Centre de Sémiotique et Rhétorique

Un /carnet/ de « recherches »

Advances on the F.R.S.-FNRS research project “Towards a Genealogy of Visual Forms”

by Adrien Deliege

Disclaimer: the images have been compressed during the publication of this article. All high-quality images are available here.

This post summarizes some works carried out on Maria Giulia Dondero’s F.R.S.-FNRS research project “Towards a Genealogy of Visual Forms”.

Background of the project. The project aims at analyzing, quantifying, clustering, characterizing the evolution of “forms” in the broad sense, that have been represented in large corpuses of paintings throughout centuries, up to modern art such as in fashion photography. For that purpose, the project

builds upon recent computer vision techniques powered by deep learning-based models trained on generic large-scale datasets. The interaction between these two worlds, computer vision on one hand, art analysis on the other hand, benefits them both, by producing insights on the capabilities and limitations of the former in the artistic domain, and by resulting in novel ways of revisiting the latter.

This post can be decomposed into 3 parts: (1) an introduction to pose estimation in computer vision and the model used, (2) an introduction to PixPlot as a visualization tool, (3) a presentation of results for the task of retrieving "similar" images to a query image.

Pose estimation in computer vision

Focus on human poses. The present article focuses on one particular type of "form" frequently encountered in classic paintings and modern photography, and in images in general: humans. More specifically, we investigate which kinds of *poses* human bodies take in the images of interest, as they usually confer a lot of structure to the image itself. By "pose", we do not mean any particular state of the character represented, such as sitting, standing, lying down, but we rather mean the overall organisation of his limbs, that is, how his skeleton is articulated. This allows for more well-defined computationally-friendly operations, while not preventing us from classifying poses in various categories when this becomes necessary.

Pose estimation models. In computer vision, the task of computing the skeleton of a person from an image of that person is called *pose estimation*. Since the deep learning revolution in the late 2000s, many "models" (which are nothing less than special types of algorithms) have been developed to handle that task, such as the popular OpenPose and DensePose. Some libraries, which regroup many models, codes, and various functionalities, also support research

activities in that field, such as MMPose and PaddlePaddle. In this work, we use the MMPose library with its RTMPose model, which we found sufficiently good in some preliminary experiments.

Skeletons and 17 keypoints. Formally, we consider that a human skeleton is composed of a collection of *keypoints*, localized at the joints between limbs or at salient important body parts. In our case, the model produces skeletons articulated around 17 keypoints: one keypoint for the nose, then two (left and right) keypoints for the eyes, ears, shoulders, elbows, wrists, hips, knees, ankles. Pose estimation models aim at providing lists of coordinates for those keypoints. Only for visual representations, these keypoints are linked together as appropriate to produce a proper skeleton-looking figure: ankles with knees, knees with hips, hips together, hips with shoulders, shoulders together, shoulders with elbows, elbows with wrists, shoulders with ears, ears with eyes, eyes together, and eyes with nose.

A two-stage operation: human detection then keypoint detection. The model used in this work is composed of two modules. Given an image to analyze, the first module aims at *detecting* the characters depicted on the image, by providing a tight *bounding box* around each of them separately. Then, the image is cropped along each bounding box, and each crop is passed to the second module. The second module aims at *detecting the coordinates of each of the 17 keypoints* of the single character represented in the crop. These coordinates are then transformed back to coordinates relative to the original (not the cropped) image.

Confidence scores for keypoints. The second module always outputs estimated coordinates for the 17 keypoints that compose a human skeleton, even if some of them are not directly visible on the image (occlusion, close-up portrait, ...), in which case the model provides its best approximation while trying to respect human body proportions. To indicate the confidence that the model has in its predictions, it also outputs a confidence score for each keypoint. This way, the

model can indicate when uncertainty arises by providing low confidence scores to keypoints that are presumably not accurately estimated. For the model used in this work, it is generally considered that keypoints with a confidence score above 30% are sufficiently reliably detected.

What if no human is present? In the case of an image where no character is represented, the first module will provide a single bounding box that corresponds to the whole image itself. The second module then still outputs a single set of 17 keypoints, but all of them very likely have a low confidence score. If there are characters on the image but the first module misses them all, then the whole image is once again provided to the second module, which may or may not output keypoints actually belonging to one or several of the previously missed characters, with potentially varying degrees of confidence.

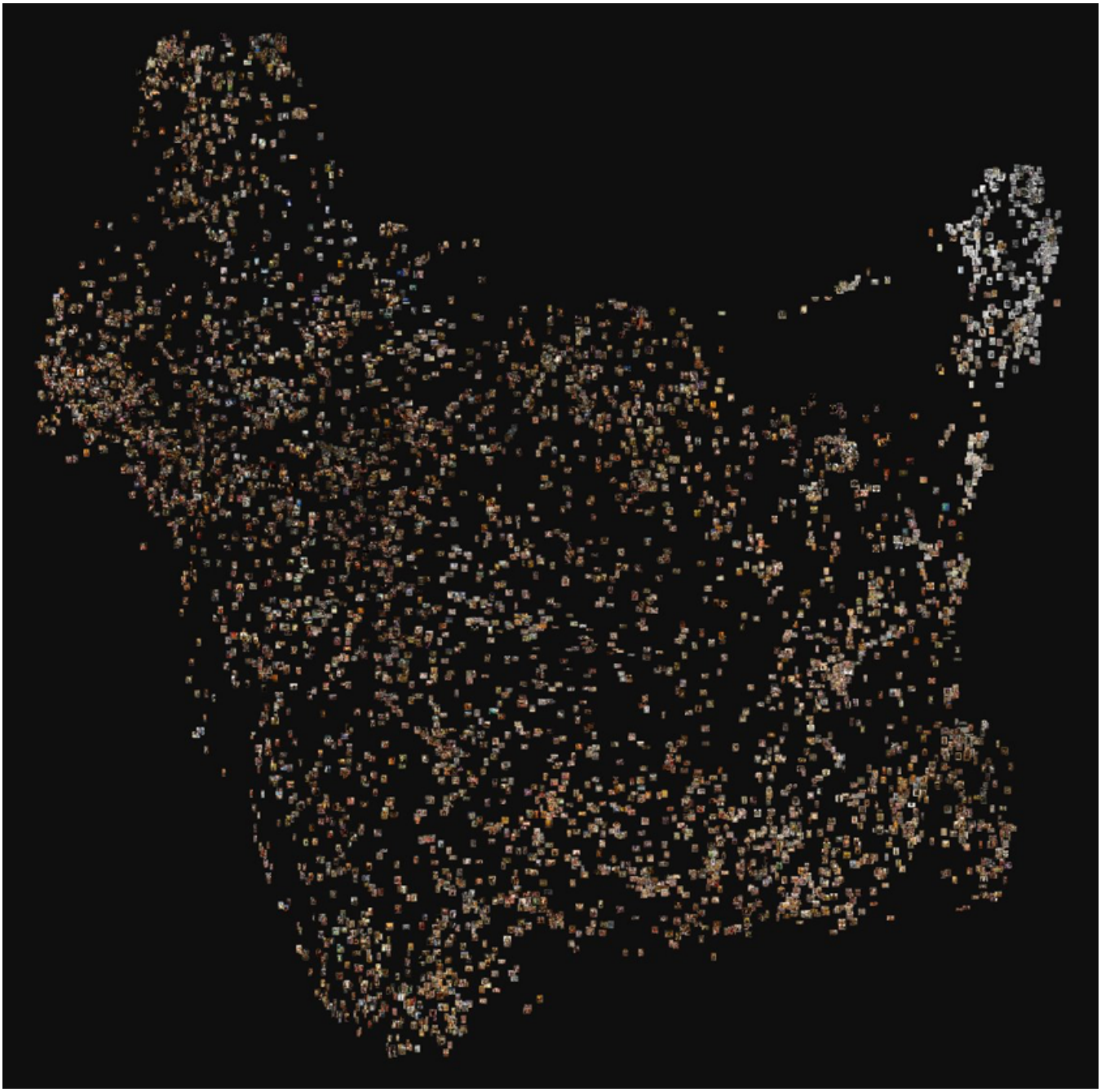
Visualizing corpuses of images: PixPlot

Images analyzed. For this study, in a first phase, we consider images from "religious paintings" of WikiArt. We downloaded the 11,980 images of this category available when we started this work. We ran MMPose on each of them to extract human poses. We found that 5,269 images contain at least one pose whose 17 keypoints have a sufficiently high confidence score, and there are a total of 8,599 such individual poses present on these images. These images and these individual poses now constitute our corpus of interest.

Visualizing large collections of images with PixPlot. In this project, we want to regroup images that display similar features (where "features" and "similar" both need to be defined). We found that a suitable tool for it is PixPlot, whose default functioning is the following. Given a collection of images, PixPlot computes an *embedding* per image (that is, a numerical representation of the image as a list of e.g. 2,048 values) with a popular neural network trained by deep learning on a generic dataset. Then, PixPlot uses the UMAP algorithm to

project all these embeddings in a 2-dimensional plane, by trying to maintain as much as possible the distances between the embeddings, that is, embeddings that are far away (respectively close) initially should remain distant (respectively close) from each other in the plane. Finally, in a web browser, PixPlot places each initial image at its corresponding position in the plane, and the browser's functionalities allow navigating through this large meta-image. Most PixPlot results can be accessed from here, but specific links to specific results will be provided as appropriate.

PixPlot applied on our images: lack of interpretability. Applying the default PixPlot to our corpus of images is interesting but yields hardly interpretable results, as it can be seen here.



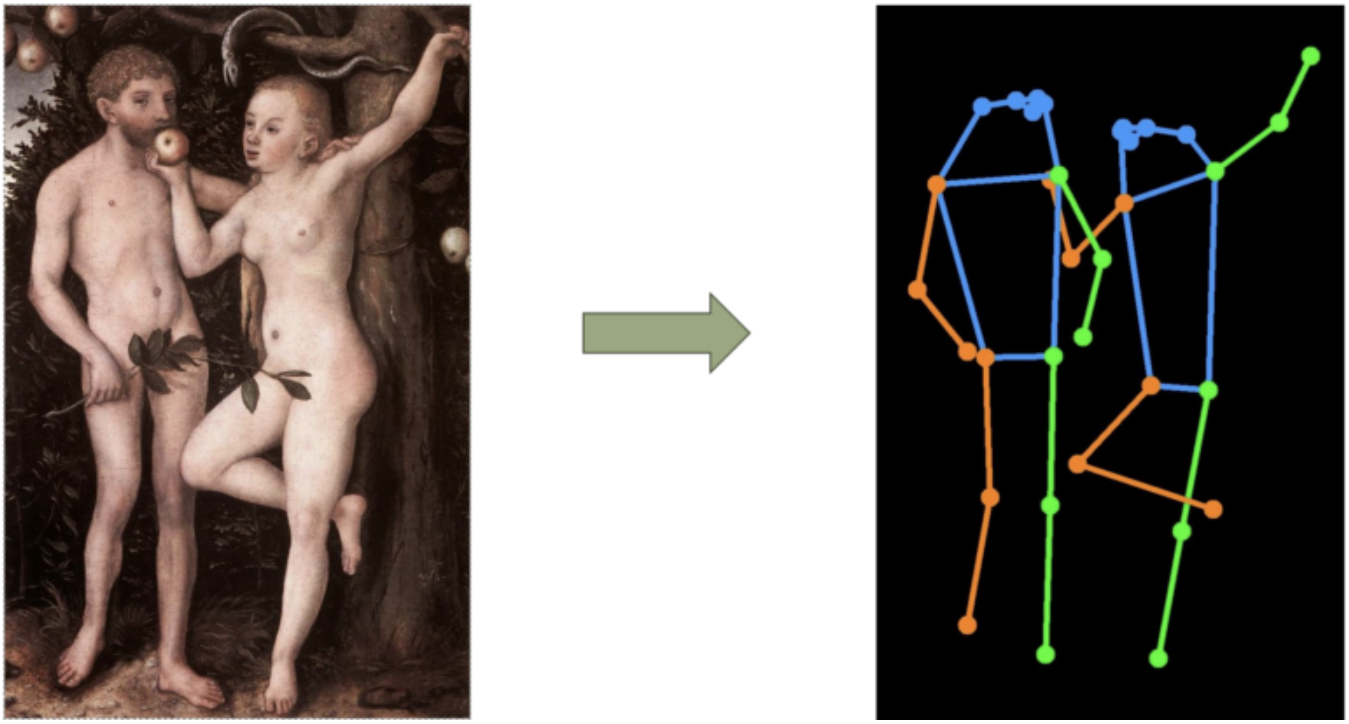
PixPlot visualization of the selected WikiArt corpus of religious paintings. [Full-size image]

Indeed, it is not always possible to "guess" why some images are located close to each other. It might be because of some elements of the content of the images (naked bodies, long clothes,...), the color palette of the images, the presence of particular shapes, or some combinations of multiple factors that are not easy to explain. For example, it might be observed that images of Jesus on his cross are scattered across the meta-image (some of them are clustered though), depending on the other elements on the images.

Example of cluster: naked bodies, and Jesus on Cross. [Full-size image]

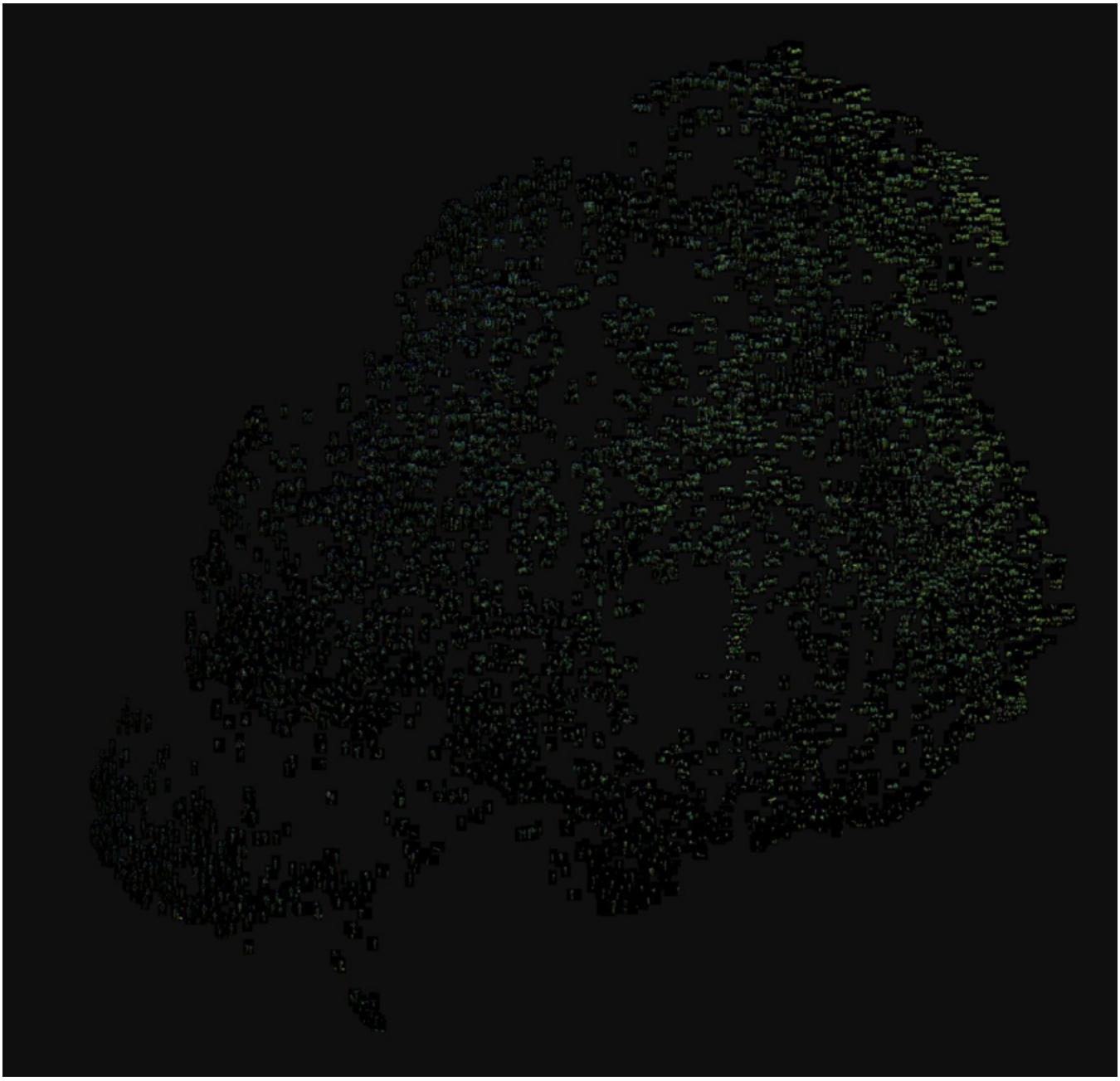
More clusters like this can be visualized on the PixPlot, and on slides 5-15 of this presentation. Given the limitations of this approach, we would like to have the possibility to focus on one modality only. For the time being, this modality will be the poses of the characters.

PixPlot applied on grouped skeleton images: little emphasis on poses. The next step is to transform our images into skeleton images. That is, blacking out the whole images except where characters are present and replace them by their skeletons.



Transforming an original image into its skeleton version. Lucas Cranach the Elder, Adam and Eve, ca. 1538. [Full-size image]

This gives a new set of images, corresponding to the pose version of the initial images. Applying the default PixPlot to this corpus is moderately interesting, because, unsurprisingly, PixPlot can only differentiate images by roughly counting their number of colored pixels, as shown here (it might be necessary to modify the "point size" setting to better see the PixPlot).

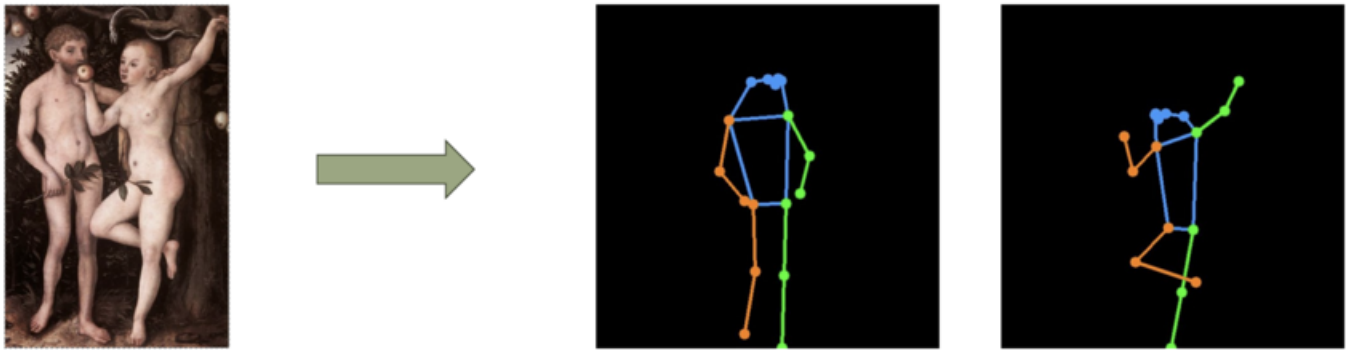


PixPlot visualization of the corpus transformed into skeletons. [Full-size image]

Therefore, images with many characters are clustered, and those with few characters are clustered as well. However, the exact poses, understood as articulations of the skeletons, barely play a role in this representation. Also, the result depends on the choice made for the color representation of the skeletons and the thickness of the lines, which is a useless dependency that does not correspond to any bodily resemblance.

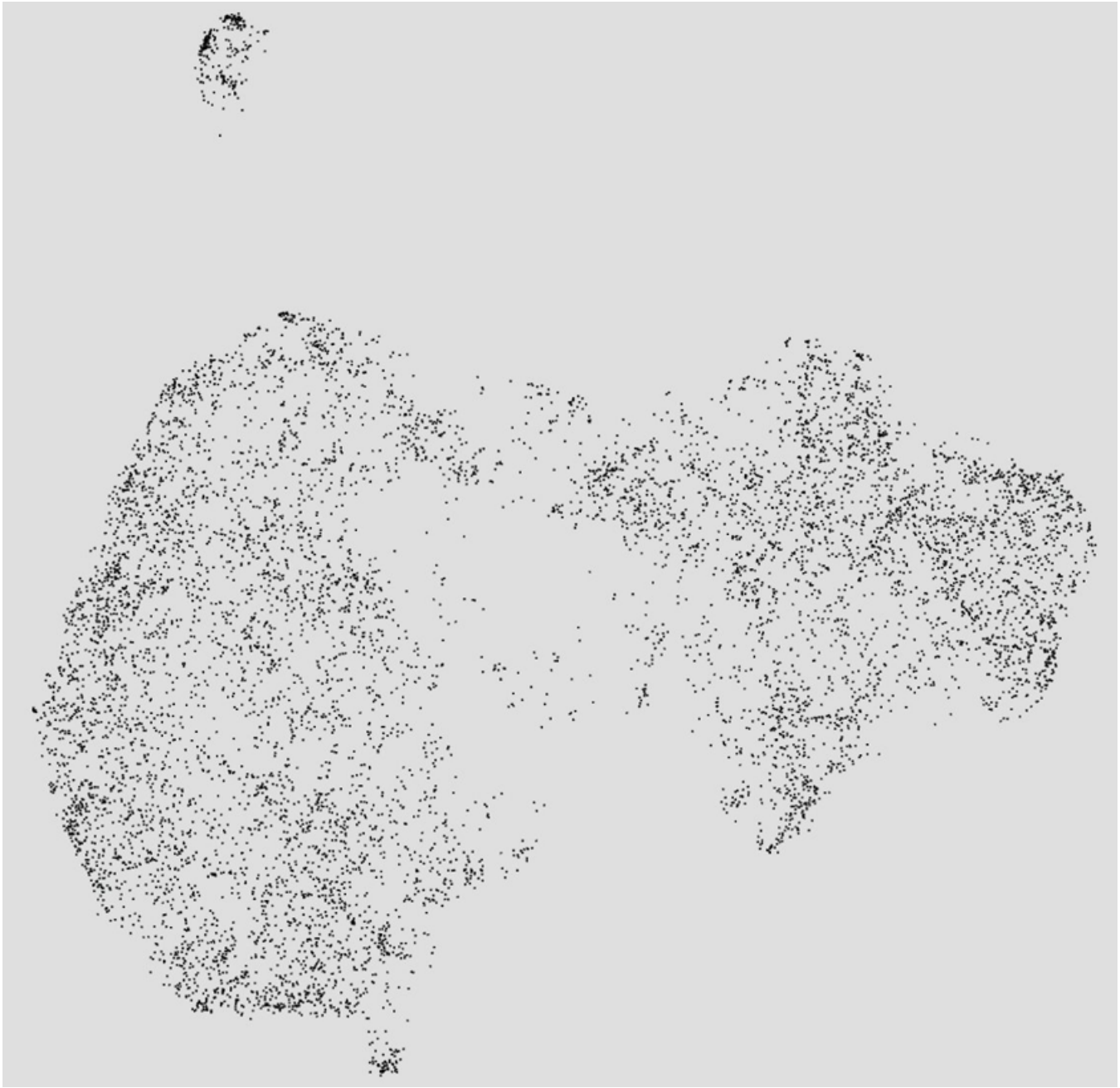
PixPlot applied on individual skeleton images: more emphasis on poses, and

normalization. As a consequence of the previous result, we decided to analyze poses individually. For that purpose, each of the 8,599 individual poses is represented in its own image. Nevertheless, to make poses comparable to each other, we need to get rid of the position of the character in the image, and of its size. Therefore, each skeleton is drawn in a square image of 512×512 pixels, such that the average across the keypoints is located at the center of the image, thus removing the dependency to the position in the image. To be scale-independent, the skeleton is enlarged or compressed such that the largest distance between the center of the image and the keypoints equals 256 pixels. This also forces to inscribe the pose within a circle centered at the center of the image, and of radius 256 pixels.



Transforming original images into separate skeletonized crops for each detected character. Lucas Cranach the Elder, Adam and Eve, ca. 1538. [Full-size image]

This gives yet another corpus of images that can be passed to PixPlot. Applying PixPlot to this corpus yields interesting results that now tend to cluster similar poses together, as shown here.



PixPlot visualization of the single skeleton crops. [Full-size image]

Cluster of Jesus on the Cross. *[Full-size image]*

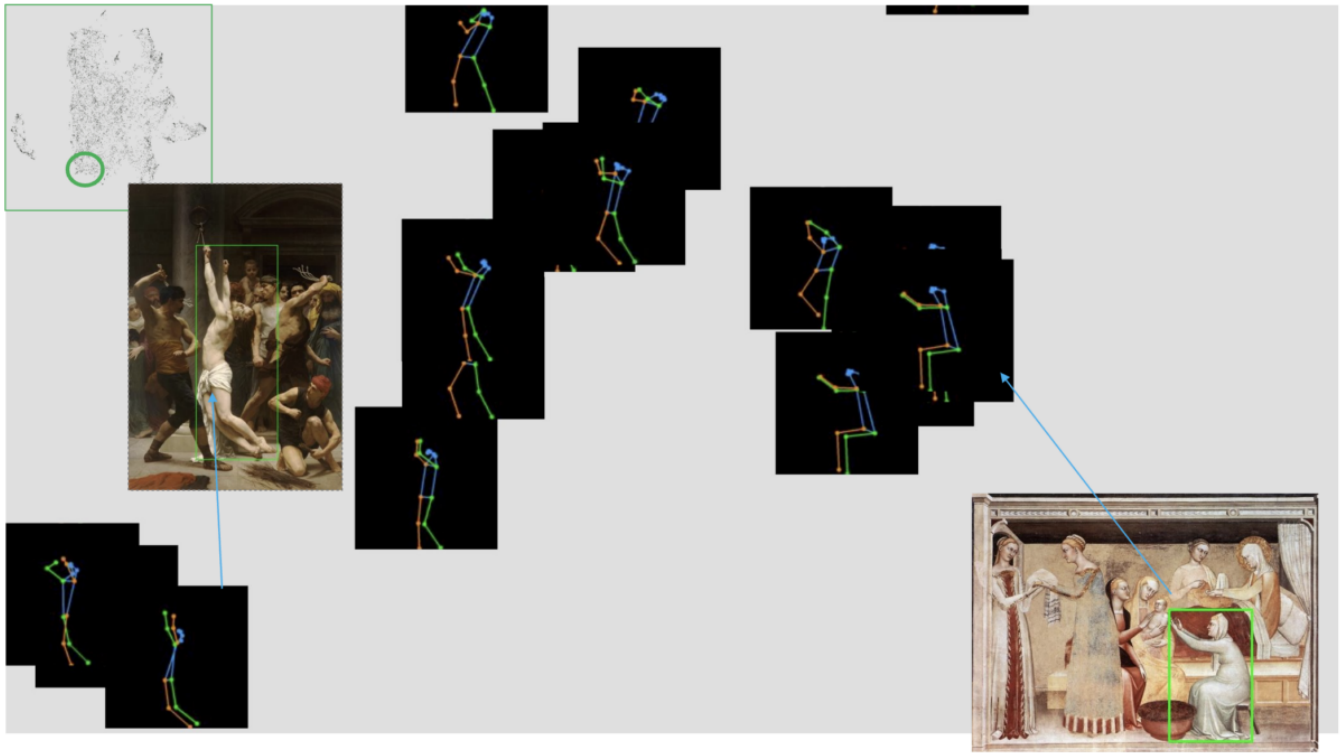
However, most of the time, there is still a lot of variability that can be observed within poses placed close to each other. This might be due to some features or artefacts present in the embeddings, over which we have so far no control. Indeed, these embeddings are still computed with a pretrained neural network, directly applied to our pose images. This means that the keypoints coordinates that we have at our disposal, which constitute a very rich and accurate source of information, are not used explicitly in the current process. This is the change that we will make next.

PixPlot applied on individual skeleton images and custom pose distance: maximum emphasis on poses. The individual pose images are just a visual representation of the keypoints extracted by the model. These images use specific colors and trait thickness that influence how the neural network sees the image and generates the embeddings. So, instead, let us consider the list of keypoints as the embeddings themselves, so that we remove the uninterpretable neural network from the equation. Our embedding dimension is thus 2×17=34 instead of 2,048. Besides, we need to circumvent the default distance metric used by PixPlot to cluster similar images together. The default

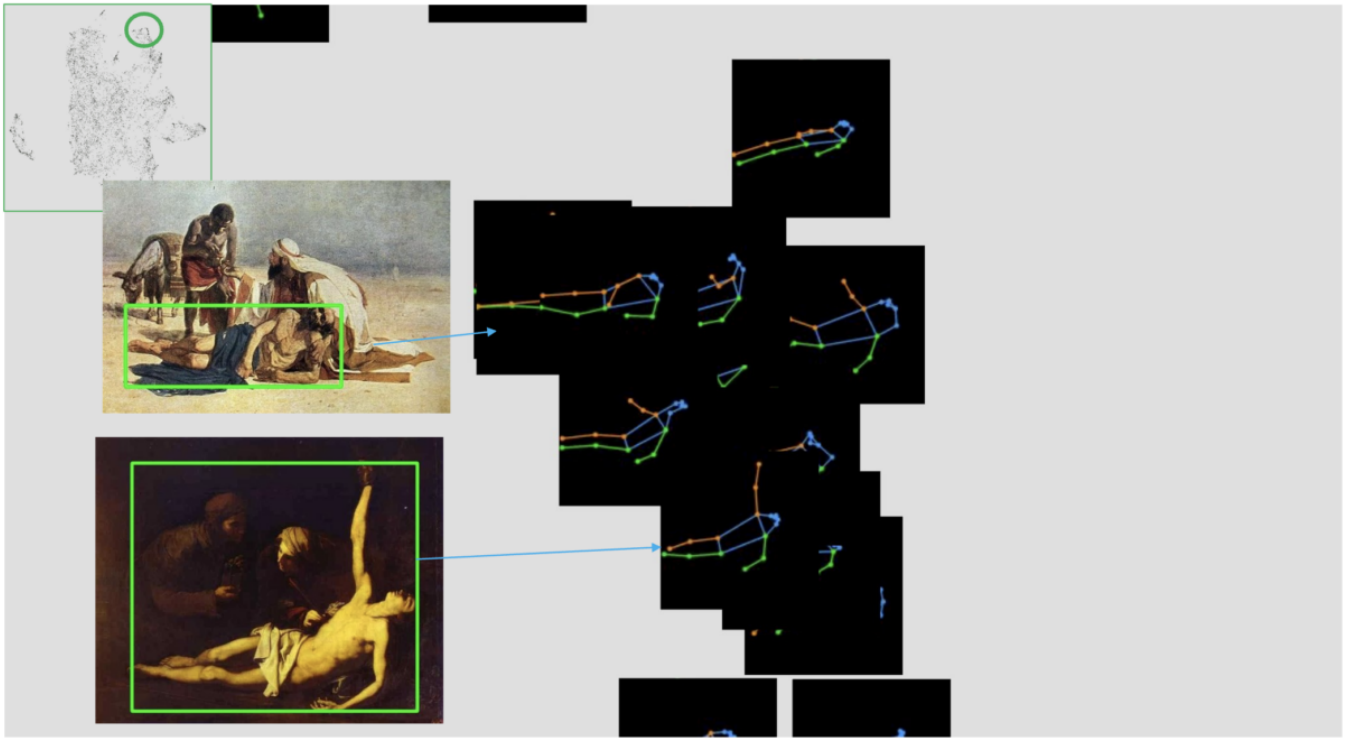
metric, namely the cosine similarity, is well-suited for embeddings that originate from neural networks. In our case, we decided to use a metric directly related to our keypoints. Given two sets of 17 keypoints, belonging to two characters, we compute their distance as the sum of the euclidean distances between the pairs of corresponding keypoints. In other words, we compute the distance between the coordinates of the nose of the first pose and of the second pose, between the coordinates of the left eye of the first pose and of the second pose, etc., and we sum all these distances. Re-wiring PixPlot with these two modifications (keypoints as embeddings and pose distance), gives a new meta-image where the focus is completely on the pose itself, shown here.

PixPlot visualization of the single skeleton crops but with the custom pose-specific comparison distance. [Full-size image]

Analysis of the pose clusters. An in-depth analysis of this meta image reveals interesting clusterings of similar poses, as desired. This is exemplified hereafter, with many more such examples on slides 33-56 of this presentation.



[Full-size image]



Some clusters and associated images in the PixPlot visualization. [Full-size image]

It can be observed that the center of the PixPlot tends to regroup poses that are relatively neutral, depicting a person standing, facing the viewer. As we move away from the center, the poses continuously vary, reaching completely

different poses in the corners of the meta-image, such as characters lying down, sitting, falling, etc. A cluster of representations of Jesus on his cross is also clearly visible, as this pose is common among religious paintings. We can also spot a cluster of images where the character is seen from the back, which is completely, and rightfully, dissociated from the rest of images. Let us also note that a body lying down with the head on the left is a complete different pose (according to the metric used) than if the head is on the right of the image. The opposite poses are also represented in opposite parts of the meta-image, namely top and bottom in this case. Finally, as for every large-scale automated analysis, there are some unfiltered errors that sneaked through this visualization; in this case as a small cluster of poses with legs cut at the knees, corresponding to characters that are not completely shown on the images.

Retrieving images similar to a query image

Similar pose retrieval with respect to a query image. PixPlot is a nice visualization tool, useful for exploring the dataset and figuring out the underlying structure of it, with respect to either generic or specific features. However, in its current state, it does not allow one to submit a query image, which might or might not belong to the corpus, and retrieve the images of the corpus that are the closest to it, again according to generic or specified features. Besides, even for a query image from the corpus, its neighbors in the PixPlot visualization might not be exactly the ones that are the closest in term of computed distance, and their ranking is uncertain, which is not convenient as it might be hard to grasp a sense of which images are the closest in densely populated areas of the meta-image. There is indeed a loss of information when projecting high-dimensional embeddings (2,048 dimensions, or even 34 as in our case) into a 2-dimensional plane, which results in possibly flawed interpretations of what a close neighbor is when looking at the PixPlot image only. As a consequence of these observations, we also developed a tool to define a query image and a pose of interest of a character in this query image, which

outputs the list of closest poses among the corpus. We provide two visualizations of these results: one simply showing the list of images retrieved, one displaying these images on a timeline, based on the metadata available. For example, here is the result for the query image *Bergström over Paris* from Helmut Newton. The poses compared are indicated by green boxes (so the similarity is not at the whole image-level but at the solo pose-level).

Left: query image. Right: unfiltered nearest religious paintings in term of estimated pose. [Full-size image]

Filtering the retrieved images. On the above figure, some retrieved images are not satisfactory. For instance, the first one (Locatelli's painting), which is estimated as the closest religious image to Newton's in term of pose of the main woman, displays a very small character. The next two images barely represent anything and accidentally end up being highly ranked. Only a couple of images are relevant: those where a human body is lying down to the right (which is often the Christ). To remove such unwanted results, we set up 4 filtering parameters (that can be modified at will to generate new analyses) on the box and keypoints features:

1. Valid box threshold: threshold above which the bounding box confidence score of the first module of the model should be to accept the box as a valid candidate. We simply select it as 0, such that images are discarded from the search only when no human character is detected on the images (this will eliminate e.g. the second-ranked image in the above figure).
2. Aspect ratio threshold (size threshold): threshold above which the area of the bounding box relatively to the image size must be to accept the box as a valid candidate. We set it to 0.05 (5%), which means that characters occupying less than 5% of the image are considered too small to be interesting enough in the search for similar poses (this will eliminate e.g.

the top ranked image above).

3. Keypoint confidence threshold: threshold above which the confidence score of the keypoints produced by the second module of the model are considered valid. As done for the PixPlot visualizations, we set this threshold to 0.3.
4. Number of valid keypoints: a pose is valid when it has at least a certain number of valid keypoints. For instance, to allow one uncertain keypoint, we set this number to 16 (this will eliminate e.g. the third-ranked image above).

As a result, the updated list of retrieved images is the following, which is much more satisfying than the previous one. Let us note that we still compare normalized (scale-independent) poses, which means that only the pose itself matters, regardless of its size (provided that is is large enough to fulfill the second filtering).

Left: query image. Right: filtered nearest religious paintings in term of scale-independent estimated pose. [Full-size image]

If we do not normalize the poses and want to keep its initial relative size with respect to the image into account, we obtain the following ranking. The top result of the previous figure is not displayed anymore, but new images that might not be relevant are now well-ranked, such as Van der Meyden's *Madonna and Child*, because the Child's occupancy of the space is very similar to Bergstrom's, and the poses are somewhat "similar".

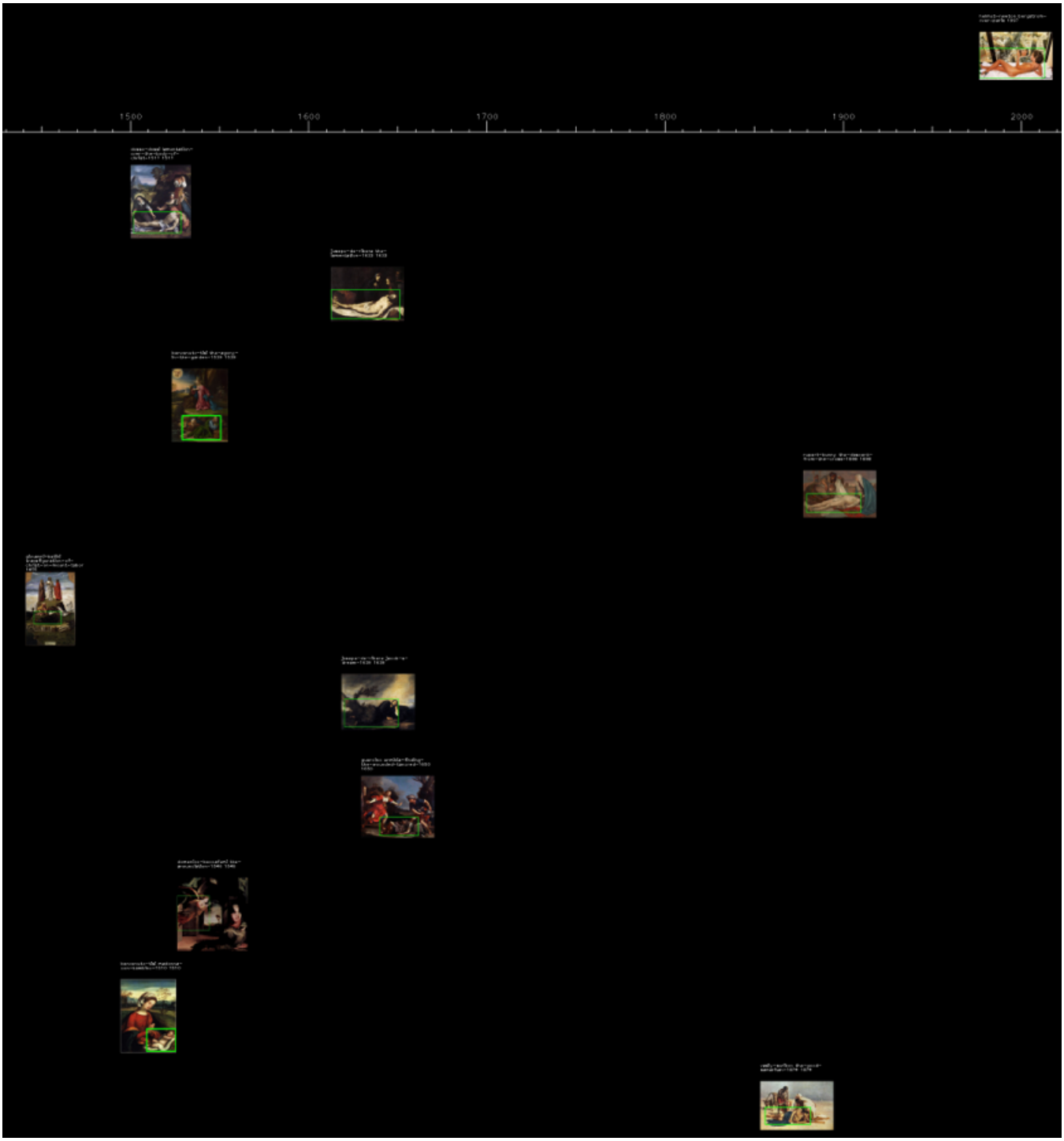
Left: query image. Right: filtered nearest religious paintings in term of scale-dependent estimated pose. [Full-size image]

As each image of the retrieval corpus has two rankings, one for the scale-independent retrieval, the other for the scale-dependent one, we can also average those rankings to obtain a balanced ranking, that takes the scale of the pose into account but that still leaves room for poses of very different sizes if they are similar enough to the query pose, as shown below. This might well be the most relevant ranking of all, in this case. This idea to combine rankings is further developed in the next sections.

Left: query image. Right: filtered nearest religious paintings in term of average combination of scale-dependent and scale-independent rankings. [Full-size image]

Let us note that, in none of these analyses, the position of the pose within the image was taken into account. In other words, the poses are compared as if they were all centered, even if they are actually located in different parts of their respective images. We noticed that taking the localisation into account may worsen the rankings by favoring too much poses that are located almost at the same spot as the query pose while completely disregarding the pose itself, which could be a person standing, sitting, praying,... Hence, our rankings can be qualified as translation-invariant.

As mentioned, we can also provide a timeline representation of a ranking. In the following figure, top-ranked images are closer to the time arrow, so they are ordered by ranking from top to bottom, for the given query image (located above the time arrow).



Timeline representation of the previous ranking. The image above the time arrow is the query image. The images below are ranked such that the best images are closer to the time arrow. [Full-size image]

Finally, it is well-known that *Bergström over Paris* is inspired by Velasquez's *The Rokeby Venus*, which is a mythological painting. Therefore, looking for similar images in religious painting might not be an appropriate choice. When looking after WikiArt's mythological paintings (whose PixPlot solo skeletonized visualization can be found here), we have the following result (after filtering

and combination of scale-dependent and independent rankings).

Left: query image. Right: filtered nearest mythological paintings in term of average combination of scale-dependent and scale-independent rankings. [Full-size image]

We can see that Velasquez's painting appears first in the ranking, and that the retrieved images better align with the query image in term of pose.

From individual poses to groups of poses. When comparing images depicting multiple characters, it might be interesting to take into account all the poses together, rather than individual poses, such that images are considered close to each other when human bodies form the same shapes globally on the image. In the same spirit, in some cases, some characters are just part of the crowd and do not play any specific role, thus might not need to be considered in this group-level analysis, and we might want to focus only on a sub-group of characters. Therefore, we also worked on the comparison of groups of poses, which presents some extra difficulties, as explained hereafter.

Combinatorial explosion. The first obstacle, not the least, is the combinatorial explosion that arises when many characters are represented on a painting. Indeed, if N characters (at least two) are detected, then the total number S of different subgroups of at least two characters that can be formed from these N detections is $S=2^N-N-1$. While this remains manageable for small values of N, e.g. for $N=2$ then $S=1$, for $N=3$ then $S=4$, for $N=4$ then $S=11$, this number grows exponentially with N (it roughly doubles for each extra character detected). Thus, with values of N that are not so large and not so rare in paintings, for example $N=10$ characters, we can form $S=1{,}013$ distinct subgroups of at least two characters, most of them being presumably not so much interesting to examine. For that purpose, in the following, when an image has more than 3

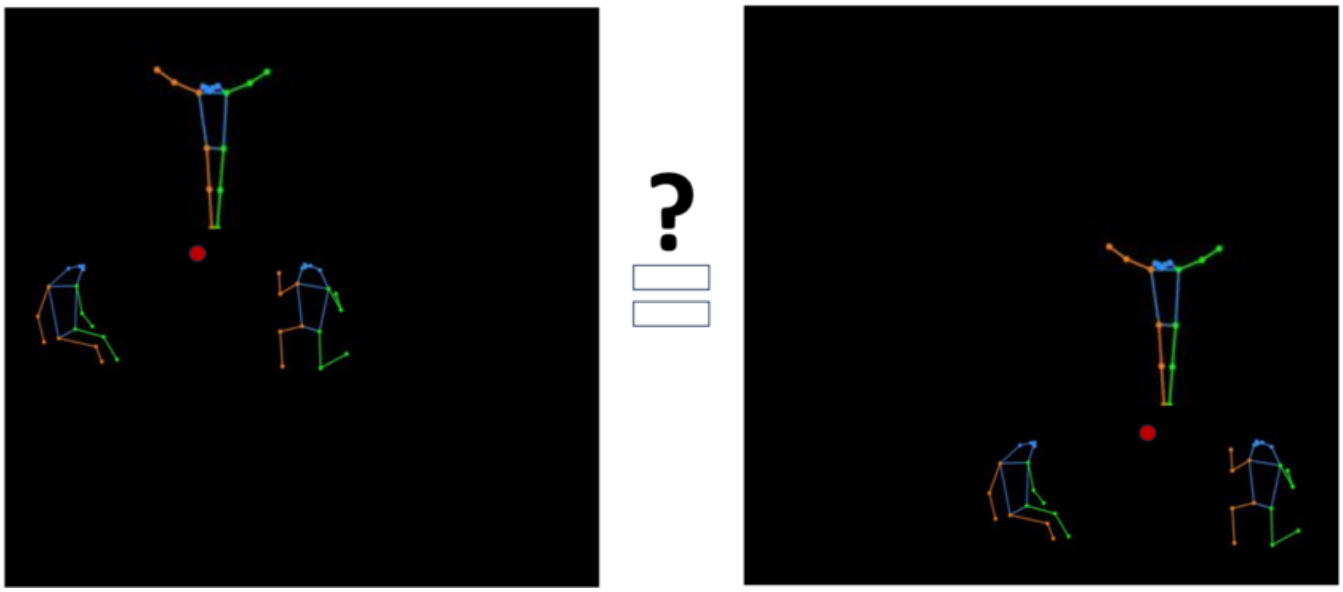
characters, we only keep the 3 most confidently detected ones.

Various ways to compare two groups of poses. Another difficulty that arises when comparing multiple poses is answering the question: what do we compare? First, we need to compare groups (or subgroups, but we will write groups in the following) with the same number of characters. So, each image of interest yields groups of poses, and we can analyze altogether all the groups of two characters, three, four, etc., but we cannot compare a group of two characters with a group of three. So, let us assume that we have two groups of characters from different paintings to compare. How do we compare them? We identified 7 criteria that allow each a unique way of comparing the poses. In detail, 3 of them focus on the configuration of the group of poses: the poses are reduced to their average point, these average points give the configuration of the group of poses. We analyze these configurations by looking at (1) its average relative position in the image, (2) its shape (e.g. triangle pointing upwards) independently of its scale, that is, for example, an upwards equilateral triangle of side 2 and another one of side 5 are both upwards equilateral triangles and are thus equivalent for the analysis, (3) its shape dependently of its scale, such that the two upwards equilateral triangles are not considered equivalent anymore. The next 4 criteria focus on the poses themselves and not on the configuration of the whole group of poses. Again, we can either keep (4)(5) or remove (6)(7) the dependency to the scale of the poses, by letting them as is or normalizing them, just as discussed for the configuration of the group of poses. Besides, when comparing two groups of equally-numbered poses, we must decide which pose of the first group is compared with which pose of the second group. The matching between the poses of the two groups can be done either by matching the poses by their appearance (which pose of the second group is the most similar, as in the individual poses analysis, to which pose in the first group?), or by their localisation (which one of the second group is the closest to which one of the first group in term of relative position in the image?). We name the first matching "best pose matching" (4)(6) and the second one "location-based matching" (5)(7). Hence, our 4 pose-based criteria for

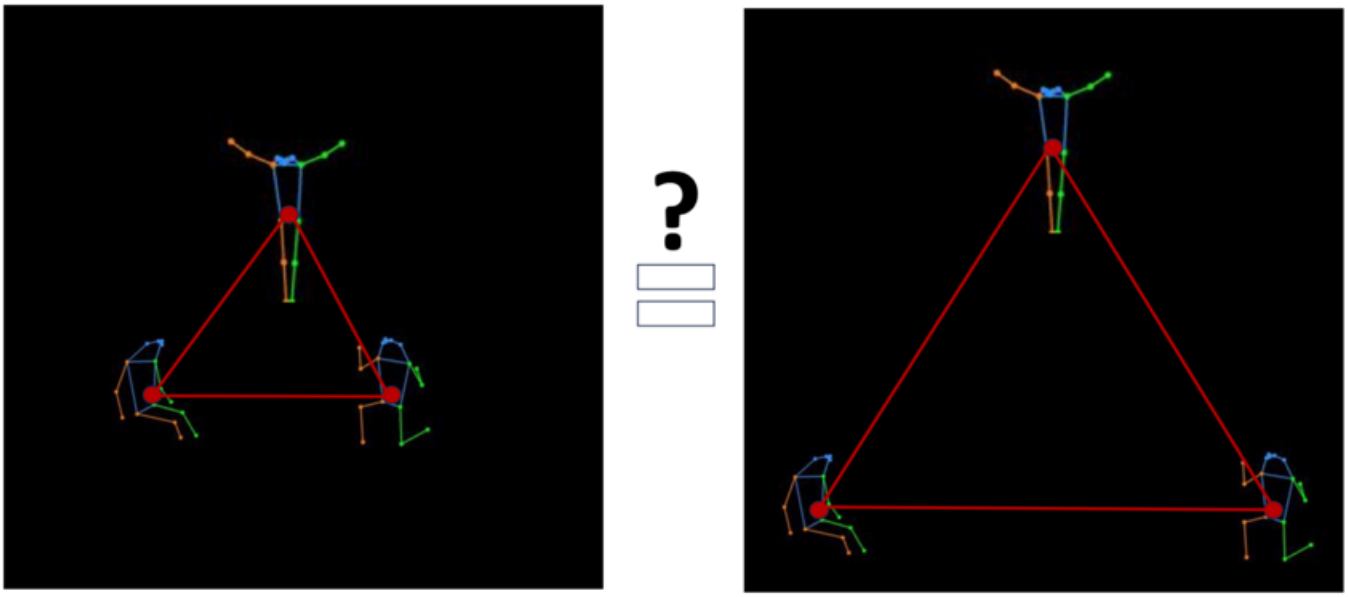
comparing groups of poses are defined by the 2×2 combinations of scale dependence/independence x best pose matching/location-based matching. In summary, the 7 criteria available are:

1. Configuration comparison: localisation of the group of poses within the image
2. Configuration comparison: shape of the group of poses, scale-dependent
3. Configuration comparison: shape of the group of poses, scale-independent
4. Poses comparison: scale-dependent, best pose matching
5. Poses comparison: scale-dependent, location-based matching
6. Poses comparison: scale-independent, best pose matching
7. Poses comparison: scale-independent, location-based matching

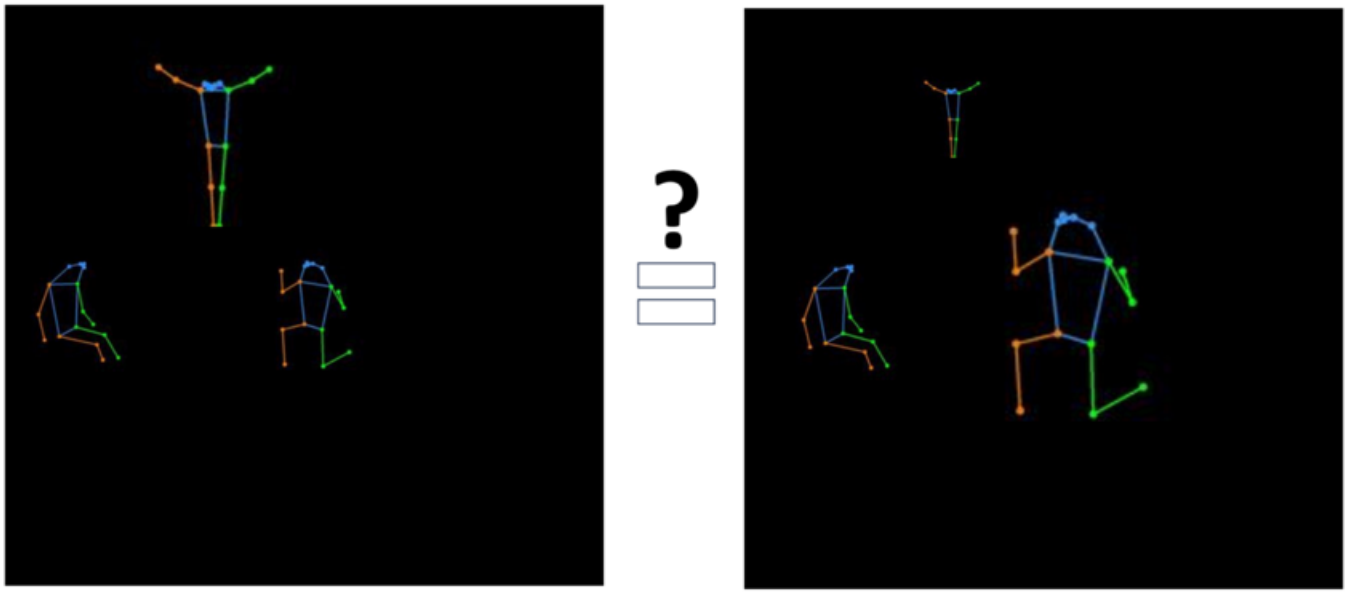
These choices are embodied in the following figures.



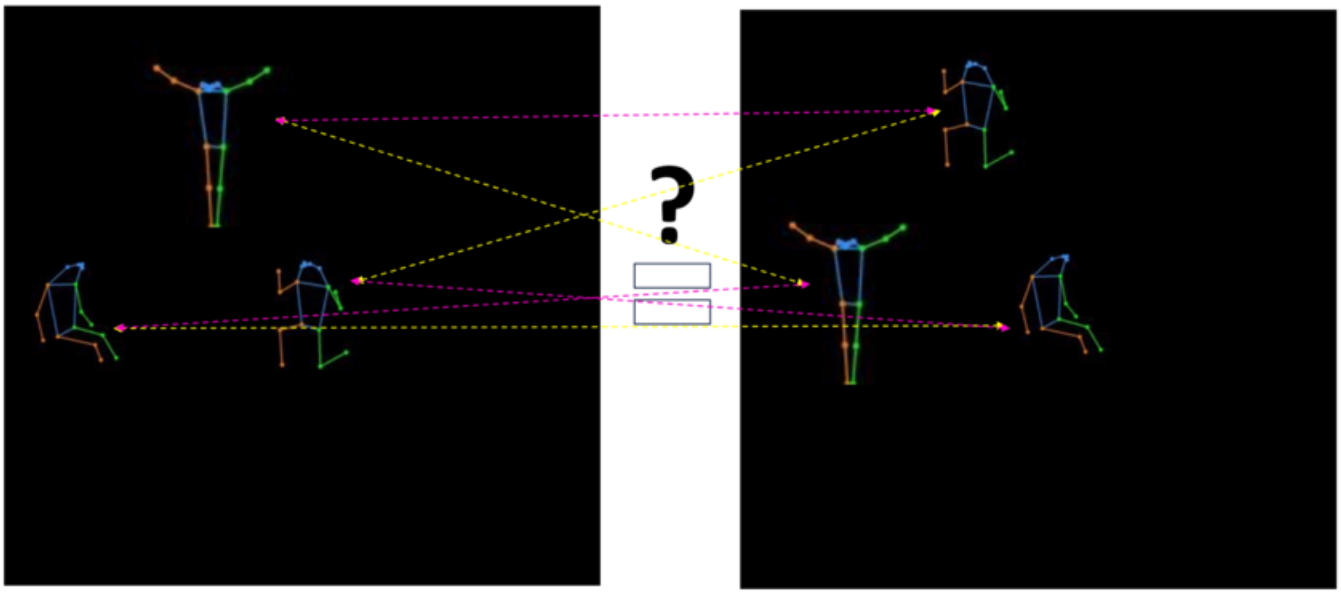
1. Configuration comparison: should we take into account the localisation (red dot) of the group of poses within the image? [Full-size image]



2. and 3. Configuration comparison: should we take into account (2.) or not (3.) the scale of the shape formed by the group of poses (in this case the red triangle)? [Full-size image]



4.,5.,6., 7. Poses comparison: should we take into account (4.,5.) or not (6.,7.) the scale of the individual poses? [Full-size image]

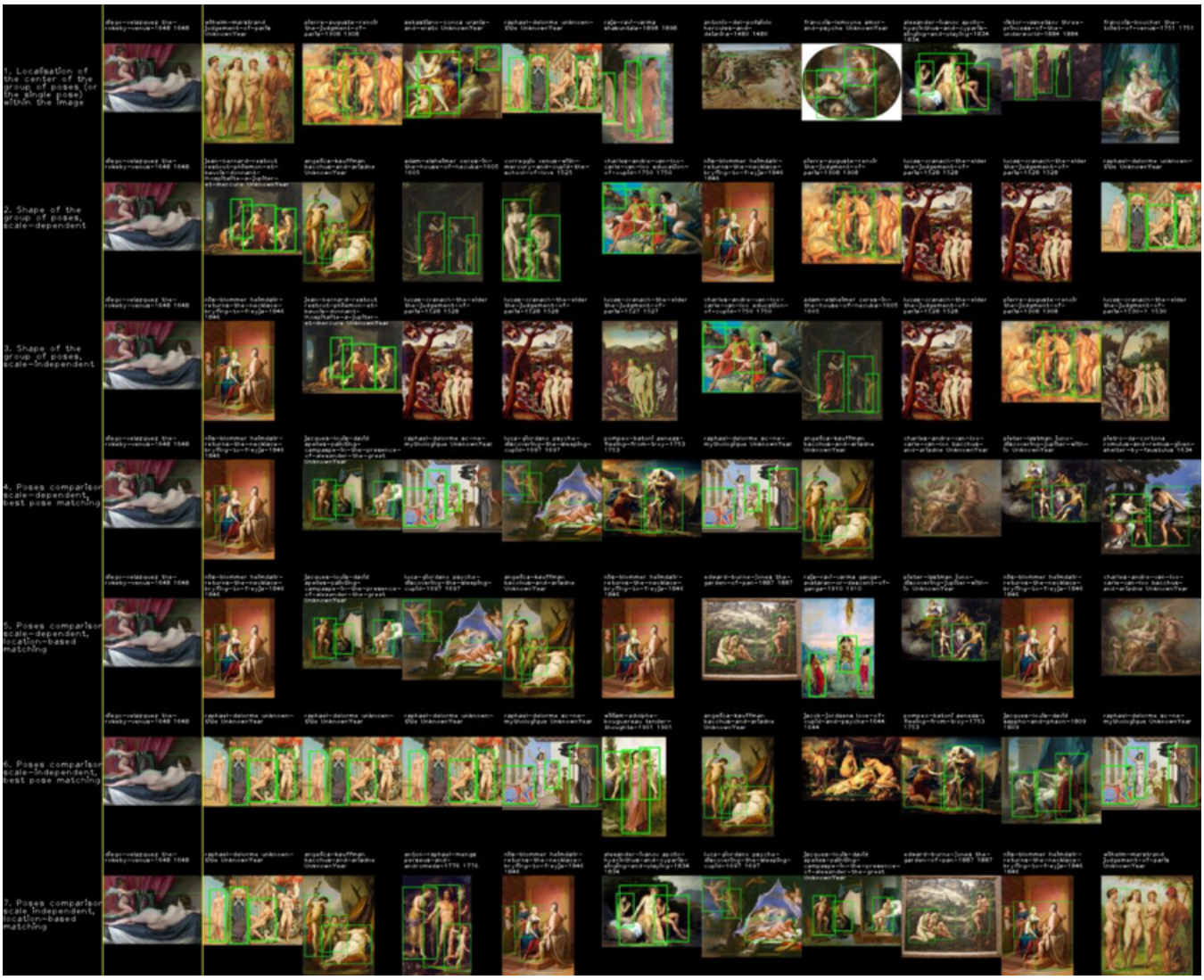


4.,5.,6.,7. Poses comparison: should we compare poses that are the most similar (best pose matching, yellow arrows, 4.,6.) or that are located at the most similar place in the group configuration (location-based matching, pink arrows, 5.,7.). *[Full-size image]*

Let us note that this naturally extends to the case of individual poses comparisons, but criteria 2 and 3 then become irrelevant since there is no such thing as "shape of the group of poses", and the matching does not matter anymore since there is only one pose to compare with another one, thus criteria 4 and 5 are equivalent, as well as 6 and 7. Thus, one needs to compute only criteria 1, 4, 6 in the case of individual poses, as already discussed and illustrated in our results on *Bergström over Paris*.

Example on *The Rokeby Venus*. As soon as at least two poses are considered, the 7 types of rankings can be made. For example, let us consider as query image Velasquez's *The Rokeby Venus*, and let us compare it with WikiArt's mythological paintings (of course we exclude the query image from the retrieval corpus). We detect 3 characters on the image: the Venus, the Angel, and the reflection of the Venus in the mirror, which is in reality not another character but the model is not trained to discriminate it, so technically it considers it as a character on its own. Let us first list the rankings that can be computed. We can consider individual query poses, thus 3 of them, and for each, we can compute a ranking for criteria 1, 4, 6, which makes already 9 rankings. Then, we can consider the 3 pairs of poses: Venus-Angel, Venus-

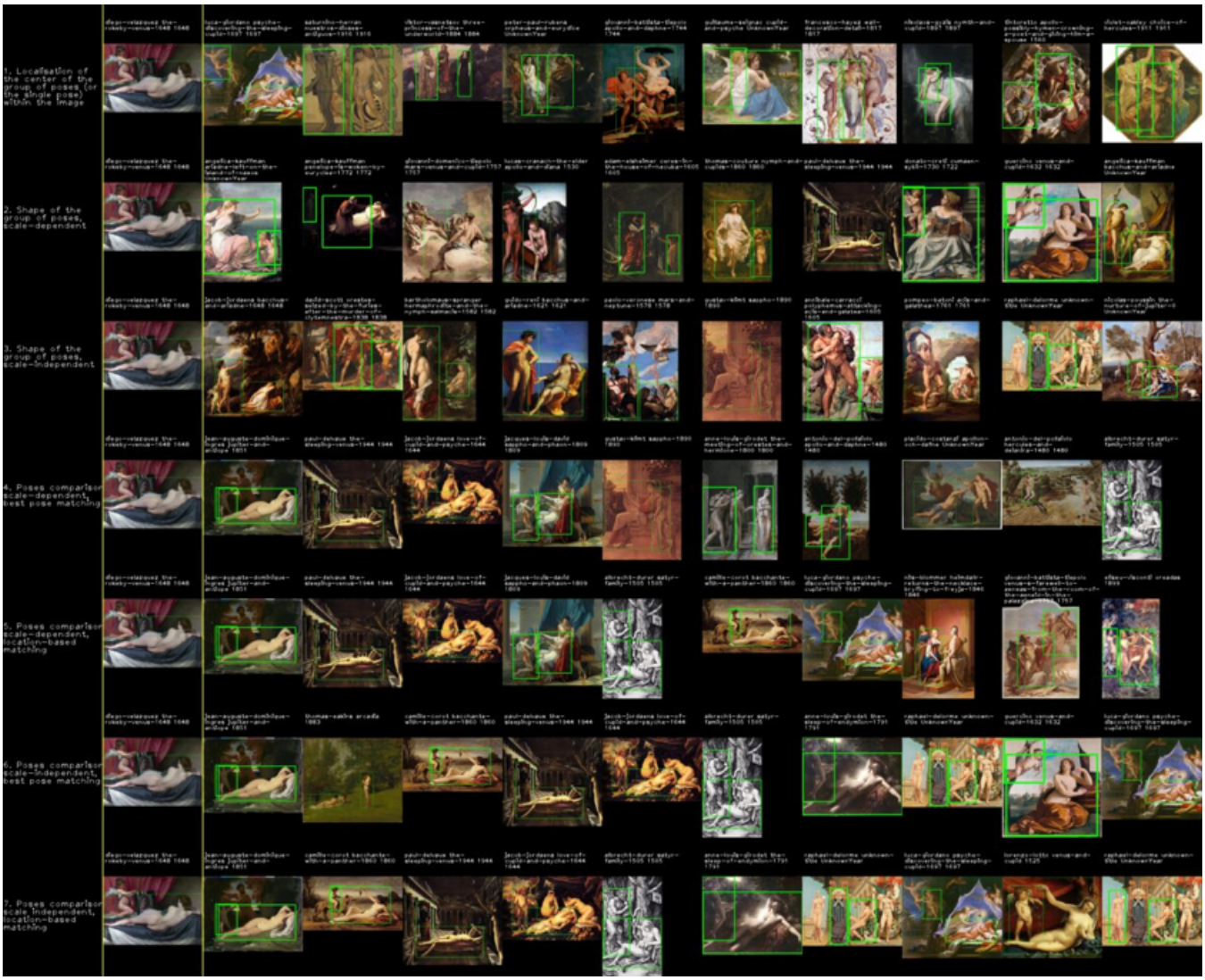
reflection, Angel-reflection. For each of them, we can compute the 7 rankings, which makes 21 additional rankings. Finally, we can consider the whole group of the 3 poses, for which we can compute 7 extra rankings. In total, we can thus compute 9+21+7 = 37 rankings for this image only (which contains only 3 detected characters), without even combining those rankings (the combinations give extra rankings, possibly in infinite amount as we could give any weight to each ranking). All these (filtered yet uncombined) rankings can be found here. The 7 rankings for the whole group of poses are represented in the following image.



The 7 rankings for the comparisons of the whole group of 3 poses of the query image. [Full-size image]

This already gives a hint on the difficulty to retrieve very similar images when many characters are present: the combinatorial explosion (37 ranking) and the

choice of what to rank give plethora of results, that need to be further refined. In this particular case, it might also be better not to consider the reflection in the mirror. Thus, keeping only the Venus and the Angel, we have the following figure.



The 7 rankings when keeping only the Venus and the Angel. [Full-size image]

Still, we should suggest a way to combine these rankings to compress all that information in a unified view.

Note: Visualization in PixPlot. Technically speaking, we can compute a PixPlot visualization as previously for each set of images containing a subgroup of any given number of poses, for each of the 7 criteria. That is, there can be 7 PixPlots for the set of images focusing on subgroups of 2 characters, then 7 again for the

set of images focusing on subgroups of 3 characters, 7 again for 4 characters, etc. This becomes quickly hard to track and analyze, albeit being technically feasible, if the number of images remains limited (as explained before, a combinatorial explosion lurks and makes a complete analysis challenging). We did not compute those PixPlots.

Combining criteria: not with distances. For a given number of characters of interest in the subgroups of poses, it might be worth combining the 7 criteria into one single summarizing criterion that measures some kind of "global distance" between groups. This combination can hardly be done at the distances-level, as we might risk comparing apples and oranges: the distance between configurations of poses is not at the same scale as the distance between poses themselves. Therefore, visualizing the results of combining modalities in PixPlot will not be possible, as the program needs to be able to compute pairwise distances, whatever the distance notion is, provided that it is well-defined and can be computed from two instances mathematically. This is not a huge issue, as PixPlot is mainly a visualization/exploration tool, as explained, and that specific computations and queries/retrievals are made analytically anyway. Moreover, it goes without saying that, generally speaking, the bigger the group of characters considered, the larger the variability in the representations, which means that finding very similar images becomes increasingly difficult, if not irrelevant. On the PixPlot visualizations, this materializes by scattered meta-images, where it becomes tricky to determine clusters (if there are any clusters at all).

Combining criteria with image rankings for query/retrieval. There is a way to mitigate the issue mentioned previously about combining criteria to measure how two images might be considered similar. Let us again consider the task of having a query image, with some number of characters detected, and retrieving its most similar-looking images, in terms of groups of poses. Each image in the dataset can be compared with the query image according to each of the 7 criteria, and thus has a ranking among the images of the dataset with respect to

these criteria. Contrary to distances, which were assimilated to apples and oranges in the previous section, rankings can be combined, as they are all “rankings” in their respective categories, thus they have the same range and scale. Consequently, we can aggregate the rankings of each image in the dataset to obtain, for each such image, a unique “score”, representing some kind of average ranking, with respect to the query image. Finally, we can output the images that are the closest to the query image based on this aggregated metric.

How to aggregate rankings? Each criteria can be assigned a weight, and the aggregation of the 7 rankings of an image is simply a weighted sum of its rankings. While many choices can be made, it seemed to us that, in the case of individual poses, a combination of equal weights of criteria 4 and 6 (pose comparisons, with scale-dependence and independence, equivalent to 5 and 7) is appropriate, as already shown with *Bergström over Paris*. This prevents putting too much emphasis on poses that look very different but have the same scale as the query image (which acts thus in its favor in the ranking while being not relevant) as well as putting too much emphasis on poses that look very similar to the query pose but are way too different in term of scale/prominence in the image (which may thus convey a different interpretative meaning). Besides, criterion 1 (the position of the pose in the image) does not seem that useful to us, as we believe desirable to consider as similar poses that are the same, with the same scale, but at different positions in the image. In the case of multiple poses, the same motivations regarding the scale of the poses lead us to consider criteria 5 and 7, with a location-based matching preferred over a pose-based matching. Indeed, we believe that, if the same poses are permuted within the pose configuration, this yields a different image, which is not captured by the pose-based matching. We also consider criteria 2 and 3 in the mix, to incorporate the shape aspect of the configuration in the final comparison, with both of them equally weighted for similar reasons as those motivating the choice of 5 and 7. We neglect criterion 1 again for the same reason as previously.

For *The Rokeby Venus*, this aggregation for the group of 3 poses (Venus, Angel, reflection of Venus), gives the following image.

Ranking for the group of 3 poses after the combination of rankings 2,3,5,7. [Full-size image]

For the pair Venus-Angel, the combination gives the following image.

Ranking for the pair Venus-Angel after the combination of rankings 2,3,5,7. [Full-size image]

For the Venus alone, we have unsurprisingly a similar ranking as for *Bergström over Paris*.

Ranking for the Venus alone, after combination of rankings 5,7 (equivalent to 4,6). [Full-size image]

For the sake of completeness, for the Angel alone, we have the following ranking.

Ranking for the Angel alone, after combination of rankings 5,7 (equivalent to 4,6). [Full-size image]

It looks that most of these rankings make sense to some extent, and that the

proposed methodology could prove helpful for extensive analyses of large image corpuses.

Future works? Overall, which retrieved image is the "most similar" to the query image? Well, that depends on which poses the viewer wants to focus on. We could further imagine another combination of these aggregated rankings, putting more weights to more prominent characters, based on the space occupied in the image. Still, these remain developments to be made, if necessary at all. Besides, it might be interesting to try to run an automatic analysis that compares two databases and finds the most similar-looking images, without the need for human intervention. This would be time-consuming (but presumably doable), and another difficulty would be to avoid receiving useless (yet strong) similarities, such as side characters simply standing. To that end, it might be possible to filter out uninteresting poses based on the PixPlot visualization: those close to the standard standing skeleton could be avoided.

Extra. In a first attempt to compare modern fashion photography and religious paintings in terms of poses, we downloaded Artsy's fashion photography catalog and produced a PixPlot of its poses, which can be visualized here. For an easier comparison, we produced a PixPlot combining Artsy's poses and WikiArt's religious paintings poses here. The user can toggle between both corpuses with the pre-established clusters on the left. Overall, it seems that Artsy's poses are more present at the edge of the PixPlot, with more acrobatic/artistic/extreme poses than regular standing characters, more characteristic of religious paintings and more present in the center of the PixPlot. Another preliminary interesting result is the presence of a couple of Artsy images in the cluster of Jesus on his Cross produced by religious paintings, at the very top of the representation:

Herb Ritts, Tatjana-Metamorphosis 2, Joshua Tree, 1988. *[Full-size image]*

Quil Lemons, Quiladelphia, 2023. *[Full-size image]*

Our previous analyses showed how to retrieve similar images to a query image (e.g. *Bergström over Paris*) in a corpus of images (e.g. WikiArt’s religious or mythological paintings). This result shows how we could search for a specific pose (e.g. Jesus on his Cross-like pose) in another corpus of images (e.g. Artsy’s fashion photography). This kind of analysis would be difficult without our tools, as there are more than 12,000 fashion photography Artsy images that would need to be browsed (and just 5000 complete poses). Computer vision really helps finding needles in haystacks!

AdrienDeliege / 12/01/2024