Supplementary Material

1 Detecting Motion Lines

To detect motion lines in a panel, we use a heuristic strategy as follows. First, we detect straight lines (line segments that are longer than 20 pixels) in the panel using Hough transform. Second, we compute intersections for each pair of the detected lines, and group the intersection points into 3 clusters using the K-mean algorithm. We refer to each cluster center as a *convergence point*. Finally, by analyzing the cluster with the largest number of members C, we determine the presence and the type of motion lines. Specifically, if $|C|$ is greater than a given number (20 in our implementation), we compute how far the convergence point is from the panel center. If it is far enough, the panel is considered to contain parallel motion lines. Otherwise, converging motion lines are present. On the other hand, if $|C|$ is smaller than the given number, we check the total number of detected lines. If the total number of detected lines is smaller than a threshold (30 in our implementation), motion lines are considered as absent; otherwise, parallel motion lines are considered to be present.

2 Fitting Key Window to ROI

We seek to fit a key window F to a region of interest (ROI) whose center is point of interest (POI). Let $\mathbf{x}_F = (x, y), w$ be the position and width of the key window. We formulate the problem as an instance of the unconstrained optimization over $\mathbf{X} = \{x, y, w\}$, with the objective function:

$$E = w_1 E_{\text{dist}} + w_2 E_{F,ROI} + w_3 E_{F,P} + w_4 E_{A_F}, \tag{1}$$

where:

E_{dist}: encourages the key window center $\mathbf{x}_F$ to coincide with the POI, and is defined as $E_{\text{dist}} = \|\mathbf{x}_F - \mathbf{x}_{POI}\|^2$, where $\mathbf{x}_{poi}$ is the location of the POI.

$E_{F,ROI}$: measures the amount of overlap between the key window F and the ROI, and is defined as $E_{F,ROI} = 1 - \frac{\mathcal{A}(F \cap ROI)}{\mathcal{A}(ROI)}$, where $\mathcal{A}(F \cap ROI)$ is the area of intersection between the key window and the ROI, and $\mathcal{A}(ROI)$ is the area of the ROI.

$E_{F,P}$: measures the fraction of the key window F that lies within the panel P and is defined as $E_{F,P} = 1 - \frac{\mathcal{A}(F \cap P)}{\mathcal{A}(F)}$

E_{A_F}: measures the size of the key window F relative to its panel P, and is defined as $E_{A_F} = \frac{\mathcal{A}(F)}{\mathcal{A}(P)}$.

$\{w_i\}$ are weights controlling the contributions of the terms. We empirically set $w_1 = 0.2$, $w_2 = 0.2$, $w_3 = 0.4$, $w_4 = 0.2$.

The key window that best fits the ROI can be obtained by minimizing the objective function in Equation 1, i.e., $\hat{\mathbf{X}} = \arg\max_{\mathbf{X}} E(\mathbf{X})$. Because of the non-linear and non-convex nature of our objective function, we minimize it using simulated annealing, which can efficiently explore the solution space and avoid getting stuck with the local optimum. Since $\mathbf{X}$ is low-dimensional, the optimization can converge after only a few iterations.

3 Principles for Guiding Camera Movements

The principles used to determine camera movements are visualized in a binary decision tree as shown in Figure 1.

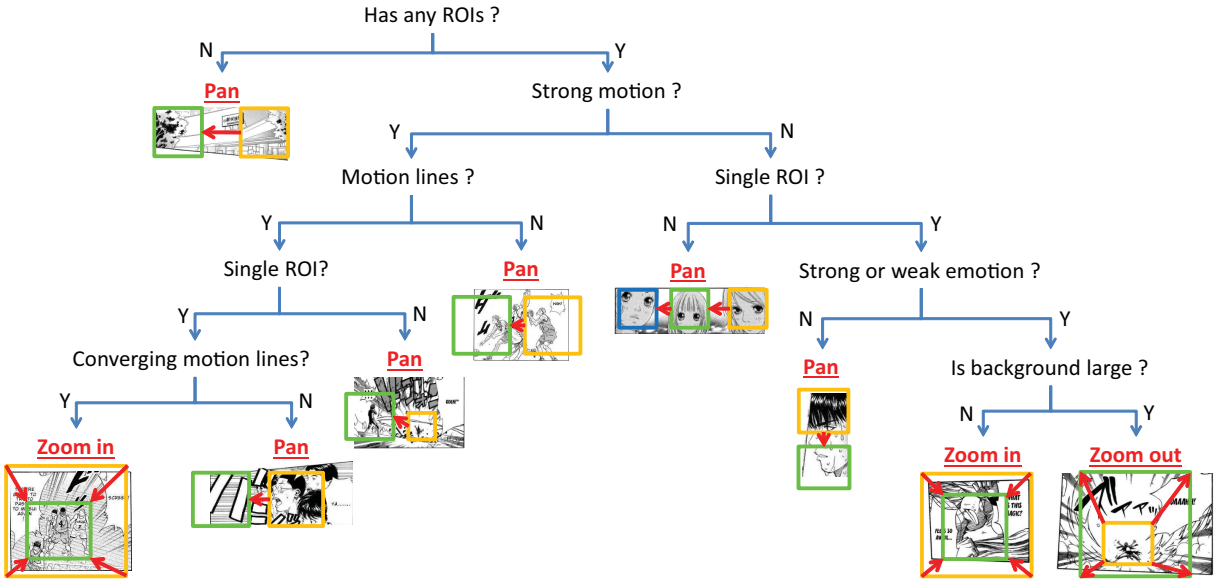


Figure 1: A binary decision tree used to determine camera movements.

4 Implementation Details of Special Effects

Motion Blur. Let $\mathbf{c}$ be the center of a foreground character. The motion blur is achieved by applying a zoom radial blur filter with the center at $\mathbf{c}$. To keep the details of the foreground character, we only blur the regions outside of the character's bounding circle R. An explosive effect around R can also be created by perturbing R's boundary using a sinusoid function. Let $\mathbf{p}_i$ be a point on R's boundary. It is perturbed as $\mathbf{p}_i + \frac{\mathbf{p}_i - \mathbf{c}}{\|\mathbf{p}_i - \mathbf{c}\|} w \sin a\theta$, where $w = 20$ and $a = 30$ controls the magnitude and the frequency of jittering, and θ is the orientation of the vector from $\mathbf{c}$ to $\mathbf{p}_i$.

Moving Motion Lines. To create the effect of moving motion lines, the user first needs to segment a panel into foreground and background layers. A cloud layer is then inserted between the foreground and background layers. To create the cloud layer, we randomly add white blobs to a transparent image and apply Gaussian blur. Finally, the cloud layer is moved in the direction from the panel center to the character center.

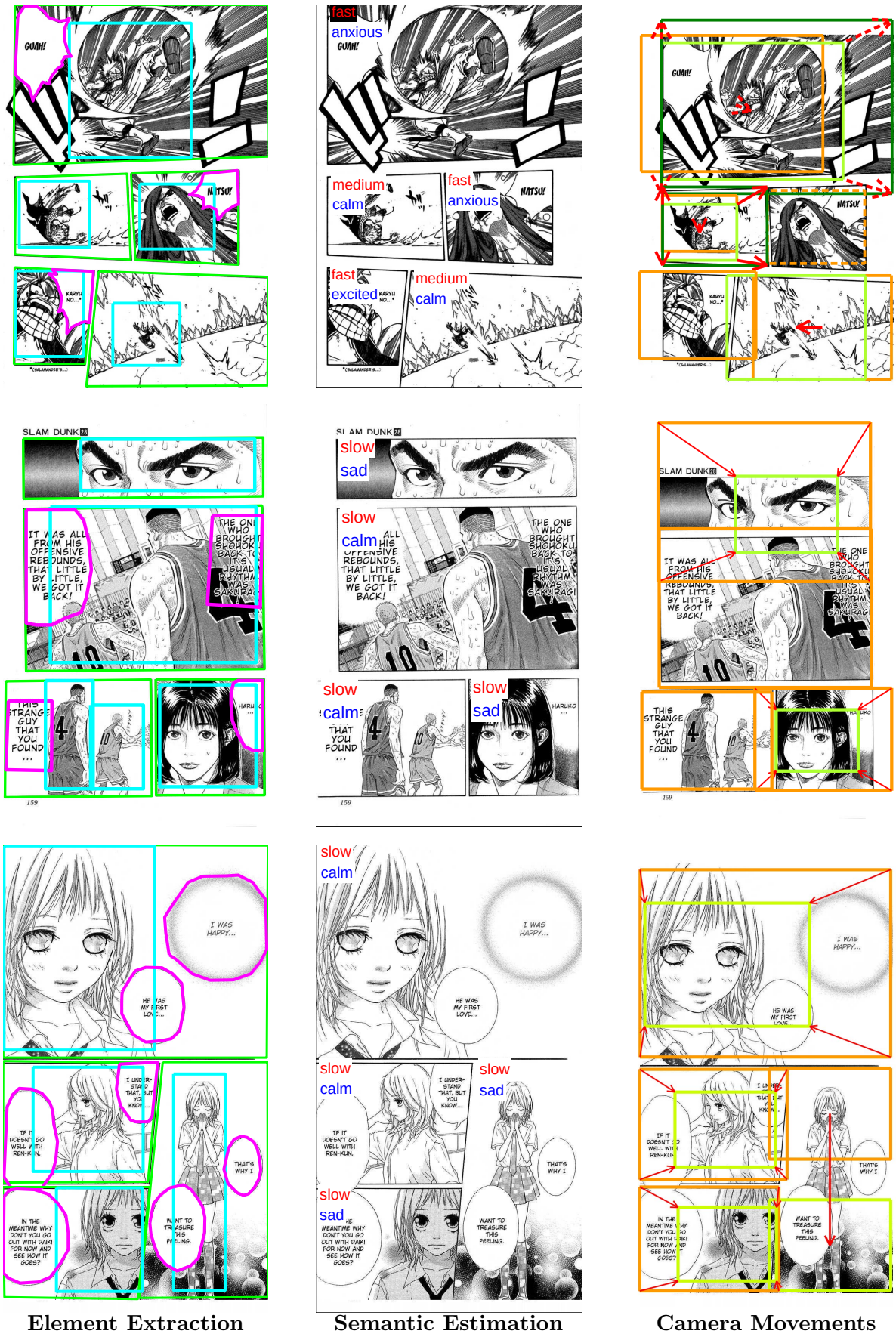
5 Additional Results by the Proposed Method

Additional results by our method are shown in Figure 2.

6 User Comments in Mobile Comic Reading Evaluation

We include some example comments left by the users when comparing our mannually-advanced dynamic manga (MDM), automatically-advanced dynamic manga (ADM), and traditional pages (TP) in the user study in Section 7.3.1.

When asked to explain why they like MDM or ADM as compared to other alternatives, some participants responded:



Element Extraction **Semantic Estimation** **Camera Movements**

Figure 2: Results on three manga pages. Manga pages at the first and second rows are from "*Fairy Tail*" and "*Slam Dunk*", and the one at the last row is from "*Strobe Edge*". The color of the windows on the right column indicates the temporal order (orange = $1st$, green-yellow = $2nd$, green = $3rd$).

- "The animations make contents look more dynamic."
- "Motion with varying speed helps me better sense what the characters are feeling and how they are moving."
- "I would choose MDM as my comic reader, mainly because the added motions well reveal the pace of the story, which helps me better immerse into the comic story."
- "I do not need to frequently move my fingers on the mobile phone any more."
- "Automatic zooming in and panning to proper contents is one of the useful features I really like. It saves me a lot of efforts."
- "The ability of moving to different contents inside a panel by simple tapping is impressive, minimizing my interactions with the phone during comic reading."

When asked to explain whey they prefer MDM over ADM, some participants commented:

- "I like advancing the story interactively so that I have control on my own reading speed."
- "Moving forward by just tapping the screen offers me with adequate control over my own reading pace with little effort."
- "Interesting animations and allowing me to control the reading progress, making MDM my first choice."

7 Description of Supplementary Videos

The supplementary videos include the animations generated by our method and alternative methods. The description of these videos is given in Table 1:

Video Name	**Description**
supplementary_video_(special effects).mp4	the animation generated by our method from comics "*Fairy Tail*" with special effects including shaky effect, motion blur and moving motion lines
supplementary_video_(comparison)(MPS).mp4 supplementary_video_(comparison)(MoS).mp4 supplementary_video_(comparison)(Ours).mp4	the animations generated by MPS, MoS and our method from the same comics "*Slam Dunk*".
supplementary_video_(comic trailer).mp4	the comic trailer generated by our method from comics "*Slam Dunk*".
supplementary_video_(western comics).mp4	the animation generated by our method using western comics "*Ultimate Spider Man*".
supplementary_video_(Expert).mp4	the animation generated by an animation expert from comics "*Slam Dunk*".

Table 1: Description of supplementary videos.

8 Comic Pages

The comic pages used to generate the animations in the supplementary videos are shown in Figure 3, Figure 4 and Figure 5.

(1) (2) (3)

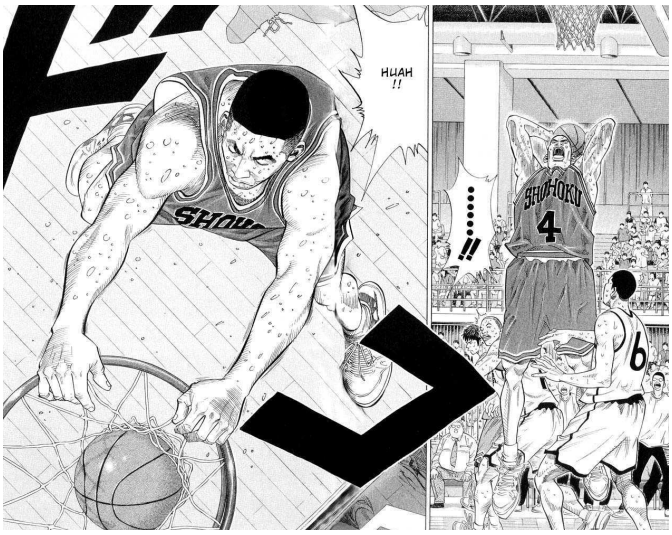
(4) (5) (6)

Figure 3: Comic pages from "*Fairy Tail*". Reading order of each page is from right to left.

(1)

(2)

(3)



(4)

(5)

Figure 4: Comic pages from "*Slam Dunk*". Reading order of each page is from right to left.

(1) (2) (3)

Figure 5: Comic pages from "*Ultimate Spider Man*". Reading order of each page is from left to right.