Automated Eye Motion Using Texture Synthesis

Zhigang Deng, J.P. Lewis, and Ulrich Neumann
University of Southern California

Modeling human eyes requires special care. Our goal is to synthesize realistic eye-gaze and -blink motion, accounting for any possible correlations between the two.

Because all humans are especially aware of the nuances of the human face, rendering a believable human face is a challenging subject for computer graphics. Of all parts of the face, the eyes are particularly scrutinized because eye gaze is one of the strongest cues to the mental state of another person. When people are talking, they look to each others' eyes to judge interest and attentiveness, and in turn look into the eyes to signal an intent to talk.

In projects that attempt to create realistic computer-animated faces, the eyes are often the feature that observers point out as looking wrong. Producing convincing eyes in computer graphics applications requires attention to several topics in modeling, rendering, and animation. (See the "Related Work" sidebar for information on other research projects in this area.)

Our technique adopts a data-driven texture synthesis approach to the problem of synthesizing realistic eye motion. The basic assumption is that eye gaze probably has some connection with eyelid motion (see Figure 1 on page 26), as well as with head motion and speech. But the connection is not strictly deterministic and would be difficult to characterize explicitly. For example, as suggested in Figure 1, gaze changes often appear to be associated with blinks. A major advantage of the data-driven approach is that the investigator does not need to determine whether these apparent correlations actually exist. If the correlations occur in the data, the synthesis (properly applied) will reproduce them.

With these assumptions, a data-driven stochastic modeling approach makes sense. The combined

Related Work

Recently, several research efforts have modeled eye-gaze motions in different scenarios. One team proposed rule-based approaches for generating animated conversation between multiple agents given specified text.[1,2] Another team proposed a framework for computing visual attending behaviors (such as eye and head motions) of virtual agents in dynamic environments.[3] Another team investigated whether eye-gaze direction clues could be used as a reliable signal for determining who is talking to whom in multiparty conversations.[4,5]

Most of the work cited here takes a goal-directed approach to gaze, focusing on major gaze changes such as those for creating conversational turn taking. This high-level direction indicates what the eyes should do and where the eyes should look, but there still is some freedom as to how particular gaze changes should be performed. The detailed timing of eye saccades and blinks can convey various mental states, such as excited or sleepy.

Recently, the Eyes Alive project presented the first in-depth treatment of these textural aspects of eye movement, demonstrating the necessity of this detail for achieving realism and conveying an appropriate mental state.[6] In the Eyes Alive model, signals from an eye tracker are analyzed to produce a statistical model of eye saccades. Eye movement is remarkably complex, however, and the Eyes Alive model does not consider all aspects of eye movement. In particular, the project used only first-order statistics and did not consider gaze–eyelid coupling and vergence. The main text addresses the first two of these issues by introducing a more powerful statistical model that can simultaneously capture gaze–blink coupling.

The problem of synthesizing eye movement can be considered to be an instance of a general signal-modeling problem. We distinguish parametric from nonparametric or data-driven approaches. In the former, the investigator proposes an analytic model of the phenomenon in question, with some number of parameters that are then fit to the data (a purely rule-based approach with no explicit dependence on the data is a further possibility). This

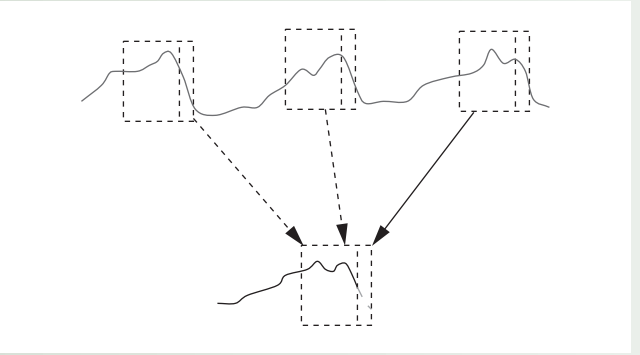
 Published by the IEEE Computer Society

technique can produce compact, economical models of the data, but it has the danger that the analytic model might fail to capture some important aspects of the data.

Data-driven approaches, on the other hand, provide an alternative in which the data itself is queried to produce new signals as desired. However, in such techniques, sufficient training data must be available and there is no explicit model of the phenomena (hence no understanding is acquired). But by using data-driven techniques, characteristics of the data are not lost as a result of choosing an incorrect or insufficiently powerful model.

Researchers have applied the data-driven approach to several problems in computer graphics recently. The "motion-capture soup" research approaches human body motion synthesis by first partitioning human motion signals into small segments and then concatenating these segments together, chosen by an optimization algorithm.[7-9] Several recent and successful texture-synthesis articles also explore a data-driven approach.[10,11] The basic idea is to grow one sample (or one patch) at a time, given an initial seed, by identifying all regions of the sample texture that are sufficiently similar to the neighborhood of the sample, and randomly selecting the corresponding sample from one of these regions (see Figure A).

These texture synthesis algorithms have some resemblance to the motion-capture soup approaches, although they differ in that the system searches the entire texture-training data for matching candidates. In the motion-capture soup case, the system divides the data into segments in advance and only searches transitions between these segments. This tradeoff assumes that possible matches in texture-like data are too many and too varied to be profitably identified in advance.



A Signal synthesis with nonparametric sampling schemes. Regions in a sample signal (top) that are similar to the neighborhood of the signal being synthesized (bottom) are identified. One such region is randomly chosen, and new samples are copied from it to the synthesized signal.

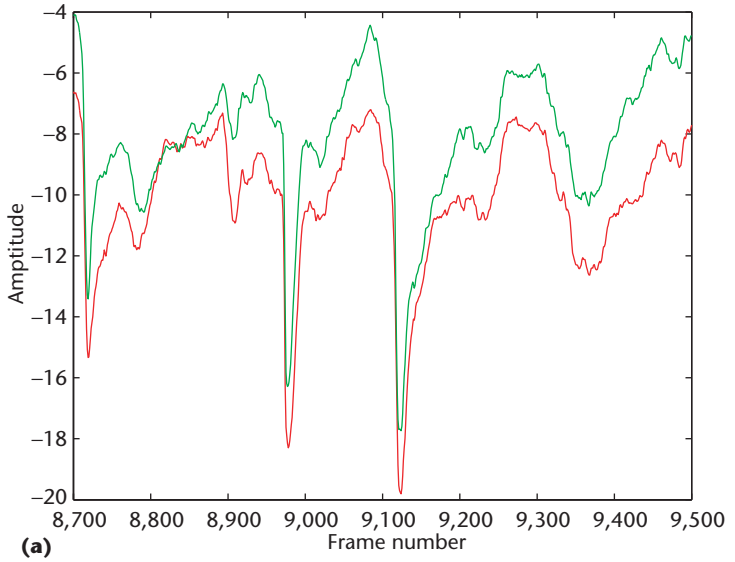
References

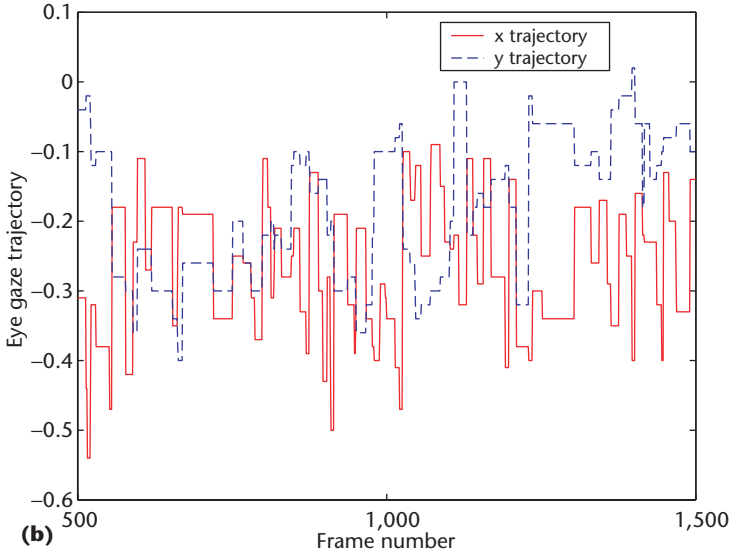
1. J. Cassell et al., "Animated Conversation: Ruled-Based Generation of Facial Expression Gesture and Spoken Intonation for Multiple Conversational Agents," *Proc. Siggraph*, ACM Press, 1994, pp. 413-420.
2. J. Cassell, H. Vilhjalmsson, and T. Bickmore, "BEAT: The Behavior Expression Animation Toolkit," *Proc. Siggraph*, ACM Press, 2001, pp. 477-486.
3. S. Chopra-Khullar and N. Badler, "Where to Look? Automating Visual Attending Behaviors of Virtual Human Characters," *Proc. 3rd ACM Conf. Autonomous Agents*, ACM Press, 1999, pp. 16-23.
4. R. Vertegaal, G.V. Derveer, and H. Vons, "Effects of Gaze on Multiparty Mediated Communication," *Proc. Graphics Interface*, Morgan Kaufmann, 2000, pp. 95-102.
5. R. Vertegaal et al., "Eye Gaze Patterns in Conversations: There is More to Conversational Agents than Meets the Eyes," *Proc. ACM CHI Conf. Human Factors in Computing Systems*, ACM Press, 2001, pp. 301-308.
6. S.P. Lee, J.B. Badler, and N. Badler, "Eyes Alive," *ACM Trans. Graphics*, vol. 21, no. 3, 2002, pp. 637-644.
7. O. Arikan and D. Forsythe, "Interactive Motion Generation from Examples," *ACM Trans. Graphics*, ACM Press, vol. 21, no. 3, 2002, pp. 483-490.
8. L. Kovar et al., "Motion Graphs," *ACM Trans. Graphics*, ACM Press, vol. 21, no. 3, 2002, pp. 473-482.
9. Y. Li, T. Wang, and H.-Y. Shum, "Motion Texture: A Two-Level Statistical Model for Character Motion Synthesis," ACM Trans. Graphics, ACM Press, vol. 21, no. 3, 2002, pp. 465-472.
10. A. Efros and T.K. Leung, "Texture Synthesis by Non-Parametric Sampling," *Proc. Int'l Conf. Computer Vision* (ICCV), IEEE CS Press, 1999, pp. 1033-1038.
11. L. Liang et al., "Real-Time Texture Synthesis by Patch-Based Sampling," *ACM Trans. Graphics*, 2001, vol. 20, no. 3, pp. 127-150.

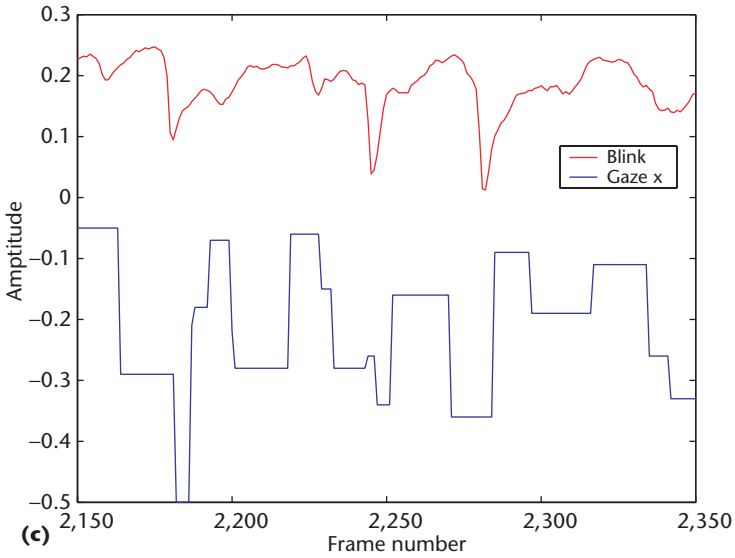
gaze–blink vector signal does not have obvious segmentation points, as is the case with body motion capture data. Thus, our technique adapts the data-driven texture synthesis approaches to the problem of realistic eye-motion modeling. Our technique considers eye gaze and aligned eye-blink motion together as an eye-motion-texture sample, which we use for synthesizing novel but similar eye motions.

To justify this choice of a texture-synthesis approach over a hand-crafted statistical model, consider the order statistics classification of statistical models. The first-order statistics $p(x)$ used in the Eyes Alive project capture the probability of events of different magnitude but do not model any correlation between different events.[1] Correlation $E[xy]$ is a second-order moment, or an average of the second-order statistics $p(x,y)$. Third-order statistics would consider the joint probability of triples of events $p(x,y,z)$ and so forth. Unfortunately, it's difficult to know the answer to the question of what order of statistics should be used to capture all the relevant characteristics of joint gaze–eyelid movement. Low-order statistics, such as probability density and correlation, clearly do not capture some visible features (see Figure 2). Higher-order models are algorithmically complex and perform poorly if derived from insufficient data.

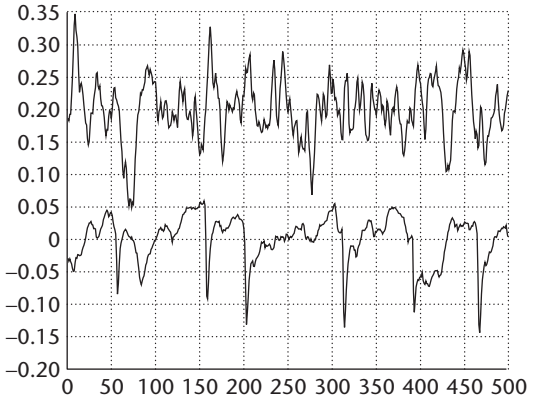
It's also possible to use a hidden Markov model because HMMs can approximate all real-world probability distributions, and (as in the case of speech) the HMM architecture also can provide a deeper model of the phenomenon. In the case of modeling eye move-







1 Connection between eye gaze and eyelid motion: (a) *y*-coordinate motion of captured left and right eye blinks (green for left and red for right). (b) Labeled eye-gaze signals. (c) Eye-blink and *x*-coordinate gaze signals plotted simultaneously.



2 Eye-blink data (bottom) and a synthesized signal with the same autocovariance (top). A simple statistical model cannot reproduce the correlation and asymmetry characteristics evident in the data.

ment, however, the required number of hidden states is not as obvious as it is in the speech case. Because the model must potentially capture subtle mental states that are manifested in eye movement (such as agitated, distracted, and so forth), the hidden states might not be easily interpretable. Although an HMM approach would probably work for our problem, the effort of designing and training a suitable HMM might not be worth the effort if the goal is simply to synthesize animated movement mimicking an original sample. By adopting a data-driven approach, we avoid this issue and let the data speak for itself while achieving movement that is indistinguishable in character from captured eye signals.

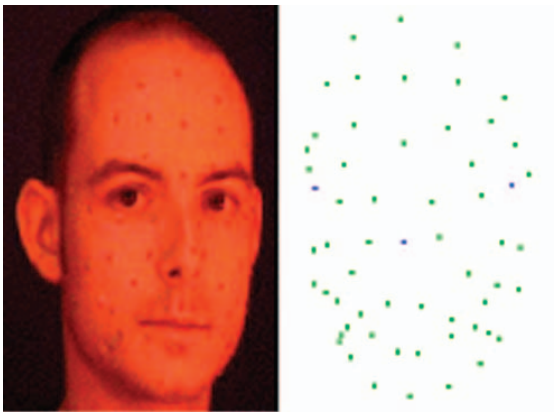
In this article, we focus on improving eye movement realism, specifically the cadence and distribution of gaze *saccades*—small jerky movements of the eyes as they jump from fixation on one point to another—with correlated eyelid motion. Our contribution to research in this area is to synthesize eye saccade signals as a 1D texture-synthesis problem and to apply recent texture-synthesis approaches to this animation domain.

Data acquisition and preprocessing

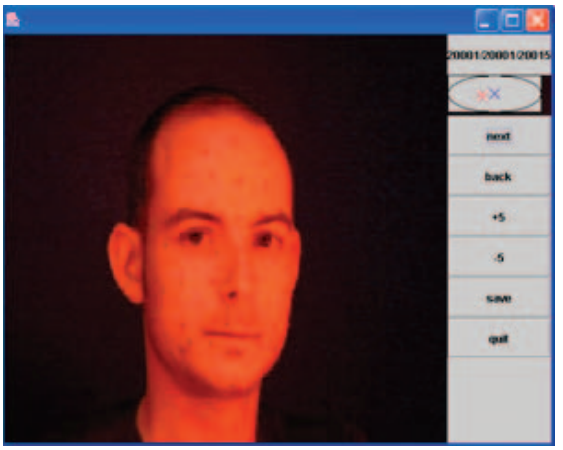
For raw data, we captured a human subject's facial motion with 59 markers on his face (see Figure 3) and recorded corresponding audio and video tracks. The recorded corpus is about two minutes in duration and consists of about 16,500 motion-capture frames. The motion-capture rig has six cameras sampling at 120 Hz. Two markers on the eyelids provided the eye-blink motion data.

After capture, we normalized the data. First we translated all data points to make a point on the nose the local coordinate center of each frame. Second, we picked one frame with a neutral and a closed-mouth head pose as reference frame. Third, we used three approximately rigid points (nose point and corner points of eyes) to define a local coordinate origin for each frame. Last, we rotated each frame to align it with the reference frame.

To extract the eye-blink motion signals from captured data, we converted the captured eyelid motion to a 1D



3 Facial motion capture markers.



4 Eye direction training signals are digitized with a custom GUI.

blink-texture signal. Because the motions in three directions (x, y, and z) are strongly correlated, we can represent the eye-blink motion in 3D with a 1D blink signal based on the dominant y (vertical) direction. Figure 1 illustrates the *y*-coordinate motions of captured left and right eye blinks.

After examining the captured data, we found that the motion of the left eyelid is nearly synchronized with that of the right. As a result, we need only the motion-capture trace for one eye to create the eye-blink texture signal. By scaling the *y*-coordinates of the eye-blink motion into the range [0,1], we get a 1D eye-blink texture signal. Here, 0 denotes a closed eyelid, 1 denotes a fully open eyelid, and any value between 0 and 1 represents a partially open eyelid. In this procedure, we ignored outliers and used interpolated neighbor points to fill the gaps.

We obtained corresponding eye-gaze direction signals by manually estimating the eye direction in training videos, frame by frame, using an eyeball-tracking widget in a custom GUI (see Figure 4). While the manually estimated direction data is not completely accurate, it qualitatively captures the character and cadence of real human gaze movement, and the gaze durations are frame accurate. Information on automatic saccade-identification techniques can be found elsewhere.[2] Figure 1 illustrates the resulting gaze signals.

Note the rectangular or piecewise-continuous character of these signals, which reflects the fact that gaze tends to fixate on some point for a period of time and then rapidly shift to another point. When doing a large change of gaze, it appears that the human eye often executes several smaller shifts rather than a single large and smooth motion. We also observed that gaze changes frequently occur during blinks.

Eye motion synthesis

After we extracted and aligned the eye-blink and gaze-motion signals, we used texture synthesis to synthesize new eye motions. We used the patch-based sampling algorithm because of its time efficiency.[3] The basic idea is to grow one fixed-size texture patch at a time. We chose the patch randomly from qualified candidate patches in the input texture sample. Figure A (in the sidebar) illustrates the basic idea. Each texture sample (analogous to a pixel in the 2D image-texture case) consists of three elements: eye-blink signal, *x* position of eye-gaze signal, and *y* position of the eye-gaze signal.

We used $t^i = (b, g_1, g_2)$ to represent one sample. First, we estimate the variance of each element:

$$V_\tau = \frac{1}{(N-1)} \sum_{i=1}^{N} (t_\tau^i - \bar{t}_\tau)^2$$
$$\tau = b, g1, g2$$

We divided each component by its variance (V_b, V_{g1}, or V_{g2}) to give it equal contribution to the candidate patch searching. We defined the distance metric between two texture blocks as follows:

$$d(B_{\text{in}}, B_{\text{out}}) = \left(\frac{1}{A} \sum_{k=1}^{A} d\_\text{tex}(t_{\text{in}}^k, t_{\text{out}}^k)\right)^{1/2}$$

and

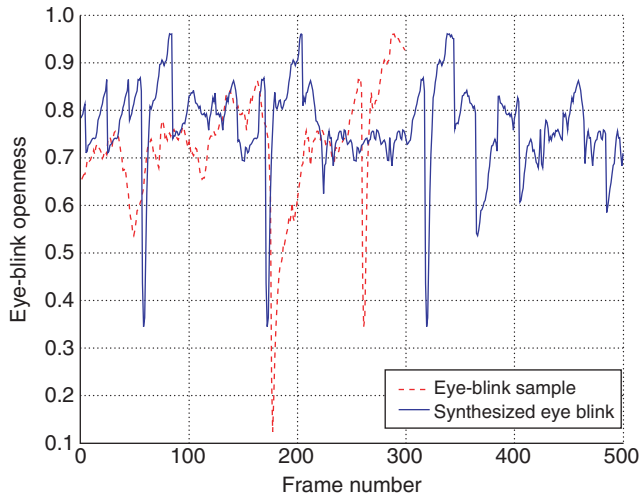
$$d\_\text{tex}(t^1, t^2) = (b^1 - b^2)^2 / V_b + (g_1^1 - g_1^2)^2 / V_{g1} + (g_2^1 - g_2^2)^2 / V_{g2}$$

Here, t_{in} represents the input texture sample, t_{out} represents the synthesized (output) texture sample, and A is the size of the boundary zone that functions as a search window. In our case, the patch size is 20 and the boundary zone size is 4. Patch size depends on the properties of a given texture. A proper choice is critical to the success of the synthesis algorithm. If the patch size is too small, it cannot capture the characteristics of eye motion, and it might cause the eye gaze to change too frequently and look too active. If it's too large, there are fewer possible matching patches and more training data is required to produce variety in the synthesized motion.

Another parameter we used in this algorithm is the distance tolerance:

$$d_{\max} = \varepsilon\left(\frac{1}{A} \sum_{k=1}^{A} (d\_\text{tex}(t_{\text{out}}^k, 0_3))^2\right)^{1/2} \quad (1)$$

In Equation 1, 0_3 is a 3D zero vector. We set the tolerance



5 Synthesized eye-blink motion (blue) versus eye-blink sample (red). The *x*-axis represents the frame number and the *y*-axis represents the openness of eyelids.

constant to $\varepsilon = 0.2$. Figures 5 and 6 illustrate synthesized eye motions. We synthesized the signals in these figures at the same time, which is necessary to capture the possible correlations between them.

Patch size selection

To determine the proper patch size, we used the transition interval distribution. For eye-blink data, we counted all the time intervals (in terms of frame number) between two adjacent approximate eye-blink actions. If the eye-blink value (openness of the eyelid) was less than the threshold value set to 0.2, we counted it as an approximate eye blink. We used this threshold only for the purpose of choosing the texture patch size. The eye-blink synthesis uses the original unthresholded data. For eye gaze data, we counted all time intervals between two adjacent large saccadic movements, which we defined as places where either the *x*- or *y*-movement is larger than a threshold value (we set it to 0.1).

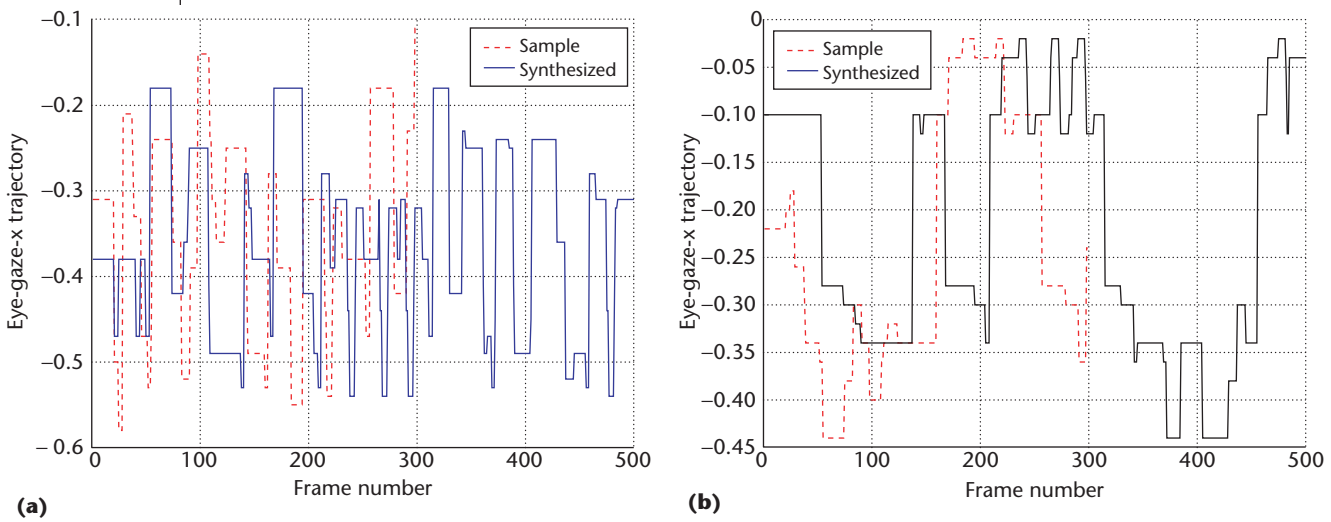
Finally, we gathered all these time intervals and plotted their distributions (see Figure 7). This figure shows us that a time interval of 20 is a transition point. The covered percentage increased rapidly when the time interval is less than 20. Beyond 20, the covered percentage slows down rapidly. Also, when the time interval limit is set to 20, it accounts for 55.68 percent of the large eye motions. We therefore used 20 as the proper patch size for the motion synthesis algorithm.

The boundary zone size is useful to control the number of texture-block candidates. If the size of the boundary zone is too large, then few candidates are available and the diversity of the synthesized motion is impaired. On the other hand, if this size is too small, some of the higher-order motion statistics are not captured, and the resulting synthesis looks jumpy. We adopted a strategy similar to patch-based sampling[3] where the size of the boundary zone is a fraction of patch size. As such, we chose four as the size of the boundary zone. In practice, that number works well.

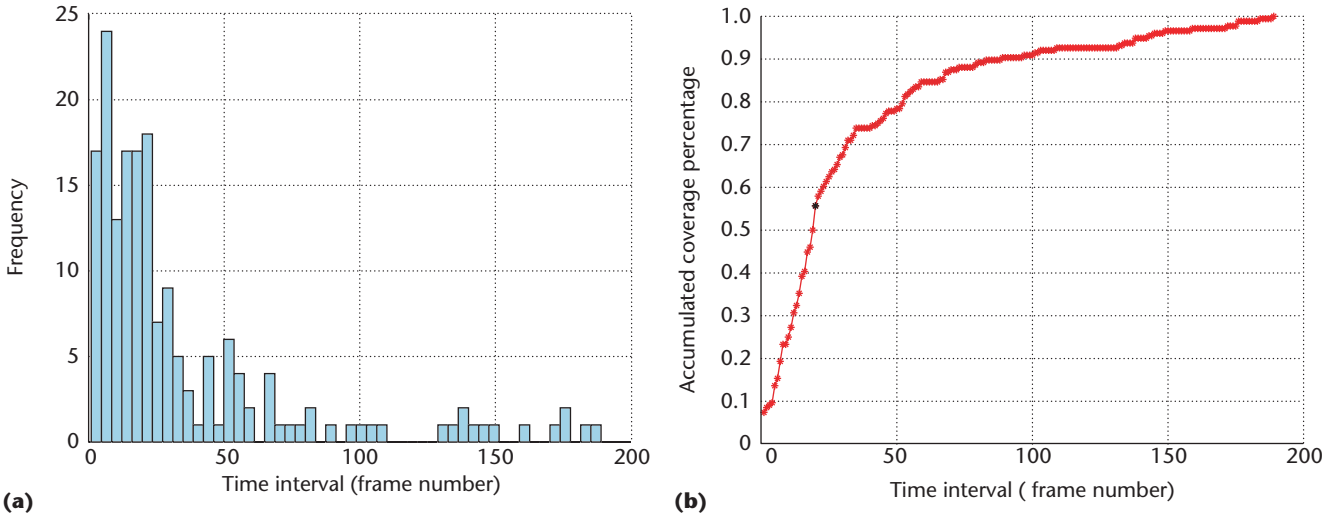
Results and evaluations

We arbitrarily used one segment from an extracted sample sequence as an eye-motion texture sample and then synthesized novel eye motions similar to this sample. We found that our synthesis produces eye movement that looked alert and lively (rather than the drugged, agitated, or schizophrenic moods that observers attribute to random or inappropriate eye movements, although the realism is difficult to judge because the face model itself is not completely photorealistic). Figure 8 shows some frames of synthesized eye motions.

To compare our method with other approaches, we synthesized eye motions on the same face model using three different methods and then conducted subjective tests. In the first experiment (method I), we used the Eyes Alive model to generate gaze motion. We sampled eye-blink motion from a Poisson distribution. A *discrete event* in this Poisson distribution means an eyelid-closed event.



6 Eye motions: (a) Synthesized eye-gaze-*x* trajectory (blue) versus eye-gaze-x sample (red) and (b) Synthesized eye-gaze-y trajectory (blue) versus eye-gaze-y sample (red).



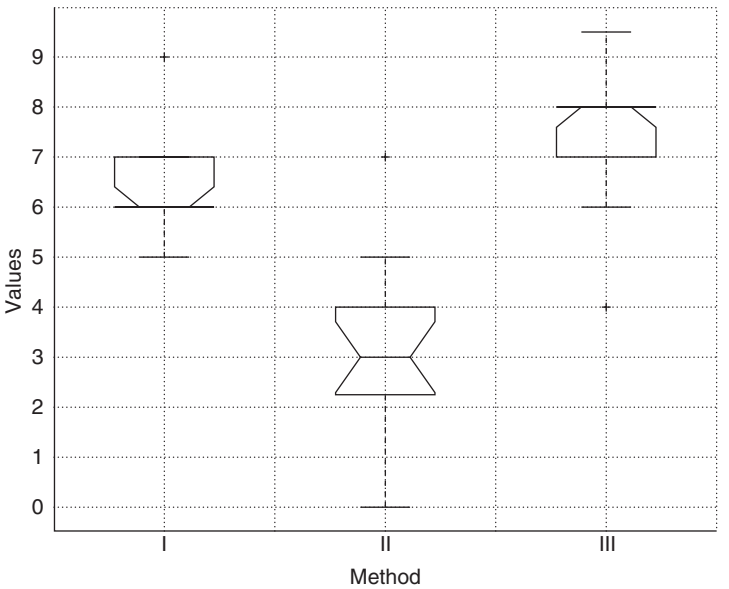
7 Time interval distributions: (a) histogram of large eye movement time intervals, and (b) accumulated covered percentage versus the time interval limit.

8 Some frames of synthesized eye motion.

In the second experiment (method II), we used random eye gaze and blink together with a Poisson distribution. In the third (method III), we used eye-blink and –gaze motion synthesized simultaneously using our method.

We presented three eye-motion videos with the same duration to 22 viewers in random order. We asked the viewers to rate each eye-motion video on a scale from 1 to 10, with 10 indicating a completely natural and realistic motion. Figure 9 illustrates the average score and standard deviation of this subjective test. As you can see from Figure 9, both method I and method III received much higher scores than method II. In fact, viewers slightly preferred the synthesized eye motion from our approach (method III) over that of method I.

We also conducted a second test to see whether it's possible to distinguish our synthesized eye motion from the original captured eye motion. In this test, we asked viewers to identify the original after they carefully watched two eye-motion videos, one being the originally captured eye motion and the other synthesized from this captured segment, both being viewed on the same computer graphic face model. Seven out of 15 made correct choices; the other eight subjects made wrong choices. Using the normal approximation to the binomial distribution with $p = 7/15$, we found that equality ($p = 0.5$) is easily within the 95 percent confidence interval. However, with this small number of subjects, the interval is too broad to conclude fully that the original and synthesized videos are truly indistinguishable.



9 One-way *analysis of variance* (Anova) results of our evaluations. The *p* value is 2.3977e-10.

Conclusions and future work

Texture synthesis techniques indeed can be applied in the animation realm to model incidental facial motion. While the synthesized results are difficult to distinguish from actual captured eye motion, a limitation of this approach is that it's difficult to know in advance how much data is needed to avoid getting stuck in the synthesis procedure. However, it's easy to evaluate the variety of synthesized movement after generating some animation, and we could always obtain more data if necessary.

Our approach works reasonably well for applications where nonspecific but natural-looking eye motions are required, such as for characters in games. We are aware that complex eye-gaze motions exist in many scenarios, especially communications among multiple agents. Realistic eye motion in these scenarios will require a combination of goal-directed gaze modeling and realistic motion quality. Such a combination is conceivable if the goal, for example, is to look away from the speaker to appear uninterested. A suitable gaze–blink signal of the required duration could be synthesized with our approach.

Realistic eye movement involves several phenomena not considered in this article, such as upper eyelid shape changes due to eyeball movement, skin deformation around the eyes due to muscle movement, and so forth. In future work, we plan to verify whether we can produce different moods (such as attentive, bored, nervous, and so forth) with our approach. To do this, we need to address head rotation (and especially rotation-compensated gaze) and vergence. We also plan to enhance the synthesis algorithm to deal with more complex scenarios by introducing constraints into the system. For example, we could generate high-level constraints (such as "look ahead for 15 seconds") with a goal-directed system in which a constrained texture synthesis approach would fill in the details of the eye motion. ■

Acknowledgments

This research was funded by the Integrated Media System Center at the University of Southern California, a National Science Foundation Engineering research center, under cooperative agreement No. EEC-9529152. We thank Jun-yong Noh and Doug Fidaleo for data capture; Hiroki Itokazu, Bret St. Clair, Pamela Fox, and Albin Cheenath for model preparation; and the anonymous reviewers for useful comments and suggestions. We also appreciate many valuable comments from other colleagues in the Computer Graphics and Immersive Technologies Laboratory at USC.

References

1. S.P. Lee, J.B. Badler, and N. Badler, "Eyes Alive," *ACM Trans. Graphics*, ACM Press, vol. 21, no. 3, 2002, pp. 637-644.
2. D.D. Salvucci and J.H. Goldberg, "Identifying Fixations and Saccades in Eye-Tracking Protocols," *Proc. Symp. Eye Tracking Research & Applications* (ETRA), ACM Press, Nov. 2000, pp. 71-78.
3. L. Liang et al., "Real-Time Texture Synthesis by Patch-Based Sampling," *ACM Trans. Graphics*, ACM Press, vol. 20, no. 3, 2001, pp. 127-150.

Zhigang Deng *is a PhD candidate in the Computer Graphics and Immersive Technologies Laboratory in the Computer Science Department at the University of Southern California. His research interests include computer graphics and computer facial animation. Deng received a BS in mathematics from Xiamen University and an MS in computer science from Peking University, China. Contact Deng at zdeng@graphics.usc.edu.*

J.P. Lewis *is a research associate in the animation group at the Computer Graphics and Immersive Technology Laboratory at the University of Southern California. His research interests include facial animation and computer vision topics. Lewis received an S.M.Vis.S. in Architecture from Massachusetts Institute of Technology. Contact Lewis at noisebrain@graphics.usc.edu.*

Ulrich Neumann *is the Charles Lee Powell Professor of Engineering, an associate professor of computer science, and director of the Integrated Media System Center at the University of Southern California. His research interests include immersive environments and virtual humans. Neumann received an MS in electrical engineering from SUNY Buffalo and a PhD in computer science from the University of North Carolina at Chapel Hill. Contact Neumann at uneumann@graphics.usc.edu.*