Real-Time Video Abstraction

Holger Winnemöller* Sven C. Olsen* Bruce Gooch*

Northwestern University

Figure 1: **Abstraction examples.** *Original*: Snapshots of a guard in Petra (left) and two business students (right). *Abstracted*: After several bilateral filtering passes and with DoG-edges overlayed. *Quantized*: Luminance channel soft-quantized to 12 bins (left) and 8 bins (right). Note how folds in the clothing and other image details are emphasized (stones on left and student's shadows on right).

Abstract

We present an automatic, real-time video and image abstraction framework that abstracts imagery by modifying the contrast of visually important features, namely luminance and color opponency. We reduce contrast in low-contrast regions using an approximation to anisotropic diffusion, and artificially increase contrast in higher contrast regions with difference-of-Gaussian edges. The abstraction step is extensible and allows for artistic or data-driven control. Abstracted images can optionally be stylized using soft color quantization to create cartoon-like effects with good temporal coherence. Our framework design is highly parallel, allowing for a GPU-based, real-time implementation. We evaluate the effectiveness of our abstraction framework with a user-study and find that participants are faster at naming abstracted faces of known persons compared to photographs. Participants are also better at remembering abstracted images of arbitrary scenes in a memory task.

CR Categories: I.3.3 [Computer Graphics]: Image Generation

Keywords: non-photorealistic rendering, visual perception, visual communication, image abstraction

1 Introduction

Many image stylization systems are designed for purely artistic purposes, like creating novel forms of digital art or helping laymen and artists with laborious or technically challenging tasks. Recently, several authors have proposed the goal of automatic stylization for efficient visual communication, to make images easier or faster to understand [DeCarlo and Santella 2002; Gooch et al. 2004; Raskar et al. 2004]. Although we also cater to artistic stylization (Figure 1, *Quantized*), our work focuses primarily on visual communication.

We present an automatic, real-time framework that abstracts imagery by modeling visual salience in terms of luminance and color opponency contrasts. We simplify regions of low contrast while enhancing high contrast regions. For adjusting contrasts, we employ several established image processing algorithms, which we modify for greater parallelism, temporal coherence, and directability.

We show that a separated approximation to a bilateral filter, applied iteratively, is an effective, parallelizable approximation to the process of simplifying images using anisotropic diffusion. We ensure that small input changes lead to similarly small output changes, on a frame-per-frame basis using several smooth quantization functions and avoid having to track object contours across frames.

A user study demonstrates the effectiveness of our framework for simple recognition and memory tasks, showing that our framework performs well even on small images, particularly on difficult subject matter like faces. We thus believe that visual communication applications will greatly benefit from our framework, as perceived fidelity is often paramount to actual fidelity for communication purposes. Possible applications include low-bandwidth videoconferencing and portable devices.

2 Related Work

Previous work in image-based stylization and abstraction systems varies in the use of scene geometry, video-based vs. static input, and the focus on perceptual task performance and evaluation.

Among the earliest work on image-based NPR was that of Saito and Takahashi [1990] who performed image processing operations on data buffers derived from geometric properties of 3D scenes. Our own work differs in that we operate on raw images, without requiring underlying geometry. To derive limited geometric infor-

*{holger|sven|bgooch}@cs.northwestern.edu

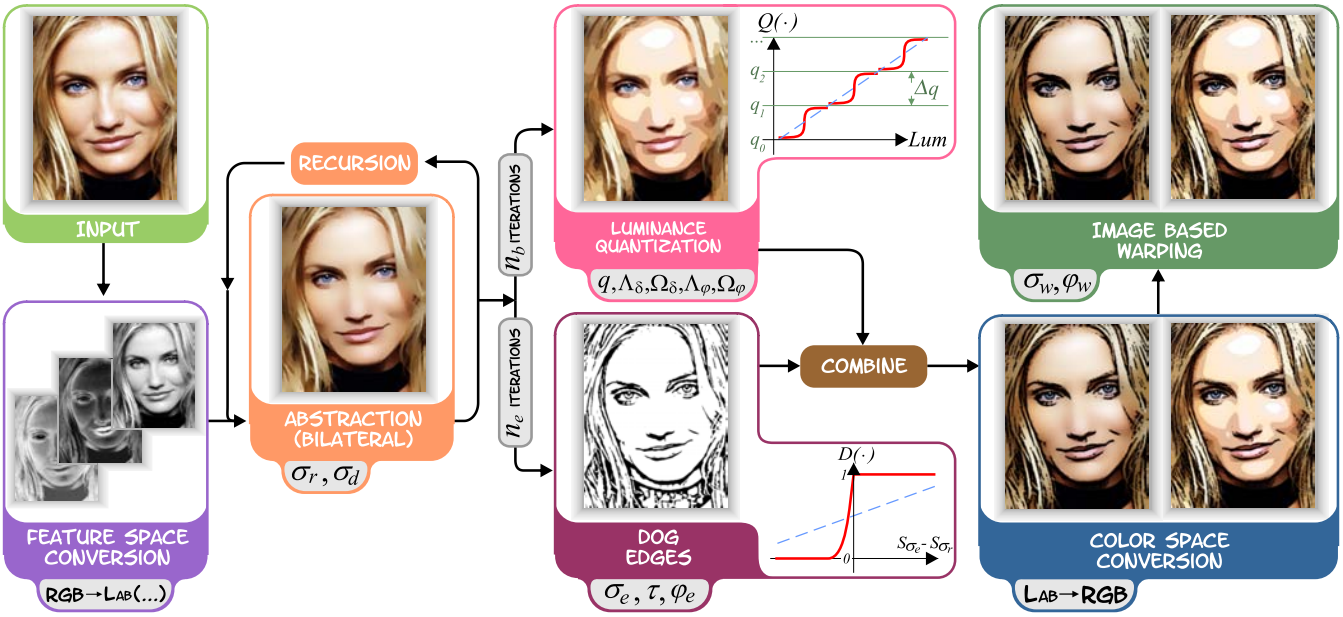

Figure 2: **Framework overview**. Each step lists the function performed, along with user parameters. The right-most paired images show alternative results, depending on whether luminance quantization is enabled (right) or not (left). The top image pair shows the final output.

mation from images, Raskar et al. [2004] computed ordinal depth from pictures taken with purpose-built multi-flash hardware. This allowed them to separate texture edges from depth edges and perform effective texture removal and other stylization effects. Our own framework does not model global effects such as repeated texture, but also requires no specialized hardware and does not face the technical difficulties of multi-flash for video.

Several video stylization systems have been proposed, mainly to help artists with labor-intensive procedures [Wang et al. 2004; Collomosse et al. 2005]. Such systems extended the mean-shift-based stylization approach of DeCarlo and Santella [2002] to computationally expensive three-dimensional video volumes. Difficulties with contour tracking required substantial user correction of the segmentation results, particularly in the presence of occlusions and camera movement. Our framework does not derive an explicit representation of image structure, thus limiting the types of stylization we can achieve. In turn, we gain a framework that is much faster to compute, fully automatic, and temporally coherent.

Fischer et al. [2005] explored the use of automatic stylization techniques in augmented reality applications. To make virtual objects less distinct from the live video stream, they applied stylization effects to both virtual and real inputs. Although parts of their system are similar to our own, their implementation is limited in the amount of detail it can resolve, and their stylized edges tend to suffer from temporal noise.

Recently, several authors of NPR systems have defined task-dependent objectives for their stylized imagery and tested these with perceptual user studies. DeCarlo and Santella [2002] use eye-tracking data to guide image simplification in a multi-scale system. In follow-up work, Santella and DeCarlo [2004] found that their eye-tracking-driven simplifications guided viewers to regions determined to be important. They also considered the use of computational salience as an alternative to measured salience. Our own work does not rely on eye-tracking data, although such data can be used. Our implicit visual salience model is less elaborate than the explicit model of Santella and DeCarlo's later work, but can be computed in real-time. Their explicit image structure representation allowed for more aggressive stylization, but included no provisions for the temporal coherence featured in our framework.

Gooch et al. [2004] automatically created monochromatic human facial illustrations from Difference-of-Gaussian (DoG) edges and a simple model of brightness perception. We use a similar edge model and evaluation study to Gooch et al. but additionally address color, real-time performance and temporal coherence.

3 Method

Our goal is to abstract images by simplifying their visual content while preserving or even emphasizing most of the perceptually important information.

Our framework is based on the assumptions that (1) the human visual system operates on different *features* of a scene, (2) changes in these features are of perceptual importance and therefore visually interesting (*salient*), and (3) polarizing these changes is a basic but useful method for automatic image abstraction.

Several image features are believed to play a vital role in low level human vision, among these are luminance, color opponency, and orientation [Palmer 1999]. A sudden spatial change (high contrast) in any of these features can represent boundaries of objects, subobject boundaries, or other perceptually important information. High contrast in these features is therefore linked to high visual salience, and low contrast to low salience. Based on this principle, several computational models of visual salience have been proposed [Privitera and Stark 2000; Itti and Koch 2001].

For our automatic, real-time implementation we implicitly compute visual salience with the following restrictions: we consider just two feature contrasts, luminance, and color opponency; we do not model effects requiring global integration; and we process images only within a small range of spatial scales. To allow for artistic control or more elaborate visual salience models, our framework alternatively accepts arbitrary scalar fields to direct abstraction.

The basic workflow of our framework is shown in Figure 2. We first exaggerate the given contrast in an image using nonlinear dif-

fusion. We then add highlighting edges to increase local contrast, and we optionally stylize and sharpen the resulting images.

3.1 Extended Nonlinear Diffusion

Perona and Malik [1991] defined a class of filters, called *anisotropic diffusion* filters, which have the desirable property of blurring small discontinuities *and* sharpening edges, as guided by a diffusion conduction function that varies over the image. Using such a filter with a conduction function based on feature contrast, we can amplify or subdue the given contrast in parts of an image. Barash and Comaniciu [2004] demonstrated that anisotropic diffusion solvers can be extended to larger neighborhoods, thus producing a broader class of *extended nonlinear diffusion* filters. This class includes iterated bilateral filters as one special case, which we prefer due to their larger support size and the fact that they can be approximated quickly and with few visual artifacts using a separated kernel [Pham and Vliet 2005].

Given an input image $f(\cdot)$, which maps pixel locations into some feature space, we define the following filter, $H(\cdot)$:

$$H(\hat{x},\sigma_d,\sigma_r) = \frac{\int e^{-\frac{1}{2}\left(\frac{\|\hat{x}-x\|}{\sigma_d}\right)^2} w(x,\hat{x}) f(x)\,dx}{\int e^{-\frac{1}{2}\left(\frac{\|\hat{x}-x\|}{\sigma_d}\right)^2} w(x,\hat{x})\,dx} \quad (1)$$

In this formulation, $\hat{x}$ is a pixel location, x are neighboring pixels, and σ_d is related to the blur radius. Increasing σ_d results in more blurring, but if σ_d is too large features may blur across significant boundaries. The range weighting function, $w(\cdot)$, determines where in the image contrasts are smoothed or sharpened by iterative applications of $H(\cdot)$.

$$w(x,\hat{x},\sigma_r) = (1-m(\hat{x}))\cdot w'(x,\hat{x},\sigma_r) + m(\hat{x})\cdot u(\hat{x}) \quad (2)$$

$$w'(x,\hat{x},\sigma_r) = e^{-\frac{1}{2}\left(\frac{\|f(\hat{x})-f(x)\|}{\sigma_r}\right)^2} \quad (3)$$

For the real-time, automatic case, we set $m(\cdot)=0$, such that $w(\cdot)=w'(\cdot)$ and Equation 1 becomes the familiar bilateral filter, where σ_r determines how contrasts will be preserved or blurred. Small values of σ_r preserve almost all contrasts, and thus lead to filters with little effect on the image, whereas for large values, $w'(\cdot) \xrightarrow{\sigma_r\to\infty} 1$, thus turning $H(\cdot)$ into a standard, linear Gaussian blur. For intermediate values of σ_r, iterative filtering of $H(\cdot)$ results in an extended nonlinear diffusion effect, where the degree of smoothing or sharpening is determined by local contrasts in $f(\cdot)$'s feature space. We use $\sigma_d=3$ throughout this paper and choose $\sigma_r=4.25$ for most images and the video.

With $m(\cdot)\neq 0$, the range weighting function, $w(\cdot)$, turns into a weighted sum of $w'(\cdot)$ and an arbitrary importance field, $u(\cdot)$, defined over the image. In this case, $m(\cdot)$ and $u(\cdot)$ can be computed via a more elaborate visual salience model [Itti and Koch 2001], derived from eye-tracking data (Figure 3, [DeCarlo and Santella 2002]), or painted by an artist [Hertzmann 2001].

Tomasi and Manduchi [1998] suggested computing the bilateral filter on a perceptually uniform feature space, such as *CIELab* [Wyszecki and Styles 1982], so that image contrast is adjusted depending on just noticeable differences. We follow this advice and our parameter values assume that $L\in[0,100]$ and $(a,b)\in[-127,127]$. Theoretically, the feature space could be extended to include additional features, such as orientation-dependent Gabor filters, although care would have to be taken to maintain perceptual uniformity of the combined feature space.

Figure 3: **Automatic vs. external abstraction**. *Top Row*: Original image by DeCarlo and Santella [2002] and their abstraction using eye-tracking data. *Bottom Row*: Our automatic abstraction, and data-driven abstraction based on the eye-tracking data.

3.2 Edge detection

In general, edges are defined by high local contrast, so adding visually distinct edges to regions of high contrast further increases the visual distinctiveness of these locations.

Marr and Hildreth [1980] formulated an edge detection mechanism based on zero-crossings of the second derivative of the luminance function. They postulated that retinal cells (*center*), which are stimulated while their *surrounding* cells are not stimulated, could act as neural implementations of this edge detector. A computationally simple approximation is the difference-of-Gaussians (DoG) operator. Rather than using a binary model of cell-activation, we define our DoG edges using a slightly smoothed step function, $D(\cdot)$ (bottom inset, Figure 2) to increase temporal coherence in animations. The parameter τ in Equation 4 controls the amount of center-surround difference required for cell activation, and φ_e controls the sharpness of the activation falloff. In the following, we define $S_{\sigma_e}\equiv S(\hat{x},\sigma_e)$ and $S_{\sigma_r}\equiv S(\hat{x},\sqrt{1.6}\cdot\sigma_e)$, with blur function $S(\cdot)$ given in Equation 5. The factor of 1.6 relates the typical receptive field of a cell to its surroundings [Marr and Hildreth 1980].

$$D(\hat{x},\sigma_e,\tau,\varphi_e) = \begin{cases} 1 & \text{if } (S_{\sigma_e}-\tau\cdot S_{\sigma_r})>0, \\ 1+\tanh(\varphi_e\cdot(S_{\sigma_e}-\tau\cdot S_{\sigma_r})) & \text{otherwise.}\end{cases} \quad (4)$$

$$S(\hat{x},\sigma_e) = \frac{1}{2\pi{\sigma_e}^2}\int f(x)c(\hat{x}-x,\sigma_e)\,dx \quad (5)$$

Here, σ_e determines the spatial scale for edge detection. The larger the value, the coarser the edges that are detected. The threshold level τ determines the sensitivity of the edge detector. For small values of τ, less noise is detected, but real edges become less prominent. As $\tau\to 1$, the filter becomes increasingly unstable. We use $\tau=0.98$ throughout. The falloff parameter, φ_e, determines the sharpness of edge representations, typically $\varphi_e\in[0.75,5.0]$. For n_b bilateral iterations, we extract edges after $n_e<n_b$ iterations to reduce noise. Typically, $n_e\in\{1,2\}$ and $n_b\in\{3,4\}$.

Canny [1986] devised a more sophisticated edge detection algorithm, which found use in several related works [DeCarlo and Santella 2002; Fischer et al. 2005]. Canny edges are guaranteed to lie on any real edge in an image, but can become disconnected for large values of σ_e and are computationally more expensive. DoG

Figure 4: **Parameter variations**. *Coarse*: Abstraction using coarse edges ($\sigma_e = 5$) and soft quantization steps ($q = 10, \Lambda_\varphi = 0.9, \Omega_\varphi = 1.6, \varphi_q = 3.1$). *Detailed*: Finer edges ($\sigma_e = 2$) and sharper quantization steps ($q = 14, \Lambda_\varphi = 3.4, \Omega_\varphi = 10.6, \varphi_q = 9.7$).

edges are cheaper to compute and not prone to disconnectedness but may drift from real image edges for large values of σ_e. We prefer DoG edges for computational efficiency and because their thickness scales naturally with σ_e.

Image-based warping (IBW) To fix small edge drifts linked to DoG edges and to sharpen the overall appearance of our final result we optionally perform an image-based warp (Figure 2, top-right). IBW is a technique first proposed by Arad and Gotsaman [1999] for image sharpening and edge-preserving expansion, in which they moved pixels along a warping field towards nearby edges. Loviscach [1999] proposed a simpler IBW implementation, in which the warping field is the blurred and scaled result of a Sobel filter of an input image. We use Loviscach's method with Gaussian blur $\sigma_w = 1.5$, and a scale factor of $\varphi_w = 2.7$.

3.3 Temporally coherent stylization

To open our framework further for creative use, we perform an optional color quantization step on the abstracted images, which results in cartoon or paint-like effects (Figures 1 and 4).

$$Q(\hat{x}, q, \varphi_q) \quad = \quad q_{nearest} + \frac{\Delta q}{2} \tanh(\varphi_q \cdot (f(\hat{x}) - q_{nearest})) \quad (6)$$

In Equation 6, $Q(\cdot)$ is the pseudo-quantized image, Δq is the bin width, $q_{nearest}$ is the bin boundary closest to $f(\hat{x})$, and φ_q is a parameter controlling the sharpness of the transition from one bin to another (top inset, Figure 2). Equation 6 is formally a discontinuous function, but for sufficiently large φ_q, these discontinuities are not noticeable.

For a fixed φ_q the transition sharpness is independent of the underlying image, possibly creating many noticeable transitions in large smooth-shaded regions. To minimize jarring transitions, we define the sharpness parameter, φ_q, to be a function of the luminance gradient in the abstracted image. We allow hard bin boundaries only where the luminance gradient is high. In low gradient regions, bin boundaries are spread out over a larger area. We thus offer the user a trade-off between reduced color variation and increased quantization artifacts by defining a target sharpness range $[\Lambda_\varphi, \Omega_\varphi]$ and a gradient range $[\Lambda_\delta, \Omega_\delta]$. We clamp the calculated gradients to $[\Lambda_\delta, \Omega_\delta]$ and then generate a φ_q value by mapping them linearly to $[\Lambda_\varphi, \Omega_\varphi]$. The effect for typical parameter values are hard, cartoon-like boundaries in high gradient regions and soft, painterly-like transitions in low gradient regions (Figure 4). Typical values for these parameters are $q \in [8, 10]$ equal-sized bins and a gradient range of $[\Lambda_\delta = 0, \Omega_\delta = 2]$, mapped to sharpness values between $[\Lambda_\varphi = 3, \Omega_\varphi = 14]$.

Another significant advantage of our pseudo-quantization implementation is temporal coherence. In standard quantization, an arbitrarily small luminance change can push a value to a different bin, thus causing a large output change for a small input change, which is particularly troublesome for noisy input. With soft quantization, such a change is spread over a larger area, making it less noticeable. Using our gradient-based sharpness control, sudden changes are further subdued in low-contrast regions, where they would be most objectionable.

Figure 5: **Sample images from evaluation studies.** The top row shows the original images and the bottom row shows the abstracted versions. All images use the same σ_e for edges and the same number of simplification steps, n_b. *Left*: Faces similar to those in Study 1. *Right*: Sample images from Study 2.

4 Evaluation

To verify that our abstracted images preserve or even distill perceptually important information, we performed two task-based studies to test recognition speed and short term memory retention. Our studies use small images because we see portable visual communication and low-bandwidth applications to practically benefit most from our framework and because small images may be a more telling test of our framework, as each pixel represents a larger percentage of the image.

Participants In each study, 10 (5 male, 5 female) undergraduates, graduate students or research staff acted as volunteers.

Materials Images in Study 1 are scaled to 176×220, while those in Study 2 are scaled to 152×170. These resolutions approximate those of many portable devices. Images are shown on a 30-inch Apple Cinema Display at a distance of 24 inches. The background of the monitor is set to white and the displayed images subtend a visual angle of 6.5 and 6.0 degrees respectively.

In Study 1, 50 images depicting the faces of 25 famous movie stars are used as visual stimuli. Each face is depicted as a color photograph and as a color abstracted image created with our framework. Five independent judges rated each pair of photograph and abstracted image as good likenesses of the face they portrayed. In Study 2, 32 images depicting arbitrary scenes are used as visual stimuli. Humans are a component in 16 of these images. Examples of stimulus images are shown in Figure 5.

Analysis For both studies, p-values are computed using two-way analysis of variance (ANOVA), with $\alpha = 0.05$.

4.1 Study 1: *Recognition Speed*

Study 1 assesses the recognition time of familiar faces presented as abstract images and photographs. The study uses a protocol [Stevenage 1995] demonstrated to be useful in the evaluation of recognition times for facial images [Gooch et al. 2004].

Procedure Study 1 consists of two phases: (1) reading the list of 25 movie star names out loud, and (2) a reaction time task in which participants are presented with sequences of the 25 facial images. All faces take up approximately the same space in the images and are three quarter views. By pronouncing the names of the people that are rated, participants tend to reduce the *tip-of-the-tongue* effect where a face is recognized without being able to quickly recall the associated name [Stevenage 1995]. For the same reason, participants are told that first, last or both names can be given, whichever is easiest. Each participant is asked to say the name of the person pictured as soon as that person's face is recognized. A study coordinator records reaction times, as well as accuracy of the answers. Images are shown and reaction times recorded using the *Superlab* software product for 5 seconds at 5-second intervals. The order of image presentation is randomized for each participant.

Results and Discussion In our study, participants are faster ($p < 0.018$) at naming abstract images ($M = 1.32$s) compared to photographs ($M = 1.51$s). The accuracy for recognizing abstract images and photographs are 97% and 99% respectively, indicating that there is no significant speed for accuracy trade-off. It can further be concluded that substituting abstract images for fully detailed photographs reduces recognition latency by 13%, a significant improvement not found by Sevenage [1995] and Gooch et al. [2004]. However, neither author used color images as stimuli.

4.2 Study 2: *Memory Game*

Study 2 assesses memory retention for abstract images versus photographs with a memory game, consisting of a grid of 24 randomly sorted cards placed face-down. The goal is to create a match by turning over two identical cards. If a match is made, the matched cards are removed. Otherwise, the cards are placed face down and another set of cards are turned over. The game ends when all pairs are matched. We created a Java program of the card game in which a user turns over a virtual card with a mouse click. The 12 images used in any given memory game are randomly chosen from the pool of 32 images without replacement, and randomly arranged. The program records the time it takes to complete a game and the number of cards turned over.

Procedure Study 2 consists of three phases: (1) a practice memory game with alphabet cards, (2) a memory game of photographs, and (3) a memory game of abstract images. All participants first play a practice game with alphabet cards to learn the interface and to develop a game strategy. No data is recorded for the practice phase. For the remaining two phases, half the participants are presented with photographs followed by abstracted images, and the other half is presented with abstracted images followed by photographs.

Results and Discussion In our study, participants are quicker ($p_{time} < 0.003$, $p_{clicks} < 0.004$) in completing a memory game using abstract images ($M_{time} = 59.95$ s, $M_{clicks} = 49.2$) compared to photographs ($M_{time} = 76.13$ s, $M_{clicks} = 62.4$). The study demonstrates that participants play the abstracted image version of the game faster than the version using photographs. In addition, using the abstracted images requires fewer cards to be turned over, possibly indicating that it is easier to remember previously revealed

Figure 6: **Failure case.** A case where our contrast-based importance assumption fails. *Left*: The subject of this photograph has very low contrast compared with its background. *Right*: The cat's low contrast fur is abstracted away, while the detail in the structured carpet is further emphasized. Despite this rare reversal of contrast assignment, the cat is still well represented.

abstractions. We thus conclude that the automatic image abstraction of our framework may produce more distinctive imagery.

5 Discussion and Conclusion

Performance We implemented and tested our framework in both a GPU-based real-time version, using OpenGL and fragment programs, and a CPU-version using OpenCV. Both versions were tested on an Athlon 64 3200+ with Windows XP and a GeForce GT 6800. Performance values depend on graphics drivers, image size, and framework parameters. Typical values for a 640×480 video stream and default parameters are $9-15$ frames per second (FPS) for the GPU version and $0.3-0.5$ FPS for the CPU version.

Limitations Our framework depends on local contrast to estimate visual salience. Images with very low contrast likely abstract too much and loose significant detail. Simply increasing contrast of the original image may reduce this problem, but can also increase noise. Figure 6 demonstrates an inversion of our general assumption, where important foreground objects have low contrast while background regions have high contrast. In practice we have obtained good results for many indoor and outdoor scenes.

Human vision operates at various spatial scales simultaneously. By applying multiple iterations of a non-linear blurring filter we cover a small range of spatial scales, but the range is not explicitly parameterized and not as extensive as that of real human vision.

Several high-contrast features that may be emphasized by our framework are actually deemphasized in human vision, among these specular highlights and repeated texture. Dealing with these phenomena using existing techniques requires global image processing, which is impractical in real-time on today's GPUs, due to their limited gather-operation capabilities.

Our fixed equidistant quantization boundaries are arbitrary, making it difficult to control results for artistic purposes. Constructing spatially varying boundaries to better account for underlying dynamic range might prove beneficial.

Compression A discussion of theoretical data compression and codecs exceeds the scope of the paper, but Pham and Vliet [2005] have shown that video compresses better when bilaterally filtered, judged by RMS error and MPEG quality score. Collomosse et al. [2005] list theoretical compression results for vectorized cartoon images. Possibly most applicable to this paper is work by Elder [1999], who describes a method to store the color information of an image only in high-contrast regions, achieving impressive compression results.

Indication *Indication* is the process of representing a repeated texture with a small number of exemplary patches and relying on

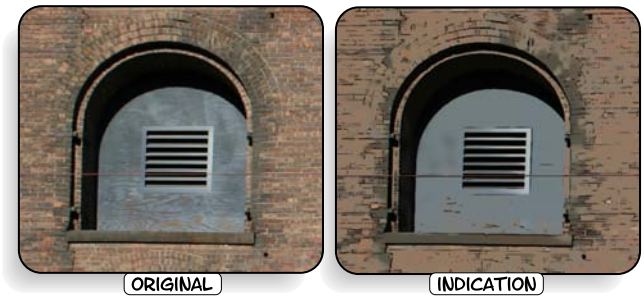

Figure 7: **Automatic indication**. The inhomogeneous texture of the bricks causes spatially varying abstraction. The resulting edges *indicate* a brick texture instead of depicting each individual brick.

an observer to interpolate between patches. For structurally simple, slightly inhomogeneous textures with limited scale variation, like the brick wall in Figure 7, our framework can perform simple automatic indication. As noted by DeCarlo and Santella [2002], such simple indication does not deal well with complex or foreshortened textures. Our automatic indication is not as effective as the user-drawn indications of Winkenbach and Salesin [1994], but some user guidance can be supplied via Equation 2.

Conclusion We have presented a simple and effective real-time framework that abstracts images while retaining much of their perceptually important information, as demonstrated in our user study. Our optional stylization step is temporally highly stable, results in effective color flattening and is much faster than the mean-shift procedures used in offline cartoon stylization for video [Collomosse et al. 2005; Wang et al. 2004]. Interestingly, several authors [Barash and Comaniciu 2004; Boomgaard and de Weijer 2002] have shown that anisotropic diffusion filters are closely related to the mean-shift algorithm. It is thus conceivable that various graphics applications that today rely on mean-shift could benefit from the much speedier anisotropic diffusion pre-process used in this paper.

Acknowledgements Many thanks to Amy Gooch, David Feng and our anonymous reviewers for their helpful writing suggestions; Jack Tumblin for inspiring discussions; Pin Ren for photographic assistance; Tom Lechner for modeling; the Northwestern GFX Group for their support; Rosalee Wolfe and Karen Alkoby for the deaf signing video; Douglas DeCarlo and Anthony Santella for proof-reading and supplying Figure 3 (top) and eye-tracking data; Marcy Morris and James Bass for acquiring image permission from Ms. Diaz. This material is based upon work supported by the National Science Foundation under Grant No. 0415083. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors. Figure 1 (left) and Figure 4 by Martina Winnemöller. Figure 2, Cameron Diaz, with permission. Figure 3 (top) by Anthony Santella, with permission of ACM. Figure 5, celebrities by Rita Molnár, Creative Commons License. Figure 7 by Hans Beushausen.

References

ARAD, N., AND GOTSMAN, C. 1999. Enhancement by image-dependent warping. *IEEE Trans. on Image Processing 8*, 9, 1063–1074.

BARASH, D., AND COMANICIU, D. 2004. A common framework for nonlinear diffusion, adaptive smoothing, bilateral filtering and mean shift. *Image and Video Computing 22*, 1, 73–81.

BOOMGAARD, R. V. D., AND DE WEIJER, J. V. 2002. On the equivalence of local-mode finding, robust estimation and mean-shift analysis as used in early vision tasks. *16th Internat. Conf. on Pattern Recog. 3*, 927–390.

CANNY, J. F. 1986. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence 8*, 769–798.

COLLOMOSSE, J. P., ROWNTREE, D., AND HALL, P. M. 2005. Stroke surfaces: Temporally coherent artistic animations from video. *IEEE Trans. on Visualization and Computer Graphics 11*, 5, 540–549.

DECARLO, D., AND SANTELLA, A. 2002. Stylization and abstraction of photographs. *ACM Trans. Graph. 21*, 3, 769–776.

ELDER, J. H. 1999. Are edges incomplete? *Internat. Journal of Computer Vision 34*, 2-3, 97–122.

FISCHER, J., BARTZ, D., AND STRASSER, W. 2005. Stylized Augmented Reality for Improved Immersion. In *Proc. of IEEE VR*, 195–202.

GOOCH, B., REINHARD, E., AND GOOCH, A. 2004. Human facial illustrations: Creation and psychophysical evaluation. *ACM Trans. Graph. 23*, 1, 27–44.

HERTZMANN, A. 2001. Paint by relaxation. In *CGI '01:Computer Graphics Internat. 2001*, 47–54.

ITTI, L., AND KOCH, C. 2001. Computational modeling of visual attention. *Nature Reviews Neuroscience 2*, 3, 194–203.

LOVISCACH, J. 1999. Scharfzeichner: Klare bilddetails durch verformung. *Computer Technik 22*, 236ff.

MARR, D., AND HILDRETH, E. C. 1980. Theory of edge detection. *Proc. Royal Soc. London, Bio. Sci. 207*, 187–217.

PALMER, S. E. 1999. *Vision Science: Photons to Phenomenology*. The MIT Press.

PERONA, P., AND MALIK, J. 1991. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. on Pattern Analysis and Machine Intelligence 12*, 7.

PHAM, T. Q., AND VLIET, L. J. V. 2005. Separable bilateral filtering for fast video preprocessing. In *IEEE Internat. Conf. on Multimedia & Expo*, CD1–4.

PRIVITERA, C. M., AND STARK, L. W. 2000. Algorithms for defining visual regions-of-interest: Comparison with eye fixations. *IEEE Trans. on Pattern Analysis and Machine Intelligence 22*, 9, 970–982.

RASKAR, R., TAN, K.-H., FERIS, R., YU, J., AND TURK, M. 2004. Non-photorealistic camera: depth edge detection and stylized rendering using multi-flash imaging. *ACM Trans. Graph. 23*, 3, 679–688.

SAITO, T., AND TAKAHASHI, T. 1990. Comprehensible rendering of 3-D shapes. In *Proc. of ACM SIGGRAPH 90*, 197–206.

SANTELLA, A., AND DECARLO, D. 2004. Visual interest and NPR: an evaluation and manifesto. In *Proc. of NPAR '04*, 71–78.

STEVENAGE, S. V. 1995. Can caricatures really produce distinctiveness effects? *British Journal of Psychology 86*, 127–146.

TOMASI, C., AND MANDUCHI, R. 1998. Bilateral filtering for gray and color images. In *Proceedings of ICCV '98*, 839.

WANG, J., XU, Y., SHUM, H.-Y., AND COHEN, M. F. 2004. Video tooning. *ACM Trans. Graph. 23*, 3, 574–583.

WINKENBACH, G., AND SALESIN, D. H. 1994. Computer-generated pen-and-ink illustration. In *Proc. of ACM SIGGRAPH 94*, 91–100.

WYSZECKI, G., AND STYLES, W. 1982. *Color Science: Concepts and Methods, Quantitative Data and Formulae*. Wiley, New York, NY.