UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

PEDRO FRONCHETTI COSTA DA SILVA

A Quality Metric for Comparing Inbetweening Algorithms

Work presented in partial fulfillment of the requirements for the degree of Bachelor in Computer Science

Advisor: Prof. Dr. Eduardo Simões Lopes Gastal

Porto Alegre
August 2024

ABSTRACT

Inbetweening is the process of generating a new video frame whose visual content represents a moment in time that lies between two existing frames. This can be used, for example, to increase the framerate of a video or hand-drawn animation. Since many algorithms exist for generating inbetween frames, methods to compare results of different algorithms are needed. Benchmarks created from video sequences can be used to compare algorithm results to the actual middle frame, however this approach has two main limitations: *(i)* if the ground-truth middle frame does not exist, the method has no reference with which to compare the generated frame; *(ii)* even if the ground-truth middle frame exists (extracted from a video or drawn by a human animator, for example), it does not necessarily represent the only viable inbetween frame (i.e., many different inbetween frames may match the movement of the input video). To address these limitations, we propose the **Feature Matching Inbetweening Score** (FMIS) metric, which rates an inbetween frame based on features present in the input frames (before and after the generated inbetween frame). Variations of this scoring system are also discussed, taking factors like image blurriness and distance between matched points in consideration. The objective of the score is to provide an evaluation of inbetween frame quality similar to the analysis of a human observer. We perform a small user study which provides evidence that our FMIS metric aligns well with human perception.

Keywords: Inbetweening. Comparison. Image Processing.

Métrica de Qualidade Para Comparação de Algoritmos de Quadros Intermediários

RESUMO

Inbetweening é o processo de geração de um novo quadro de vídeo, cujo conteúdo visual representa um momento temporal que fica entre dois quadros existentes. Isso pode ser usado, por exemplo, para aumentar a taxa de quadros de um vídeo ou animação feitas a mão. Já que múltiplos algoritmos de geração de quadros intermediários existem, métodos para comparar seus resultados são necessários. Métricas de teste criadas a partir de sequências de vídeo podem ser usadas para comparar resultados de algoritmos com o quadro intermediário real, porém, essa abordagem possui duas limitações principais. *(i)* se não existir um quadro intermediário real, não há referência para comparar o quadro gerado; *(ii)* mesmo que o quadro exista (extraído de um vídeo ou desenhado por um animador humano, por exemplo), isto não representa necessariamente o único quadro intermediário viável (i.e., muitos quadros intermediários diferentes podem corresponder ao movimento do vídeo de entrada).

Para lidar com essas limitações, nós popomos a métrica **Feature Matching Inbetweening Score** (FMIS), que avalia um quadro intermediário com base nas propriedades presentes nos quadros de entrada (antes e depois do quadro intermediário gerado). Variações deste sistema de pontuação também são abordadas, levando fatores como nitidez e distância entre regiões características em consideração. O objetivo da pontuação é fornecer uma avaliação da qualidade do quadro intermediário que seja similar à análise de um observador humano. Nós conduzimos um pequeno estudo de usuário que fornece evidências de que nossa métrica FMIS se alinha bem com a percepção humana.

Palavras-chave: Inbetweening. Comparação. Processamento de Imagens.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

PSNR	Peak Signal-to-noise Ratio
SSIM	Structural Similarity Index Measure
AMT	All-Pairs Multi-Field Transforms
FILM	Frame Interpolation for Large Motion
GMFSS	GMFlow Based Anime Video Frame Interpolation
RIFE	Real-Time Intermediate Flow Estimation
sepconv	Adaptive Separable Convolution
ORB	Oriented FAST and rotated BRIEF
FMIS	Feature Matching Inbetweening Score
DA-FMIS	Distance Aware Feature Matching Inbetweening Score
BA-FMIS	Blur Aware Feature Matching Inbetweening Score
BDA-FMIS	Blur and Distance Aware Feature Matching Inbetweening Score

CONTENTS

1 INTRODUCTION

The animation industry has expanded exponentially in recent years, and with all the advancements in streaming and entertainment technology, the industry still has a lot of room to grow. Even though the consumption of animated content is at an all time high, the production process behind them is really demanding, especially for 2D animation, since frames have to be mostly drawn by hand. With the increase in demand and tight release schedules for the animation industry, many studios employ techniques such as mixing 3D graphics with 2D animation, in order to speed up the work. However, the use of 3D sometimes does not look good when mixed with 2D animation, so some studios still prefer doing everything in 2D, although it takes significantly more time and a lot more work by the artist team. Alternatives to drawing each frame in detail by hand usually lie in the process called *inbetweening*, in which artists draw "smear frames" or intermediate frames with less detail, to smooth the movement of the animated characters in the final product. Still, even if these frames are less detailed, the amount of them and the level of detail in the frames can result in a time-consuming inbetweening process.

With the recent advancements in machine learning and computer processing power, it becomes increasingly viable to use software in order to automate the generation of inbetween frames, as illustrated in Figure 1.1. However, current tools for frame interpolation often output very different results for the same animation, therefore, a method of gauging the quality of the generated frame becomes necessary, so that both animators and tool developers can compare different methods easily. Through these comparisons, a choice can be made between the many alternatives and the best-performing option can be selected.

The simplest approach to evaluate the quality of an inbetweening method is to rely on a ground-truth frame to evaluate the quality of the generated interpolation. However, this approach has two main limitations:

1. There are many cases (which includes the majority of commercially-produced animation), where there is no ground truth frame to be used as a reference for comparison. This limitation makes this method much harder to use for individual cases, in a production workflow (or even in video-framerate-increasing uses, where the middle frame doesn't exist yet);
2. When comparing to a specific ground-truth frame, the algorithms are being compared to one correct output, while there may be more than one inbetween frame that fits both input frames. This means the comparison to one ground truth may hinder some

Figure 1.1 – Example video frames with an inbetweening frame. Inbetweening frame is in the middle of the other two frames.

Source: Author

algorithms that provide an output that is different from the reference frame, but may not be incorrect. This is more pronounced in the case of animations, where sometimes, different techniques (such as deformation to convey movement and impact, for example) are applied and may result in different results that may be equally good.

In this work we propose a different way to compare outputs from frame interpolation methods, which can be used comparatively without a ground-truth frame, using any set of input frames. The goal is to provide an objective way to gauge how good an inbetween frame looks for a human observer, so that the best generated frame between many models can be selected.

2 RELATED WORKS

We give a brief overview of recent state-of-the-art automatic inbetweening methods, and how they are currently benchmarked. Throughout this manuscript, we use the terms "inbetweening" and "frame interpolation" interchangeably.

2.1 Frame Interpolation Methods

Many methods for inbetween frame generation have been proposed through recent years, seeking to solve different problems. Some methods benefit from the advancements in machine learning in order to approximate an adequate inbetweening method. These methods sometimes aim to tackle a specific type of video to interpolate, with some having optimizations for animation and some for real videos. The methods that are tested in this work are AMT (Li et al., 2023), FILM (Reda et al., 2022), Practical-RIFE (Huang et al., 2022), GMFSS_Fortuna (98mxr, 2023) and sepconv (Niklaus; Mai; Wang, 2021).

AMT (All-Pairs Multi-Field Transforms): The AMT method works by extracting features and image flow to build correlation volumes, refined by predictive convolutions, which generate many different flow fields. These flow fields are then used to generate the intermediate frame for the input images. For the purposes of model result reproduction and testing in our own data, we use the AMT-S pre-trained model, available at https://github.com/MCG-NKU/AMT.

FILM (Frame Interpolation for Large Motion): The FILM method uses a machine learning network, trained with true intermediary frames, to effectively generate a video frame. The algorithm uses feature extraction to estimate movement flows, and combines these features and flows to generate the inbetween frame for the video. We use the pre-trained film_net model (obtained in https://github.com/google-research/frame-interpolation) in order to generate inbetween frames with FILM.

GMFSS_Fortuna (GMFlow Based Anime Video Frame Interpolation): GMFSS_Fortuna also generates an inbetween frame based on the flow of the image, obtained via GMFlow (Xu et al., 2022), which estimates image flow using a transformer-based approach. We use the pre-trained union model to generate inbetween frames. We use the "fortuna_union_ft_animerun" (available at https://github.com/98mxr/GMFSS_Fortuna)

Practical-RIFE (Real-Time Intermediate Flow Estimation): RIFE (Huang et al., 2022) works by estimating intermediate flows and a fusion map using an Intermediate

Flow Network. The intermediate frame is obtained by warping the input frames based on the estimated flows and then combining them. Practical-RIFE aims to implement a practical version of RIFE, adding features and using new models to refine the frame generation further. For the purposes of model result reproduction and testing in our own data, we use the pre-trained model version 4.13.1, provided in the Practical-RIFE repository: https://github.com/hzwer/Practical-RIFE.

Sepconv (Adaptive Separable Convolution): sepconv aims to generate an intermediate frame using adaptive separable convolutions. After input processing, the input frames are plugged into a neural network, that generates the interpolated frame.

2.2 Evaluation of Frame Interpolation Methods

Many of the current methods are benchmarked using datasets like Vimeo 90K (Xue et al., 2019). The Vimeo 90K benchmark usually gives results based in a ratio between the PSNR and SSIM values for a ground-truth frame and a given interpolation. However, as previously mentioned, this evaluation depends on a ground-truth frame, which may not always exist. Also, these benchmarks usually provide good metrics to evaluate an inbetweening method, but they are hard to adapt to specific examples, which is sometimes a necessity when dealing with a variety of animation styles and different types of video in general.

3 SCORING AND COMPARING INBETWEENS

In this section, a method to compare and evaluate the quality of generated inbetween frames will be described, so that multiple approaches can be compared and the one that provides the best results can be automatically selected. Our goal is to provide an objective numerical score to make analysis easier, and this score should correlate with human perception of quality, giving preference to images that would be positively evaulated by a human observer. This method also aims to work without requiring a ground-truth frame and evaluate frames comparatively.

The proposed scoring method generates a number for a given inbetween image. This image is computed, by a given inbetweening method, from a pair of input images, representing the starting and ending frame. To explain the scoring of the inbetween frame, we will refer to the *starting image* as **A** and the *ending image* as **C**, while *the generated inbetween image* will be called **B**. Our numerical score takes the following metrics into account: visual features that are shared between all three images (feature matching), the distance between matched regions in different images, the number of matches, and the inbetween image's sharpness. The following subsections describe each of these metrics in more detail.

3.1 Feature Matching Inbetweening Score (FMIS)

We first find feature points in each image using the ORB algorithm (Rublee et al., 2011), resulting in a set of points $P(X)$ for each image $X \in \{A, B, C\}$. The ORB algorithm was chosen due to its rotation-agnostic nature and fast execution time, since we are going to run many comparisons between images and we need these comparisons to be fast. Points from each image are then matched against points in the other images, using a brute force matcher with Hamming distance. This results in a collection of matches $M(X, Y)$ of pairs of points $(p_1, p_2) \in M(X, Y)$ which are matched between images X and Y, such that $p_1 \in P(X)$ and $p_2 \in P(Y)$. The score for the inbetween image B is then defined as follows:

- For each point that has been matched between A and C, and is represented in B as well, the score in incremented by 2 units. Mathematically, the points that satisfy this

condition can be represented by the set $S_{A\wedge C}$, defined as follows:

$$\begin{aligned} S_{A\wedge C} = \{p_B \in P(B) \mid \exists p_A \in P(A) \wedge \exists p_C \in P(C) \text{ where} \\ (p_A, p_C) \in M(A, C) \wedge \\ (p_A, p_B) \in M(A, B) \wedge \\ (p_B, p_C) \in M(B, C)\}. \end{aligned} \tag{3.1}$$

- For each point that has been matched between A and B or B and C, but does not occur in all three images, the score is incremented by 1 unit. The mathematical definition for this set of points $S_{A\vee C}$ is (where "\" denotes set difference):

$$\begin{aligned} S_{A\vee C} = \{p_B \in P(B) \mid (\exists p_A \in P(A) \text{ where } (p_A, p_B) \in M(A, B)) \vee \\ (\exists p_C \in P(C) \text{ where } (p_B, p_C) \in M(B, C))\} \setminus S_{A\wedge C}. \end{aligned} \tag{3.2}$$

- Finally, for each point that is present in the inbetween B but not in any of the base images (A and C), the score is decremented by 1 unit. This set of points, named $S_{\varnothing}$, is represented by the following formula:

$$\begin{aligned} S_{\varnothing} = \{p_B \in P(B) \mid (\nexists p_A \in P(A) \text{ where } (p_A, p_B) \in M(A, B)) \wedge \\ (\nexists p_C \in P(C) \text{ where } (p_B, p_C) \in M(B, C))\}. \end{aligned} \tag{3.3}$$

After all the matches are processed, the results can be used to calculate a score, which we call **FMIS** (Feature Matching Inbetweening Score). This score, denoted by Σ_{FMIS}, represents a baseline approach for determining which inbetween images are better when compared with other inbetweens. It is computed by considering the cardinality of the sets ($S_{A\wedge C}$, $S_{A\vee C}$, $S_{\varnothing}$), as shown in Eq. (3.4). This score is numerical and doesn't have a defined range, larger numbers mean better results:

$$\Sigma_{\text{FMIS}} = 2|S_{A\wedge C}| + |S_{A\vee C}| - |S_{\varnothing}|. \tag{3.4}$$

3.2 Distance-Aware Feature Matching Inbetweening Score (DA-FMIS)

When observing the matches made by the ORB algorithm, it is possible to see that while most of them are correct, in a few instances, the points matched are not related to each other. For example, Figure 3.1 represents all the matches between the input frames

A and C, being A the image on the left, and C the image on the right. While the figure is very hard to read due to the number of matches, it is still possible to spot some wrong matches, such as lines that match the character's feet in drawing A to the character's torso in drawing C.

To circumvent this problem, when reviewing the matches, it is possible to filter matches by distance, removing matches with two points that are too distant, which may possibly be a mismatch. To define the distance threshold, it is also ideal to take image dimensions into account, since a fixed threshold might work well for images of a certain size, but it is going to be too big or too small for images that are bigger or smaller. In practice, a percentage of the image's diagonal is used as parameter to define an adaptive distance threshold. So considering d to represent the size of the image's diagonal and p to be the parameter (a number between 0 and 1 representing the percentage of the diagonal to be used), $d\,p$ is used as the distance threshold, so only matches that have a distance smaller or equal to $d\,p$ between their coordinates will be considered in the scoring process. More precisely, for a match $(p_1, p_2) \in M(X, Y)$, if $p_1 \in P(X)$ has coordinates (x_1, y_1) in image X and $p_2 \in P(Y)$ has coordinates (x_2, y_2) in image Y, the match is only considered if:

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \leq d\,p. \tag{3.5}$$

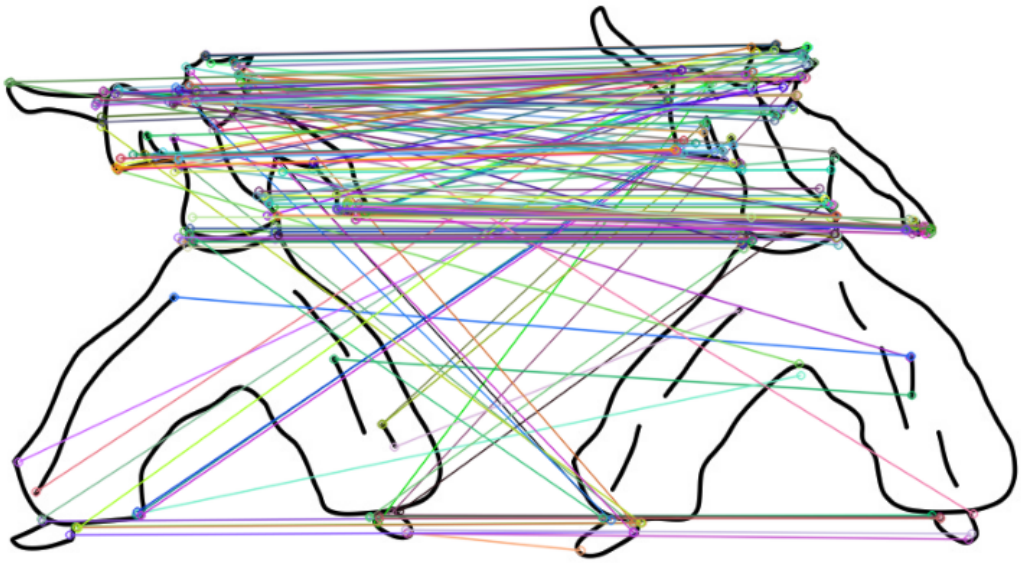
This method results in a variation of the original scoring system which is aware of the distances between points in a match, but in its core, it uses the same scoring presented in FMIS. Therefore, we call this method DA-FMIS (Distance-Aware Feature Matching Inbetweening Score).

Figure 3.2 shows the effect of limiting the distance between matches. As it is possible to see, many of the wrong matches were removed, so that now FMIS can focus on the features that are true matches.

3.2.1 Choosing the relative threshold parameter p for DA-FMIS

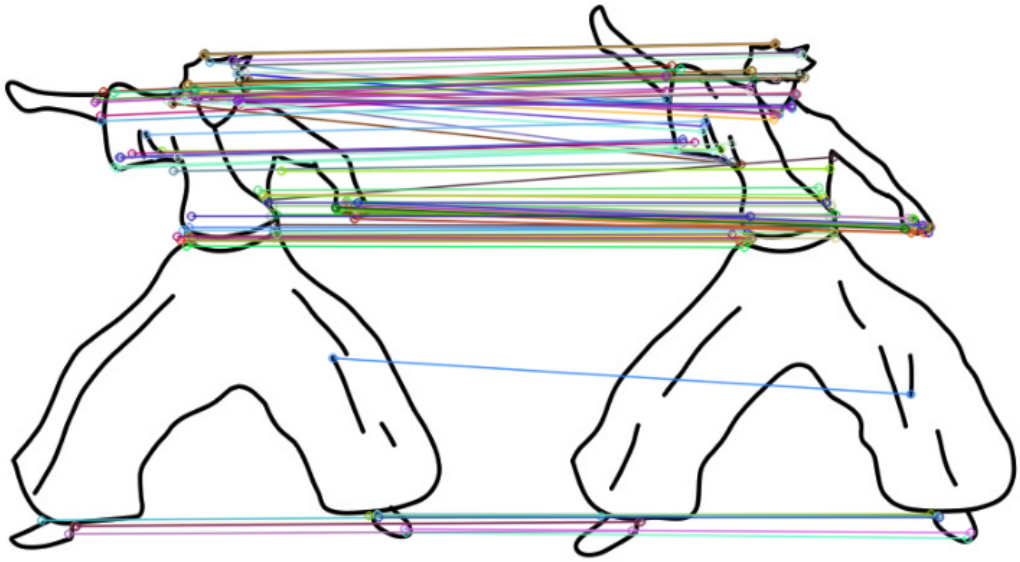
A k-fold cross-validation inspired approach was used, where the image set used was divided in 5 folds (with 3 images each), 3 used for training and 2 for testing, in all possible fold combinations. For each fold, using a brute-force search, the best p value for the training images was discovered through comparing the results for DA-FMIS using a set p value, to results obtained in a user study related to inbetween frame preference (this

Figure 3.1 – All ORB feature matches between frame A (left) and C (right), used in FMIS. Matching points are color coded and connected by straight lines.



Source: Author

Figure 3.2 – All ORB feature matches used in DA-FMIS (with distance parameter $p = 0.1$), which removes matches that are too far apart in the image space. Matching points are color coded and connected by straight lines.



Source: Author

will be detailed in Section 4.2). This value was applied to the images in the test set of the fold, resulting in a Kendall Tau metric (discussed in Section 4.2) which is averaged across folds, and the average τ of the rankings made using DA-FMIS. The obtained result that seems to work the best is $p = 0.27$.

3.3 Blur Aware Feature Matching Inbetweening Score (BA-FMIS)

There is another factor that is important to determine the quality of an image, which is image blurriness. If a blurry inbetween frame is generated for a pair of sharp input images, the result is often not desirable, as it will often contrast too much with frames A and C. There is another case to be considered, which is the case of images employing motion-blurring, which is a technique in animation that reduces details of objects and deforms their shapes, in order to represent fast movement. If one of the input frames is using this technique, it may result in a blurrier inbetween, therefore it would be unfair to penalize the score of image B for being blurry as well.

This problem can be solved by applying a Laplacian filter with kernel size 3 to images A, B and C, in order to quantify their sharpness (by detecting edges). The sharper an image is, the higher the Laplacian filter's return values are going to be. The average of the Laplacian filter values for every pixel of the images A, B and C will be referred to as L_A, L_B and L_C respectively. To better utilize these values, the function below is used to place the blurriness values in a scale of 0 to 1:

$$f(L_B, L_A, L_C) = \min\left(1, \frac{L_B}{\min(L_A, L_C)}\right). \tag{3.6}$$

The use of the minimal number between L_A and L_C as divisor ensures that we select the most blurry of the input frames as the expected blurriness of the inbetween frame. This function is also capped at 1, so even if the inbetween image turns out to be sharper than the blurriest input frame, it will not receive any additional score for this fact.

After the value of the function $f(L_B, L_A, L_C)$ is obtained, it is used to increment the inbetween's FMIS score. However, since the score is not contained in a defined range, it is hard to gauge the impact an arbitrary number will have in each case. Considering this, it is better to use the function's result as a multiplier to the score already calculated by FMIS. Assuming Σ_{FMIS} to be the FMIS score for an image, the formula for the new score

calculation is the following:

$$\Sigma_{\text{BA-FMIS}} = \Sigma_{\text{FMIS}} + f(L_B, L_A, L_C) \times \frac{\Sigma_{\text{FMIS}}}{2}. \tag{3.7}$$

This score now reflects feature matches and takes the image's sharpness into account, therefore, this method is called BA-FMIS (Blur Aware Feature Matching Inbetweening Score). This approach can also be combined with DA-FMIS, to offer a metric that takes into consideration both distance between matches and image sharpness. To combine them, we can replace Σ_{FMIS} by the score $\Sigma_{\text{DA-FMIS}}$ of DA-FMIS in Eq. (3.7). We call the resulting metric BDA-FMIS.

Finally, as was done with DA-FMIS, the value of the parameter p needs to be adjusted, and for that, the same approach was used as when determining the value of p for DA-FMIS. The value chosen for BDA-FMIS is $p = 0.25$.

4 RESULTS

In this section we evaluate the capacity of our FMIS, DA-FMIS, BA-FMIS and BDA-FMIS metrics in evaluating the quality of a generate inbetween frame.

4.1 Image Dataset

Different methods of inbetween frame generation work in their own way internally, but all of them have two inputs: a starting frame and an ending frame. By using the same pairs of images for the inputs of the various methods, we can generate comparable inbetween frames for those pairs. For our experiments we chose images with diverse styles, some being real life photographs, some being frames from traditional animations and some being black and white line art. Even if some of the tested methods have a specific focus, they were run on the whole image set, so they can be better compared with the other methods. In total, the image set used to test the methods was composed of a total of 30 images, representing 15 pairs of starting and ending frames, shown in Figures 4.1 to 4.3.

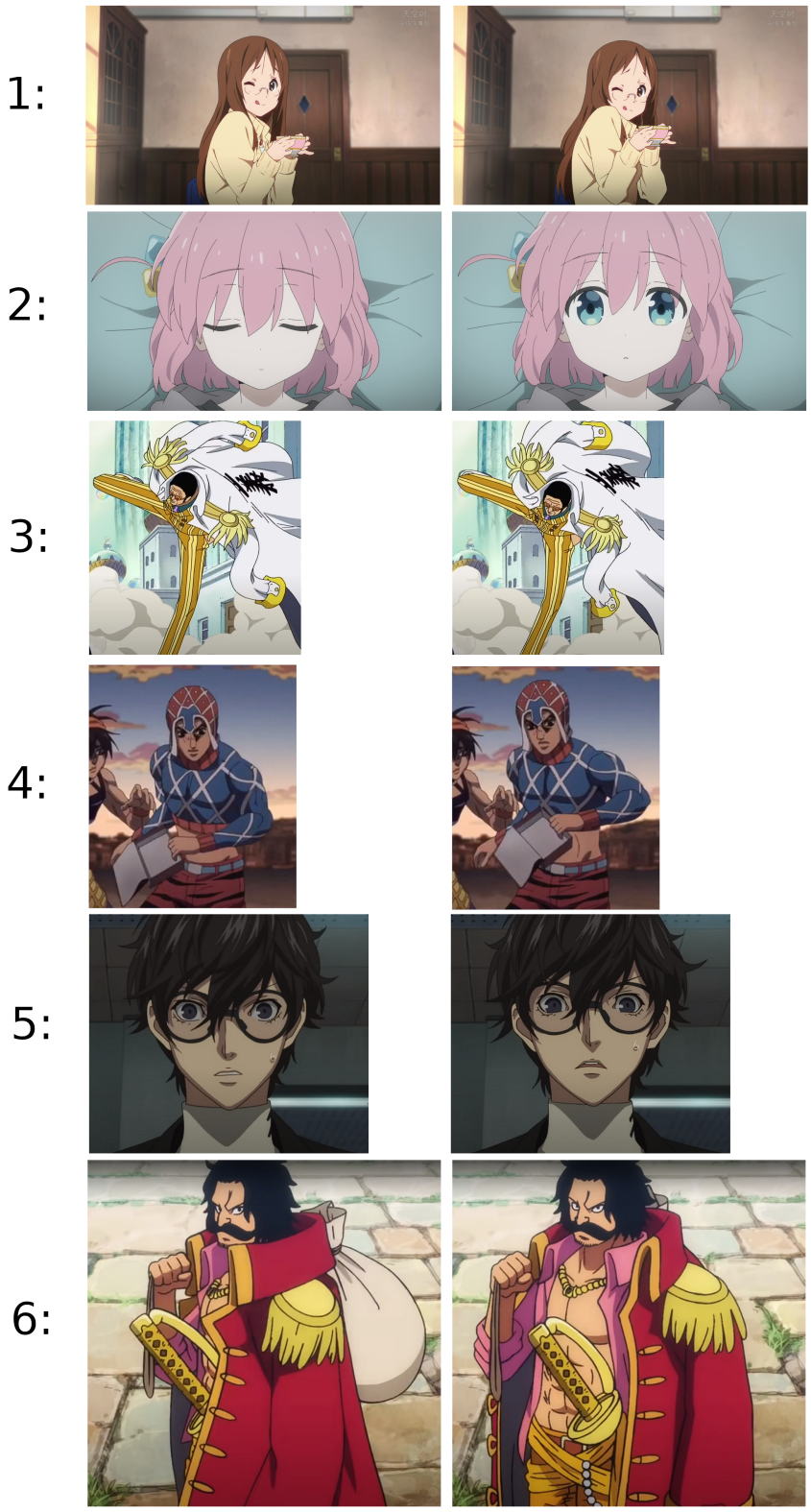
These images can each be described as trying to test something specific, and are labeled accordingly. Images named Animation have been extracted from traditional animation and are colored drawings. Line Arts have been drawn with black and white only, with much less detail than Animation images, but still easily comprehensible by humans. Photos contain pictures of real people, animals or objects. All images can be accessed in their original resolution and compression format at https://github.com/pefcos/fmis. A short description of each image is specified in Table 4.1.

4.2 Inbetweening Methods Used to Evaluate Our Metrics

A total of five inbetween frame generation methods were used:

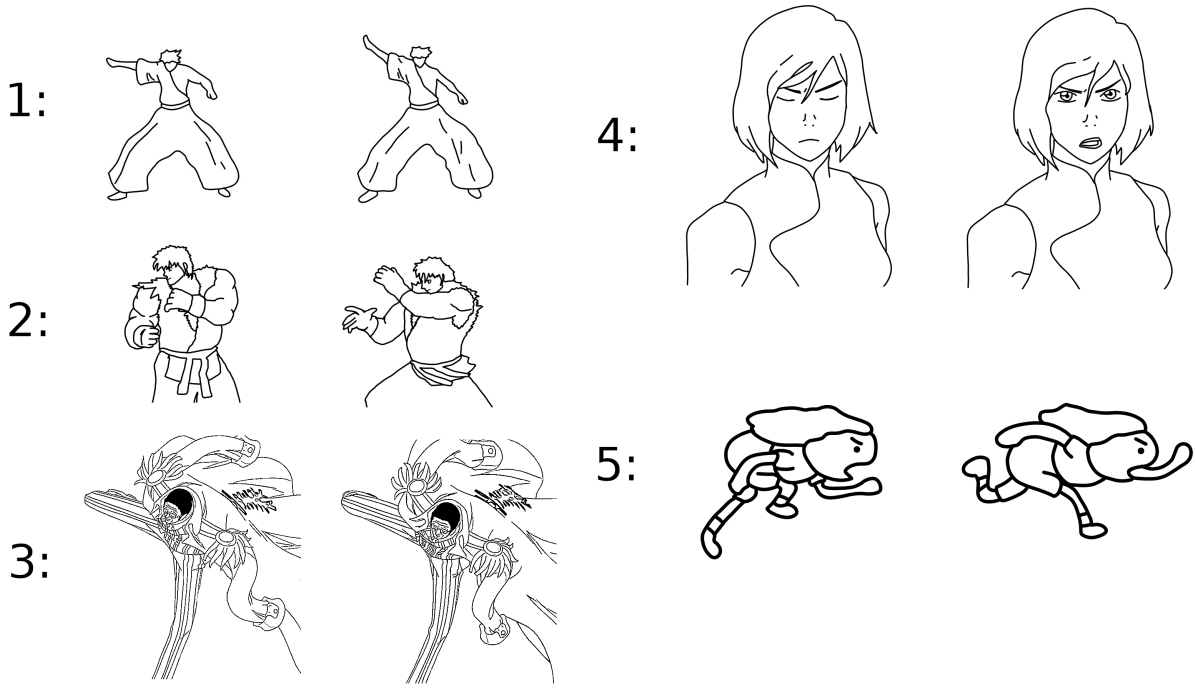
- AMT (All-Pairs Multi-Field Transforms) (Li et al., 2023);
- FILM (Frame Interpolation for Large Motion) (Reda et al., 2022);
- GMFSS-Fortuna (GMFlow Based Anime Video Frame Interpolation) (98mxr, 2023);
- Practical-RIFE (Real-Time Intermediate Flow Estimation) (Huang et al., 2022);
- sepconv (Adaptive Separable Convolution) (Niklaus; Mai; Wang, 2021).

Figure 4.1 – Test images labeled as "Animation", 1 to 6. The left column shows the first input image (frame A) and the second columns shows the corresponding second input image (frame C).



Source: (Kyoto Animation, K-On!), (CloverWorks, Bocchi The Rock!), (TOEI, One Piece), (David Productions, JoJo's Bizarre Adventure), (Atlus, Persona 5), (TOEI, One Piece)

Figure 4.2 – Test images labeled as "Line Art", 1 to 5. The left column shows the first input image (frame A) and the second columns shows the corresponding second input image (frame C).



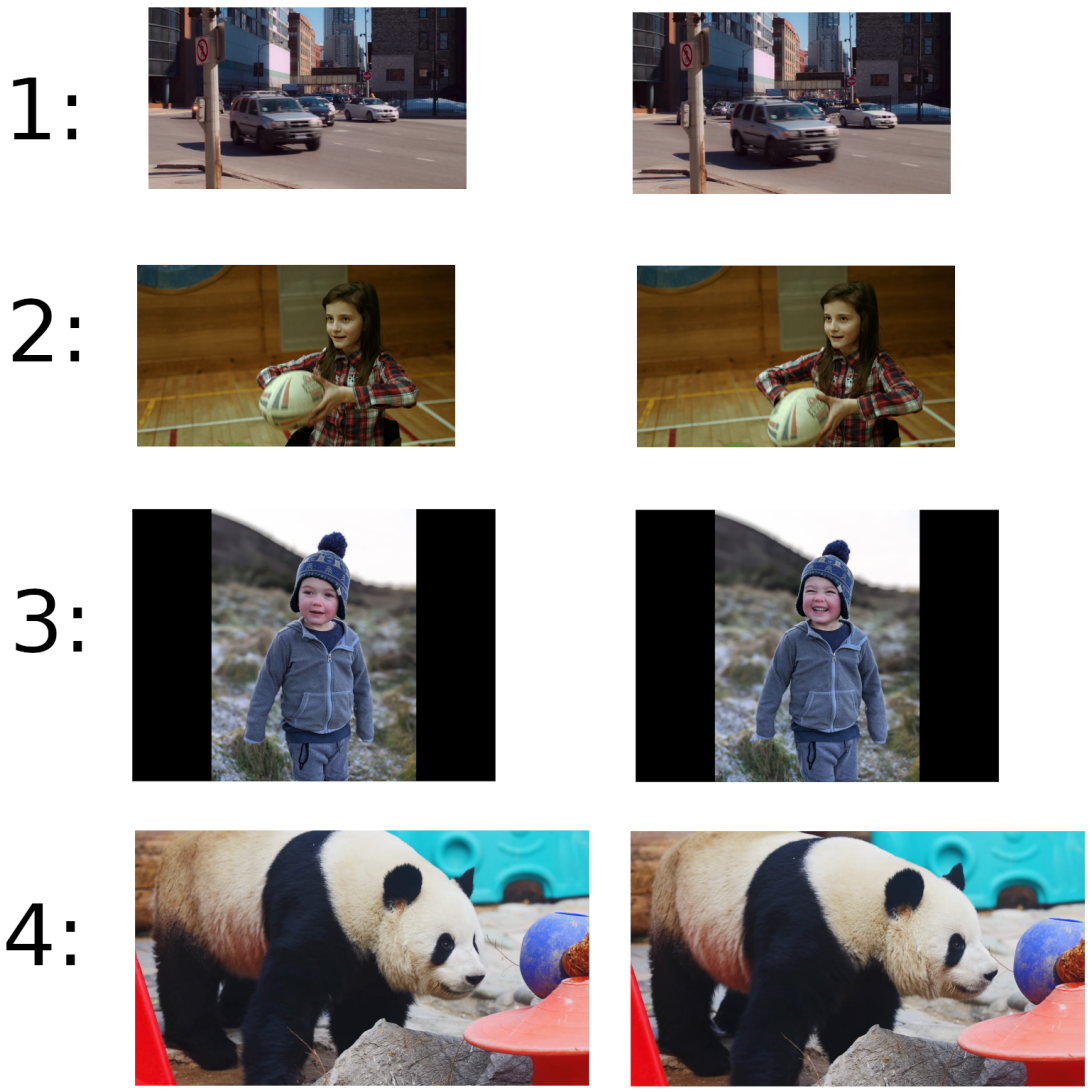
Source: Author, Author, (TOEI, One Piece, Edited by Author), Author, Author

Table 4.1 – Test image set description

Image Name	*Description*
Animation 1	Animation frame with small movement and little occlusion
Animation 2	Large image with eyes opening, inner eye occlusion
Animation 3	Animation frame with a character rotating a small amount, clothing movement
Animation 4	Animation frame with a character moving their arm back
Animation 5	Animation frames with very little change
Animation 6	Animation frames with heavy occlusion and a wide character rotation
Line Art 1	Line art with character raising their hand and clothing creases
Line Art 2	Very different frames with character rotation, pose change and hair movement
Line Art 3	Line art frame with a character rotating a small amount, clothing movement
Line Art 4	Line art with character opening their eyes and has little hair movement
Line Art 5	Very different frames of a cartoon character running sideways
Photo 1	Motion-blurred picture of a car moving
Photo 2	Girl lifting a ball with her hands
Photo 3	Child smiling, little movement but has expression change
Photo 4	Panda bear moving slightly, little movement as well

Source: Author

Figure 4.3 – Test images labeled as "Photo", 1 to 4. The left column shows the first input image (frame A) and the second columns shows the corresponding second input image (frame C).



Source: (Huang et al., 2022), (Huang et al., 2022), (Reda et al., 2022), (Li et al., 2023)

Number of participants:	10
Number of rankings created per participant:	15 (one for each image in Table 4.1)
Number of inbetween images per ranking:	up to 5 (one for each inbetweening algorithm)

Table 4.2 – User study setup.

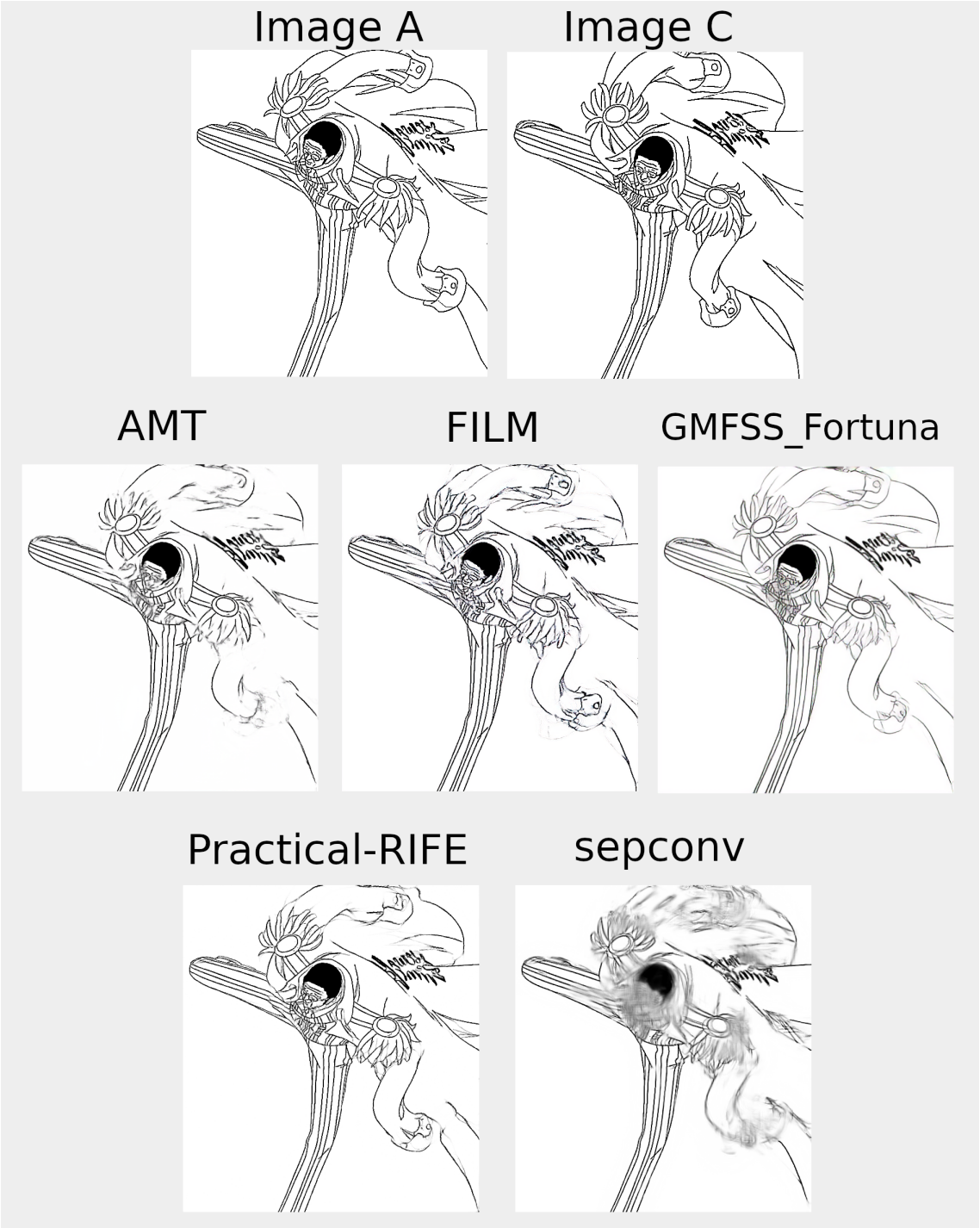
Some of them did not work as intended for some image pairs, so some of the test image sets have less than 5 generated frames. An example of the generation results for all of these methods can be seen in Figure 4.4.

4.3 Evaluation Metric and Comparison Baseline

In order to accurately measure the effectiveness of the four scoring systems, it is necessary to compare it against some reference. Since the desired effect of the animation is looking good to the final consumer, it is desirable to have the metrics proposed compared to human rankings of the inbetween frames. To achieve this, we performed a user study where a survey was answered by 10 participants, who were asked to rank the inbetweens generated by different methods from best to worst, in a set of 15 A-C image pairs, each with up to 5 inbetween frames to rank (one for each inbetweening method) – Table 4.2. This resulted in 10 different rankings for each input pair (one for each user in the study). Considering that all rankings have the same elements and the only change is the order in which they are ordered, the Kendall Tau rank correlation coefficient (KENDALL, 1938), denoted by τ, was chosen to compare the different rankings. The coefficient τ between any two rankings is a number between -1 and 1, where -1 signifies completely different rankings, and 1 signifies completely equal rankings.

Comparing participants' rankings among themselves revealed some interesting information about the human opinion on some of the example images. This data is organized in the form of a boxplot graph, to make it easier to visualize, represented in Figure 4.5. Each of the 15 images has its own column with a statistic of all the rank-comparisons (τ values) made for the image (for each image, the ranking from each of the 10 participants is compared with the rankings of every other participant). This data is divided into 4 quartiles, with the box representing the two middle quartiles, and the horizontal line inside the box representing the median. The vertical whisker lines range from the minimum to maximum τ values, with outliers denoted by unfilled circles. The closer an image's ranking comparison distribution is to 1, the more users agree on the ranking of the inbetweening methods for that image. If the lines and box are too tall, users

Figure 4.4 – Demonstration of the inbetween frames generated by different methods (for Line Art 3).



Source: Author

have more divergent opinions on the correct ranking. For example, one can see that all users agree on the ranking of the inbetween frames for the “Animation 2” test image, as it is clear that Practical-RIFE generates the best result (Figure 4.6). On the other hand, there is strong disagreement between users on the correct ranking of the inbetween frames for the “Animation 5” test image, as all methods perform relatively well in this case (Figure 4.7).

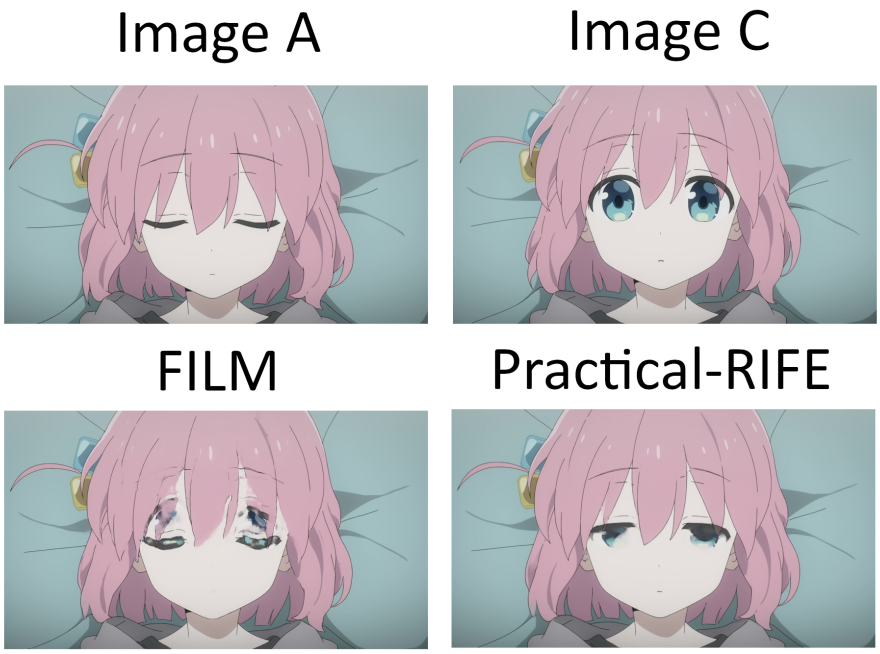
Figure 4.5 – Comparison of human inbetween image rankings

Source: Author

4.4 Evaluation of FMIS, DA-FMIS, BA-FMIS and BDA-FMIS

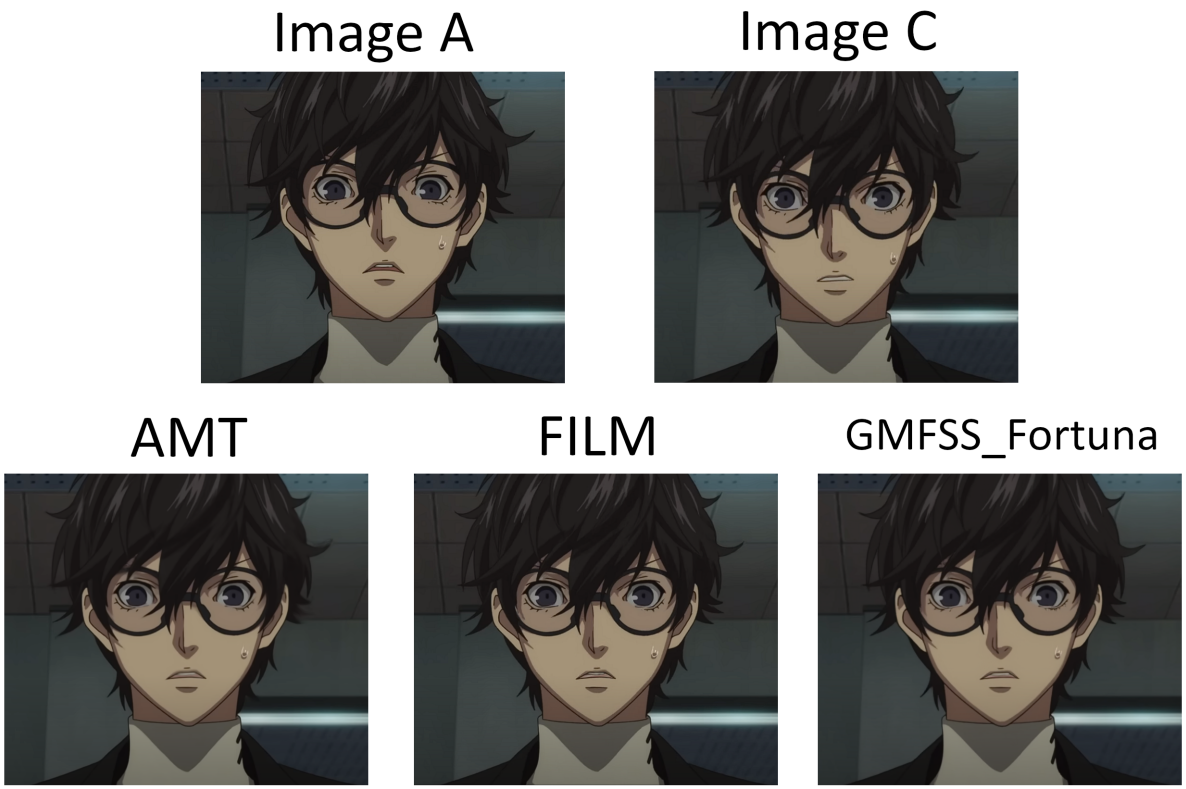
To evaluate how well our FMIS metric represents the order established by the survey participants, a plot of the same boxplot was made, but now with light blue dots indicating the average τ value for comparisons of FMIS rankings to human rankings for each image. This can be seen in Figure 4.8. For a FMIS score to be considered good, it needs to be inside of the box plotted for its respective image. This means that the FMIS score agrees with humans in the same measure as humans agree with themselves. Values at the upper bounds or above the boxes are also desireable, which means that our score’s selection of inbetweening method would please most, if not all humans in the survey, despite their different opinions among themselves. In summary, values inside the box are

Figure 4.6 – Inbetween frames generated by different methods for the “Animation 2” test image. Practical-RIFE clearly generates a better result when compared to FILM.



Source: Author

Figure 4.7 – Inbetween frames generated by different methods for the “Animation 5” test image. All methods generate comparable results.



Source: Author

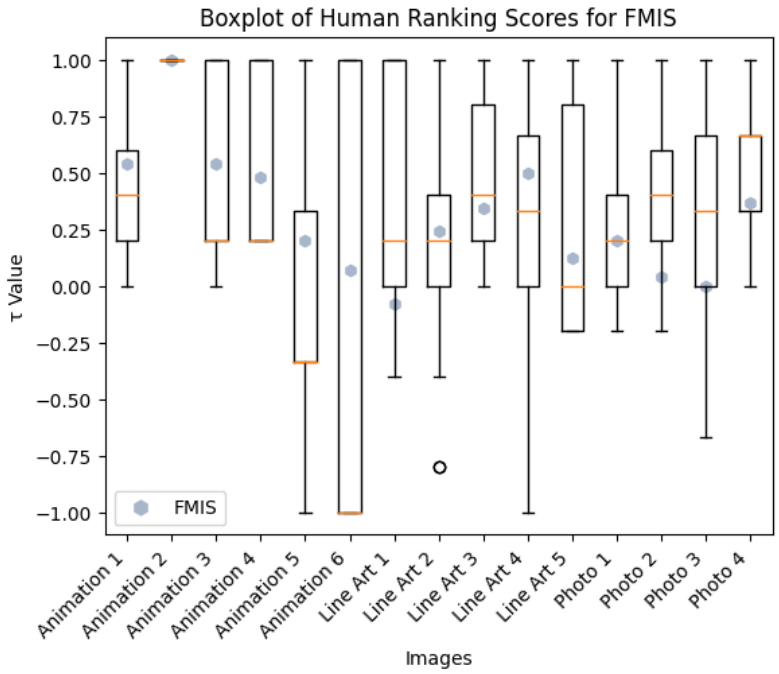
good, but the closer to the upper bound of the box, the better. In the case of FMIS, 2 values out of the 15 are outside and below the boxes, specifically, for the tests conducted with Line Art 1 and Photo 2. This means that while FMIS generally has an opinion similar to humans, for these test images it tends to agree more with a minority of users.

The same statistical comparison can be performed for our DA-FMIS metric (Distance Aware FMIS), using the parameter $p = 0.27$. This plot is represented in Figure 4.9 by the red markers, and is comparable to the plot of the FMIS score, represented by light blue markers. DA-FMIS presents itself to be as good as FMIS in most cases, but proves to be better than it in some cases (Such as Line Art 1 and Photo 4), only failing to touch the box in Photo 2. The removal of matches that are too distant might bring the algorithm's vision closer to our own, removing any wrong matches that a human wouldn't make. However, this approach might hinder the scoring of images whose starting and ending frames have large movement between them, since it cuts all matches more distant than a certain length. Given that many of the algorithms tested for inbetween frame generation don't perform ideally for ample movements, the use of DA-FMIS is a compromise usually worth making.

Now for BA-FMIS (Blur Aware FMIS), the same base boxplot will be used, however, it must be compared with FMIS and with DA-FMIS. Firstly, by comparing it to FMIS in Figure 4.10, some improvements can be seen, but while in the comparison between FMIS and DA-FMIS there was a clear DA-FMIS prevalence, in this graph there are some cases where BA-FMIS performs worse than FMIS (such as Animation 1 and Line Art 3). This data alone might seem to indicate that DA-FMIS is the better method, but when comparing both DA-FMIS and BA-FMIS, as seen in Figure 4.11, it becomes harder to indicate a clear winner, as each method performs better in different cases. BA-FMIS also touches the box in every image, including Photo 2, where both FMIS and DA-FMIS failed to do so, indicating its scoring is closer to human standards.

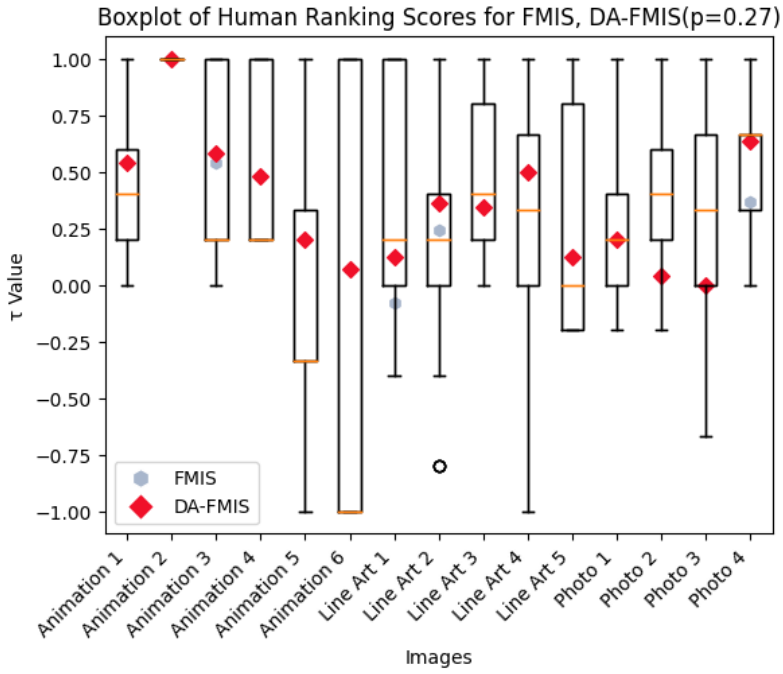
As mentioned before, both DA-FMIS and BA-FMIS can be combined into an approach BDA-FMIS, that uses both distance and image blurriness to determine a score. Through the comparison with all the other methods in Figure 4.12, it is clear the approach outperforms most others, with the exception of Animation 1, in which DA-FMIS remains the better metric, possibly because the blurring of the inbetween frame doesn't impact the comprehension of the animation. With BDA-FMIS, all markers are at least on the lower boundary of the box for their image, and many of them are inside them, with no marker outside the box or its borders. It is also notable in the comparison of the four methods, that some images did not benefit from improvements in relation to distance and blurriness, as is

Figure 4.8 – Comparison of average FMIS to human rankings



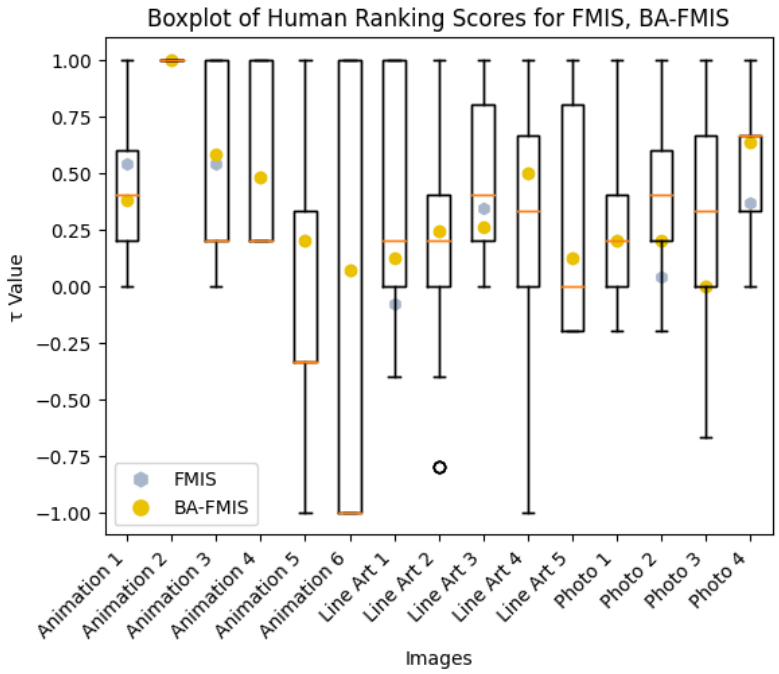
Source: Author

Figure 4.9 – Comparison of average DA-FMIS and FMIS to human rankings



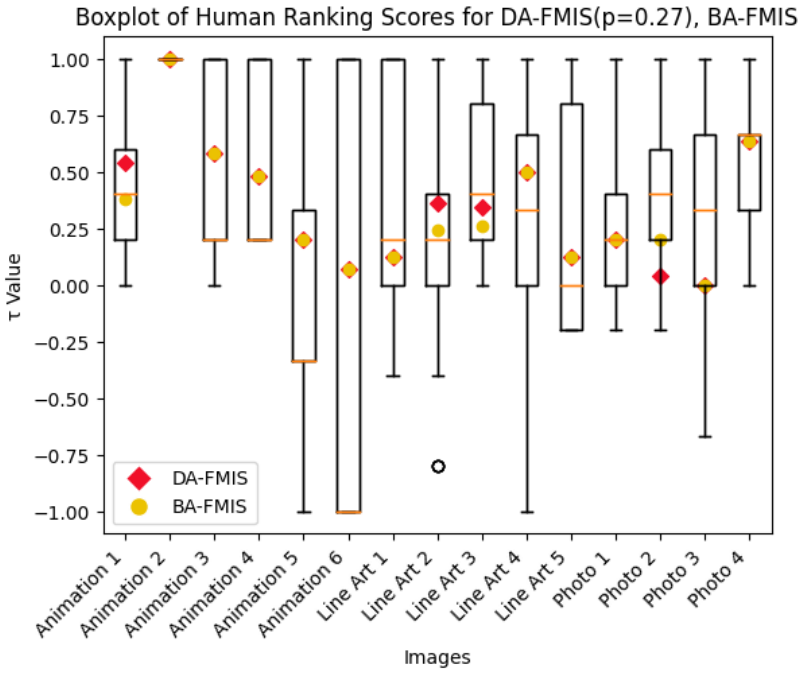
Source: Author

Figure 4.10 – Comparison of average BA-FMIS and FMIS to human rankings



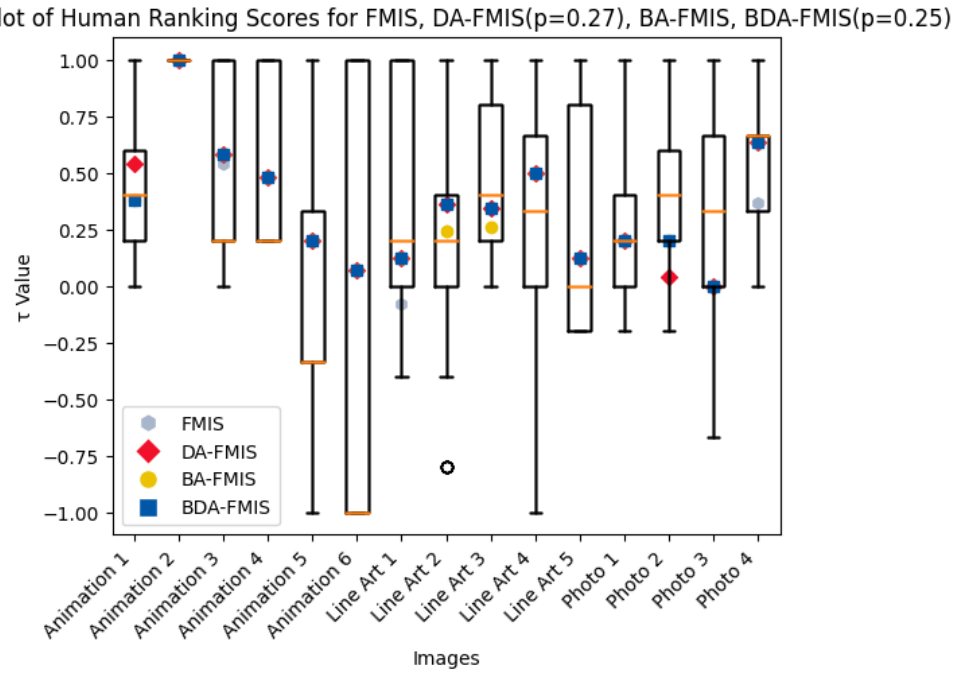
Source: Author

Figure 4.11 – Comparison of average BA-FMIS and DA-FMIS to human rankings



Source: Author

Figure 4.12 – Comparison of average FMIS, DA-FMIS, BA-FMIS and BDA-FMIS to human rankings



Source: Author

the case with image pairs Animation 1, Line Art 4 and Photo 3. This is due to inbetweens having very similar levels of blurriness and having few mismatched features.

5 CONCLUSION

When analyzing the results, it becomes clear each factor has its own influence in determining the feature matching score. While FMIS was a flawed metric, it was a good starting point to improve on. Both improvements made in regards to distance (DA-FMIS) and blurriness (BA-FMIS) could be combined to create a metric (BDA-FMIS) that can be used reliably to compare different inbetweens. Applying the BDA-FMIS comparison score to the full image dataset, using all compared inbetweening methods, results in the scores seen in Table 5.1, which when ordered in a ranking, result in Table 5.2.

Table 5.1 – $\Sigma_{\text{BDA-FMIS}}$ comparison for each inbetweening generation method for each image pair. Locations marked with a "–" represent images for which the respective inbetweening method generated an error and was not capable of generating an output (the hardware we had available for our tests did not have enough memory for these methods to process the respective input images).

Image Pair	AMT	FILM	GMFSS_Fortuna	Practical-RIFE	sepconv
Animation 1	492.69	538.14	433.56	527.31	252.58
Animation 2	–	-71.27	–	-18.63	–
Animation 3	300.0	333.48	261.34	352.47	120.53
Animation 4	464.45	546.40	367.78	512.0	265.04
Animation 5	513.26	597.0	435.44	–	–
Animation 6	–	96.56	56.14	64.37	–
Line Art 1	206.32	258.21	118.05	294.84	221.54
Line Art 2	-139.03	-112.30	-233.23	-220.24	-223.89
Line Art 3	97.82	34.72	25.38	11.91	-13.75
Line Art 4	617.89	611.90	400.46	631.79	–
Line Art 5	-98.46	-66.12	-116.94	-102.97	-93.46
Photo 1	612.08	607.46	351.86	612.57	483.12
Photo 2	491.85	477.25	276.74	500.09	436.61
Photo 3	455.12	504.42	378.68	485.90	–
Photo 4	–	624.10	460.97	576.26	351.29

Source: Author

The detailed algorithm ranking analysis enables some insight into which approaches generally give the best results, in a way that mostly aligns with human spectator perception when watching an animation. It is possible to see that FILM and Practical-RIFE usually performed best in most test cases, while sepconv and GMFSS_Fortuna were comparatively worse than the alternatives. AMT also had a satisfactory performance and generates good looking inbetweens. Upon further analysis, with a specific focus in animation examples, FILM performed the best in 4 out of 6 cases, being the second best approach in the remaining 2.

Although the metrics proposed proved to be capable of comparing the methods in

Table 5.2 – $\Sigma_{\text{BDA-FMIS}}$ based ranking for each inbetweening generation method for each image pair

Image Pair	1st	2nd	3rd	4th	5th
Animation 1	FILM	Practical-RIFE	AMT	GMFSS Fortuna	sepconv
Animation 2	Practical-RIFE	FILM			
Animation 3	Practical-RIFE	FILM	AMT	GMFSS Fortuna	sepconv
Animation 4	FILM	Practical-RIFE	AMT	GMFSS Fortuna	sepconv
Animation 5	FILM	AMT	GMFSS Fortuna		
Animation 6	FILM	Practical-RIFE	GMFSS Fortuna		
Line Art 1	Practical-RIFE	FILM	sepconv	AMT	GMFSS Fortuna
Line Art 2	FILM	AMT	Practical-RIFE	sepconv	GMFSS Fortuna
Line Art 3	AMT	FILM	GMFSS Fortuna	Practical-RIFE	sepconv
Line Art 4	Practical-RIFE	AMT	FILM	GMFSS Fortuna	
Line Art 5	FILM	sepconv	AMT	Practical-RIFE	GMFSS Fortuna
Photo 1	Practical-RIFE	AMT	FILM	sepconv	GMFSS Fortuna
Photo 2	Practical-RIFE	AMT	FILM	sepconv	GMFSS Fortuna
Photo 3	FILM	Practical-RIFE	AMT	GMFSS Fortuna	
Photo 4	FILM	Practical-RIFE	GMFSS Fortuna	sepconv	

Source: Author

a similar way to the average human analysis, there are still some limitations that might need addressing. Even with the distance awareness introduced in DA-FMIS, some of the matches detected by the brute force matcher are still inaccurate, while a human is certainly able to match features with near perfect certainty. Also, the DA-FMIS approach of cutting too distant matches may remove some accurate matches that are just representative of wide movements in the animation frames. Both of these issues are still present in the final BDA-FMIS implementation. Also, since there were relatively few testing images (30 images in 15 pairs), further testing and comparing with larger image sets can reveal some edge-cases in which BDA-FMIS might perform poorly when compared to human preferences.

The BDA-FMIS metric (or any of the other proposed metrics) could also be used to automate inbetween frame selection when using many methods for the same animation. This workflow could greatly improve animators' time and ensure that the best method for automated inbetween frame generation is always used, saving resources and providing more fluid looking animations, without any significant image quality loss. Of course, it can also be purely used as a benchmark for inbetweening method developers, to compare their technique to other significant techniques. This approach does not only apply to animation, but could be valuable to streamline comparisons in AI generated video creation and image animation as well.

REFERENCES

98MXR. *GMFSS_Fortuna*. 2023. <https://github.com/98mxr/GMFSS_Fortuna>. Accessed: 2024-05-20.

HUANG, Z. et al. **Real-Time Intermediate Flow Estimation for Video Frame Interpolation**. 2022.

KENDALL, M. G. A NEW MEASURE OF RANK CORRELATION. **Biometrika**, v. 30, n. 1-2, p. 81–93, 06 1938. ISSN 0006-3444. Available from Internet: <https://doi.org/10.1093/biomet/30.1-2.81>.

LI, Z. et al. **AMT: All-Pairs Multi-Field Transforms for Efficient Frame Interpolation**. 2023.

NIKLAUS, S.; MAI, L.; WANG, O. Revisiting adaptive convolutions for video frame interpolation. In: **IEEE Winter Conference on Applications of Computer Vision**. [S.l.: s.n.], 2021.

REDA, F. et al. **FILM: Frame Interpolation for Large Motion**. 2022.

RUBLEE, E. et al. Orb: An efficient alternative to sift or surf. In: **2011 International Conference on Computer Vision**. [S.l.: s.n.], 2011. p. 2564–2571.

XU, H. et al. Gmflow: Learning optical flow via global matching. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2022. p. 8121–8130.

XUE, T. et al. Video enhancement with task-oriented flow. **International Journal of Computer Vision**, Springer, v. 127, p. 1106–1125, 2019.