction.

The Primatte Keyer was originally developed by Yaz Mishima in 1992 [Sey05] and presented at the 8th NICOGRAPH Conference. Primatte uses polyhedron shapes with 128 faces to separate pixels, like figure 14 illustrates. Instead of only two shapes for tolerance and softness, the Primatte algorithm includes an additional shape for defining the space where color spill is suppressed. The shape refinement process is done in the background and not visualized when running the software.

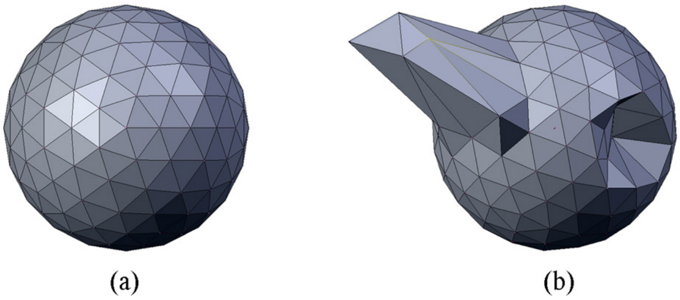


Figure 14: An example of polyhedron shape with 128 faces: unmodified (a), and modified after user interaction (b).

[6] http://www.imagica.co.jp , accessed July 2006

The Modular Keyer was the first 3D Keyer produced by discreet/autodesk in 1998. The developers reacted on the demands of the artists and operators using their visual effect software – the need to have a fast and precise keyer that gives direct response [Sey05]. “The Modular Keyer shows graphically for the first time the actual 3D histogram and allowed for direct manipulation of the 3 dimensional point cloud solution” [Sey05]. Figure 15 shows an example of the color space visualization including the 3D selection shapes.

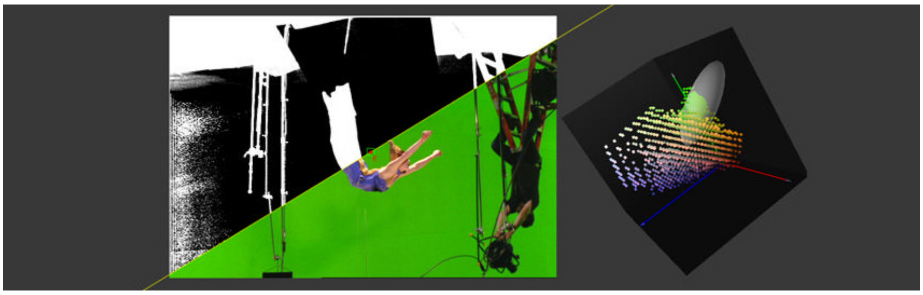


Figure 15: The Modular Keyer visualizes the image in a 3D color cube including the 3D selection shapes (source: [dis06]).

However, the modeled convex hull shapes are too complex for a fast calculation and responds. Hence, the Modular Keyer automatically “calculates the smallest possible ellipsoid that includes all selected points; this is called the minimal shape” [Lot03, p.71]. Like shown in figure 16, the convex hull object is covered by a minimal ellipsoid shape that is more simple and faster to calculate.

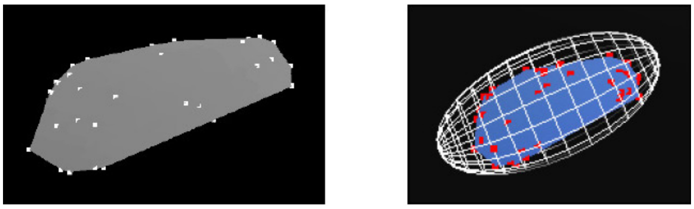


Figure 16: To reduce calculation time the Modular Keyer creates a minimal ellipsoid from the convex hull shape (source [Lot03, p.76]).

3.6 Alternative Keying Methods

The approaches listed above all focus on conventional recording (e.g. with a conventional video camera) and creating a mask using image processing. Beside them, there exist keying techniques employing special recoding hardware to enhanced the source footage. Gathering additional data of the set raises the information we can use to solve the general keying equation (see equation 2).

[BE00] lights the set with polarized light which is usually reflected unpolarized by skin or normal surfaces, but reflected polarized by special textiles or material. The difference of the reflected signal is used to create a matte. The advantage is the color independency of the foreground object to the background. It is not required to light the set precisely, to avoid certain colors for the foreground object or to pay attention to color spill. The disadvantage is the high financial costs for the reflection material and the recording-/processing hardware of the polarized signal.

Another approach uses infrared light to calculate the distance of each pixel to the camera, so that this technique is also called “z-keying” [KOY+95]. Using the depth information it is possible to set a threshold and to select only objects according to the wanted z-range.

The method used by [YNH03] records also the emitted temperature of objects and creates a matte by separating warm from cold image regions. The disadvantage of “thermo matting”is that only living objects can be keyed out and fine details like hair are not correctly keyed.

3.7 Summary

To sum up the described keying approaches table 1 presents an overview.

Approach	Principle	Software
Luma Keying	Calculates pixel transparency according to its luminance value	Included in most video effects software like *AfterEffects* or *Combustion*
Difference Keying	Calculates the matte by subtracting the foreground image from the plain background image	Included in most video effects software like *AfterEffects* or *Combustion*
Simple Chroma Keying	Calculates matte based on hue difference of a pixel to the given keying value	Included in consumer video editing tools like *Ulead MediaStudio*
HLS Keying	Calculates matte based on difference of pixel value to keying value regarding hue, luminance and saturation	*Combustion Discreet Keyer*, *Combustion Diamond Keyer*, *shake* or *AfterEffects*
Color Difference Keying	Calculates a pixel transparency value based on the difference of its red, green and blue value	*Ultimatte*
3D Keying	Represents all pixels of an image in 3D space (RGB), separates them using 3D shapes and calculates their transparency based on the spacial distance to a keying value	*Primatte* and *Autodesk Modular Keyer*
Alternative Approaches	Other methods record additional data of the set using infrared light, heat measurement or polarized light	not yet implemented in commercial software

Table 1: An overview of most common keying approaches

The players on the visual effect software market can be roughly clustered into three groups as follows:

1. **Professional effect systems** – like *discreet/autodesk Flint*, *Flame*, *Inferno*, *Quantel iQ* or *Avid Media Composer*. These high-end products are mostly sold including a specially configured (UNIX) hardware, hence they are called "systems". They often integrate a self-developed keying module based on color difference or 3D keying and enables the user to set a variety of options for the keying process like matte refinement tools or a complex keying GUI representation.

2. **Consumer effect software** – like *Adobes After Effects*, *discreet/autodesk combustion* or *Apple shake*. These software packages often provide HLS keyers and mostly offer a plug-in interface for more sophisticated keying modules.

3. **Keying plug-in distributors** – like *Ultimatte* or *Primatte*. The developers of the color difference method and the 3D keying approach offer their software as add-ons for most effect software packages.

Taking the products of the visual effect software industry as an indicator what keying approaches provide the most promising results we can differentiate between two main approaches. Over the last years the visual effects industry has specialized the three main keying approaches HLS keying, color difference keying and 3D keying. Even if there exists several keying approaches with different advantages and disadvantages, it seems that the industry prefers these three keying tools.

The reason for this could be the the long tradition of blue- and greenscreen technique that has set a standard for keying for the film industry. This fact excludes alternative keying methods for daily keying production in film industry. Furthermore, keying approaches with many requirements for recording like difference keying or thermo keying seem to be inferior to other, more flexible approaches due to their limitations. The market seems to have more demand for keying tools that offer a broad field of application than for tools offer solutions only for specialized situations.

4 TZI Keyer

Due to the relatively new approach of 3D keying this work concentrates on implementing such a matting application. Since digital video (DV) equipment is used by the majority of semi-professional users, the application should offer keying of typical DV footage. With “typical” it is refered to not perfectly lighted background footage recorded in small studios or in front of foldable blue or green screens. The software requirements covered

- keying of a self produced DV blue screen footage using 3D keying methods,
- automatic determination of the keying color in order to minimize the user interaction,
- decrease the blocky effect of DV compression,
- automatic removal of color spill,
- controlling over a simple GUI (front / back / matte / composite image),
- rendering of the composite and matte into new video files,
- processing with 32 bit instead of usual 8 bit in order to gain more precise mattes,
- outputting the keying matte and final composite as video file.

In the following, solutions and results for these requirement aspects are explained and the final keying software is presented.

4.1 Keying Software Architecture

The architecture of this software is method-based, not object-oriented. A segmentation of the keying process into small units and methods offered enough control and flexibility for the implementation. Beside this, an object-oriented concept could cause problems for future integration in other software. For example, our method-based architecture software is more comfortable to convert into a plug-in for Adobe AfterEffect than converting object-oriented software.

Table 2 gives an overview of the implemented functions and their purpose. The program flow is illustrated in figure 17.

Since digital video footage usually has 8 bit per channel (24 for RGB image), most keying tools calculate the matte based on 8 bit internally. To create high-quality keys, TZI Keyer process is realized with 32 bit precision. Input videos and images are converted and processed within a value range of 0.0 – 1.0.

The application was implemented in C++ and includes the open source computer vision library *OpenCV*[7] which offers an easy accessing and processing of image data. The source code was compiled using the *cygwin* compiler.

[7] `http://www.sourceforge.net/projects/opencvlibrary` , accessed July 2006

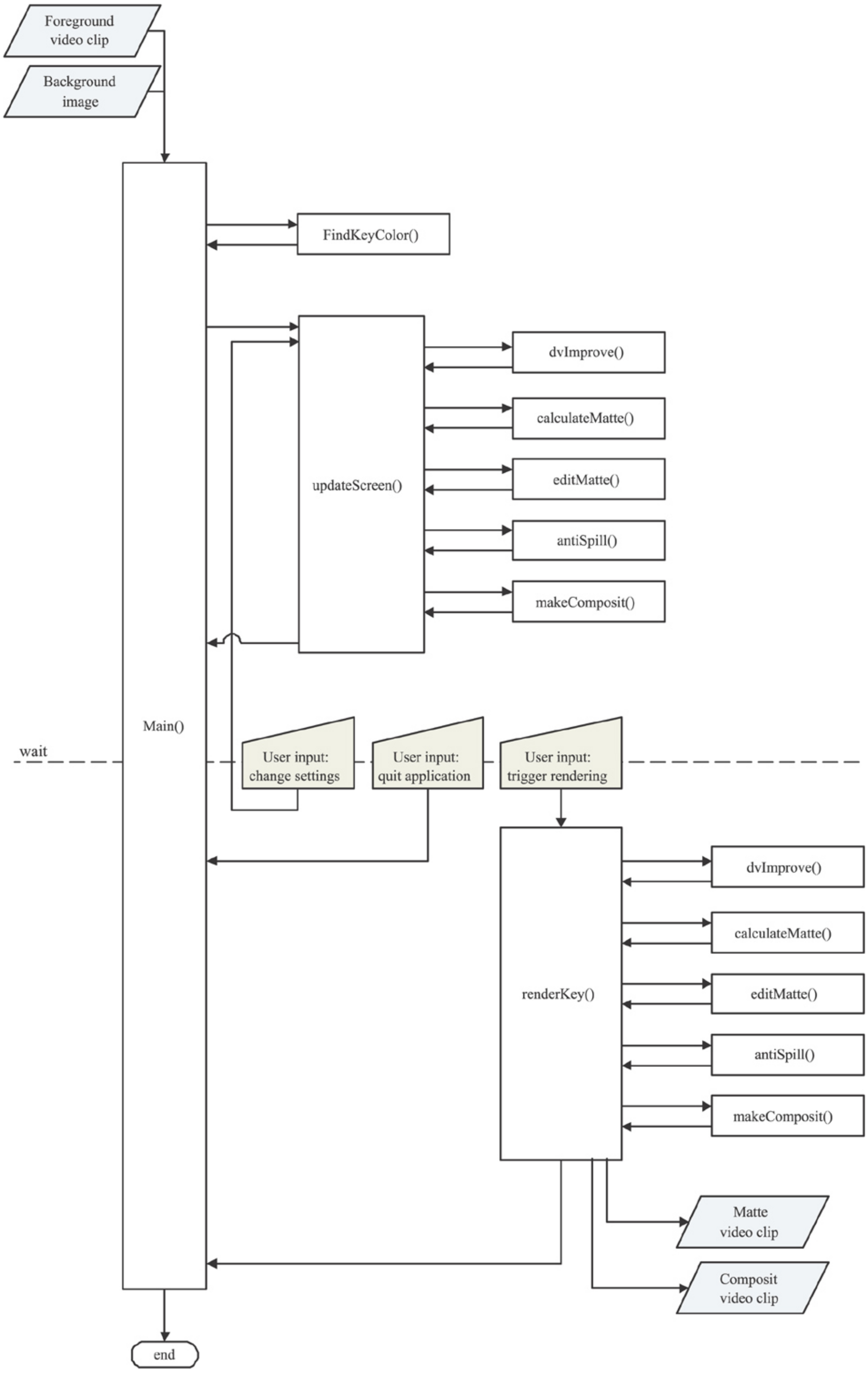


Figure 17: The schematic program flow of TZI Keyer.

Method	Purpose
main()	loads the footage and initializes the keying process
findKeyValue()	determines the keying color for the video clip (see section 4.2)
dvImprove()	optimizes DV footage for keying (see section 4.4)
calculateMatte()	processes the 3D key and creates a matte (see section 4.3)
antiSpill()	reduces color spill on the foreground object (see section 4.5)
editMatte()	various post processing features to edit the matte (erode, dilate and gamma correction)
makeComposit()	creates a composition of the foreground and background image based on the matte
updateScreen()	shows input images and attribute sliders for manual changes on a GUI (see section 4.6)
renderKey()	renders the composite and matte into video files of the output folder
convertRGB2HLS()	converts a RGB value set into HLS color space

Table 2: The methods implemented in TZI Keyer software and their purpose.

4.2 Key Color Finder

To reduces the user input, the application can determine the keying color of the foreground image automatically. Since the keying color of footage can be any color we cannot assume a certain key color (like blue tones only). Instead, the keying color is calculated based on a color histogram. The first image of the video sequence is converted into HLS color space and a histogram of the H channel is calculated. This histogram represents the distribution of color values of all pixels in one image. Figure 18 illustrates an example of a color histogram.

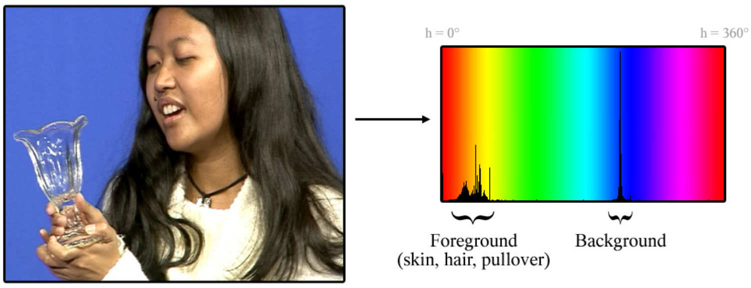


Figure 18: A color histogram of a sample keying image.

In most situations the background color is the most occurring H value. Hence, the keying color is determined by selecting the maximum of the color histogram. Due to film grain or compression artifacts, the background often is not perfectly solid. Like figure 18 shows, the background is then distributed over a small range around the maximum value. Considering this, all pixel of the image are selected holding a H value within a certain tolerance around the histogram maximum value. Since the keying application is based on RGB, R-, G- and B values are averaged of the selected pixel which results the averaged keying color.

4.3 3D Keying Module

As described in section 3.5, 3D keying requires two objects to cover pixel values in 3D space. In contrast to professional systems, we did not implement convex hull objects but used simple spheres. The calculated keying color is used as reference point for the two spheres. Like figure 12 illustrates, the inner sphere defines the tolerance area, while the outer sphere represents the softness area.

The visibility (v) for a pixel is calculated based on its Euclidian distance to the reference point, like figure 19 illustrates. Since the 3D RGB space is normalized to 1 on its x-, y- and z-axis, the maximum distance between two points is $\sqrt[2]{x^2 + y^2 + z^2} = \sqrt[2]{1 + 1 + 1} = \sqrt[2]{3} \approx 1.732$.

In contrast to hull objects, a sphere has the peculiarity that all points on its surface have the same distance to the center point. This allows visibility value calculation using the spheres diameter as global thresholds: diameter D_T defines the threshold up to where the visibility is set 0.0 (fully invisible), while diameter D_S defines the threshold from where the visibility is set 1.0 (fully visible). Between these thresholds the visibility is calculated linear according to the distance (compare figure 19).

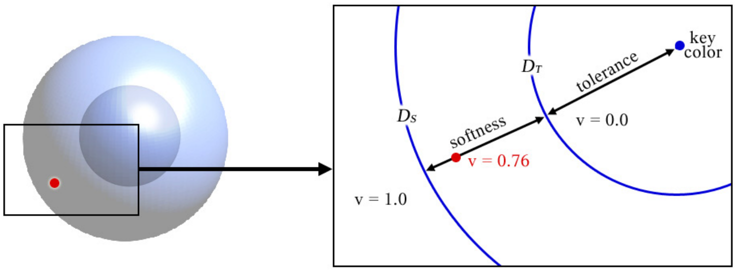


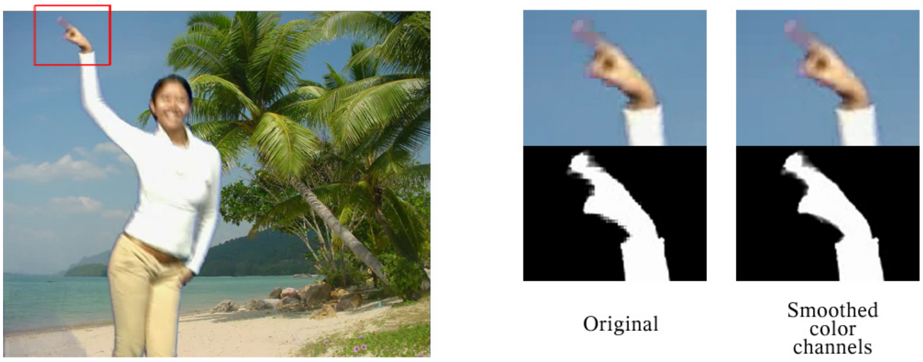
Figure 19: Visibility calculation for a pixel using spheres for tolerance and softness area.

The advantage of using spheres is that D_T and D_S are the same for all pixel, in contrast to hull objects where D_T and D_S can vary according to the hull object shape. Like this, visibility calculation can be performed fast and easily. The disadvantage of this approach is that areas can be only included or excluded by modifying the diameter of a sphere (which has an influence on the whole surface), which means less control and less precise results than modifying only certain parts of the object like hull objects allow it.

4.4 DV Improvement

As described in section 3.3, blocky keying results can occur due to artifacts of DV compression. To improve the keying results, TZI Keyer offers a “DV improve option”. Based on the concept explained in [Thu00], the compressed and blocky channels of an image are smoothed. This method seems to be very helpful to improve the keying results for DV footage, so that consumer products like *dvMatte*[8] or *Primatte 3.0* have included this feature into their keying software.

Internally, the keying application converts the footage into Y’CbCr color space since this is the original recording color space of DV footage. A gaussian blur of 3x3 pixel kernel is applied to the color channels Cb and Cr. This leads to softer matte edges and reduces the artifacts, like the comparison in figure 20 shows.



Original

Smoothed color channels

Figure 20: A sample video before and after applying the “dvImprove” option.

Since DV footage is use interlaced frames, it is theoretically more precise to blur both half-frames independently. Like this, for example motion blur can be better preserved.

4.5 Auto Spill Removal

The term “color spill” describes light reflected by the (blue or green) background shining at the foreground object. This changes the color of the border area – especially fine details like hair or semitransparent objects are often tinted into the background color. Since color spill is a basic and crucial problem of keying, consumer and professional keying applications have integrated antispill tools.

The approach used in our software decreases the saturation of the affected color spill areas automatically. According to the color tone (H), the saturation of a pixel is adjusted. This concept is similar to a look-up-table. If H_{pixel} is within a tolerance range around the keying color H_{key} (=the pixel is a background pixel), its S value is decreased. If the H_{pixel} is not within this range (=the pixel is a foreground pixel) its S value remains unchanged. The decrease of S_{pixel} depends on the difference of H_{pixel} to the keying color H_{key} value.

[8] `http://www.dvmatte.com` , accessed July 2006

Since hue describes a color cycle, the antispill concept has to deal with *overlapping color ranges*: as described in section 3.3.1, a color range that overlaps the border of H (< 0 or > 360) has to be mapped to the other side of H.

This problem was solved by using the degree function sine. Sine repeats with a fixed interval which was set to 1 in order to be used as a look-up-graph for the antispill method. To place the maximum at the H value of the keying color, $h_{Key} + (0.25 - h_{pixel})$ shifts the function maximum to the wanted value.

Represented on a coordinate system, the x-axis represents hue while the y-axis represents the desaturation factor. Values on both scales are limited to 0.0 and 1.0. In order to create a tolerance value around the functions maximum, its amplitude was increased (t). The bigger t is, the bigger is the amplitude. Values above 1.0 are limited to 1.0, like this a tolerance is created around the maximum of the function. Test showed that it helps to shift the function vertically ($t - 1.5$) in order to decrease the tolerance area compared to the rest of the graph. Figure 21 shows the graph of the modified sinus function:

$$f = t * \sin(2.0 * \pi * (h_{key} + (0.25 - h_{pixel}))) - (t - 1.5) \tag{6}$$

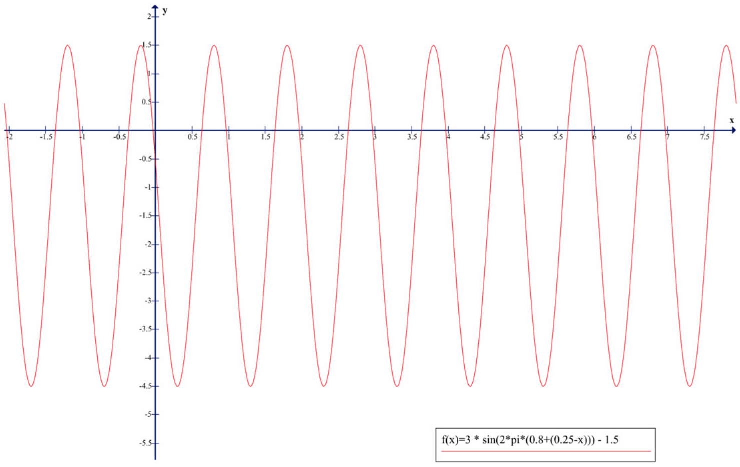


Figure 21: The sinus function used for the antispill method.

To perform the antispill process, the sinus function uses the H value of a pixel to return the corresponding decrease factor. Due to t the range of the returned values vary. Hence, the factor has to be limited to the range of 0.0 – 1.0 (compare figure 22). The pixels S value is then decreased by this factor.

As the comparison in figure 23 proves, automatic spill correction significantly reduces color spill on hairs or semitransparent objects. In test it turned out that it furthermore helps to improve the visual impression of scenes where background pixels are not fully keyed away. The left over background pixels lose their color and become gray – like this they are more neutral and do not peak out of the composite image.

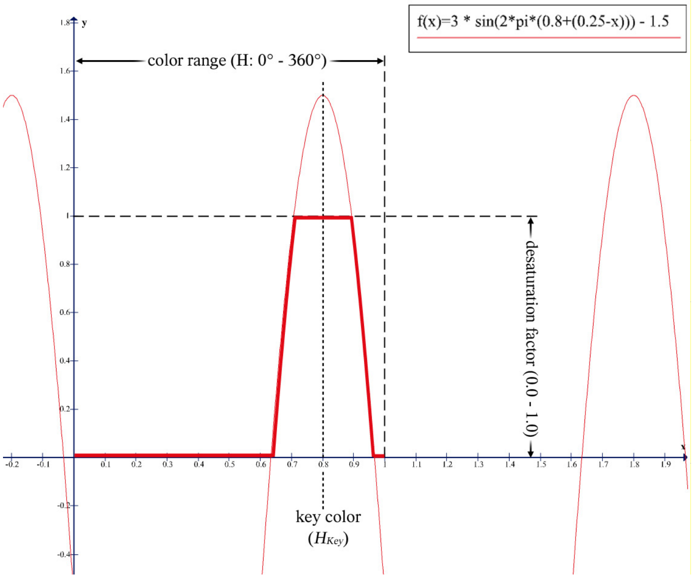
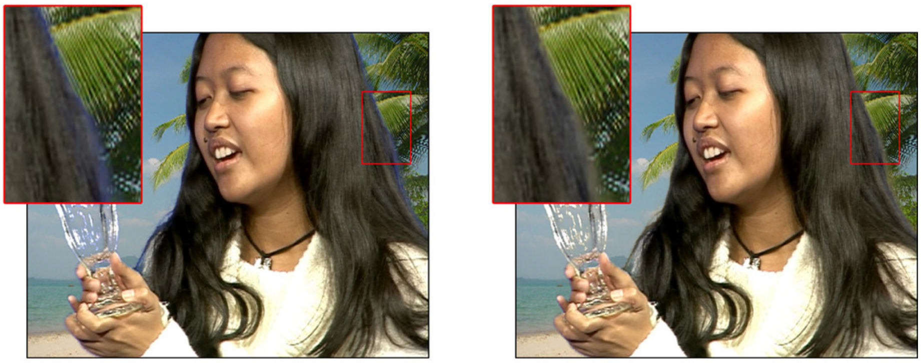


Figure 22: The sinus function used for the antispill method.



without spill suppression

with automatic spill suppression

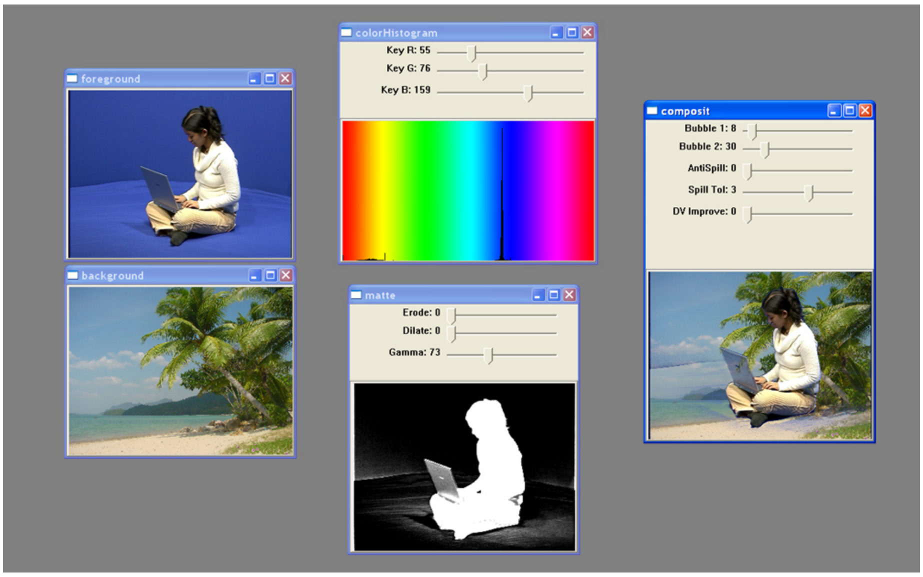
Figure 23: DV footage keyed without and with automatic spill suppression.

4.6 Keying GUI

To improve and simplify the control of the keying process it is necessary to give visual feedback of the keying results (for example showing the matte), and to offer a possibility for manual change of parameters of the keying process (for example editing the values for sphere diameters).

A GUI was developed displaying the foreground, background, matte and composite image, like shown in figure 24. The color histogram of the first image of the video sequence is displayed as well. Furthermore, the GUI allows user interaction by using sliders to control parameter values. The sliders are set to the element they influence, i.e. matte control variables to the matte window, or a manual keying color picker to the color histogram window.

As the flowchart in figure 17 illustrates, the parameters can be changes for the first image of the video sequence. Pressing the 'r' button opens the render menu, where the codec for rendering the composite and matte video is selected. For each keying render process a new output folder is created with a label of the current date and time. The keying process can be quited by pressing 'q'.



Key R: 55
Key G: 76
Key B: 159

Erode: 0
Dilate: 0
Gamma: 73

Bubble 1: 8
Bubble 2: 30
AntiSpill: 0
Spill Tol: 3
DV Improve: 0

Figure 24: The user interface of the keying application.

5 Evaluation

In order to know "how good" our keying software works, it was evaluated against other software according to certain criteria. The aspects of interest are described in section 5.1. In section 5.2 the choice and setup of the alternative keying systems are explained. The collected test results are presented in section 5.3.

5.1 Evaluation criteria

The evaluation focuses on testing two aspects of interest about the performance of a keying system: (A) the required time to render a keying job, and (B) the quality of the resulting matte and composite.

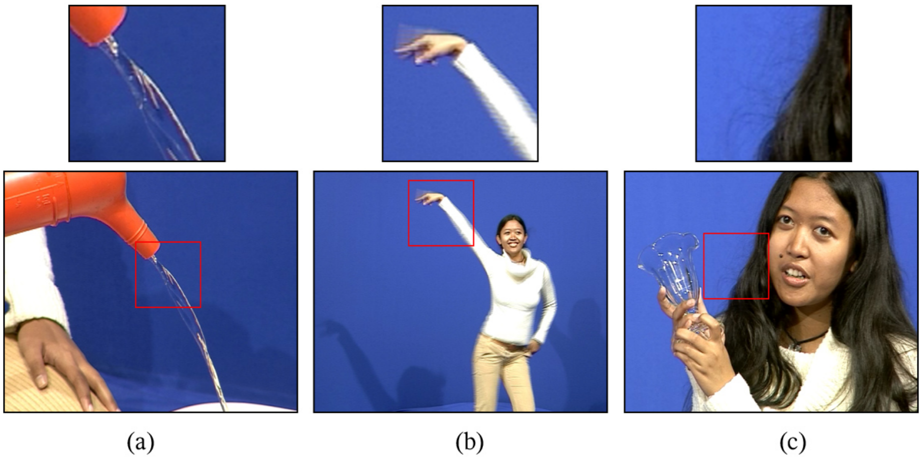
Hence, the evaluation process is divided into two parts. The first testing session (A) measures and compares the time required by the keying systems to perform the same keying job. Beside this, the average time of our software for calculating the same keying jobs with different options was measured.

The second testing session (B) is comparing the results of the keying systems of the same keying job. Hereby, four critical aspects of keying systems were tested: *transparency* requires handling of semitransparent objects, *motion blur* requires processing of unsharp outlines of an object, *fine details* require matting of small and semitransparent elements, and *reflections* require color correction of objects with color spill.

To evaluate keying systems according to these four critical aspects, it was necessary to use appropriate video footage. Hence, the following eight sequences we recorded in the university blue screen studio using a 3-CCD-chip DV camera (sample screenshots are shown in figure 25):

1. *Transparency*
   1.1 – water poured out of a watering can
   1.2 – an actress holding a crystal glass
2. *Motion blur*
   2.1 – an actress jumping and dancing (interlaced frames)
   2.2 – the same sequence but with not interlaced frames.
3. *Fine details*
   3.1 – an actress with open hair sitting
   3.2 – an actress with open hair turning the head and hair fast
4. *Reflections*
   4.1 – an actress working with a metallic notebook
   4.2 – an actress balancing with a metal bar

The footage was captured via Firewire interface and trimmed to sequences of 80 – 140 frames. Since OpenCV does not support DV format, the footage was stored uncompressed, with 720 x 576 pixel, 24 bit and no audio. Using another codec would not make sense, since it would recompress the footage and take away the typical attributes of DV footage, for example the 4:1:0 compression artifacts shown in figure 6. Uncompressed footage keeps the image structure so that DV attributes are remained. The resulting test collection requires 1.26 GB storage.



(a) (b) (c)

Figure 25: Screenshots of the testing sequences: 1.1 (a) / 2.2 (b) / 3.1 (c).

5.2 Evaluation setup

The selection of alternative keying software for this evaluation was limited to commercial software. A comparison with professional systems (see section 3.7) was not possible, since they require special hardware. Most commercial keying software is offered as trial version online.

The video effect software discreet/autodesk Combustion[9] was used as platform for all alternative keys. Combustion includes several keying modules and can integrate third-party keyers via plug-in. The selected keyers offer the same features like TZI Keyer: manual tolerance and softness option, antispill option and matte postprocessing.

- **Combustion HLS Keyer** – an HLS keyer as described in 3.3.1, including a variety of options for antispill filtering and matte processing.
- **Combustion Diamond Keyer** – another HLS keyer. Since it does not include antispill options like the other keying modules, an antispill filter was added to equalize the testing conditions.
- **Combustion Primatte Keyer Plugin**[10] – a 3D keyer as described in 3.5, including antispill and matte processing options.

Although, a keying job is usually manually prepared and post-edited[11], or consists of keys for several image regions which are finally combined, the test sessions were performed by using the keyers like an image filter: the parameters were fine tuned manually and affected the whole image and sequence. All keying jobs were rendered on a Intel centrino$^{\text{TM}}$processor with 1.7 GH, 512 MB RAM and Microsoft® Windows XP®.

[9] Trial version: `http://autodesk.com` , accessed July 2006

[10] Trial version: `http://www.redgiantsoftware.com` , accessed July 2006

[11] Often unwanted areas of the image are manually masked out using "garbage masks". Like this the amount of pixel to be keyed are reduced, the process is fastened and optimized since unwanted pixels are not disturbing the calculation. Furthermore, the resulting matte can be retouched by animated paint tools. Like this resisting spots can be patched by hand.

5.3 Evaluation results

As stated, the evaluation was split into two parts: section 5.3.1 presents results of render time test sessions, while section 5.3.2 describes the comparison of the resulting mattes and composites.

5.3.1 Time comparison

A complete keying process consists of two steps: (1) the manual corrections of keying parameters, and (2) the following rendering. Since the time for fine tuning parameters depends on the know-how and experience of the user, it cannot be included to the temporal evaluation of several keying systems. Hence, only rendering time was taken into consideration.

Although on the average one frame is processed in under a second by all test applications, there are differences in render performance. Like table 3 and figure 26 show, our software performs faster than the other systems for nearly all test keys. Combustion Diamond is second fastest, while Primatte requires significantly more time to perform keys. This tendency is also proven by the average time, calculated based on the eight test key videos: TZI Keyer performs keying jobs 17% faster than the Primatte keyer. As figure 27 shows, Combustion Diamond is only slightly slower than our software (3%), while Combustion HLS requires 8% more time.

Sequence	Frames	TZI Keyer	Comb. HLS	Comb. Diamond	Comb. Primatte
1.1.	95	51	60	60	67
1.2.	107	63	78	61	72
2.1.	100	58	62	60	73
2.2.	100	56	63	60	74
3.1.	121	73	67	68	80
3.2.	83	49	48	48	60
4.1.	146	80	88	84	95
4.2.	98	56	62	60	69

Table 3: Render time (in seconds) to perform test keying jobs.

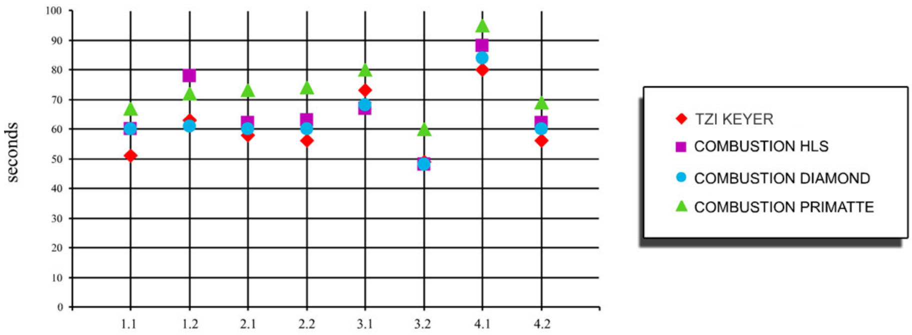


Figure 26: Render time (in seconds) to perform test keying jobs.

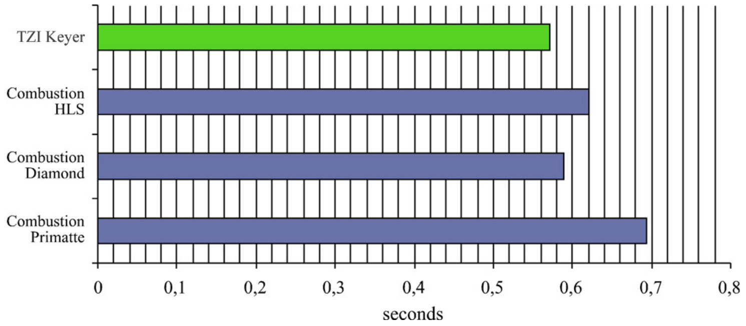


Figure 27: The time needed to perform a key process for one frame (averaged over all keying jobs).

This results can be explained with the application size: TZI Keyer is a stand-alone software, implemented and optimized for keying only. The alternative systems are run by Combustion, a complete video effect framework which offers a multitude of features. Compared to the "slim implementation" of our software, the render process of Combustion might be implemented more "open and complex".

The render time difference between Combustion HLS and Combustion Diamond surprises, since both keyers work based on HLS color space. However, Combustion HLS includes several more features to control and correct the matte. Even if not all of them were used for this evaluation, they might affect calculation time.

A reason why Primatte performs slowest in this evaluation might be the fact that it is used as plug-in, which might slow down the render process due to possible data conversion. Additionally, Primatte is a 3D keyer using complex 3D shapes to separate pixels in 3D space – this geometrical calculation requires costs significantly render time.

Furthermore, we measured the time needed by our software to perform a key with several options activated. Like figure 28 shows, every activated option requires additional render time, since additional image operations have to be processed. Antispill option requires the most additional render time (+38 %).

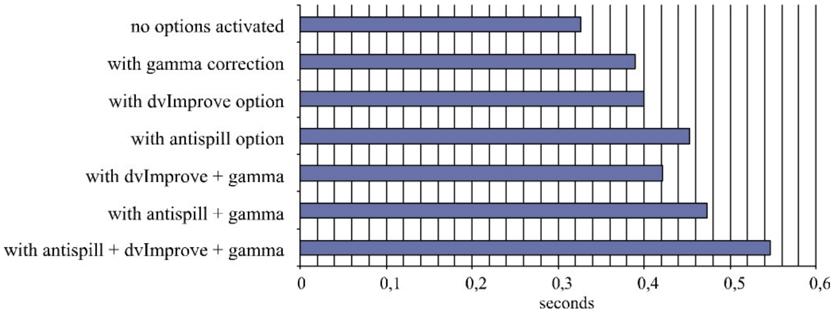


Figure 28: The averaged time of the TZI Keyer software needed to perform a key for one frame with several options.

5.3.2 Visual result comparison

A quality evaluation of the resulting mattes and composites is difficult, since there exist no reference material for the recorded footage. This means, a quality measurement based on mathematical calculations is not possible. Hence, a visual comparison of resulting matte and composite videos of all applications was performed according to the aspects of interest (see section 5.1). Although visual comparisons are subjective, they help to determine differences and tendencies.

In general, Primatte and Combustion Diamond keyer produce much softer mattes than TZI Keyer or Combustion HLS. Semitransparent image areas of Primatte mattes look blurred, which is not necessarily of disadvantage. Compared to this, the most important difference to mattes produced by our software is the lack of soft edges which makes them look very contrasty (compare figure 29). The main reason for this might be the different separation objects used for 3D keying: the 128 faced polyhedron of Primatte is more flexible and modifiable than the simple sphere shapes we use. Hence, background pixels are more precisely covered by Primatte.

Furthermore, it turned out that Primatte reduces black outlines automatically. These black outlines occur when DV cameras perform an automatic contrast correction of a bright object and a dark background. Primatte automatically softs these matte edges, which results in a more realistic integration of an object into the new background.

- *Transparency* – Primatte handles transparent element best. It is the only keyer that includes the crystal glass completely, while all others set parts of its rim invisible.
- *Motion Blur* – Primatte and Combustion Diamond handle motion blur best because they create very soft edges which helps for matting motion blur. Our software does not mask motion blurred area very well, so that for example a fast moving hand is cut instead of semitransparent. Even with "dvImprove option" enabled, TZI Keyer shows artifacts. Also, "dvImprove" does not show significant improvement of quality for not interlaced footage.
- *Fine Details* – Combustion Diamond and HLS create the best mattes for fine hairs. Due to Primattes automatic edge softening its mattes lose details so that fine hairs cannot be identified any more. Because of the missing softness in mattes of our software, several fine hairs are still preserved, while others are simply cut away.
- *Reflections* – Keying the metallic notebook seems to be the hardest job for all keying applications. Combustion HLS and our software did not succeed so that the notebook was only set semitransparent. Furthermore, TZI Keyer did not compete to key out the background completely. Combustion Diamond and Primatte were able to set the notebook visible, but only after long fine tuning of the keying parameters. However, the edges of the notebook are disturbed and shacking over time. Blue spill on the actress was suppressed by all applications so that no blue shade was left over.

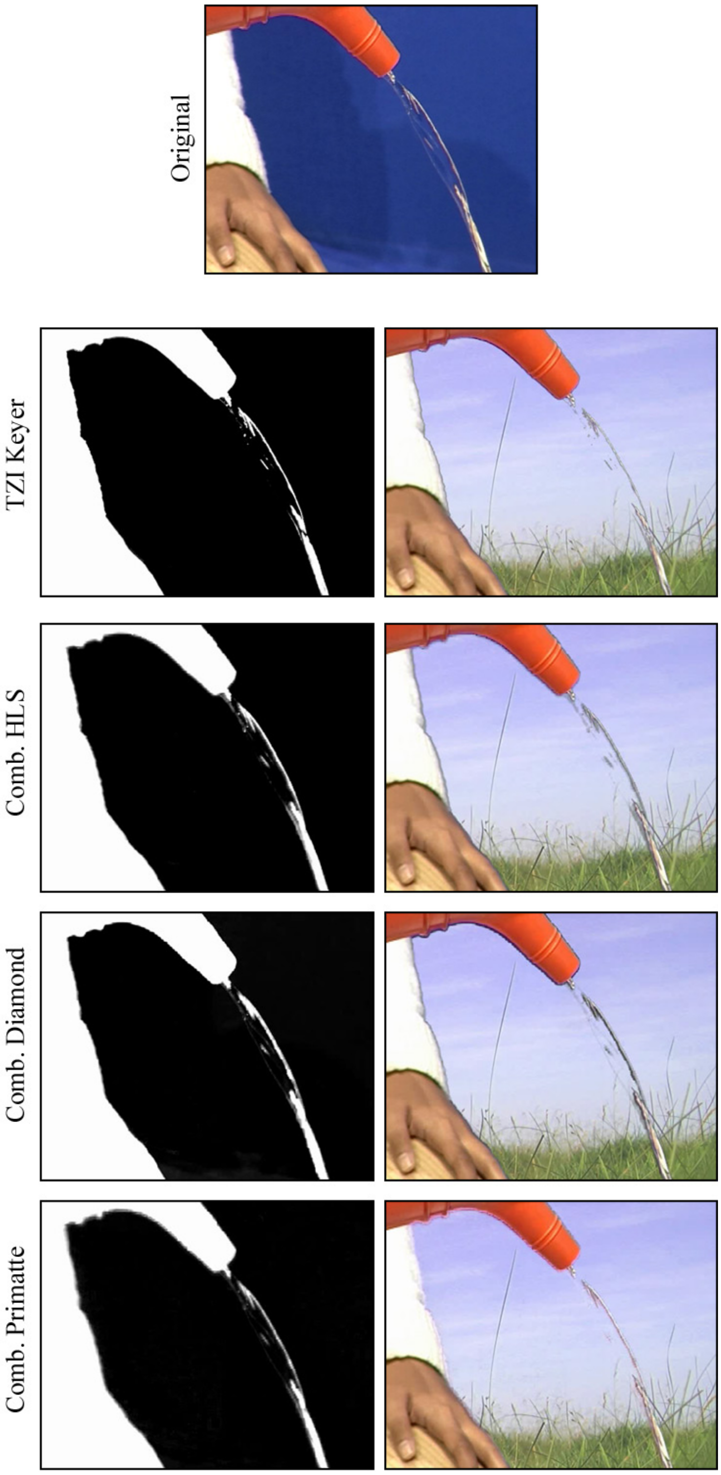


Figure 29: Resulting mattes and composites of four test systems for an evaluation key job.

6 Conclusion

This report focuses on keying – the process of separating an object from the rest of an image by creating a matte which represents transparency information about the original image. Therefore, the general keying problem is described and mathematical basics of a keying process are explained in section 2.

In section 3, state-of-the-art keying approaches are presented in detail. Since movie industry has specialized in keying green or blue screen footage, this overview focuses mainly on methods for controlled backgrounds. Furthermore, a classification of commercial keying systems on the market is presented and several major keying applications are described. Industry seems to prefer keying systems that apply for a broad field of keying situations than keying approaches that work only for special situations. Especially color difference keying, HLS keying and 3D keying meet these expectations.

Section 4 presents a self developed 3D keying application. Beside the general software architecture its features are described in detail. The automatic key color selection helps to determine statistically the optimal keying color. The keying calculation is based on spherical tolerance and softness areas in 3D space. "dvImprove" reduces artifacts of DV compression. To correct color spill on fine details and semitransparent objects "antiSpill" automatically despills the footage. All parameters can be controlled using a GUI.

Our software (TZI Keyer) was evaluated against three commercial keyers (Combustion HLS, Combustion Diamond and Primatte). Section 5 presents the results of the evaluation. Summing it up, Primatte outperforms the other test systems in regard of quality, however requires most render time. TZI Keyer performs fastest of all systems, but the rendered mattes lack of softness due to the use of simple 3D shapes for separating background in 3D space. Combustion Diamond renders slightly slower than our software, but creates high quality mattes close to Primatte results. Depending on how much weight each criteria is given, the best performer is either Primatte (more weight for quality than for render time) or Combustion Diamond (more weight for render time than for quality).

For future work, the lack of matting softness could be reduced by implementing convex hull elements instead of simple spheres as 3D separation objects. Furthermore, DV artifacts could be even better reduces using a bigger blur kernel. For a more sophisticated visualization, the GUI could show the images plotted into a 3D cube where user interaction gives live feedback of the modified 3D separation objects (compare Modular Keyer). Finally, a conversation of the source code into a Combustion plug-in would offer more equal conditions for further evaluation.

References

[BE00] Moshe Ben-Ezra. Segmentation with Invisible Keying Signal. In *IEEE Computer Vision and Pattern Recognition*, pages p. 32–37, July 2000.

[Bri99] Ron Brinkmann. *The Art and Science of Digital Compositing.* Morgan Kaufmann, 1999.

[CAC^{+}02] Yung-Yu Chuang, Aseem Agarwala, Brian Curless, David H. Salesin, and Richard Szeliski. Video Matting of Complex Scenes. *ACM Transactions on Graphics*, 21(3):243–248, July 2002.

[CCSS01] Yung-Yu Chuang, Brian Curless, David H. Salesin, and Richard Szeliski. A bayesian approach to digital matting. In *Proceedings of IEEE CVPR 2001*, volume 2, pages 264–271. IEEE Computer Society, December 2001.

[Chu04] Yung-Yu Chuang. *New Models and Methods for Matting and Compositing.* PhD thesis, University of Washington, 2004.

[Chu06] Yung-Yu Chuang. Matting and compositing (lecture presentation), 2006. `http://www.csie.ntu.edu.tw/$\sim$cyy/courses/vfx/06spring/lectures/handouts/lec10_matting.pdf` , accessed July 2006.

[Con95] Consultative Committee for International Radio. Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios, 1995. ITU-R Recommendation BT.601-5.

[Dat06] Internet Movie Database. All-Time Non-USA Boxoffice, June 2006. `http://www.imdb.com/boxoffice/alltimegross` , accessed July 2006.

[dis06] discreet/autodesk. Example screenshot of the Modular Keyer, 2006. `http://www.discreet.ru/news/imagelib/images/flame/ModularKeyer.tif` , accessed August 2006.

[KOY^{+}95] Takeo Kanade, Kazuro Oda, Atsushi Yoshida, Masaya Tanaka, and Hiroshi Kano. Video-Rate Z Keying: A New Method for Merging Images. Technical report, Carnegie Mellon University, 1995.

[Lot03] Helmut Lottenburger. Analyse professioneller Keyingtechnologien und -methoden (diploma thesis), 2003. Fachhochschule Kiel.

[PD84] Thomas Porter and Tom Duff. Compositing Digital Images. In *Computer Graphics*, pages 253–259, July 1984.

[Pho05] Photron. Why a 128-faced polyhedron?, 2005. `http://www.primatte.com/whitepapers/popup.cfm?id=9` , accessed July 2006.

[SB96] Alvy Ray Smith and James Blinn. Blue Screen Matting. In *Proceedings of SIGGRAPH 96*, pages 259–268, August 1996.

[Sey05] Mike Seymour. Art of Keying, November 2005. `http://www.fxguide.com/article314.html` , accessed July 2006.

[Thu00] Earl R. Thurston. Clean DV Bluescreen/Greenscreen Composites, September 2000. `http://www.neopics.com/bluescreen/` , accessed July 2006.

[Vla82] Petro Vlahos. Comprehensive Electronic Compositing System. U. S. Patent 4,344,085, August 1982.

[Vla83] Petro Vlahos. Encoded Signal Color Image Compositing. Patent 4,409,611, October 1983.

[Vla86] Petro Vlahos. Comprehensive Electronic Compositing System. U. S. Patent 4,625,231, November 1986.

[Wik06] Wikipedia. Broadcast Television System, 2006. `http://en.wikipedia.org/wiki/Broadcast_television_system` , accessed August 2006.

[Wri02] Steve Wright. *Digital Composition for Film and Video.* Focal Press Visual Effects and Animation, 2002.

[YNH03] Kazutaka Yasuda, Takeshi Naemura, and Hiroshi Harashima. Thermo-key: Human Region Segmentation from Video Using Thermal Information. In *ACM SIGGRAPH*, July 2003.