

Coded Projection and Illumination for Television Studios

A. Grundhöfer, M. Seeger, F. Häntsch and O. Bimber

Bauhaus-University Weimar, Germany

Abstract

We propose the application of temporally and spatially coded projection and illumination in modern television studios. In our vision, this supports ad-hoc re-illumination, automatic keying, unconstrained presentation of moderation information, camera-tracking, and scene acquisition. In this paper we show how a new adaptive imperceptible pattern projection that considers parameters of human visual perception, linked with real-time difference keying enables an in-shot optical tracking using a novel dynamic multi-resolution marker technique.

Categories and Subject Descriptors (according to ACM CCS): H.5.1 [INFORMATION INTERFACES AND PRESENTATION]: Multimedia Information SystemsArtificial, augmented, and virtual realities; I.4.8 [IMAGE PROCESSING AND COMPUTER VISION]: Scene AnalysisTracking

1. Introduction and Motivation

Virtual sets have become a standard technology for many modern studio productions. Blue screens and chroma keying enable the composition of computer generated backgrounds with the recorded image of moderators or actors. Several problems, such as realistic keying and matting, or spill compensation have been solved for creating professional effects. Some advanced systems even support tracking the studio cameras and the integration of rendered 3D graphics perspective correct into the recorded footage. However, virtual studio setups restrict television productions to a special recording environment, like a blue box. Integrating additional real props realistically is challenging. Although virtual studios have the enormous advantage of reducing the production cost, they still appear synthetic. It is problematic to apply them for shows with a live audience, and it remains difficult for actors and moderators to actually not see the composite content directly.

Augmented reality holds the potential of integrating computer generated graphics into video recordings of real studio settings. Synthetic backgrounds for keying are not required. Special effects can also be created for live shows or recordings that cannot be pushed into a blue screen environment. According to virtual studios, we want to refer to this as *augmented studios*. This, however, leads to even more difficult problems: How to perform background keying and how to support camera tracking within arbitrarily complex studios?

Although several solutions to the one or the other problem exist, we want to propose an entirely different concept: We envision a technical extension to existing studio technology that enables new effects and control possibilities. Projectors allow a spatial and temporal modulation of light and displayed pictorial content that can be computer controlled and synchronized with the capturing process of studio cameras. Consequently, we propose the application of coded projection and illumination in modern television studios - either exclusively or in combination with existing analog lighting and projection displays. We believe that this holds the opportunity to solve several of the problems mentioned above, and that it promises to open new possibilities for future TV productions:

1. Integration of imperceptible coded patterns into projected images that support continuous online-calibration, camera tracking, and acquisition of scene properties.
2. Dynamic presentation of un-recorded direction, moderation and other information spatially anywhere within the studio - not being limited to inflexible screens, such as teleprompters.
3. Computer controlled, projector-based re-illumination of studio content without physical modification of the lighting equipment.
4. Temporal coded illumination to support keying of foreground objects.

While the early idea of this concept was outlined in [[?]],

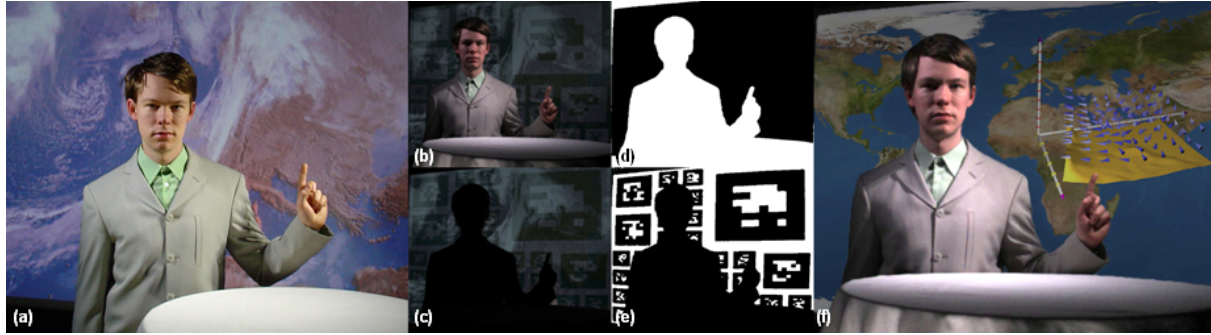


Figure 1: Real studio setting with physical back projection encoding adaptive imperceptible patterns (a), images captured by camera at 120Hz (b, c), computed foreground matte from real-time difference keying (d), extracted multi-resolution marker pattern for in-shot camera pose estimation (e), and composite frame with virtual background and 3D augmentation (f).

we now want to present first concrete realizations and techniques.

Our main contributions in this paper are an *adaptive imperceptible pattern projection technique* that overcomes limitations of existing approaches, and a *dynamic multi-resolution marker tracking method* that ensures a continuous in-shot camera tracking despite possible occlusions. Together, these techniques allow displaying an arbitrary projected content within real studios that is visible to participants in the familiar way. The poses of synchronized studio cameras, however, can be estimated through the extracted code patterns. We have also combined our methods with *real-time difference keying* using a high-speed white-light LED illumination or the coded projection itself.

The application of large multi-projection displays in real studios has become very common. Projection screens are used in many television shows. This represents an already established technological foundation for the techniques presented in this paper.

2. Related and Previous Work

The following subsections discuss only the related work that is most relevant to our approach. A full state-of-the-art review within the different areas is out of the scope of this paper.

2.1. Embedded Imperceptible Pattern Projection

Besides a spatial modulation, a temporal modulation of projected images allows integrating coded patterns that are - due to limitations of the human visual system- not perceivable. Synchronized cameras, however, are able to detect and extract these codes. This principle has been described by Raskar et al. [RWC*98], and has been enhanced by Cotting et al. [CNGF04]. It is referred to as *embedded imperceptible pattern projection*. Extracted code patterns allow, for in-

stance, the simultaneous acquisition of the scenes' depth and texture for 3D video applications [WWC*05], [VVSC05].

The most advanced technique was presented in [CNGF04], where a specific time slot of a DLP projection sequence is occupied exclusively for displaying a binary pattern within a single color channel. Multiple color channels are used in [CZGF05] to differentiate between multiple projection units. However, unless the DLP mirror flip sequences within the chosen time slot are not evenly distributed over all possible intensities (which is not the case in practice) this technique can result in a non-uniform fragmentation and a substantial reduction of the tonal values. Since the patterns are encoded independently of visual perception properties, local contrast reductions and flickering effects should be visible in unfavorable situations, such as low intensities and low spatial image frequencies, as well as during the temporal exchange of the code patterns. Modifying the color channels of individual pixels differently can also lead to slightly miscolored image regions.

Instead of increasing or decreasing the intensity of a coded pixel by a constant amount or by an amount that depends purely on technical parameters (such as mirror flip sequences), our method considers the capabilities and limitations of human visual perception. It estimates the Just Noticeable Difference and adapts the code contrast on the fly - based on regional properties of projected image and code, such as intensities and spatial frequencies. Thus, only the global image contrast is modified rather than local color values. This ensures an imperceptible coding while providing a maximum of intensity difference for decoding. Yet, it enforces only a small and linear contrast compression. Intensity coding can also be supported in our case, rather than being limited to pure binary patterns. Furthermore, a projector individual calibration is not necessary. A temporal code blending technique is used for seamlessly exchanging individual codes.

2.2. Camera Tracking

One of the main challenges of virtual and augmented studios is the robust and fast tracking of the studio cameras [BT03]. While some approaches apply special tracking hardware, others try to estimate the cameras' pose by observing natural features (e.g., ceiling-mounted studio lights or the studio content itself) or artificial tags [TJU97] with additional cameras.

Optical tracking approaches are becoming more and more popular. This can be contributed to their robustness against most environmental disturbances, speed and precision. Such techniques can be categorized into marker-less and marker-based methods. Marker-less techniques strongly rely on the robust detection of natural scene features [FKGK05]. They will fail for uniformly structured surfaces or under dim lighting conditions. This limits the application of marker-less tracking in TV studios to optimized situations. Marker-based tracking provides artificial visual features by integrating detectable tags. A very common technique for virtual studios is to integrate markers directly into the blue screens by painting them in a different blue tone that does not effect chroma keying. An example is the widely used ORAD system or more advanced techniques, such as [XDD01]. They support efficient in-shot camera tracking in virtual sets.

However, within real and augmented studios markers should neither be directly visible to the audience, nor appear in the recorded video stream. Consequently, marker-based tracking within such environments is usually restricted to observing out-shot areas such as the ceiling or the floor which are normally covered by studio equipment, like light installations, cables, and mountings. Occlusions and dynamic re-configurations of the installations cause additional problems for marker-based tracking.

We present a camera tracking technique that integrates imperceptible markers into background projections within real studio settings. Thus, they can be displayed directly within the field of view of the camera without being directly perceptible by the audience or moderator. Visibly projected markers have been used earlier for geometric projector calibration on planar screens [Fia05b]. We have developed a dynamic multi-resolution approach to ensure a continuous in-shot camera tracking rather than a projector calibration. In contrast to similar nested marker techniques [TKO06] that embed several code scales in a single printed pattern, our multi-resolution markers are projected and can consequently be automatically exchanged and modified depending on the actual camera pose and possible occlusions of the background projection (e.g., by the moderator).

2.3. Keying

A variety of different keying techniques exist. Traditional luma keying or chroma keying cannot be applied in our context due to the lack of a static or uniform background. Vari-

ations of difference keying, such as flash keying, however, are applicable. While some complex flash keying approaches such as [SLKS06] are only applied offline, a technique that uses pulsed blue LEDs for real-time flash keying was described in [AT05]. We present two types of difference keying techniques that are adapted for our purpose. The first one considers the temporal differences in the background projection while assuming a constant foreground. The second method adopts the technique described in [AT05] to generate temporal differences in the foreground. However, instead of blue LEDs that provide an additional chrominance masking, we apply pulsed white-light LEDs for creating a uniformly white studio illumination. We also discuss how the projectors themselves can be used for supporting flash keying instead of installing additional LED illumination. This, however, assumes that the studio is equipped with a projector-based illumination system. While difference keying can be realized with a single camera system, two coaxial aligned cameras can support a real-time depth-of-field based keying, similar as in [RK05], and to overcome focus issues in cases of short focal depths.

2.4. Projectors in TV Studios

Video projectors have been used before for displaying direction information in virtual studio setups.

Fukaya et al. [FFY*03], for instance, project images onto a blue screen located in a TV studio. They are alternately blocked and transmitted by an LCD shutter mounted in front of the projector lens. A separate shutter control unit synchronizes projection and exposure time of a studio camera in such a way that images are only captured when the projection is blocked. Chroma keying can then be applied in a conventional way.

Shirai et al. [STK*05] apply chrominance and luminance keying instead of a shuttered projection for solving the same problem. Projecting an image onto a blue screen enables computing both a luminance key and a chrominance key that allow masking out the blue screen area for a final composition. This is possible only if the projected image is not brighter than the studio illumination.

Grau et al. [GPA04] use a retro-reflective coating instead of diffuse blue or green screens. This allows the projection of direction information onto the screens. Mounting a blue or green light source near or around the camera lens, however, ensures that key-colored light is re-directed directly towards the camera lens by the retro-reflective coating, while the projected images -although mainly reflected back towards the displaying projectors- are partially diffused.

All of these techniques are used for virtual sets, while our approach addresses real studios. The temporal projection concept of Fukaya et al. [FFY*03] comes closest to our idea. However, we propose the integration of invisible code patterns into arbitrary projections within real sceneries, rather

than projecting visible information in blue screen setups that are not recorded by the camera.

3. System Overview

Our current prototype is illustrated in figure 2. A moderation desk and a rear-projection screen serve as backdrop. An off-the-shelf stereo-enabled DLP projector (InFocus DepthQ) displays an arbitrary and dynamic background at a speed of 120Hz. Our camera system consists of two optically aligned CCD cameras (a). Depending on the applied keying technique and supported focal depth, only one or both cameras are used (details are presented in section 5.1). For real-time flash keying, a dimable 4800 Lumen LED illumination system (c) has been built.

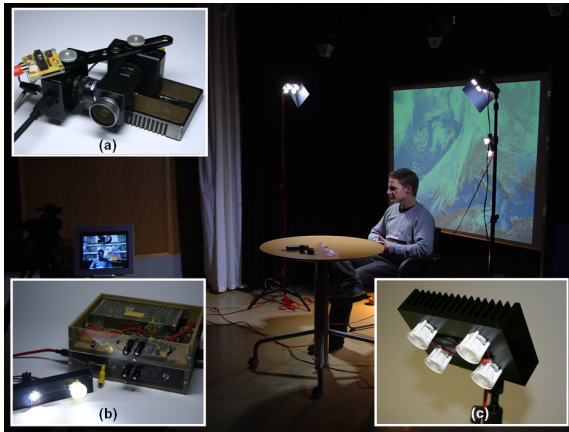


Figure 2: Overview over system components and studio setting: moderation desk with rear-projection screen and LED illumination, camera system (a), synchronization units (b) and LED module (c).

A customized synchronization electronics (b) receives the shutter signal that is generated by the graphics card (a Nvidia Quadro FX 1500 in our case) of a PC that triggers the stereo projector. This signal is then being used for triggering both - the camera and the illumination system at 120Hz. The illumination can be switched to a flash mode (i.e., on-off sequences) or to a demodulated (i.e., rectified) constant lighting. Point Grey Dragonfly Express cameras deliver raw-format images in VGA resolution over Firewire 800.

Instead of using the demosaicing functionality offered by the camera driver, we implemented a pixel grouping demosaicing algorithm that is optimized for reducing color seams at intensity boundaries. This is a good trade-off between quality and performance. The algorithm is implemented as fragment shaders on the GPU and delivers a better quality at significantly higher frame rates compared to the driver's internal CPU based algorithms.

4. Dynamic Δ -Coded Projection

Based on the initial suggestions of embedded imperceptible patterns [RWC*98] we have developed an enhanced method that projects and captures encoded images and their complements at 120Hz. However, instead of increasing or decreasing the intensity of a coded pixel by a constant amount of Δ , we compute the Just Noticeable Difference and adapt local Δ values on the fly - based on regional image intensity and spatial resolution. This ensures an imperceptible coding while providing a maximum of intensity differences for decoding. The advantages of this approach in contrast to existing methods have been discussed in section 2.1.

4.1. Static Codes

In case a static binary code image C is embedded into the displayed original image O we simply compute the projected image with $I=O-\Delta$ and its complement with $I'=O+\Delta$. Projecting both images at a speed that is above the critical flicker frequency, a human observer will perceive roughly $(I+I')/2$ which approximates O (cf. figure 1a). Depending on the binary code in C we decide whether Δ is positive or negative on a per-pixel basis.

To avoid clipping at lower and higher intensity levels when subtracting or adding Δ , O has to be scaled. Theoretically a contrast reduction of 2Δ is sufficient. However, for our currently applied projector and camera the brightness of the original image has to be increased by approximately 10% to reduce camera noise in dark regions. Practically, this leads to a maximum contrast reduction of ~ 10 -20% at the moment. However, this can be reduced significantly by applying cameras and optics that are less sensitive to noise, or brighter projectors. Compared to other approaches, such as [CNGF04] (where large tonal shifts for lower intensities in individual color channels or maximum dynamic range reductions of up to 50% are reported), O is linearly scaled in our case.

Synchronizing the camera to the projection enables capturing both images separately (cf. figures 1b-c). Dividing or subtracting them allows identifying the encoded state per camera pixel (cf. figures 1e): The ratio of both images are above or below one, while the difference of both images is above or below zero - depending on the integrated bit. It is essential that camera and projector are linearized to avoid an effect of their transfer or response functions. A gamma correction can be applied after linearization to ensure color consistency. Thus, projecting and capturing I and I' at 120Hz leads to perceiving O and reconstructing C at 60Hz.

Despite the integration of binary codes, our approach allows to embed and reconstruct multiple code intensities at each pixel up to a certain extent. This gives the opportunity to encode more information at the same time.

One problem with this simple approach is that for fast

camera movements I and I' might be no longer geometrically aligned. Reconstructing the code bits on a per-pixel basis fails in these situations. To ensure a correct alignment, we apply a Canny edge detector to both images and compute the optical flow from the result. This is used for estimating a homography matrix that allows re-registering both images. By this, we assume that the images are projected onto a planar surface.

A more critical problem is that both images also appear misregistered during fast eye movements which makes the embedded code well visible. In visual perception research this is known as *phantom array effect*. It also appears during the temporal color modulation of DLP projectors, where it is better known under the term *rainbow effect*. The strength of this effect and consequently the perception of the integrated code during eye movements can be reduced and even eliminated by using small amounts of Δ . If too small, however, the code bits are perished by camera noise.

Note that the phantom array effect is not a problem of related techniques that do not compensate the coded images temporarily [CNGF04]. For approaches that do perform a temporal compensation to avoid contrast artifacts and tonal shifts, such as in our case, this effect can be overcome.

It is important to note that the Just Noticeable Difference (JND) of the phantom array effect and consequently the largest tolerable amount of Δ depends on several parameters: the regional brightness and spatial frequency of O , the spatial frequency of C , the temporal frequency of I and I' , and the speed of the eye movements. Knowing the relation between these parameters enables a dynamic and content dependent regional adaption of Δ . Since we have not found any literature that reports on an exact function which correlates these parameters we have carried out an informal user experiment to approximate this function. Since the speed of eye movements can, in the normal application case not be measured, we want to assume fast eye movements for the following. Slower eye movements reduce the effect.

4.2. Δ -Function

For estimating the Δ -function, we asked four subjects (one female, three male) to carry out a user experiment. The subjects were asked to identify Δ at the JND point for different projected images with integrated codes. They were sitting at a distance of 95.262cm in front of a 110cm high and wide back projection screen - covering the entire foveal field of view of 60 degrees. The displayed images contained regular checkerboards representing a two dimensional box function with spatial frequencies (Fb) ranging from 1/120 to 1/8 cycles per degree (cycl/deg) of visual field in both directions, and intensities (Lb) ranging from ~ 4 -27 candela per square meter (cd/m^2). The embedded codes were also represented by a box function with spatial frequencies (Fm) ranging from 1/32 to 1/2 cycl/deg. Code patterns and image patterns were always phase shifted to avoid a cancelation.

To guarantee equal conditions, the subjects were given time to adapt to different luminance levels first. Then they were asked to follow a target on the screen that moved up and down quickly at a constant speed to enforce the phantom array effect for fast eye movements. While changing Δ , the subjects were asked to indicate the point at which the code could just not be perceived anymore (i.e., the JND point). This process was repeated about eighty times per subject to cover a combination of five different image frequencies over five luminance levels, and four different code frequencies. Each experiment took about 4-5 hours for each subject. The results of all four subjects were averaged and are presented in figure 3a.

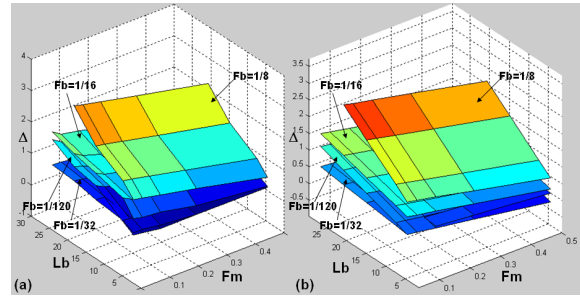


Figure 3: Average Δ responses at the JND point for a combination of four subjects, four discrete image frequencies (Fb), five luminance levels (Lb), and five code frequencies (Fm) (a). Plane function fitted to sample points (b) for each considered Fb .

Due to their mainly linear behavior, the sample points were fitted to planes using multidimensional linear regression (figure 3b). The four parameters of each plane shown in figure 3b are plotted as circles in figure 4.

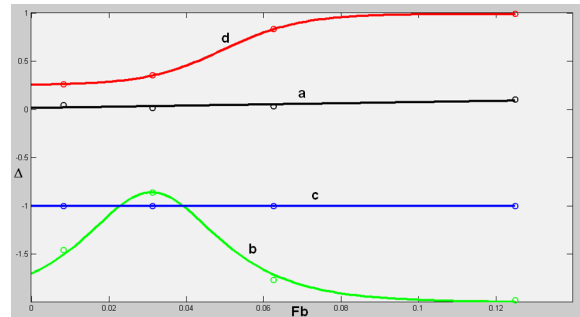


Figure 4: Approximated discrete plane parameters and fitted continuous functions.

Applying the general plane equation $\Delta = -(aLb + bFm + d)/c$ for parameterizing the fitted functions in figure 3b requires to find continuous functions that approximate the discrete plane parameters (a, b, c, d) over all image frequencies Fb . Figure 4 illustrates the result of a one-dimensional curve fitting:

$$a = 0.6108Fb + 0.0139 \quad (1)$$

$$b = 1.144/\cosh(65(Fb - 0.031)) - 2 \quad (2)$$

$$c = -1 \quad (3)$$

$$d = -0.73914/(1 + \exp((Fb - 0.04954)/0.01)) + 1 \quad (4)$$

While the parameters a and b correspond to the gradients of the planes in directions Lb and Fm , d and c represent a shift and a scaling of Δ . The scalar $c=-1$ is relative to our experiments with a temporal frequency of 120Hz. For other temporal frequencies, it has to be adapted (increased for higher frequencies, and decreased for lower frequencies).

Note that we chose a straight line to fit a , a trigonometric function to approximate b , and an exponential function to represent d . With this, the average deviation of our analytical solution with respect to the experimentally acquired values is 0.266cd/m^2 (this equals 0.89% of the projected intensity levels, or ~ 2 projected gray scales).

Besides comparing the analytical solution with the results of the user experiment, it was also exploited for values outside our discrete test samples. It was confirmed by the subjects that the function approaches the *JND* point in these cases as well.

4.3. Computation of Δ

Regionally adapting the Δ values using our experimentally derived function requires the real-time analysis of O and C .

For acquiring the spatial frequencies of particular image regions, we apply the Laplace-pyramid approach presented by [BA83]. In our case we found six levels of the Laplacian pyramid to be sufficient. As described in [RPG99a] we use the absolute differences of each level of the Gaussian pyramid and normalize each of the resulting Laplacian pyramid levels. The results are the ratios of spatial frequencies within each of the generated frequency bands. This is converted to units of cycl/deg , which depend on the observers' distance to the image plane and the physical size of the projection. The input image is transformed into its physical luminance representation in cd/m^2 (the responds function of the projector has been measured with a photometer). With these parameters we can apply our Δ -function to compute the largest non-visible Δ value for an arbitrary region within O and C .

The visibility of the encoded patterns can be significantly decreased by reducing Δ in the green channel. This is due to the fact that humans are most sensitive to the wavelength of green light. Decreasing the Δ in the green channel down to a fourth of the red and the blue channels did not lead to a quality reduction of the extracted patterns when the maximal difference of all three color channels was used for decoding.

Note, that this does not result in a tonal shift of O since the embedded code (no matter how large Δ in different color channels is) is always compensated. In practice, Δ ranging from 0.29 to 1.45 cd/m^2 (i.e., 1-5% of the projected intensity levels, or ~ 2.5 -13 projected gray scales) were computed.

4.4. Temporal Code Blending

Besides the phantom array effect that is caused by eye movements, another visual effect can be observed that leads to the perception of the code patterns in cases when they are temporally exchanged. This is illustrated in figure 5.

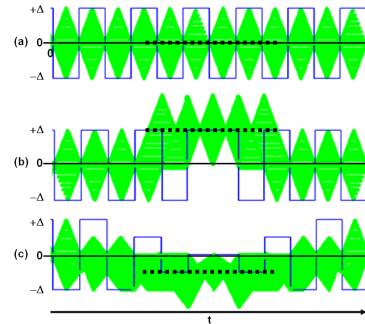


Figure 5: Possible amounts of perceived relative intensities for different temporal code states: static code (a), abruptly switched code (b), temporally blended code (c).

For photopic vision, it can be assumed that the integration times of the human eyes are between 16ms and 50ms, depending on the perceived brightness (shorter for bright situations). If the projected image and its compensate contain a static code over a period of time the subtraction or addition of Δ at each pixel of both images I and I' does not change. Figure 5a visualizes this situation. Plotting the relative amount of integrated light for all possible integration times between 16ms and 50ms, and for all possible phase shifts (in contrast to the camera, the integration process of the eyes is not in synchronization with the projection) leads to the presented green surfaces. The average integration amount (dotted line) is zero in figure 5a (assuming no eye movements). Exchanging the code at a particular point in time (i.e., switching from a binary 0 to a binary 1) leads to the integration results shown in figure 5b. The average integration amount during code switching is Δ , which leads to a visible flickering during this time.

To overcome flickering caused by code transitions, we do not switch between code states abruptly, but temporally transfer from one stage to another stage over multiple blending steps. As illustrated in figure 5c, the average integration amount reduces to $\Delta/2$ for three blending steps. In general we can say that it reduces to Δ/s for $s+1$ blending steps if we continuously decrease Δ by Δ/s in each step until $\Delta=0$, and then increase Δ by Δ/s until the original amount is reached.

During the center stage (i.e., when $\Delta=0$ and $I=I'=O$) the code switched.

The maximal average integration amount Δ/s that cannot be detected, and consequently the number of blending steps, depends on the just noticeable luminance and contrast difference which can be derived from the *threshold-vs-intensity (TVI) function* and the *contrast sensitivity function* as explained in [PFFG98], [Lub95]. They are functions of local spatial image frequency and luminance level. Consequently, the optimal number of blending steps for a particular region in O can be computed from O 's local spatial frequency and luminance level by using these functions.

We use the average spatial frequencies and luminance levels of image regions that are already computed for the estimation of Δ (see section 4.3). The TVI function and the contrast sensitivity function are applied and their results are multiplied as described in [RPG99b] for computing the largest not-detectable luminance difference Δ/s . This leads to the number of individually required blending steps s for each marker region. If the content in O changes during a blending sequence (e.g., in case of videos or interactive content), then the original Δ and s are adapted and the blending is continued until Δ first decreases to a value ≤ 0 (for switching the code) and then increases again until it reaches the new original Δ value. Varying Δ only by the maximum non-perceivable luminance difference ensures that the code cannot be detected during blending. In practice, 10-20 bending steps were derived (i.e., 3-6 marker transitions per second were supported at 120Hz).

5. Adaptive Code Placement

For supporting optical in-shot camera tracking we embed imperceptible two-dimensional markers of different scales into the projected images (cf. figure 1e). Thereby the Δ values and the number of blending steps are computed individually for each single marker by averaging the corresponding image luminance and spatial frequency of the underlying area in O and the spatial frequency of the the corresponding area in C . For spatial frequencies, the values located within the marker regions of each of the six frequency bands are averaged. The peak frequency is then approximated by choosing the frequency band containing the largest average.

To ensure a continuous tracking despite possible occlusions or different camera perspectives, the code image C is dynamically re-generated and marker placement as well as marker sizes are adapted. Consequently, the foreground objects have to be keyed and related to the projected image for determining occlusions.

5.1. Real-Time Difference Keying

To separate foreground (e.g., the moderator) from background (the Δ -coded background projection), we support

two difference keying techniques: *real-time flash keying* and *a background difference keying*.

By using high performance LED flash illumination we are able to lighten the scene 60 times per second by short flashes with a length of 8ms. Thus, each other captured frame contains an illuminated foreground (cf. figure 1b), while the remaining frames contain a dark foreground (cf. figure 1c), which allows separating both (cf. figure 1d). Due to their high frequency the flashes are not detectable. In contrast to [AT05] we use white-light LEDs with a white point of 5600K for direct studio illumination, rather than applying blue LEDs for chrominance masking. Color filters can be used in addition for supporting the required studio lighting.

The matting process in this case is straightforward: Due to the fact that one of the captured images is taken under full illumination and the other one under no illumination, we can easily extract the pixels belonging to the foreground by analyzing the difference between corresponding camera pixels and comparing it with a predefined threshold. To ensure that both images remain registered during fast camera movements, they are corrected using a homography transformation as explained for I and I' in section 4.1. However, care has to be taken because the delta coded projection in the background also differs in its intensity. The difference threshold has to be set to a value that is larger than twice the largest encoded Δ . We evaluate the maximum difference in the three color channels for thresholding instead of using the average difference of the gray channel.

In case that the camera resolution is lower than the projector resolution, individual pixels might be misclassified at the transitions of marker boundaries. This can be contributed to the fact that an integration over several Δ -coded projector pixels (also during fast camera movements) can lead nearly to the same intensity in both images. These defects can be removed efficiently by applying a median filter to the generated matte image. In a final step the matte is smoothed by a 5x5 Gaussian filter kernel to soften the transitions between foreground and background.

Instead of applying an LED illumination, video projectors themselves can be used to support flash keying if installed in the studio environment. In contrast to simple LEDs, projector-based illumination [RWLB01], [BGWK03] supports generating a synthetic, spatially varying illumination on real objects on the fly. Thus, in addition to a temporal illumination coding, a virtual lighting situation can be defined, computed and physically approximated within the studio using projectors - without changing the actual light sources.

Besides flash keying, the coded background projection itself allows another form of difference keying. If the foreground objects are illuminated with a demodulated (i.e., rectified) constant lighting, the intensity differences in I and I' can be analyzed. While camera pixels with constant intensities belong to foreground objects, pixels with variations that

are due to the Δ -coded projection belong to the background. We call this approach *background difference keying*. It does not require a synchronization between camera and illumination system but has to deal with potential misclassifications of corresponding pixels during camera movement.

In both cases, difference keying is supported at a capturing speed of 60Hz for both images. One camera is normally sufficient. However, if the Δ -coded projection is out of focus (e.g., due to a short focal depth when focussing on the foreground) marker tracking might fail. As mentioned earlier, two coaxially aligned cameras (cf. figure 2a) can be used for avoiding this problem: While one camera is focussed on the background, the other camera is focussed on the foreground. Registering both camera images and synchronizing the capturing process supports recording the focussed foreground while processing the focussed background. Furthermore, this allows to evaluate relative defocus values of corresponding pixels in both images to enable a depth-of-field based keying, as in [RK05]. A real-time keying from defocus has not yet been implemented but belongs to our future work.

5.2. Dynamic Multi-Resolution Markers

The stability of the optical tracking strongly depends on a constant visibility of a certain amount of markers with optimal sizes. While tracking will not be possible if the complete projection is occluded from the camera's point of view, for partial occlusions an adaptive marker placement leads to a more robust tracking compared to static markers.

Hence we adjust the projected imperceptible markers within C in each frame by analyzing the visibility of the projected pixels from the camera's perspective. To keep the visibility of the embedded markers during switching at a minimum we use the temporal blending techniques described in section 4.4.

For optical tracking the ARTag library [Fia05a] is used which offers the possibility to generate arbitrary array sets out of 1024 predefined markers. This feature is used to define a multi-resolution marker array containing different sized markers for the same spatial locations - all sharing the same coordinate system.

We pre-compute a quad-tree that contains multiple markers at different scales in each level. From a higher to the next lower level, the number of markers doubles while their size decreases by factor 2. We refer to this as the *marker tree*. Adaptive marker placement is implemented in several steps (cf. figure 6).

First, a full screen quad is rendered in projector resolution and a projective transform is computed that maps the generated foreground matte from the perspective of the camera (a) onto it. This is achieved by using the model-view matrix that results from tracking of the previously displayed frame.

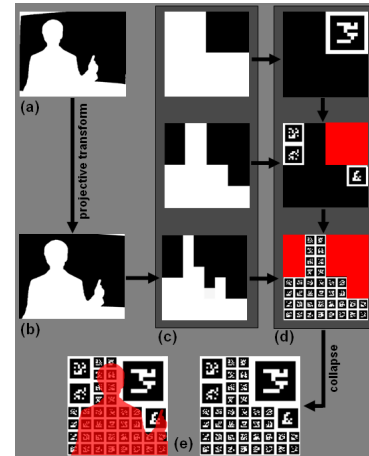


Figure 6: Generation of markers: projective transform of foreground matte from camera perspective (a) to projection screen (b), construction of visibility tree (c) and labeling of marker tree (d), collapsing of labeled marker tree (e).

The result is a binary image containing the visibility of each projector pixel from the camera's view, which we want to refer to as *visibility map* (b). This technique is analogue to conventional shadow mapping.

The initial visibility map is then used to analyze the smallest possible marker size that will be used by geometrically determining the number of projector pixels that are visible in the camera image from the previous perspective.

We sub-sample the visibility map into an image pyramid that covers the largest possible marker size in the highest level (e.g., by definition 2×2 markers in C) down to the determined smallest possible marker size in the lowest level (e.g., 16×16 pixels per marker in our case). This leads to a multi-resolution visibility map that we call *visibility tree* (c).

During runtime, the marker tree and the visibility tree are combined at corresponding levels (d): In a top-down direction, only entries that are neither occluded (i.e., marked as visible in the same visibility tree level) nor already occupied by markers of higher levels are processed. The remaining entries are then labeled as occupied within the current level of the marker tree. Regions which are not visible throughout all levels are labeled at the bottom level of the marker pyramid. If the bottom is reached, the labeled marker tree is collapsed and the non-overlapping entries that are occupied by different levels are combined. This results in a code image C that contains the set of optimally scaled and placed markers with respect to foreground occlusions and camera perspective (e). The same constellation from the perspective of the camera is shown in figure 1e.

As explained in section 4.4, local marker regions have to

be temporally blended if a code transition within a particular area in C occurs.

6. Summary and Future Work

Optical in-shot camera tracking is a common technique for virtual sets. For many situations, such as recordings or broadcasts that host a live audience, however, virtual studio technology cannot be applied. Yet, many of such productions already use large projection displays in real studio environments. In this paper we propose a coded projection and illumination that enables optical in-shot camera tracking and keying within non-blue-screen equipped sets.

Our contributions are a novel imperceptible embedded code projection technique that, in contrast to previous work, considers parameters of human perception for optimal encoding and decoding of integrated patterns. We demonstrate two real-time difference keying approaches in combination with a temporally coded projection for efficient foreground extraction. By combining both techniques, a dynamic multi-resolution marker method was introduced that ensures a continuous and optimal tracking, despite possible occlusions and different camera perspectives. This supports a flexible in-shot camera tracking and real-time keying in real studios without installing permanent physical markers anywhere in the setting, such as at the ceiling. Thus our approach offers a similar portability as commonly used projection backdrops.

All of the described techniques were implemented on modern GPUs to achieve interactive frame-rates. While the coded projection and illumination, as well as the capturing process are synchronized at a speed of 120Hz, projected dynamic content was presented at a speed of 60Hz. The final image composition that includes tracking, keying, matting, and rendering of augmented content (i.e., foreground / background / 3D / composite) was carried out 10-20 frames per second on our current single-PC prototype (depending on the number of detected markers, the ARTag library requires 20ms-50ms for processing). This is clearly not acceptable for a professional application. Distributing the different steps to multiple PCs (especially the rendering of the graphical augmentations) will lead to a significant speed-up. If the tracking data is shared among a PC cluster, high-performance frame-grabbing enables the efficient exchange of high resolution image data.

Our two implemented difference keying techniques can be combined with depth-of-field based keying, such as in [RK05], to support stable real-time matting. Furthermore, the tracking performance and quality needs to be improved significantly for professional applications. Since our approach is widely independent of the utilized marker tracking library, further investigations have to be carried out to find alternative solutions. At the moment, our system is limited to the performance and precision of the ARTag library (see [Fia05a] for details). Currently, we support online and

offline augmentations. In the latter case, the captured images I and I' are only recorded to disk during run-time. During a post-production step, tracking, keying, matting and rendering can be carried out at a much higher quality level.

In the short term, increasing the tracking precision and the overall performance of our system are the main tasks of our future work. In the long term, we envision the combination of projector-based and analog illumination in modern television studios [?]. Together with appropriate image correction techniques, such as geometric warping, radiometric compensation, and photometric calibration, this holds the potential to display imperceptible code patterns, such as the markers used for camera tracking, which are integrated into pictorial content or into the projected illumination spatially anywhere within the studio. A temporally coded projector-based illumination would also support an ad-hoc synthetic re-illumination as already shown in the small scale ([RWLB01], [BGWK03]), and the extraction of depth-information, such as explained in [WWC*05], [VWSC05].

A technical challenge will also be to adapt current studio camera technology to support fast capturing and synchronization. Today, such cameras are synchronized to external displays via the standard BlackBurst signal at a speed of 50Hz for PAL or 60Hz for NTSC. Thus, the capturing at a field rate 60Hz would decrease the extraction of the embedded code patterns to a maximum speed of 30Hz. The projection speed and consequently the the perception characteristics, however, is not effected by slower cameras. Future projectors will certainly provide even higher frame rates.

Acknowledgments

Special thanks to Daniel Kolster, Stefanie Zollmann, Hongcui Lu, and Steffen Hippeli for contributing to the user experiments and to various investigations. The investigation and realization of concepts and techniques outlined in this paper have been proposed to the Deutsche Forschungsgemeinschaft (DFG) under reference number BI 835/2-1.

References

- [AT05] ALEXANDER T. G., THOMAS R. R.: Flash-based keying. *European Patent Application EP1499117* (2005).
- [BA83] BURT P. J., ADELSON E. H.: The laplacian pyramid as a compact image code. *IEEE Transactions on Communications COM-31,4* (1983), 532–540.
- [BGWK03] BIMBER O., GRUNDHÖFER A., WETZSTEIN G., KNÖDEL S.: Consistent illumination within optical see-through augmented environments. In *ISMAR '03: Proceedings of the The 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality* (Washington, DC, USA, 2003), IEEE Computer Society, pp. 198–207.

- [BT03] BERNAS M., TEISLER M.: Determination of the camera position in virtual studio. In *Proceedings of the 3rd IEEE International Symposium on Signal Processing and Information Technology, 2003. ISSPIT 2003* (2003), pp. 535–538. ISBN 0-7803-8292-7.
- [CNGF04] COTTING D., NÄF M., GROSS M. H., FUCHS H.: Embedding imperceptible patterns into projected images for simultaneous acquisition and display. In *Proceedings of the Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'04)* (2004), pp. 100–109. ISBN 0-7695-2191-6.
- [CZGF05] COTTING D., ZIEGLER R., GROSS M. H., FUCHS H.: Adaptive instant displays: Continuously calibrated projections using per-pixel light control. In *Proceedings Eurographics 2005* (2005), pp. 705–714. Eurographics 2005, Dublin, Ireland, August 29 - September 2, 2005.
- [FFY*03] FUKAYA T., FUJIKAKE H., YAMANOUCHI Y., MITSUMINE H., YAGI N., INOUE S., KIKUCHI H.: An effective interaction tool for performance in the virtual studio - invisible light projection system. In *Proceedings International Broadcasting Conference (IBC03)* (2003), pp. 389–396.
- [Fia05a] FIALA M.: Artag, a fiducial marker system using digital techniques. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2* (Washington, DC, USA, 2005), IEEE Computer Society, pp. 590–596.
- [Fia05b] FIALA M.: Automatic projector calibration using self-identifying patterns. In *Proceedings of the IEEE International Workshop on Projector-Camera Systems (Procams 2005)* (San Diego, USA, 2005).
- [FKGK05] FRAHM J.-M., KOESER K., GREST D., KOCH R.: Markerless augmented reality with light source estimation for direct illumination. In *Conference on Visual Media Production CVMP, London, December 2005* (2005).
- [GPA04] GRAU O., PULLEN T., A. THOMAS G.: A combined studio production system for 3-d capturing of live action and immersive actor feedback. *IEEE Transactions on Circuits and Systems for Video Technology* 14, 3 (2004), 370–380.
- [Lub95] LUBIN J.: A visual discrimination model for imaging system design and evaluation. *Vision Models for target detection and recognition* (1995), 245–283.
- [PFFG98] PATTANAIK S. N., FERWERDA J. A., FAIRCHILD M. D., GREENBERG D. P.: A multiscale model of adaptation and spatial vision for realistic image display. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1998), ACM Press, pp. 287–298.
- [RK05] REINHARD E., KHAN E. A.: Depth-of-field-based alpha-matte extraction. In *APGV '05: Proceedings of the 2nd symposium on Applied perception in graphics and visualization* (New York, NY, USA, 2005), ACM Press, pp. 95–102.
- [RPG99a] RAMASUBRAMANIAN M., PATTANAIK S. N., GREENBERG D. P.: A perceptually based physical error metric for realistic image synthesis. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1999), ACM Press/Addison-Wesley Publishing Co., pp. 73–82.
- [RPG99b] RAMASUBRAMANIAN M., PATTANAIK S. N., GREENBERG D. P.: A perceptually based physical error metric for realistic image synthesis. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1999), ACM Press/Addison-Wesley Publishing Co., pp. 73–82.
- [RWC*98] RASKAR R., WELCH G., CUTTS M., LAKE A., STESIN L., FUCHS H.: The office of the future: A unified approach to image-based modeling and spatially immersive displays. *Computer Graphics 32*, Annual Conference Series (1998), 179–188.
- [RWLB01] RASKAR R., WELCH G., LOW K.-L., BANDYOPADHYAY D.: Shader lamps: Animating real objects with image-based illumination. In *Proceedings of the 12th Eurographics Workshop on Rendering Techniques, London, UK, June 25-27, 2001* (2001), pp. 89–102. ISBN 3-211-83709-4.
- [SLKS06] SUN J., LI Y., KANG S. B., SHUM H.-Y.: Flash matting. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers* (New York, NY, USA, 2006), ACM Press, pp. 772–778.
- [STK*05] SHIRAI A., TAKAHASHI M., KOBAYASHI K., MITSUMINE H., RICHIR S.: Lumina studio: Supportive information display for virtual studio environments. In *Proceedings of IEEE VR 2005 Workshop on Emerging Display Technologies* (2005), pp. 17–20.
- [TJU97] THOMAS G., JIN J., URQUHART C. A.: A versatile camera position measurement system for virtual reality tv production. In *International Broadcasting Convention* (1997), pp. 284–289. ISBN 0-85296-694-6.
- [TKO06] TATENO K., KITAHARA I., OHTA Y.: A nested marker for augmented reality. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Sketches* (New York, NY, USA, 2006), ACM Press, p. 152.
- [VVSC05] VIEIRA M. B., VELHO L., SA A., CARVALHO P. C.: A camera-projector system for real-time 3d video. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops* (Washington, DC, USA, 2005), IEEE Computer Society, p. 96.

- [WWC*05] WASCHBÜSCH M., WÜRMLIN S., COTTING D., SADLO F., GROSS M. H.: Scalable 3d video of dynamic scenes. *The Visual Computer* 21, 8-10 (2005), 629–638.
- [XDD01] XIROUHAKIS Y., DROSPOULOS A., DELOPOULOS A.: Efficient optical camera tracking in virtual sets. *IEEE Transactions on Image Processing* 10, 4 (2001), 609–622. ISSN 1057-7149.