Predicting Moves-on-Stills for Comic Art using Viewer Gaze Data

Eakta Jain[1], Yaser Sheikh[2], and Jessica Hodgins[2,3]

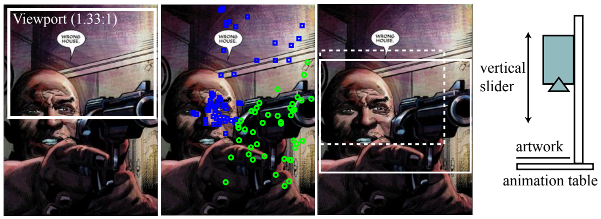
[1]University of Florida
[2]Carnegie Mellon University
[3]Disney Research Pittsburgh

***Abstract*—Comic art consists of a sequence of panels, each a different shape and size, that visually communicate the narrative to the reader. When comic books are retargeted to fixed size screens (e.g., handheld digital displays), the technique of moves-on-stills is employed to avoid scaling, cropping or otherwise distorting the artwork. Today, moves-on-stills are created by software where a user inputs the parameters of a desired camera move. We consider the problem of how the parameters for a good move-on-still may be predicted computationally. We observe that artists who work with visual media deliberately direct the flow of viewer attention across the scene. We present an algorithm to predict the move-on-still parameters on a given comic panel from the gaze data of multiple viewers looking at that panel. We demonstrate our algorithm on a variety of comic book panels, and evaluate its performance by comparing with a professional DVD.**

***Index Terms*—I.3.3 Picture/Image Generation, I.3.7.a Animation, L.1.0.a Attention**

I. Introduction

Integrating stills into motion pictures via camera moves is often known as the *Ken Burns* effect. This effect was traditionally used to engage the viewer through motion when photographs or paintings were shown on television or film. With the advent of personal handheld digital displays, presenting a still picture as a sequence of images seen through a viewport allowed for users to view pictures with varied aspect ratios on a fixed size screen (Figure 1). For comic book art in particular, moves-on-stills offer the advantage of engaging a strength of digital displays (the presented material can be animated) while maintaining the defining characteristic of sequential art (each panel is a moment frozen in time).



Comic book panel (1:1.37) Viewer gaze data (two clusters) Our result for move-on-still based on gaze data Rostrum camera

Fig. 1. Left: A move-on-still moves a viewport (e.g., 1.33:1) over original artwork (e.g., 1:1.37) to best display the art. Middle: Our method uses viewer gaze data as input. Fixation points are clustered based on their locations (green and blue). Right: The move-on-still predicted by our algorithm starts at the dotted rectangle and ends at the solid rectangle, revealing the mouth of the gun. This digital effect mimics the move-on-still traditionally created by a rostrum camera setup. Original comic panels ©MARVEL.

Moves-on-stills were created by a camera operator filming the still with a rostrum camera on an animation table. Today, software such as Adobe Premiere and iMovie allow anyone with a computer to create this effect digitally. While the mechanics have been made simpler, a central question remains unexplored: what makes for a good camera move on a still? For example, if the picture contains a man holding an object in his hand, the camera move should probably present the man's face, and the object in his hand, either moving from the face to the object, or, vice versa. Of these two moves, which one should be selected? Will Eisner famously wrote that (for comic art), "…the artist must…secure control of the reader's attention and dictate the sequence in which the reader will follow the narrative…" [1]. Therefore, a good camera move should respect this artistic intent.

However, generating a camera move that supports the narrative in the original still picture, is a complex process: it involves guessing the

A portion of this work was done while Eakta Jain was a graduate student at Carnegie Mellon University.

Manuscript received April 19, 2005; revised August 26, 2015.

artist's intent by recognizing objects such as 'man' and 'hand', understanding relationships such as 'holding', and figuring out how these relationships influence a camera move. Though computational algorithms are making strides towards some of the subtasks, how to create a camera move that supports the narrative is still not understood. Our key insight is that comic book artists make a conscious effort to direct the viewer's gaze, therefore, recording the gaze data of viewers should tell us the visual route that was used to discover (and conversely, may be used to present) the narrative in the picture. Figure 1 shows the viewer gaze data collected on a comic book panel, and our result.

The first challenge posed by this insight is that viewer eye movements are governed both by characteristics of the visual stimuli, and by individual-specific viewing strategies. As a result, no two viewers will have identical eye movements on a given visual stimulus. This inter-observer variability, and imperfect eyetracker calibration yield noisy measurements of the regions of interest in the scene being observed. The second challenge is that eye movements do not directly map to camera moves. The eye movements used to traverse still images are typically saccadic movements, i.e., rapid movements of the eye to center the fovea at a region of interest; fixations occur as the eye rests on the region of interest between saccades [2]. Repeated saccadic sequences (scanpaths) are involved when an observer visually processes an image. While eye movements occur to efficiently process visual information, camera moves are intended to present information. In the example of the man holding the object, if the camera were to move as rapidly as the eye does, from the face of the man to the object in his hand, and then back and forth, the resulting video would be jarring to look at.

Previous work has shown that inter-observer variability is reduced in comic art [3], likely as a result of expert direction by skilled artists. This finding suggests that viewer gaze could be used to drive moves-on-stills for comic art. We present an algorithm to predict the parameters of a camera move-on-still from recorded viewer gaze data. Gaze data is converted into fixations and saccades by a dispersion based method using proprietary software provided with the eyetracker. Our method extracts features from fixation locations, saccade directions, and dwell times to predict the parameters of a virtual rostrum camera move (Figure 1). We demonstrate results on a variety of comic books, and evaluate our algorithm by comparing the predicted parameters with the parameters of professionally created camera moves.

II. Related Work

Viewer eye movements are driven both by bottom-up cues, and top-down influences [2]. Artistic wisdom is that photographers, graphic artists, and film directors are able to influence viewers to look at certain regions versus others. Researchers have recognized this difference in the visual perception of artistic stimuli versus 'natural' stimuli. For example, Dorr and colleagues found that the characteristics of eye movements on static images are different from those on continuous videos, and the movement patterns in Hollywood movies are significantly more coherent than natural movies [4]. The work of Jain and colleagues [3] suggests that the artist is able to purposefully direct the visual attention of readers through the pictorial narrative. We are inspired by these findings to use viewer gaze on comic art as an input to predict camera move parameters.

Computer graphics researchers have previously recognized that gaze is a source of high-level information about the underlying scene. Researchers have used viewer gaze to simplify three-dimensional models [5], guide non-photorealistic rendering [6], and crop images [7]. Recently, Jain and colleagues proposed algorithms to retarget widescreen videos based on eyetracking data [8]. We further this past research by observing that, in addition to the attended locations, the order in which people attended to the locations, and the amount of time they spent looking is also informative of the underlying pictorial composition.

Previous work in computer graphics algorithms relating to comic art includes methods to summarize interactive game play as a sequence of comic images [9], convert a movie clip into a comic [10], and automatically generate manga layout [11]. In our work, we show that instead of a mouse based interface to move a camera or annotate the image, recording the eye movements of viewers implicitly provides the necessary information to create moves-on-stills for comic book images.

III. Predicting Moves-on-stills

We present an algorithm to predict the parameters of moves-on-stills on comic book images from the recorded viewer gaze data on those images. Moves-on-stills were traditionally filmed on a rostrum

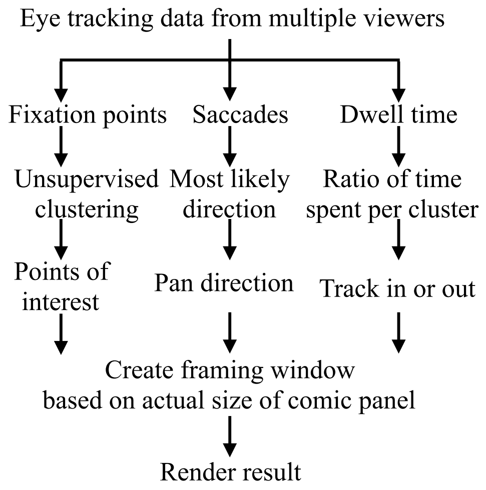


Fig. 2. An overview of our method.

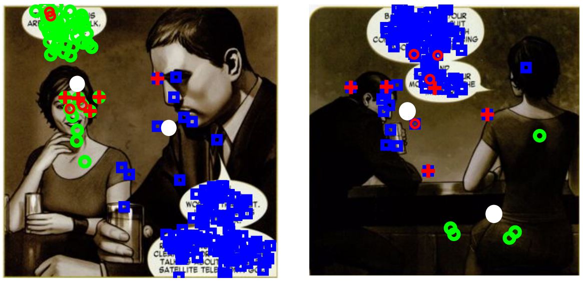


Fig. 3. The white dots mark the computed cluster centroid for each color-coded cluster. The red crosses and circles indicate the first and second fixations for each of the observers. Left: Noise in data capture leads to reading fixations that are outside the word bubble. Right: Reading and non-reading fixations can be quite close to each other, as in the case of the word bubble and the man's head. Original comic panels ©MARVEL.

camera (Figure 1). The artwork was placed so that the camera framed a point of interest. Then it was slowly moved while being filmed so that the camera viewport tracked the point of interest, or, panned from one point of interest to the next.

The degrees of freedom for a digital move are the same as those that were afforded by the physical setup: one degree of vertical movement, toward or away from the animation table, that yields a *track in* or *track out* shot, and two degrees of movement on the animation table, North-South and East-West, that result in a *pan* shot. Rotation of the artwork about the vertical axis is rarely used for comics, and thus, we focus on only the translations. In the digital setup, these degrees of freedom are characterized by the position of the center of the viewport (or framing window), and the dimensions of the viewport around the point of interest. Figure 2 provides an overview of our method. The eyetracking data used in our results was collected as part of previous work, and details of the data collection can be found in [3].

A. *Points of interest*

Points of interest are the semantically meaningful elements of the picture, which the artist used to tell the story, and which the viewport should frame. In the example image in Figure 3 (left), the points of interest are the two faces. It naturally follows that the locations of the recorded fixation points indicate regions of important semantic content. Fixations on word bubbles are not usually informative of the points of interest because word bubbles are placed to avoid occluding a pictorial element. Accordingly, we color segment the word bubbles, identify the fixations which lie inside the segmented region, and remove them. Unsupervised clustering is performed on the locations of the remaining fixation points, based on squared Euclidean distance.

We denote the ordered fixation locations for observer s on picture p by $\mathbf{X}^{ps}$. Then, $\mathbf{X}^{ps} = [\mathbf{X}_1^{ps}, \mathbf{X}_2^{ps}, \ldots, \mathbf{X}_{\gamma_{ps}}^{ps}]^T$, where $\mathbf{X}_l^{ps}$ is the l-th fixation location, and γ_{ps} is the total number of fixations on this picture. The value γ_{ps} could be different because two observers might have different reading speeds, or, because one of them fixated longer at each location, while the other jumped back and forth between the locations, for example. In addition, the same observer might attend to a different number of regions on different pictures, or, change her speed of perusal.

We perform unsupervised k-means clustering to associate each fixation location $\mathbf{X}_l^{ps} \in \mathcal{R}^2$ with its cluster via the label $z_l^{ps} = k, k = \{1, 2, \ldots, K\}$, where K is the total number of clusters and Ω_k is the set of all fixations belonging to the cluster k. Then, the point of interest μ_k is the centroid of all the fixation locations belonging to a cluster Ω_k,

$$\mu_k = \frac{1}{|\Omega_k|} \sum_{s=1}^{S} \sum_{l=1}^{\gamma_{ps}} \mathbf{X}_l^{ps} \delta(z_l^{ps} - k), \qquad (1)$$

where δ refers to the Kronecker delta function. The dwell time τ_k on a point of interest is the total duration of all fixations in its cluster. Let t_l^{ps} be the duration of each fixation. Then,

$$\tau_k = \sum_{s=1}^{S} \sum_{l=1}^{\gamma_{ps}} t_l^{ps} \delta(z_l^{ps} - k). \qquad (2)$$

In practice, eyetracking data is noisy and it is not straightforward to identify the word bubble fixations. These fixations may lie outside the colored word bubbles because of calibration error. For example, the green colored fixation points in Figure 3 (left), lie slightly outside the word bubble, and are a source of error in the computed centroid (white circle). Sometimes though, the fixation points right outside the word bubbles might not be 'reading fixations', as in Figure 3 (right), where the fixations on the man's face are about as far away from the word bubble as the erroneous data. Thus, simple heuristics, such as, dilating the color segmented word bubble, or, identifying reading fixations by the typical saccade length between them, are not guaranteed to always work, and we opt to let the clustering method take care of such errors.

Because finding the number of meaningful clusters automatically is not a well-defined problem, we ascertained the number of clusters K based on detailed observations of the professionally produced *Watchmen* motion comic DVD from DC Comics. Two independent coders coded the camera moves for panels from four pages in the first chapter (31 panels). On average 77% of the camera moves were marked as camera pans between two points of interest. The coders marked a single-point move when the camera tracked in or out but did not pan significantly. They marked a three or more point move when the scene was not a still picture, for example, the objects in the panel were animated and the camera followed them. For moves on stills, two points of interest are largely sufficient. Therefore, we specified $K = 2$ for all our examples.

B. Pan across the image plane

The panning shot is generated by moving the viewport from one point of interest to the other. Let the direction of the pan be represented by the indicator variable $\Theta \in \{0, 1\}$,

$$\Theta = \begin{cases} 0 & \text{if} \quad \mu_0 \rightarrow \mu_1, \\ 1 & \text{if} \quad \mu_1 \rightarrow \mu_0, \end{cases} \tag{3}$$

where $\mu_0 \rightarrow \mu_1$ denotes that the start point of interest is μ_0 (the centroid of the first cluster) and the end point of interest is μ_1 (the centroid of the second cluster), and $\mu_1 \rightarrow \mu_0$ denotes the opposite direction. This direction is predicted based on the onset cluster ϕ, and the dominant direction ξ.

The onset cluster is the region that a new observer will likely look at when they first look at the given still image, that is, it is the cluster that contains the onset fixations of the observers. The onset cluster $\phi \in \{0, 1\}$ takes the value $\phi = 0$ if the first and second fixations fall in the first cluster, arbitrarily denoted cluster 1, and takes the value $\phi = 1$ if the first and second fixations fall in the second cluster, denoted cluster 2. The dominant direction is the direction in which an observer is likely to peruse the image, that is, the more likely saccade direction. The direction indicator $\xi \in \{0, 1\}$ takes the value $\xi = 0$ if the saccade direction is more often from cluster 1 to cluster 2 and $\xi = 0$ otherwise. Because people often look back and forth between two clusters, when the number of saccades from cluster 1 to cluster 2 is equal to the number of saccades from cluster 2 to cluster 1, the onset cluster is the tie-breaker.

The cues are integrated into a likelihood L that the pan will follow the direction represented by Θ,

$$L(\Theta = i) = p(\phi = i)p(\xi = i). \tag{4}$$

We denote the number of first and second fixations in a cluster by $n(\phi = i)$. Then, $p(\phi = i)$ is computed as

$$n(\phi = i) = \sum_{s=1}^{S} \delta(z_1^{ps} - i) + \sum_{s=1}^{S} \delta(z_2^{ps} - i) \tag{5}$$

$$p(\phi = i) = \frac{n(\phi = i)}{\sum_{j=1}^{K} n(\phi = j)}. \tag{6}$$

Both the first and second fixations are included while computing ϕ because the first fixation may not be captured reliably, and may be subject to center bias, that is, the tendency for observers to fixate in the center of the screen when the visual stimulus switches.

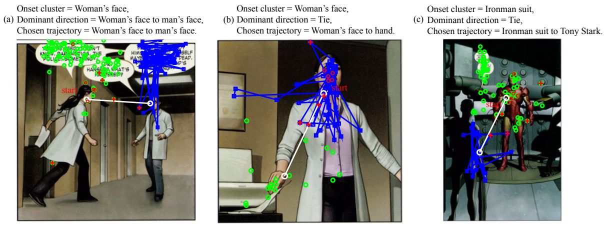
The probability that the dominant direction of eye movement is $\xi = 0$ is proportional to the number of times people look from the cluster $k_1 = 1$ to the cluster $k_2 = 2$, and vice versa for $\xi = 1$. Let us denote this count by $n(\xi = i)$. Then,

$$n(\xi = i) = \sum_{s=1}^{S} \sum_{l=1}^{\gamma_{ps}-1} \delta(z_l^{ps} - k_1)\delta(z_{l+1}^{ps} - k_2) \tag{7}$$

$$p(\xi = i) = \frac{n(\xi = i)}{\sum_{j=0}^{1} n(\xi = j)}. \tag{8}$$

We estimate the direction of the pan across the image as

$$\Theta_{MLE} = \arg\max_{\Theta} \quad L(\Theta). \tag{9}$$


(a) Onset cluster = Woman's face, Dominant direction = Woman's face to man's face, Chosen trajectory = Woman's face to man's face.

(b) Onset cluster = Woman's face, Dominant direction = Tie, Chosen trajectory = Woman's face to hand.

(c) Onset cluster = Ironman suit, Dominant direction = Tie, Chosen trajectory = Ironman suit to Tony Stark.

Fig. 4. Fixation data overlaid on example panels. The green and blue markers are for the different clusters. The red crosses indicate the first fixation for an observer and the red circles indicate the second fixation. The start cluster is marked with a red label. Original comic panels ©MARVEL.

The examples in Figure 4 illustrate the various possibilities for the two cues. In panel (a), ϕ and ξ both vote for cluster 2 as the starting cluster. In panels (b) and (c), there is no clear dominant direction (the observers look back and forth between the clusters) and the onset cluster ϕ is the tie-breaker for the estimated direction of pan. In panel (d), the onset cluster ϕ is the man's head, while the dominant direction ξ is bottom to top. The onset cluster is the man's head because the first fixations were mostly at the bottom part of the screen, but the second fixations were on the man's head. However, the observers all finished looking at the image from bottom to top, leading to a strong dominant direction. The chosen direction for the pan is bottom to top for this panel.

C. *Track in or out*

The track in or out shot refers to the movement of the camera towards or away from a subject "...to increase or decrease its importance within the context of a story..." [12]. For a move-on-still, the camera should track in if the end point is more important than the start point and track out if the start point is more important. Because people dwell longer on regions that are more interesting or important to them, the dwell time τ_i on a point of interest μ_i is a natural cue to drive the track shot.

Let μ_i be the start point of interest and μ_j be the end point of interest $(i, j \in \{0, 1\})$. Then, the tracking shot is a track in if $\tau_i < \tau_j$ and track out if $\tau_i > \tau_j$. This shot is generated digitally by increasing, or, decreasing the size of the framing window based on the ratio of the dwell times. For a track in, the size of the start window is the largest window of the desired aspect ratio that can fit inside the panel, and the size of the end window is made smaller by scaling with the scaling factor λ_j,

$$\lambda_j = c + (1 - c)\frac{\tau_i}{\tau_j} \qquad \text{where} \quad \tau_i \leq \tau_j. \qquad (10)$$

For a track out, the size of the end window is the largest window of the desired aspect ratio that can fit inside the panel and the size of the start window is smaller by the scaling factor λ_i,

$$\lambda_i = c + (1 - c)\frac{\tau_j}{\tau_i} \qquad \text{where} \quad \tau_i > \tau_j. \qquad (11)$$

The parameter $c = 0.75$ for the *Watchmen* panels, and $c = 0.5$ for all other comic panels. This parameter allows the user to control how sharply the camera will track in or out. For example, a higher value of c would mean that there is not much difference in the sizes of the start and end window, even if the cluster dwell times are unequal. A lower value of c would result in a larger difference in window sizes, to create a more dramatic effect, for example. In our examples, we chose the values of c visually, based on how dramatic a track was comfortable to watch and the typical sizes of faces in each set. The final dimensions $\mathbf{D}_i$ and $\mathbf{D}_j$ of the framing windows are computed as

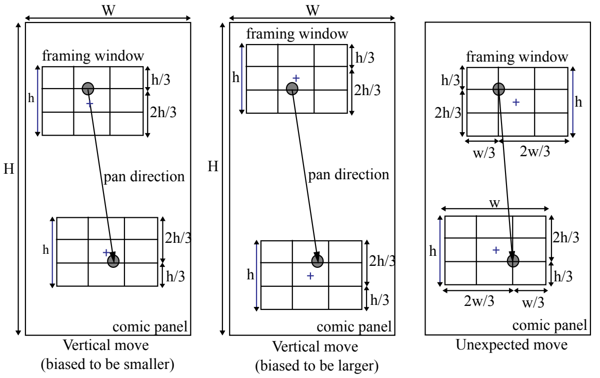
$$\begin{aligned} \mathbf{D}_i &= \lambda_i \mathbf{D}, & (12) \\ \mathbf{D}_j &= \lambda_j \mathbf{D}, & (13) \end{aligned}$$

where $\mathbf{D}$ is the render size, e.g., $\mathbf{D} = [640, 480]^T$.

The computation thus far fixes the direction the camera will move in a plane parallel to the still image, and whether it will move into or away from the still image. The final step in rendering the move-on-still is to compute the trajectory of the center of the framing window.

D. *Rendering the move-on-still*

Because gaze data contains semantic information, we can identify the points of interest in a still picture, the order in which they should be traversed, and their relative importance for the viewer. Gaze data does not, however, provide any information about how the rostrum camera operator would frame these points of interest within the viewport (or framing window). We use the well-known photography rule of thirds to resolve this ambiguity. According to this rule, the framing window should be divided into a 3×3 grid with vertical and horizontal guides; the point of interest should lie on either a vertical or horizontal guide for a visually pleasing effect (see Figure 5).



Vertical move (biased to be smaller)

Vertical move (biased to be larger)

Unexpected move

Fig. 5. Left and middle: The center of the frame is computed so that the point of interest is framed according to the rule of thirds, and such that the point of interest has the same relative spatial location inside the framing window as it does in the panel. Right: We could use scripted rules to place the framing window, for example, the start point of interest lies on the left intersection of the guides and the end point lies on the right intersection of the guides. However, this type of scripted rule can result in the effect is of a camera moving in the opposite to desired direction because of the selection of guide points.

Our method classifies the predicted move-on-still as a North-South move if the pan direction is between 15° and 165° and as a South-North move if the pan direction is between $-15°$, and $-165°$ (similarly for East-West, and West-East). For a North-South move (Figure 5, left), the x position of the framing window is computed such that the ratio of the distances of the point of interest to the boundaries is the same in the framing window as in the original panel. Let the start window be parametrized by the top-left corner (x_1, y_1) and the bottom-right corner (x_2, y_2). Its size $\mathbf{D_i}$ was computed in Equation 13. Recall that the position of the point of interest is $(\mu_1(1), \mu_1(2))$. Then,

$$x_1 = \mu_1(1) - \frac{\mu_1(1)}{W}\mathbf{D_i}(1), \qquad (14)$$

$$x_2 = \mu_1(1) + \frac{W - \mu_1(1)}{W}\mathbf{D_i}(1). \qquad (15)$$

The y position of the framing window is computed such that the point of interest lies on the horizontal third guides. We experimented with two variants: first, using the upper third guide for the start window and lower third guide for the end window (a smaller camera move, as in Figure 5 (left)), and second, using the lower third guide for the start window and the upper third guide for the end window (a larger camera move, as in Figure 5 (right)). For the first case,

$$y_1 = \mu_1(2) - \frac{1}{3}\mathbf{D_i}(2), \qquad (16)$$

$$y_2 = \mu_1(2) - \frac{2}{3}\mathbf{D_i}(2). \qquad (17)$$

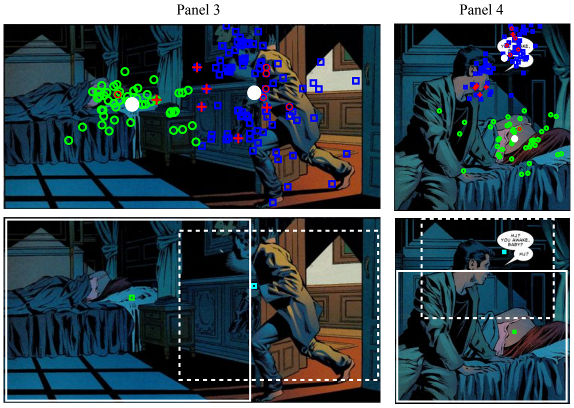


Fig. 6. Two panels from a *Spiderman* comic book. Original comic panels ©MARVEL. Top: Fixation locations (green circles and violet squares), computed points of interest (white dots). Bottom: In the camera move predicted by our algorithm, the dotted rectangle is the start window, the outlined rectangle is the end window.

The equations for the second case are similar. We found the larger camera move to be the more pleasing option as it displayed a greater amount of the underlying comic art. In Section V, we also numerically compare the two cases with a professional DVD.

Figure 5 (right) shows a possible failure case if we rely only on scripted rules based on the direction of movement and the horizontal and vertical guides (frame the start point of interest on the upper left guide intersection and the end point of interest on the lower right guide intersection). The start point of interest is to the left of the end point of interest, but placing them on the guides leads to an opposite camera move. To avoid such cases, we use the relative spatial location of the point of interest in the still picture to place the framing window.

IV. Results

We present moves-on stills-for panels taken from a variety of comic books: *Watchmen*, *Iron-man: Extremis*, *The Three Musketeers*, *Hulk:WWH*, *Black Panther*, and *The Amazing Spiderman: New Avengers*. Our results are best seen as video, and are presented as part of the accompanying movie. We show a subset of the complete set of results as figures in this section. The fixation locations are either green circles, or violet squares based on their cluster. The red markers mark the first and second fixation points, which are used to compute the onset cluster probability for the pan shot. The large white dot markers are the computed points of interest. There is only one user-defined parameter in our

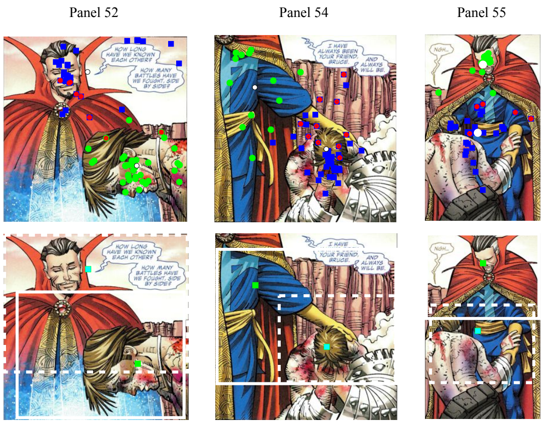

Panel 52
Panel 54
Panel 55
HOW LONG HAVE WE KNOWN EACH OTHER?
HOW MANY BATTLES HAVE WE FOUGHT, SIDE BY SIDE?
I HAVE ALWAYS BEEN YOUR FRIEND, BRUCE.
AND I ALWAYS WILL BE.
NGH...

Fig. 7. Three panels from a *World War Hulk* comic book. Original comic panels ©MARVEL. Top: Fixation locations (green circles and violet squares), computed points of interest (white dots). Bottom: In the predicted camera move, the dotted rectangle is the start window, the outlined rectangle is the end window.

method. The parameter $c = 0.75$ for the *Watchmen* panels, and $c = 0.5$ for all other comic panels.

Figure 6 shows two example panels from the *Spiderman* comic book. In Panel 3, the points of interest are the sneaking man and the sleeping woman, and in Panel 4, the points of interest are the face of the man and the sleeping woman. These panels demonstrate why gaze is a valuable cue—relying on a face detector or similar non-contextual cues would likely have missed the sleeping lady as a point of interest, yet we can see that the artist intended for the viewer to look at her. For Panel 3, the camera panned from right to left and in Panel 4, the camera pans from top to bottom.

Figure 7 shows three panels from *Hulk: WWH*. Our algorithm identifies the points of interest as the kneeling man and the cloaked man, and predicts a downward pan, a track with a pan, and an upward pan respectively. Sample frames from the result are shown in Figure 8. Figure 9 shows example panels from the *Ironman* graphic novel. In Panel 3 and Panel 7, the points of interest are clearly the face and hand, and the suit and face, respectively. In Panel 31, the starting point of interest is shifted slightly to the right of the woman's face because of stray fixation points. The camera tracks out for Panels 3 and 7 to reveal the paper in the woman's hand and the man looking at the suit. The camera tracks in towards the man's face for Panel 31, correctly capturing the semantics of the scene.

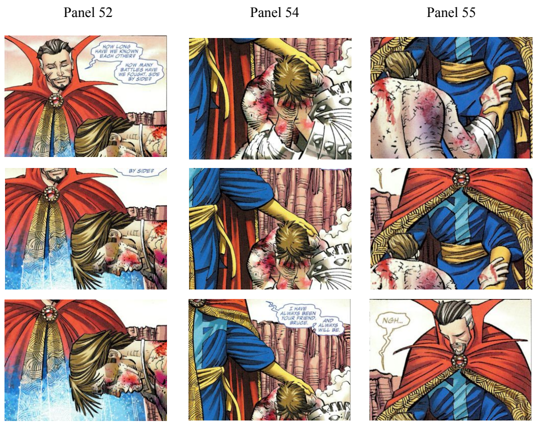


Fig. 8. Three panels from a *World War Hulk* comic book. Original comic panels ©MARVEL. Sample frames from the result movie.

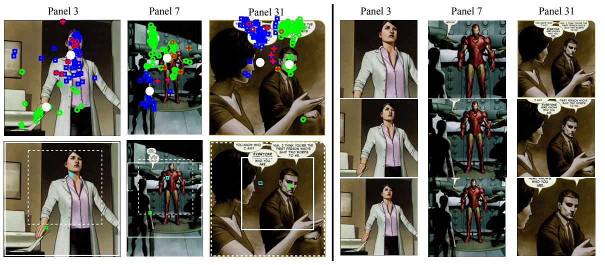


Fig. 9. Three panels from *Ironman*. Original comic panels ©MARVEL. Top: Fixation locations and the computed points of interest. Bottom: Camera move predicted by our algorithm. The dotted rectangle is the start window, the outlined rectangle is the end window.

V. Evaluation

We evaluated the camera moves predicted by our algorithm by comparing with the camera moves created by a professional artist for the *Watchmen* motion comic DVD by DC Comics. Two independent coders watched an excerpt from the DVD and marked the center of the framing window on the corresponding comic panel. The coders coded four comic book pages (31 panels). Three panels were discarded because a coder made a mistake, or because the panel was not included in the DVD.

Amongst the remaining 28 panels, coder 1 marked 19 panels with two-point moves and coder 2 marked 24 panels with two-point moves (that is, the camera panned linearly between a start and an end point). Because our algorithm generates two-point moves, we numerically evaluate panels coded with multiple-point moves by considering only the largest segment of camera movement. The average root mean squared distance between the

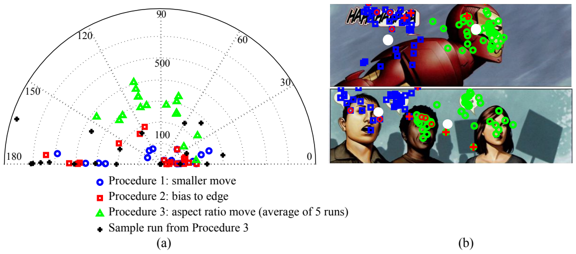


Fig. 10. Evaluation and limitations. (a) Evaluation of our method based on ground truth obtained from a professional DVD. We show two versions of our method (red squares and blue circles), along with a naive procedural camera move, shown as green triangles. Points closer to the origin and the 0° line represent a good match with the DVD. (b) Top: because people fixate on the back of the head as well as the face, the cluster center shifts towards the ear. Bottom: though the camera move may be a two-point move, two clusters may not accurately extract the points of interest. Original comic panels ©MARVEL.

window centers marked by the two coders was 104.8 (in pixels), which is approximately 10% of the largest dimension of a panel. We filtered out the cases where the coders were not consistent by discarding the panels where the distance between their annotations was greater than 10% of the panel dimension (105 pixels). Thus, we have reliable ground truth for 20 panels.

Figure 10 (a) is a polar plot where the radial distance is the root mean squared distance between the window center predicted by our method and the window center for a professional DVD, averaged over both coders, and the angular value is the included angle between the direction predicted by our method and the professional DVD. Each marker represents one of the evaluated panels. The blue circles mark the version of our algorithm which is biased towards smaller camera moves, while the red squares show the version of our algorithm which is biased towards showing more of the underlying artwork (consequently, a larger camera move). The green triangle markers are a procedural camera move: the camera moves vertically or horizontally based on the aspect ratio of the input comic panel, the direction (whether up or down, left or right) depends on a coin toss. The plotted points were obtained by averaging five runs. An example run is shown with black plus signs.

Points closer to the origin and the 0° line represent good performance because the predicted camera parameters match the DVD both in the location of the framing windows and the direction of the move. Points on the 180° line show the panels where our algorithm predicted a pan direction opposite to the DVD. Our gaze-driven method predicts the same direction as the DVD in 12 of the 20 test panels. In 2 panels, we predict a pan, whereas the coders annotate the camera move to be an in-place track. For 6 panels, we predict a direction opposite to the DVD. This might occur because people tend to look at faces first; if there is no clear dominant direction, the onset cluster is responsible for the direction of the camera move.

To better understand how well we were able to identify the points of interest (independent of direction), we computed the mean distance by ignoring the order of the start and end centers. Procedure 2 (our method, biased towards showing more of the comic panel) and procedure 1 (our method, smaller move) are equivalent in terms of center distance, averaged over the test panels (115.8 and 115.6), and are closer to the DVD than the aspect-ratio-based move (average distance to DVD = 180). The advantage of our method is especially clear in the panels where the method is able to correctly identify that the points of interest are not located near the top or the bottom of the panel.

VI. Discussion

Camera moves are the product of sophisticated creative thought; higher order cognitive processes are involved in recognizing visual elements, inferring their relationship, and figuring out a camera move that supports the narrative in the still picture. Though the cognitive processes are not well-understood, a behavioral cue is available to be recorded: gaze. Our key insight is that recorded gaze data contains the information needed to predict plausible moves-on-stills for comic art pictures (given that the comic book artist was successful in directing the flow of viewer attention). We demonstrated an algorithm to extract the parameters of a two-point camera move from the gaze data of viewers. We evaluated our algorithm by comparing the predicted parameters with the parameters of professionally created moves-on-stills.

A limitation of our method is that although human fixations are clustered around objects of interest in the image, the computed centroid is a partial representation of this object. For example, in Figure 10(b)(top), people fixate on the back of the head as well as the face, and the k-means algorithm computes the cluster centroid to be closer to the ear than the front of the face. To fully describe the object of interest, we would need to compute a bounding box in addition to the centroid, to ensure that the camera frames the entire face. Our current implementation also assumes that there are (for the most part) two points of interest in a comic book panel, and thus, it is sufficient to extract two clusters from the recorded viewer fixation points. We observed that even though the camera move may be well specified as a two-point move, the eye movements may not cluster well into two clusters, as in Figure 10(b)(bottom). An algorithm that continues to pan till it has enclosed all the fixated locations might capture this scene better. Saccade amplitudes and orientations could also be used to refine the computed clusters.

We create the track in or out shot from the ratio of the total fixation durations on the associated clusters based on the principle that people tend to fixate longer on the region that is more interesting to them; they want to look at this region more closely, therefore a camera should track in towards it. Other cues to predict the size of the window could include metrics that describe the spread of the fixation locations in the associated cluster, such as the distance between the two farthest points in a cluster, or, the standard deviation of the distribution of fixation locations. However, these metrics could be noisy when the fixation points in a cluster are not well modeled as a Gaussian (for example, the green cluster in Figure 4(b)). We note that the ambient-focal theory could explain how fixation duration, and spread are related. According to this theory, there are two modes of visual processing: ambient processing, which is characterized by larger saccade amplitudes and shorter fixation durations (perhaps to explore the spatial arrangement of the scene), and focal processing, which is characterized by longer fixation durations and shorter saccades (probably to gather detailed information about a particular object) [13], [14], [15]. Thus, we expect that longer total fixation duration on a cluster would likely be associated with a smaller spread in the fixation locations.

Finally, it may happen that camera moves created by a cinematographer, who knows where the story is going, will contain intentional reveals. Because our algorithm uses information from gaze data on only the current panel (and the viewer does not know what is going to happen next), the predicted moves will contain only as much surprising revelation as the comic book artist was able to achieve through manipulating viewer gaze.

References

[1] W. Eisner, *Comics and Sequential Art*. W.W. Norton & Company, 2008.

[2] S. E. Palmer, *Vision Science, Photons to Phenomenonlogy*. The MIT Press, 1999.

[3] E. Jain, Y. Sheikh, and J. Hodgins, "Inferring artistic intention in comic art through viewer gaze," in *ACM Symposium on Applied Perception (SAP)*, 2012.

[4] M. Dorr, T. Martinetz, K. Gegenfurtner, and E. Barth, "Variability of eye movements when viewing dynamic natural scenes," *Journal of Vision*, vol. 10, 2010.

[5] S. Howlett, J. Hamill, and C. O'Sullivan, "Predicting and evaluating saliency for simplified polygonal models," *ACM Transactions on Applied Perception (TAP)*, vol. 2, 2005.

[6] D. DeCarlo and A. Santella, "Stylization and abstraction of photographs," *ACM Transactions on Graphics (TOG)*, vol. 21, no. 3, 2002.

[7] A. Santella, M. Agrawala, D. DeCarlo, D. Salesin, and M. Cohen, "Gaze-based interaction for semi-automatic photo cropping," in *ACM Conference on Human Factors in Computing Systems (CHI)*, 2006.

[8] E. Jain, Y. Sheikh, A. Shamir, and J. Hodgins, "Gaze-driven video re-editing," *ACM Transactions on Graphics (TOG)*, 2015.

[9] A. Shamir, M. Rubinstein, and T. Levinboim, "Generating comics from 3d interactive computer graphics," *IEEE Computer Graphics & Applications*, vol. 26, no. 3, 2006.

[10] M. Wang, R. Hong, X.-T. Yuan, S. Yan, and T.-S. Chua, "Movie2comics: Towards a lively video content presentation," vol. 14, no. 3, 2012.

[11] Y. Cao, R. W. Lau, and A. B. Chan, "Look over here: Attention-directing composition of Manga elements," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, 2014.

[12] S. D. Katz, *Film directing, shot by shot, visualizing from concept to screen*. Michael Wiese Productions, 1991.

[13] J. R. Antes, "The time course of picture viewing." *Journal of Experimental Psychology*, vol. 103, no. 1, 1974.

[14] P. J. A. Unema, S. Pannasch, M. Joos, and B. M. Velichkovsky, "Time course of information processing during scene perception: The relationship between saccade amplitude and fixation duration," *Visual Cognition*, vol. 12, 2005.

[15] B. Follet, O. Le Meur, and T. Baccino, "New insights into ambient and focal visual fixations using an automatic classification algorithm," *i-Perception*, vol. 2, no. 6, 2011.

Eakta Jain is an Assistant Professor of Computer and Information Science and Engineering at the University of Florida. She received her PhD and MS degrees from Carnegie Mellon University, and B.Tech. from IIT Kanpur. Her research interests are perceptually-based computer graphics algorithms to create and manipulate artistic content, including traditional hand animation, comic art, and films.

Yaser Sheikh is an Associate Professor at the Robotics Institute, Carnegie Mellon University. His research interests span computer vision, computer graphics, and robotics. He has won Popular Sciences Best of Whats New Award, the Honda Initiation Award (2010), and the Hillman Fellowship for Excellence in Computer Science Research (2004). He received his PhD in 2006 from the University of Central Florida, and his BS degree from the Ghulam Ishaq Institute of Engineering Science and Technology in 2001.

Jessica Hodgins is a Professor in the Robotics Institute and Computer Science Department at Carnegie Mellon University, and Vice President of Research, Disney Research. She received her Ph.D. in Computer Science from Carnegie Mellon University in 1989. Her research focuses on computer graphics, animation, and robotics with an emphasis on generating and analyzing human motion. She has received a NSF Young Investigator Award, a Packard Fellowship, and a Sloan Fellowship. In 2010, she was awarded the ACM SIGGRAPH Computer Graphics Achievement Award.