NBB's Extension : morphing sequence between images using cross-domain correspondence

Tal Hadar, Matan Richker, and Jhonathan Hadas

GAN-Workshop, Raymond Beverly Sackler Faculty of Exact Sciences , Tel Aviv University

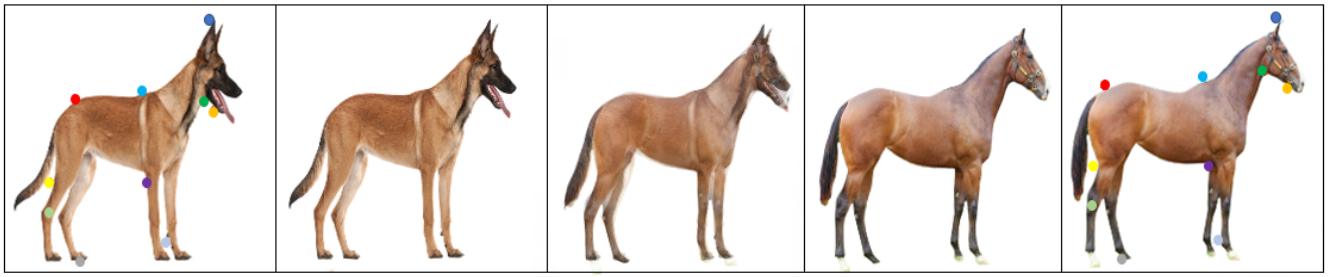


Fig. 1. Cross domain image morphing example: Top 10 Neural Best-Buddies for two cross-domain images(colored in corresponding unique colors). The central image represents the hybrid creature created by the morphing process between the images.

Image Morphing is a special effect that changes (or morphs) one image or shape into another through a seamless transition. Traditionally such a depiction would be achieved through cross-fading techniques on film. Since the early 1990s, computer techniques often produced more convincing results began to be widely used. These involved distorting one image at the same time that it faded into another through manually marking corresponding points and vectors on the "before" and "after" images used in the morph. Until this day, the use of morphing was mainly done for morphing of images from the same domain, a problem that was not directly studied is the morphing problem of images from differentiated domains(cross domain image morphing). The solution to this problem contains an essential part: finding correspondence points between two images from different domain.

Cross Domain Image Morphing | NBB's | Image Correspondence | Image Distance | Patch distance | Image Ordering

Correspondence: ***talhadar1@mail.tau.ac.il , matanrichker@mail.tau.ac.il jonatanhadas@mail.tau.ac.il***

Introduction. When you try to perform morphing from images A to B that are from the same domain (e.g.two human faces of different people), our first task is to find pairs of matching points such as left eye in image A and left eye in image B, nose in both images, chin and so on . The second part is to preform seamless transition between these two images. Till today most of the image morphing is primarily done for the purpose of morphing between images from the same domain (Person A to Person B / Cat A to Cat B). Our goal is to create a tool that will allow image morphing between set of images form distinct domains and at the same time to create the most seamless transition between the source image and the target image. To preform this task we will create a measure of the distance between two images based on pairs of appropriate matching points, then using this distance to arrange the images in sequence so that selecting source and target images will allow the arrangement of the subset of the images in logical sequence from the closest to the farthest and finally to create video that morph logically between source to target.

Related Work. Our work relies on a paper named Neural Best Buddies: Sparse Cross-Domain Correspondence by Kfir Aberman et al.[2018]. This paper deals with a fundamental images correspondence problem in computer vision and gave an effective method to extract correspondence between cross domain objects with sparsity. It uses convolutional neural networks(CNNs) to extract deep features and analyze them further with the nearest neighborhood. Earlier feature matching methods like SIFT algorithm by David G. Lowe[2004] deals with images of the same class at different morphological viewpoints with dense correspondence. Kfir's algorithm focuses on cross domain objects with 'meaningful' feature matching points at different shapes and appearance.
The authors of the paper overcome three major challenges:

1. Cross Domain Correspondence: In general, how can we map a wing of an airplane to wing of a flying bird or human ears to rabbit ear. This is a fundamental problem. This could be a challenging task even for human. The basic assumption here is that the objects contains at least some semantically related parts of geometrically similar regions, otherwise the correspondence task cannot be considered well defined (see bird vs airplane fig from the original paper). They introduced a concept called neural best buddies where each pair of neurons between objects have same activation value(correspondence) only under the assumption of semantic similarity.

2. Finding best matches: Every feature obtained in imageA may not have corresponding feature in imageB like there is no inevitable meaning to map lion's tail to somewhere in a human picture. Thus dense correspondence may not be a good approach for cross domain correspondence. The authors approach is, since we cannot map every feature between images as the cross domain images have drastic difference in shape, pose and viewpoint so it may not be possible all the time to densely map between them, hence, this method only maps features that are 'meaningful' (sparse) in both objects using CNNs and KNN.

3. Lower layers of CNNs are highly variant to color and appearance. Since cross domain objects in images never appear same and their color may not be same always. This is highly considerable for cross domain. As CNNs lower layers are variant to color and appearance, the authors took outputs of deeper layers of CNNs as a 'kind of ground truth' that represents the whole object. The benefit of using deeper layers are high invariant to color, pose and location of the object. Applying **Style Transfer**, the authors wants to use style transfer algorithm of images to compensate visual differences at lower layers of CNNs. They do style transfer at local region of interest for performance consideration. This method uses a kind of neural style transfer algorithm by normalizing feature statistics that says the style is mainly contained in the mean and standard deviation of deep feature channels (eq-4 in paper).

Neural Best Buddies: Neural Best Buddies(NBB) are pairs of neurons' activations that are mutually nearest neighbors between objects. Mapping correspondence between two images involves extracting individual features and performing metric based matching. Let say P and Q are local regions(obtained from l-1 layer) of feature vectors $F_A l$ and $F_B l$ of images A and B at layer l. We say neurons p belongs to P and q belongs to Q are nearest neighbors if and only if p under the set Q under a similarity distance function(eq.2). This measure is defined as local style transfer of regions P's and Q's, say C_A and C_B, representations corresponding in input images A and B over their $L2$ norm distance of both stylized images.

ALGORITHM: Cross-Domain Deep Correspondence

Input: Two RGB images: I_A, I_B
Output: A set of corresponding point pairs $\Lambda^0 = \{(p_i, q_i)\}_{i=1}^N$
Preprocessing: Extract $\{F_A^\ell\}_{\ell=1}^5$, $\{F_B^\ell\}_{\ell=1}^5$ by a feed forward of I_A, I_B through the VGG-19 network.
Initialization: Set $R^5 = \{P^5, Q^5\}$ to the entire domain of F_A^5 and F_B^5, $C_A^5 = F_A^5$ and $C_B^5 = F_B^5$.
for $\ell = 5$ *to* 1 **do**
  Extract Λ^ℓ from C_A^ℓ, C_B^ℓ within corresponding regions R^ℓ.
  Extract $\tilde{\Lambda}^\ell$ by filtering Λ^ℓ based on neural activations
  if $\ell > 1$ **then**
    Refine the search regions, $R^{\ell-1} = G_{\ell-1}^\ell(\tilde{\Lambda}^\ell)$, using (9)
    Generate common appearance features $C_A^{\ell-1}$, $C_B^{\ell-1}$, using (4).
  end
end
$\Lambda^0 = \tilde{\Lambda}^1$

image morphing technique:
Jing Liao et al [2014] described a basic algorithm for optimized image morphing technique in their paper "Automating Image Morphing using Structural Similarity on a Halfway Domain". In their work They address several challenges. Their optimization technique let's the user specify very few correspondences to avoid ghosting artifacts. It provides per pixel alignment based on structure similarity to avoid boundary artifacts during morphing. It ensures that the entire rectangular domain of the image is well defined finally it enables the user to interactively and effectively control the appearance of the morph. Traditional approaches construct a map from one image to the other but this leaves the secluded regions undefined due to this asymmetry. A reverse map is often necessary by maintaining consistency requires expensive bookkeeping, they introduce a simpler approach the halfway . The idea is to define a vector field V over a domain that is intermediate between the two images thus the intermediate image is parameterize by a single continuous function they minimize an energy function consisting of three terms. Structural similarity between the images, smoothness of the mapping and deviation from user-specified correspondence terms.

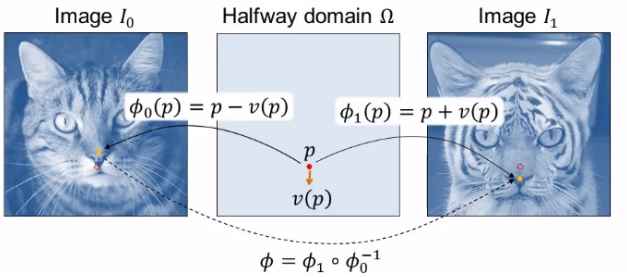


Fig. 2. halfway parameterization: The SIM energy term measures structural similarity of corresponding warped neighborhoods N0,N1 of the two input images.

The **similarity energy** attempts to match pixel neighborhoods of each halfway image pixel in the two deformed input images the algorithm use a modified structure similarity energy to match regions with similar edge structure such as the object boundaries. Shown in Figure 3(circle to star example)

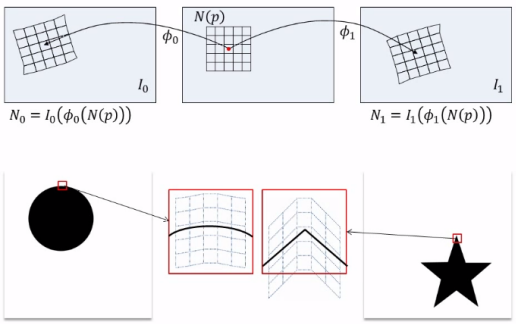


Fig. 3. similarity energy. The SIM energy term measures structural similarity of corresponding warped neighborhoods N0,N1 of the two input images.

UI energy term ensures that user specified correspondence points are matched in this example the center of the circle is mapped to the center of the star as specified by the red correspondents point.

smoothness energy ensures that the resulting vector field favors a fine functions in the mappings to both left and right images.

in order to render a morph sequences they presented a novel per pixel iterative algorithm that efficiently determines corresponding halfway image the main points and blends the colors from each input image To rasterize intermediate images a common approach is to render a triangle mesh that is deformed using the vector field our direct pixel valuation approach does not require the restoration of a fine triangle mesh and more importantly can produce plausible values beyond the boundary of the halfway image thus filling the entire rectangular domain of all intermediate image frames. The simplest approach is to assume when generating the intermediate image, is that each pixel follows a linear path, however in some situations a linear path may grow or shrink image regions. To avoid this they use quadratic motion paths it allows the extra degrees of freedom allow them to globally optimize for motion paths that reduce pixel neighborhood deformations in the intermediate images. In this article we use Jing Liao algorithm in our automatic version, instead of the user entering the Correspondence points between the images manually, we use the NBB algorithm to do so.

Method. The distance function In order for us to arrange the images in a logical order that enables seamless morphing between images, we have to create an order between the images, to do so we developed a distance function between the pairs of images that uses the output of the NBB's Correspondence algorithm between the pair to determine how different the images are.

Simple version: Since we seek a distance function that expresses the difficulty of morphing between two images, we use the distances between pairs of corresponding points. If we have a set of corresponding pairs $\{(pi,qi)\}$ then our first attempt at a distance function would be the sum of l_2 distances. however, we also need to consider the similarity between each pair. (Note: we choose the scale so the maximal distance is 1).

$$\sum_i \|p_i - q_i\| \tag{1}$$

Patch distance: To do this we use the patch distance function used in the original NBBs algorithm on patches around the points. For this we take two patches A, B and normalize them to cancel some of the effects of the style, then we take the dot product, which is a good measure of the angle.

$$d(A,B) = |N(A) \cdot N(B))| \tag{2}$$

This gives 1 or -1 for correlated patches, and 0 for uncorrelated patches, so we take the absolute value.

combined distance: We want points with correlated patches to be considered closer than uncorrelated patches, but we also want strongly correlated patches to have more effect, so for A_i, B_i denoting the patches around p_i, q_i we get the final distance function.

$$\sum_i (1 - \|p_i - q_i\|) \cdot d(A_i, B_i) \tag{3}$$

Image arrangement Assumptions When we make the logical arrangement of the images:
1. The distance function between a pair of images A and B will always return a non-negative value. where 0 distance is obtained if the images are identical.
2. Given the collection of images A and pair of images, source image and destination image S, T respectively. The best logical arrangement between S and T is:
(a) The direct transition(distance) path between S and T.
(b) The path in which all the transitions we perform will be smaller than the direct transition between the source image and the target image. If the condition (b) is met , we say that morphing pathway is preferable over direct morphing pathway.

Algorithm 1 Image Ordering algorithm(A, Ps, Pt, S, T)

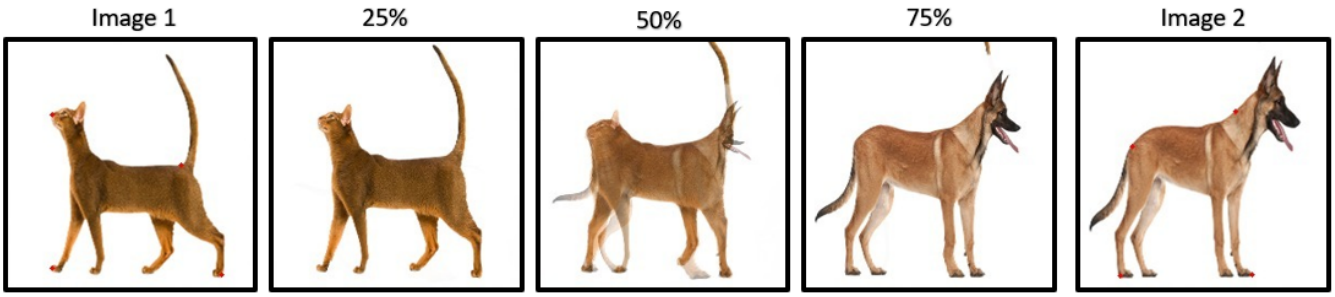
A-set of images
Ps-path list from source, Pt-path list from source target
S-source image
T-target image
1. Calculate the distance from S to all other $x \in A$
2. Calculate the distance from T to all other $x \in A$
3. Initiate a temporal variables, temp source $S' = S$ and temp target $T' = T$
4. **if** $D(S,T) > \frac{D(S,g)+D(g,T)}{2}$ **then**:
    4.1.Add the shortest edge from $\{(S,g),(g,T)\}$
        if $D(S,g) < D(g,T)$ **then**:
            Add (S,g) to Ps **and** $S' = g$
        else:
            Add (g,T) to Pt **and** $T' = g$
5. **if** $S' = S$ and $T' = T$, **return** $Ps^\frown Pt$
6. **Run** Image Ordering algorithm (A, Ps, Pt, S', T')

Experiments. All the images in the experiment part are not taken from the training set of the neural network. However, the neural network learned the images from ImageNet data set for a classification task, so we see no problem in using images that were part of this learning data set in our morphing task. But in our opinion it was right to use a foreign image collection for the morphing task.
It is not possible to quantify the way in which hybrids produced in this process (the middle image, at 50% of the process) are indeed "good representative creatures". User research can be done by a dichotomous diagnosis of whether they think the image production is real or hybridized Because we believe the diagnosis is subjective.

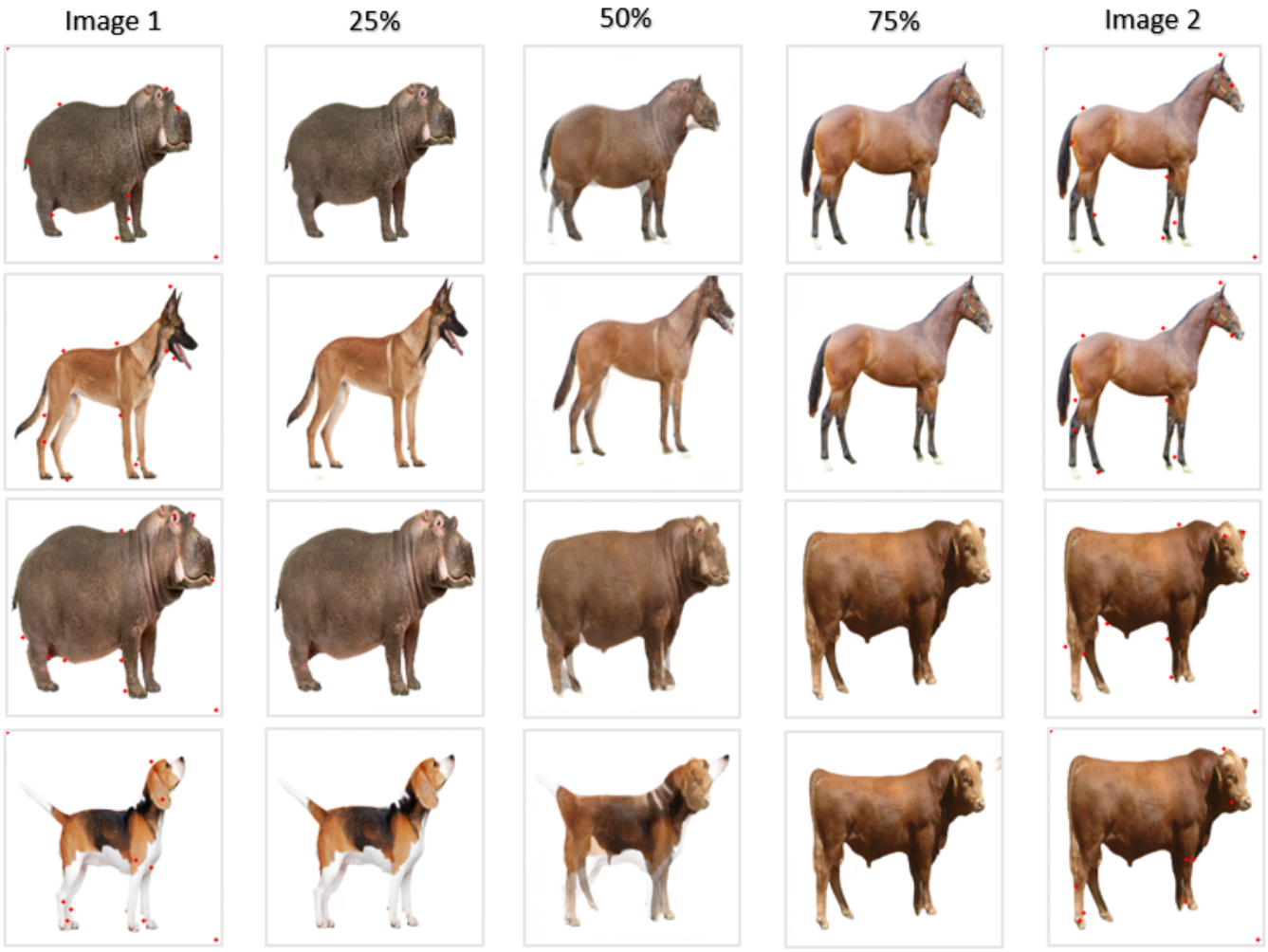
Direct Morphing - Our first attempt is to take two images from different fields and perform morphing directly between them.

In both of the above examples, lion-cat, and donkey-cat it can be observed that when the original images were inserted (with the background) the morphing action was significantly affected, to see the effect of the background, we compared the hybrid mid-production images in each pair of cases for the same images. At lion-cat example can be seen that the lion sandy background is distorted at the hybrid image, this distortion significantly affects the legs and tail area of hybrid production, which can be explained by Liao et al. [2014] morphing algorithm that we use. this algorithm uses "Cut the edges of the image" operation. after this operation is made there is a Completion using gradient-domain optimization. In the resulting extended region, for each pair of adjacent pixels, we compute their finite-difference. The effect is that the extended region of each image captures detail from the other image while seamlessly matching the appearance of its self inner region. Understanding that the background of images will affect the morphing even if it is mainly at the edges of the image and to eliminate this effect on our experiments we remove all backgrounds from the images from this step on. The background removal operation is done with an image recognition online-tool(remove.bg) that automatically removes the background from a set of images. The action it performs is to identify the main object in the image and remove the background or objects that do not belong to this object. We then chose a white background as a default for the images rather of the original background.

Opposite orientation - During the tests we performed, we came across several examples where object orientation in images was contradictory. Identifying two objects facing in different directions is an easy task for a human being, but the NBB's algorithm that we used does not take into account that the orientation of the images can be opposite. And if so, it does make a good fit for some images that are in small biased orientation from each other. Today there is a lot of research in the object orientation field, that can determine the image orientation vector (for example, there is an object orientation induction model 'OOI' that allows object orientation detection by using number of adjacent frames from a video shooting of the object) in this article we chose to do the image alignment manually due to lack of time , but of course this can be automatically integrated into the system pipeline future.).

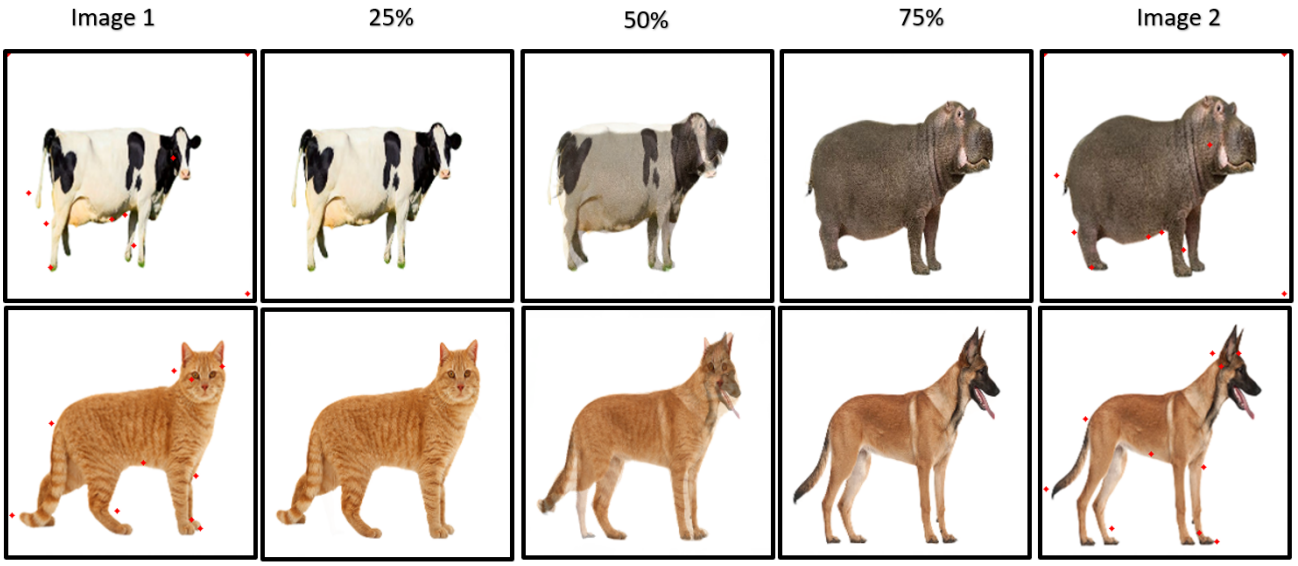


Aligned orientation - In the example below of aligned image pairs with white background. On each pair, we run the algorithm to perform morphing of image1 to image2.



The four examples above demonstrate the algorithm performance after removing the background and aligning the images. It can be seen that the algorithm can achieve good morphing results (the 50% image in each example). however In the first example (hippo - horse) and the last example (dog - cow), it can be seen that despite the changes and adjustments made to the images, the output hybrid creatures created may be "imperfect", The algorithm did not perform a seamless transition in certain areas of the image.

Morphing incorrectly - In the example below of aligned image pairs with white background. On each pair, we run the algorithm to perform morphing of image1 to image2.



In several cases, the morphing algorithm performed improperly, the hybrid creature that is distorted and cannot be said to be a "real" hybrid. In the above examples, the objects are aligned, but only one animal head faces 90 degrees vertically. We conclude that the poor performance of the algorithm is since the head in the image is not aligned if this is the rest of the body of the hybrid creature formed satisfactorily. To make a seamless transition in such a situation, we will be forced to identify the different part of the object and their orientation (the heads in the picture). This can be done using continuous object orientation estimation task, which requires prediction of 0 to 360 degrees orientation of the objects. There are today several algorithms that can do this successfully. But that's beyond this article.

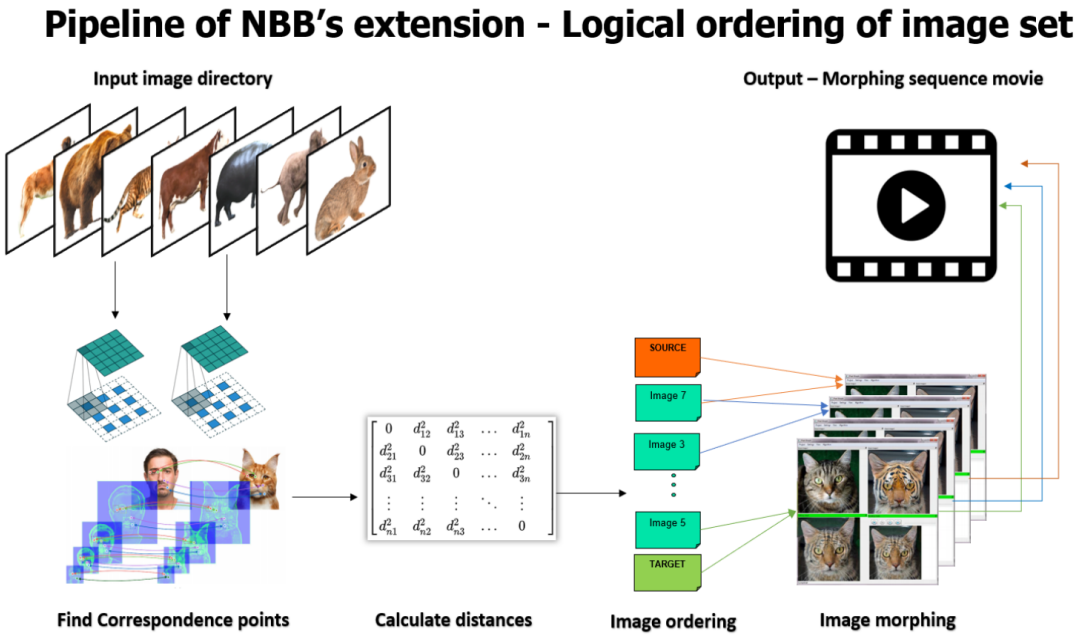
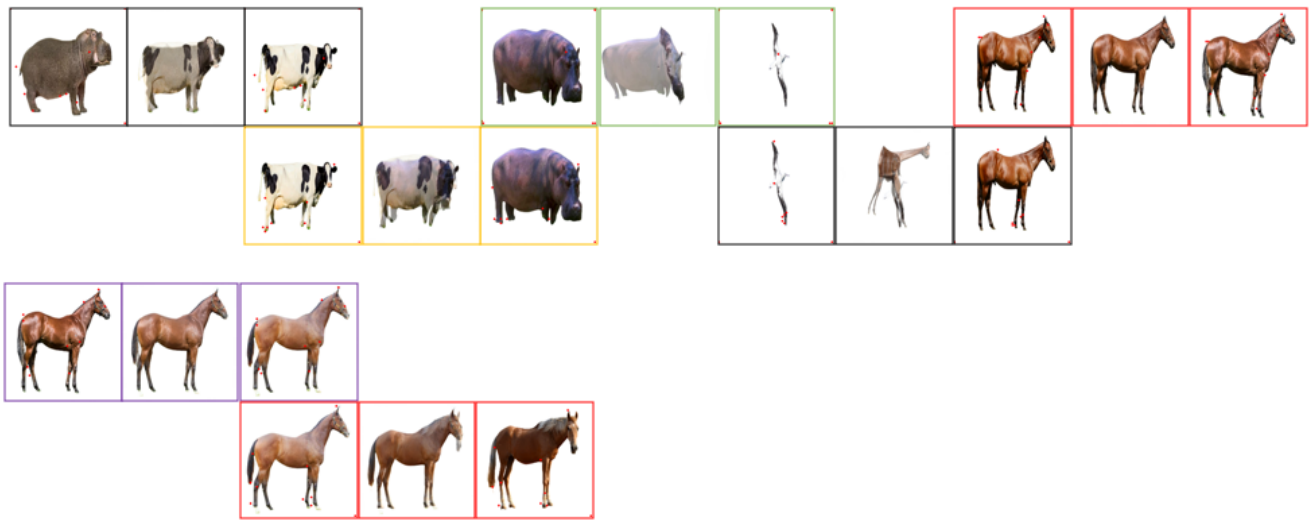


Fig. 4. NBB's extension pipeline
1.Input image directory - reshape images to 224X224 pixel and remove original background and color it white.
2. Find correspondence points - Using NBB's Sparse Cross-Domain correspondence algorithm to find all Correspondence points between all image pairs in the directory.
3. Calculate distances – use the correspondence to measure the distance between two images.
4. Image ordering – Using the distance to arrange the images in sequence so that selecting any image will allow the arrangement of the other images in logical sequence from the closest to the farthest.
5. Image morphing – using the correspondence points to morph between pair of images and output the Morphing sequence result.

Results of our pipeline

The results below show the sequences we obtained by running the pipeline on different source and target images. We have compared our image arrangement algorithm to the Dijkstra algorithm that brings the shortest route between two images. We also ran these two algorithms in combination with domain manipulation, so that the distances between images within the same domain were divided by 2, to give more weight to logically close images. Each example is accompanied by a sequence of picture numbers from the dataset, so picture 00 is the first picture of cats, and picture 10 in dogs is the first, and for birds it is the 20th picture and so on. Each transition is represented by 3 images (surrounded by the same color frame) The left image is the source image, the right is the target image and the middle image is the hybrid creature created during the transition.

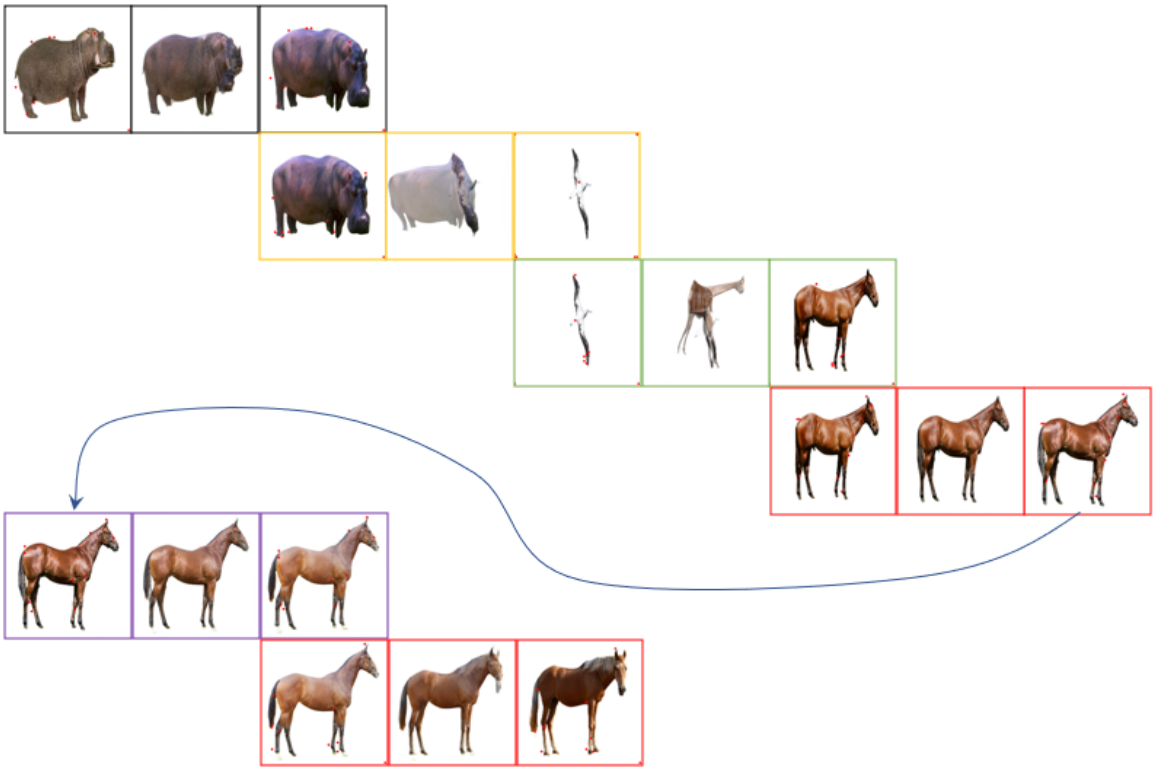
Our morphing path (50,81,53,23,96,94,98,95)



Same results: morphing Dijkstra and morph Dijkstra with domain manipulation (50,98,95)

our order: [50, 81, 53, 23, 96, 94, 98, 95]
Avg. transition distance: 2612.62
Median. transition distance: 3325.4
dijsktra order: [50, 98, 95]
our order with domain manipulation: [50, 53, 23, 96, 94, 98, 95]
Avg. transition distance: 1998.51
Median. transition distance: 2654.73
dijsktra order with domain manipulation: [50, 98, 95]

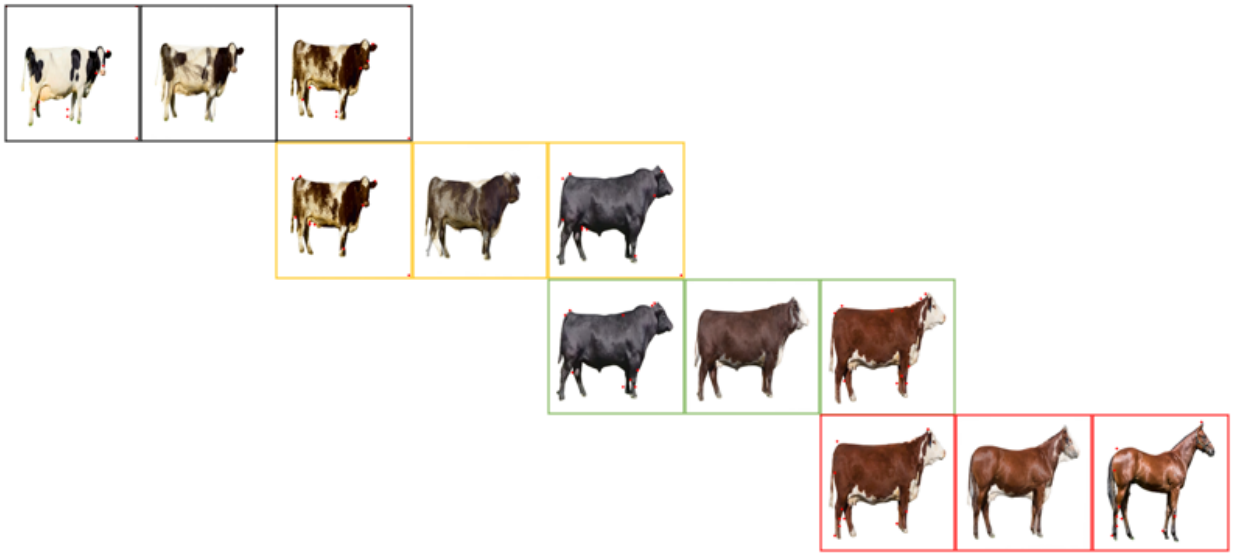
Our morphing with domain manipulation path (50,53,23,96,94,98,95)



Our Morphing path (81,87,83,94)

Morphing Dijkstra (81,50,98,94)

Our morphing path with manipulation (81,87,85,88,94)

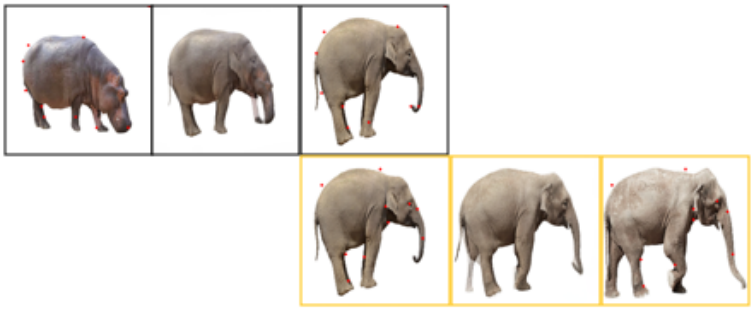


our order: [81, 87, 83, 94]
Avg. transition distance: 3199.97
Median. transition distance: 3646.83
dijsktra order: [81, 50, 98, 94]
our order with domain manipulation: [81, 87, 85, 88, 94]
Avg. transition distance: 1806.51
Median. transition distance: 2898.00
dijsktra order with domain manipulation: [81, 88, 94]

Morphing Dijkstra with domain manipulation (81,88,94)

Same results in all paths (20,36,26)

Best morph Dijkstra (56,40,41)



Morphing with Dijkstra got best results in intra-domain images (70,77)

Note: In all experiments,we stopped the NBB's algorithm in step 2 of the run (total of 4 layers without reaching the fifth layer) and extracted the matching points from that layer. Because we did not see a significant difference in obtaining the result from the fifth layer over the fourth. It is possible to achieve a more accurate resolution (pixel level) but it requires a longer computation time, for example, two 224x224 pixel images running our algorithm fully until the fifth layer lasted about 10 minutes against only one minute when stopping at the fourth layer.

Image Data Set

Ten domains of images, each domain represented by one row of ten images of animals that belong to it. all images were aligned horizontally, and faced to the right side.

Conclusions and future work. So far we have not accepted that our arrangement algorithm does the arrangement better than finding the shortest route. Indeed, several examples could be observed where reducing the distance of the images made the transition significantly more smooth.
In the results of the above, the use of domain manipulation appears to have positively affected the transitions and eliminated logically unnecessary transitions, so in our opinion, using this manipulation whether dividing the distance by 2 or any other factor can improve the performance of the algorithm. One of the lessons we learned during the experiment is that there is room to add the object orientation diagnosis, and we should consider adding a bias in our distance calculation function for different directions.
To use this algorithm to make transitions between different animals faces, we must train the network on the faces of animals (the network is trained on image classification).
The algorithm we introduced allows you to arrange images in a logical order and to make seamless transitions between the images along the way. The output of the algorithm is a video showing the transitions.
It is difficult to show the results of the transitions based on a limited number of frames, so we recommend that you try the algorithm yourself or see our examples at the paper repository

.

References.

- This paper code for NBB's extention image sequence morphing: https://github.com/talhadar1/NBBs$_{e}xtension_{m}orphing_{s}equence_{u}sing_{c}ross_{d}omain_{c}orrespondence$
- The code for the image morphing is taken from the original paper: https://liaojing.github.io/html/data/pixmorph.pdf
- and can be found in the original repository: https://github.com/liaojing/Image-Morphing
- NBB's Cross Domain Correspondence paper: https://arxiv.org/pdf/1805.04140.pdf
- NBB's Cross Domain Correspondence repository: https://github.com/kfiraberman/neural-best-buddies
- Remove Background from Image tool: https://www.remove.bg

ACKNOWLEDGEMENTS