Semi-Automated Stereo Image Patches Generation and Labeling Method Based on Perspective Transformations

Diego Patricio Durante[1], Ramiro Verrastro[1], Juan Carlos Gómez[1,2], and Claudio Abel Verrastro[1,3]

[1] Grupo de Inteligencia Artificial y Robótica (GIAR), Univ. Tecnológica Nacional Fac. Regional Bs. As., Ciudad Autónoma de Buenos Aires, Argentina.
{ddurante,ramiroverrastro}@frba.utn.edu.ar
[2] Centro de Electrónica e Informática, Instituto Nacional de Tecnología Industrial (INTI), San Martin, Buenos Aires, Argentina.
juanca@inti.gob.ar
[3] Investigador consulto, Comisión Nacional de Energía Atómica (CNEA), Centro Atómico Ezeiza, Buenos Aires, Argentina.
cverra@cae.cnea.gov.ar

Abstract. In computer vision, Wide Baseline Stereo (WxBS) refers to Vision System configurations on which their images come from cameras with non parallel and widely separated views.
One common task in reconstruction algorithms of WxBS consists of subvididing the stereo images in multiple image patches and then associate homologous patches between homologous images. Multiple approaches can be used to associate homologous patches. To train and test supervised learning algorithms for this tasks, a labeled dataset is required.
In this work, a semi-automated method to generate patches and their labels from WxBS images is presented. It allows to calculate thousands of positive and negative pairs of patches with a score of correspondence between a pair of potentially homologous image patches. This method largely solves the problems of traditional approach, which requires a lot of hand labeled work and time. To apply the method, images from different viewpoints of objects with planar faces and their corner locations are required.

Keywords: Computer Vision · Machine Learning · Wide Baseline Stereo · Labeling Tool · Siamese Convolutional Neural Networks

1 Introduction

To train a supervised learning task in machine learning algorithms, deep learning neural networks, and also to benchmark computer vision algorithms, a labeled dataset is needed. Data labeling in computer vision consists of annotating from simple text tags for categorical classifications, to bounding boxes or more complex polygons for image segmentation. Keypoint based mapping vision system,

or image registration methods have the capability to find homologous points between multiple images (keypoint matching) [20].

A typical keypoint association pipeline is shown in figure 1, the task can be separated into 3 stages: keypoint detection, description and matching [9].

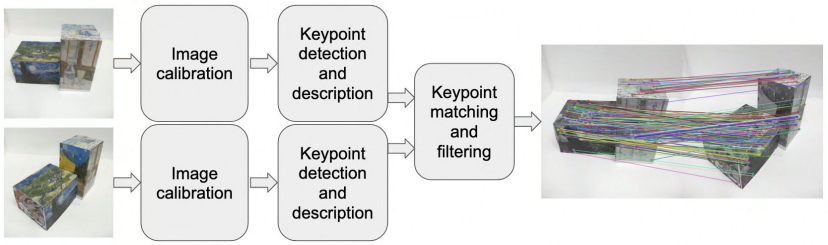


Fig. 1. Pipeline for image calibration, image detection, description and matching.

The keypoint matching stage can be studied as a classification task in which the degree of correspondence between two homologous points is sought to be established.

Keypoint description is a central issue, algorithms such as SIFT [14] and SURF [2] have proven to be robust to multiple condition changes, such as lighting, noise, rotation and little perspective changes. They have been used successfully in many applications and can be customized through the adjustment of a few parameters.

For strong perspective changes, there exists algorithms based on SIFT, such as ASIFT [24]. ASIFT simulates affine transformations on the images before applying SIFT in order to approximate the effects of the variations of camera positions. With the advance of convolutional neural networks (CNN) and the computing power, there exists solutions that outperform traditional algorithms, such as Siamese CNN architectures [5]; however, these require a large amount of data to train and tuning them to the different problems.

To organize the datasets and generate annotations for the different problems, there are numerous tools, which allow manual labeling, automatic labeling and semi-automatic labeling assisted by Artificial Intelligence (AI) algorithms [19, 10, 22]. However and even using these tools, for this particular problem, labeling pairs of homologous points entails a great demand of manual work.

In this work[4], a method to automatically generate a dataset of great amount of labeled stereo image crops is presented. The method is based on taking advantage of the geometry of planar faces. It requires the user to mark the corners of the planar faces in multiple images. By using the corners of homologous faces in different images, a map matrix from one image to the another one is obtained. Starting from any point inside a marked face and by using the map matrix, the location of the homologous point can be obtained easily. Taking advantage of this principle, the overlap between a keypoint and another potentially homologous in another image is calculated. Finally, the label of the pair belongs to

[4] The code might be available at https://bit.ly/3B1MRiA

the overlapping score. For each image, keypoint candidates have been obtained using ASIFT keypoint detectors.

Finally, the marked structure can be used to generate synthetic datasets: It allows to easily increase the amount and the diversity of data from new images.

In the following sections: the main problem is presented and the usefulness of the method is raised; The set of images called Van Gogh are described; The necessary calibration step is mentioned; The algorithm and the perspective transformation is presented; A metric to qualify the match is established; The construction of a dataset and a strategy to increase the quantity of labeled data is described. In the final section, conclusions are presented.

2 Methods

2.1 Problem Statement

Usually, matching problems from images in WxBS configuration [17] needs to use a labeled dataset, either for training and/or test supervised learning models or to compare the performance of some methods to solve the problem. If the detection, description and matching strategy is used, thousands or hundreds of pairs of crops and its labels are required. Hand labeling those crops is very expensive and requires a lot of work and time. This problem also appears in the case of a siamese CNN strategy [8, 1, 26].

For this reason, a method to generate automatically a dataset of thousands pairs of stereo crops (or pairs of image patches), and its label or level of correspondence from two base images on WxBS configuration is proposed.

2.2 Van Gogh Dataset

For this project, the *Van Gogh* dataset has been created. The dataset consists of 24 RGB images with a resolution of 4000 x 3000 pixels. Each one capturing the same object which consists of two joined cubes. For each image, the camera has been placed in different perspectives, focusing on the center of the object. Two samples of Van Gogh dataset images can be seen in figure 2a.

The captured object on each image is based on the union of two cuboids, making a figure with planar faces. Also, each face of the cuboids contains a picture: almost all faces have a Van Gogh painting printed on them, except for one of those faces that consists of a picture that is normally used as a benchmark for patch matching problems [15].

Additionally to the 24 images of the object, the dataset has pictures of chessboard patterns for camera calibration (see samples on figure 2b).

Image Calibration Step While ideal cameras don't have optical aberrations, images obtained with real cameras have different sources of distortions (optical aberrations), like Barrel and Pincushion distortion. There exists multiple approaches to correct this problem [13, 20].

(a) Two samples of Van Gogh dataset

(b) Two samples of chessboard patterns for camera calibration

Fig. 2. Samples of Van Gogh dataset and chessboard patterns taken with the same camera.

One of the most used image undistortion techniques consists of using camera calibration patterns to calculate intrinsic and extrinsic camera parameters and subsequently by applying a function based on those parameters to undistort the image. *Zhang et al.* have presented a widely known method [25] to calibrate images by using chessboard calibration patterns. Those patterns are pictures taken with the same camera to be calibrated. Next, the parameters obtained on the calibration stage can be used to undistort all images taken by this camera. A drawing of undistorting a distorted chessboard image can be seen in figure 3.

Fig. 3. Camera calibration step using chessboard patterns.

2.3 The Semi-Automated Labeling Algorithm

The main purpose of this work is to easily generate and label in a semi-supervised manner a dataset from at least two input images. Those inputs consist on pictures of the same object with planar faces (such as images of Van Gogh dataset). The generated dataset consists of multiple pairs of crops of the original images and its label, indicating its overlapping metric. The label should be close to 0 if the crops don't correspond to the similar part of the image in different images or close to 1 if they are homologous.

Given multiple images and some information to calculate their homologous transformations between zones, the algorithm 1 returns a list of relevant output pairs of image crops and an overlapping metric. The overlapping metric is adjusted to be the label of the pair.

In next sections, different parts of the presented algorithm 1 should be described. It can be summarized as:

Algorithm 1 Algorithm for Candidates Generation and Auto-Labeling Dataset

```
1: function LABELDATA(Image_1, Image_2, Img1_Quads, Img2_Quads)
2:     Kp_1 ← CandidateDetector(Image1)   ▷ Get Matching candidates for image 1
3:     Kp_2 ← CandidateDetector(Image2)   ▷ Get Matching candidates for image 2
4:     labeledData ← ∅
5:     for face_1 in Img1_Quads do
6:         for face_2 in Img2_Quads do
7:             M_21 ← getMapping(face_2, face_1)
8:             Kp1_filt ← Kp1_all ∩ mask(face_1)   ▷ Get just keypoints inside quadr 1
9:             Kp2_filt ← Kp2_all ∩ mask(face_2)   ▷ Get just keypoints inside quadr 2
10:            for Kp_1 in Kp1_filt do
11:                for Kp_2 in Kp2_filt do
12:                    IoU_pair ← IoU_H(Kp_1, Kp_21, M_21)
13:                    if IoU_pair > IoU_min and r_Th ≤ r_coef(Kp_1, Kp_21) then
14:                        labeledData.append(Kp_1, Kp_2, IoU_pair)
15:                    end if
16:                end for
17:            end for
18:        end for
19:    end for
20:    labeledData ← addNegativeCases(labeledData)   ▷ Optional, see discussions
21:    return labeledData
22: end function
```

1. Detect candidates (image crops) to be labeled in the images (lines 2 and 3).
2. Subdivide input images on their planar faces to be processed (from line 5).
3. Use perspective transformations to get a mapping from one face to its homologous face projecting the images to a common space (line 7).
4. Measure the overlapping factor between each keypoint of the planar face on the left image and each keypoint on its homologous planar face. This step generates a list of positive samples (lines 10 to 17).
5. Generate negative cases (line 20).

2.4 Candidate Detection Stage

In this stage, multiple keypoint candidates are detected. Keypoints are structures compounded by the keypoint localization and a group of features that represents its surrounding local region. Particularly, keypoints need to be robustly detected.

Usually, the structure of the keypoints are compounded by:

• **Keypoint location** is a point that represents its coordinates.

• **Keypoint orientation** is a reference angle that depends on the method. It is used to make comparisons between keypoints aligning its orientations.

• **Keypoint corners** Are 4 points that defines the corners of a polygon (quadrangle). It represents a squared region of the image where the keypoint is located.

By using the original image and any keypoint, one image patch can be obtained. The image patch is the image obtained by cropping the original image on the keypoint corners, and aligned with the keypoint orientation.

Keypoint detectors are methods to locate those keypoints. Those methods are powerful when they can detect the same keypoint despite some changes in the images such as: lighting, rotation, perspective and noise. There exists a list of well known keypoint detectors, starting from the popular SIFT detector and its derivatives like SURF, KAZE [7], AKAZE [6], ORB [18, 4], BRISK [12], and ASIFT [24] to new methods based on Deep Learning and Convolutional Neural Networks [8, 1]. Tareen, S.A.K. and Saleem, Z. made a comparison between SIFT, SURF, KAZE, AKAZE, ORB and BRISK [21]. Also Zheng et al. made a survery [26] on SIFT and CNN keypoint description and matching.

Considering that images of WxBS have strong changes of perspectives, the keypoint detector algorithm should be robust to this condition. ASIFT [24] has been selected for this stage because it was created to be robust to such a changes.

ASIFT algorithm is based on the well known SIFT algorithm. It applies affine transformations to the image simulating multiple perspective transformations and then detecting keypoints on multiple spaces. Detections on different scales are made using SIFT algorithm. Figure 4a shows a detected keypoint using SIFT detector at some ASIFT simulated perspective. Figure 4b shows the keypoint previously detected on its real space. Figure 4c shows the image patch of the keypoint detected previously and aligned to -90^o.

(a) Image with patch on the detected space.

(b) Image with patch on the image space.

(c) Patch rectified and cropped.

Fig. 4. A keypoint detected by ASIFT on different stages.

2.5 Perspective transformation

An important part of the presented algorithm is the perspective transformation. Perspective transformation is a common linear transformation in computer vision [16]. Given a point on one frame of reference and the transformation matrix (a.k.a. map matrix), it is possible to get the homologous point on other frame of reference. This relationship is expressed in equation 1.

$$\begin{bmatrix} x' \\ y' \\ s_f \end{bmatrix} = M \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & b_1 \\ a_3 & a_4 & b_2 \\ c_1 & c_2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{1}$$

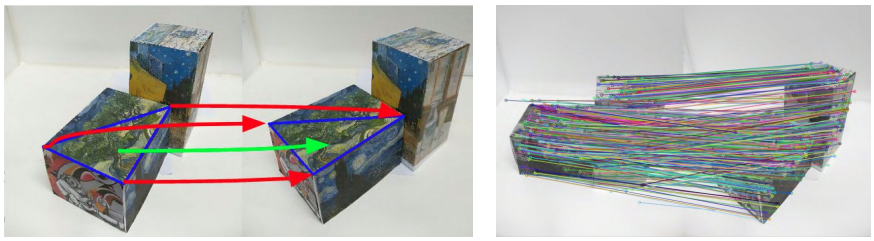
M is the map matrix to transform the point $(x; y) \rightarrow (x'; y')$. It can be decomposed as:

- $\vec{A}$ defines basic transformations (rotation, scaling, etc).
- $\vec{B}$ defines the translation vector.
- $\vec{C}$ defines the projection vector.
- s_f is the scale factor.

Van Gogh dataset is made by pictures of the same object with planar faces.

For each pair of homologous planar faces, it is possible to get one map matrix M. On figure 5a, green arrow maps an arbitrary point in a planar face of the left image and its transformed position on the same face of the right image. For each planar face, by using perspective transformations on images, straight lines and polygons in the left image are mapped to straight lines and polygons in the right image. As it is shown on the same figure, red arrows maps blue straight lines from left to right image. On figure 5b, multiple homologous points are mapped.

Once M matrix is known, transformations are made in an automatic way. For each pair of planar faces, the corners of the face are needed to calculate M matrix.



(a) Point and polygon mappings from multiple images.

(b) Corresponding keypoints in between left and right image.

Fig. 5. Perspective Transformation for hand labeled points and inferred points.

Map matrix or transformation matrix (M) between homologous planar faces of two images can be calculated analytically, solving the linear system of equations 1 for four homologous points [20]. If instead of knowing 4 homologous points, many potentially homologous points were known, statistical methods to estimate M can be used. Those methods should be robust to outliers. RanSAC [23] algorithm allows to filter outliers and estimate M.

2.6 Points and Corners to obtain the M Matrix

In this work, the M matrix is an essential part of the algorithm 1. Given the M matrix, it is possible to know if a pair of keypoints previously detected in right and left images are homologous or not. Therefore, it is necessary to calculate or estimate the map matrix M.

To get the map matrix, two approaches have been evaluated.

The first approach needs to hand-labeling corners of faces on both images. Using those points the M matrix is obtained. It is important to label as few points

as possible. Note that for every planar face, the minimum quantity of points to estimate M matrix is 4. Given that points, map matrix can be analytically calculated. Infinite combinations of points can be used to get the M matrix, but if the corners of every planar faces were chosen:

– Their corners can be used to know the geometrical limits of every face too.

– Normally, to reduce the error, it is better to select points as far apart as possible. With this assumption, corners are good candidates to calculate M matrix.

– Multiple faces share corners. So multiple labels will be shared and reused.

For this stage, the manual labeling of the corners has been strategically chosen. On figure 6 two hand-labeled images are shown. Note that faces have an internal line subdividing quadrangles in triangles. Those triangles can be calculated automatically and this step is optional. Triangle numbers on each image indicates homologous triangles in left image and right image.

For this strategy, the number of corners to be hand-labeled is very low and the error of manual labeling is low too. No special skills are needed for this task.

The second approach detects faces and map matrices automatically. For this approach, two alternatives have been evaluated:

– The first alternative is detecting perspective transformations using keypoint detector descriptor and matching strategies [9]. To make robust this option, multiple filtering stages should be added (ratio match filtering, KNN filtering) [3]. Finally, the M matrix can be estimated using RanSAC [23]. In this case, planar faces should be estimated too. This task can be simplified using another type of dataset, on which just images have only one planar face. The whole precision of the final system depends on the quality of all the estimations and it can be worse than expected.

– The second alternative is using fiducial marks or tags to label the corners of the faces and estimating its location [11]. This alternative has not been programmed because it requires modified images.

For the final version of this work, the first approach has been chosen: Hand-labeling corners of faces strategy was simpler and more precise. The corners of the planar faces have been hand-labeled. This criterion has been selected to reduce errors due hand-labeling corner positions and due to numerical precision.

2.7 Overlapping Metric

The presented method generates a list of pairs of keypoint candidates and the information to know its location and its label. Candidate patches are labeled using an overlapping metric. The label indicates the overlapping between two candidates, tending to 1 when the keypoints are perfectly homologous or to 0 when the candidates are not. The overlapping is calculated as the percentage between the location of the patch and the location of the projection of its candidate to the same image. On figure 7, two candidates are drawn in green. The overlapping metric of this sample is 0.6.

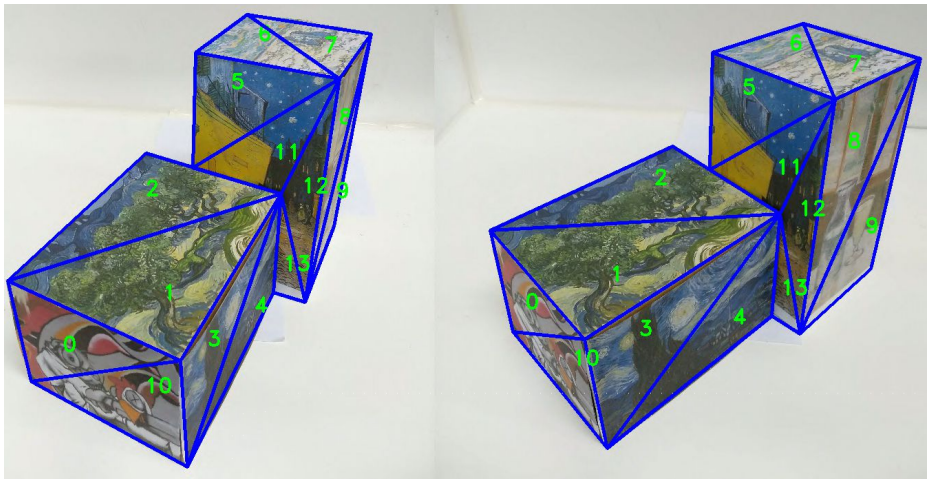


Fig. 6. Each planar face in the left image and its correspondence in the right image. Quadrangles have been splitted into two triangles

Fig. 7. Matched pair with $\text{IoU}_H = 0.6$.

Jaccard index Is a general overlapping metric that indicates the degree of overlapping between two sets. It is calculated as the intersection of the sets over the union of the sets, as it is expressed on equation 2.

$$\mathcal{J}(A, B) = \frac{\mid A \cap B \mid}{\mid A \cup B \mid} \tag{2}$$

Intersection over Union (IoU) Is a metric commonly used in segmentation problems of computer vision. It is used to know the level of overlapping between

some target mask (the ground truth) and the prediction of this mask. The general formula to calculate IoU in this context is expressed in equation 3. It can be seen as the Jaccard index when it is evaluated on images.

In that case, the intersection is the area in common between target and prediction and the union is the area in which at least one of the masks exists, either target mask or prediction mask.

$$IoU(target, prediction) = \frac{target \cap prediction}{target \cup prediction} \tag{3}$$

Intersection over Union for Homologous Images For this problem, given two image keypoints and a map matrix that projects from the frame of reference of the second image to the first one, the metric IoU has been used with some modifications on the definition:

• Both images are projected to the same frame of reference by using M_{21} : $projection(x_2; y_2; M_{21}) \mapsto (x_1; y_1)$. The map matrix is calculated using the corners of the face on which the keypoint exists. By convenience, the target frame of reference is the frame of reference of the first image.

• Then, the intersection over union is calculated for the area occupied by the keypoint of the first image (Kp_1) and the area occupied by the keypoint of the second image (Kp_2) projected into the frame of reference of the first image.

This metric is a number between 0 and 1, where 0 indicates not overlapping and 1 indicates fully overlapping. The equation 4 represents this calculation.

$$IoU_H(Kp_1, Kp_2, M_{21}) = \frac{area(mask(Kp_1) \cap (projection(mask(Kp_2), M_{21})))}{area(mask(Kp_1) \cup (projection(mask(Kp_2), M_{21})))} \tag{4}$$

(a) $Patch_1$ (b) $Patch_{21}$ (c) Intersection (d) Union

Fig. 8. Patches on frame of reference of Image 1, Intersection and Union

On fig. 8a, the mask for the $Patch_1$ is shown. On fig. 8b, the $Patch_{21}$ is plotted (the $Patch_{21}$ is the $Patch_2$ projected on frame of reference of the image 1). On fig. 8c, their intersection and on fig. 8d their union are plotted.

An example of a bad and good matches are shown in figures 9a and 9b respectively. The label for the bad match is $IoU_H = 0$ and for the better match is $IoU_H = 0.85$. For each one, left crop is a plot of the result of the $Patch_2$ projected to the frame of reference on the image 1 after its rectification ($Patch_{21}$). The

center crop is a plot $Patch_2$ on its original frame of reference. Finally, the right crop is a plot of the rectified $Patch_1$ on its original frame of reference. Note that if the $IoU_H = 1$, the pair of candidates should be very similar.

(a) Sample of bad match with label $IoU_H = 0$.

(b) Sample of better match with label $IoU_H = 0.85$.

Fig. 9. Samples of patches with low (left) and high (right) IoU_H

2.8 Skipping misaligned patches

The IoU_H metric calculates the overlapping of the area projected to the same frame of reference, but it doesn't use angle information. In case of $IoU_H > 0$, the angular alignment defines if the pair can be valid or not. A way to discard misaligned patches is used. Given:

•$Kp_{1(i,n)}$: the location on the $i = x$ or y axis of the corner index $n = 0..3$ for the keypoint 1 (Kp on image 1).

•$Kp_{21(i,n)}$: the location on the $i = x$ or y axis of the corner index n of the keypoint 2 (Kp on image 2) projected on the frame of reference of image 1.

Equation 5 defines the sum of distances between keypoint 1 corners on image 1 and the projection of the keypoint 2 corners into the frame of reference of image 1.

Equation 6 defines the sum of distances between keypoint 1 corners on image 1 and the projection of the keypoint 2 of image 2 to the frame of reference of image 1 on the minimal distance when the corners have been interchanged.

$$alignBase(Kp_1, Kp_{21}) = \sum_{n=0}^{3} \sum_{\forall i \in \{x,y\}} (Kp_{1(i,n)} - Kp_{21(i,n)})^2 \tag{5}$$

$$swMinDist(Kp_1, Kp_{21}) = \min_{m \in [1..3]} \sum_{n=0}^{3} \sum_{\forall i \in \{x,y\}} (Kp_{1(i,(n+m) \mod 4)} - Kp_{21(i,n)})^2 \tag{6}$$

For a pair of keypoints, if the relationship of distances between non swapped and swapped corners (equation 7) is higher than a threshold ($r_{coef} \geq r_{Th}$), the pair is discarded, avoiding adding rotated pairs of patches to the labeled dataset. Else, it is a labeled pair with score IoU_H. In this work, $r_{Th} = 0.5$ has been used.

$$r_{coef}(Kp_1, Kp_{21}) = \frac{alignBase(Kp_1, Kp_{21})}{swMinDist(Kp_1, Kp_{21})} \tag{7}$$

In summary, if two image patches are similar, but one is rotated with respect to another one, the pair is discarded beyond its high overlapping factor.

2.9 Final Dataset

The final output of the generated dataset consists of thousands of positive samples and negative samples. Positive samples are samples on which the patches are homologous. Negative samples are samples on which patches are not homologous.

Positive samples The output of the system for labeled pairs of patches is a list of pairs in which $IoU_H \geq IoU_{min}$ and $r_{coef} \leq r_{Th}$ for the labeled pairs given by the algorithm 1

Negative samples Negative samples are samples with $IoU_H = 0$. In some cases, it could be good to include samples with $IoU < IoU_{low}$ to get more complex negative samples. The quantity of negative samples is not strictly defined, but a good simplification can be to add the same quantity of negative and positive samples. Some strategies can be used to define negative samples:

- Negative samples can be selected in each batch of the training step.
- Negative samples can be selected considering cases in which they are complex to recognize (probably false positives). Also, they can be

Finally, images are taken from planar faces, but it is possible to apply image processing tasks to deform and simulate perspective changes if needed.

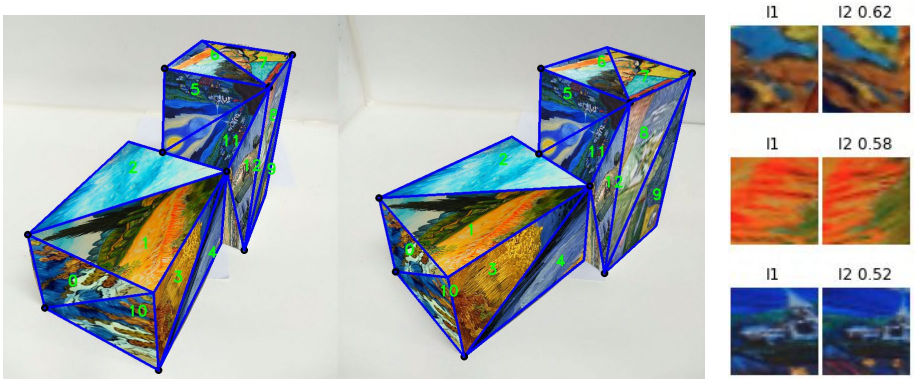
As example, for another work, currently in progress, a sublist of 1039 pairs with true labels and 1039 pairs with false labels have been generated. They have been used to train convolutional neural networks in siamese configurations.

2.10 Dataset Augmentation Strategy

If the quantity of pairs for the labeled data is not sufficient, the method allows to easily generate an augmented dataset. The data previously generated to calculate the map matrix for pairs of homologous faces on the Van Gogh dataset can be used, by replacing entire faces with new images. After this replacement, a new dataset can be obtained by using the method presented along this work. This is called Dataset Augmentation. The proceedings are:

- The picture of each face is replaced in multiple images by another picture (not necessarily plain) by using perspective transformations. Each face should have the same picture and orientation in homologous faces. Next, the algorithm should be run.
- To improve the diversity of the new dataset, each image can be subtly modified with data augmentation strategies (adding lighting effects, noise effects, etc) before this augmentation.

On figure 10a results using this procedure by replacing images on faces of figure 6 can be appreciated. On figure 10b, samples of pairs of the generated dataset with its corresponding IoU_H as label have been plotted.



(a) Input images with their labeled faces transferred from Van Gogh dataset.

(b) Samples of labeled candidates.

Fig. 10. Modified homologous inputs and 3 output samples of an augmented dataset.

3 Conclusions

Along this work, it has been demonstrated that it is possible to generate a labeled dataset of thousand of stereo image crops by using just a pair of WxBS base images with objects of planar faces and their marked corners. Note that hand making and hand labeling the complete dataset is not viable because it requires a lot of time, work and effort. An alternative has been presented. Also, a technique to generate thousands of labeled crops from new images that are not in stereo configuration, by reusing the structure and generating multiple synthetically modified datasets with the implication of a very low marginal cost and high diversity of images has been presented.

This job allowed to make another work, currently in progress, to train CNN in siamese configurations and to compare and validate it with traditional keypoint detection, description and matching strategies.

References

1. Altwaijry, H., Veit, A., Belongie, S.: Learning to detect and match keypoints with deep architectures. pp. 49.1–49.12 (01 2016)
2. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. vol. 3951, pp. 404–417 (07 2006)
3. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (2006)
4. Calonder, M., Lepetit, V., Ozuysal, M., Trzcinski, T., Strecha, C., Fua, P.: Brief: Computing a local binary descriptor very fast. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(7), 1281–1298 (2012)
5. Chicco, D.: Siamese Neural Networks: An Overview, pp. 73–94. Springer US, New York, NY (2021)
6. Fernández Alcantarilla, P.: Fast explicit diffusion for accelerated features in nonlinear scale spaces (09 2013)
7. Fernández Alcantarilla, P., Bartoli, A., Davison, A.: Kaze features (10 2012)

8. Han, X., Leung, T., Jia, Y., Sukthankar, R., Berg, A.: Matchnet: Unifying feature and metric learning for patch-based matching (06 2015)
9. Hassaballah, M., Ali, A., Alshazly, H.: Image Features Detection, Description and Matching, vol. 630, pp. 11–45 (02 2016)
10. Intel: CVAT: Computer vision annotation tool (2019), open source software available from https://github.com/openvinotoolkit/cvat
11. Kalaitzakis, M., Cain, B., Carroll, S., Ambrosi, A., Whitehead, C., Vitzilaios, N.: Fiducial markers for pose estimation. Journal of Intelligent & Robotic Systems **101**(4), 71 (2021)
12. Leutenegger, S., Chli, M., Siegwart, R.Y.: Brisk: Binary robust invariant scalable keypoints. In: 2011 International Conference on Computer Vision. pp. 2548–2555 (2011)
13. Li, X., Zhang, B., Sander, P.V., Liao, J.: Blind geometric distortion correction on images through deep learning (2019)
14. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60**, 91– (11 2004)
15. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Van Gool, L.: A comparison of affine region detectors. International Journal of Computer Vision **65**, 43–72 (11 2005)
16. Page, G.: Multiple view geometry in computer vision, by richard hartley and andrew zisserman, cup, cambridge, uk, 2003, vi+560 pp., isbn 0-521-54051-8. Robotica **23**, 271 (03 2005)
17. Pritchett, P., Zisserman, A.: Matching and reconstruction from widely separated views. In: Koch, R., Van Gool, L. (eds.) 3D Structure from Multiple Images of Large-Scale Environments. pp. 78–92. Springer Berlin Heidelberg, Berlin, Heidelberg (1998)
18. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An efficient alternative to sift or surf. In: 2011 International Conference on Computer Vision. pp. 2564–2571 (2011)
19. Sager, C., Janiesch, C., Zschech, P.: A survey of image labelling for computer vision applications (04 2021)
20. Szeliski, R.: Computer vision - algorithms and applications, second edition (2022)
21. Tareen, S.A.K., Saleem, Z.: A comparative analysis of sift, surf, kaze, akaze, orb, and brisk (03 2018)
22. Tkachenko, M., Malyuk, M., Holmanyuk, A., Liubimov, N.: Label Studio: Data labeling software (2020-2022), open source software available from https://github.com/heartexlabs/label-studio
23. Yang, S., Li, B.: Outliers elimination based ransac for fundamental matrix estimation. pp. 321–324 (09 2013)
24. Yu, G., Morel, J.M.: Asift: An algorithm for fully affine invariant comparison. Image Processing On Line **1** (02 2011)
25. Zhang, Z.: A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**, 1330–1334 (December 2000)
26. Zheng, L., Yang, Y., Tian, Q.: Sift meets cnn: A decade survey of instance retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence (08 2016)