

Université de Montréal

**Apprentissage de descripteurs locaux pour  
l'amélioration des systèmes de SLAM visuel**

par

**Johan Luttun**

Département d'informatique et de recherche opérationnelle  
Faculté des arts et des sciences

Mémoire présenté à la Faculté des arts et des sciences  
en vue de l'obtention du grade de  
Maître ès sciences (M.Sc.)  
en informatique

Décembre, 2023

© Johan Luttun, 2023

Université de Montréal  
Faculté des études supérieures

Ce mémoire intitulé :  
Apprentissage de descripteurs locaux pour l'amélioration des systèmes  
de SLAM visuel

présenté par

Johan Luttun

a été évalué par un jury composé des personnes suivantes:

---

Jean Meunier  
(Professeur)

---

Max Mignotte  
(Professeur)

---

Sébastien Roy  
(Directeur)

Mémoire accepté le \_\_\_\_\_

## RÉSUMÉ

---

Le présent mémoire traite du sujet de mise en correspondance entre deux images dans un contexte de SLAM visuel ou de SfM. Ces problèmes reposent généralement sur une représentation vectorielle de points saillants d'une image, appelée descripteur, et qu'on cherche à mettre en correspondance avec les points saillants d'une autre, en utilisant une mesure de similarité pour comparer les descripteurs.

Cependant, il reste difficile de réaliser cette mise en correspondance avec succès, en particulier pour les scènes difficiles où des changements d'illumination, des occultations, des mouvements, des éléments sans texture, et des éléments similaires sont présents, conduisant à des mises en correspondance incorrectes. Nous développons dans ce mémoire une méthode d'apprentissage profond contrastif auto-supervisé pour calculer des descripteurs robustes, particulièrement à ces situations difficiles. Nous utilisons le jeu de données TartanAir construit explicitement pour cette tâche, et dans lequel ces cas de scènes difficiles sont présents.

Nos résultats montrent que l'apprentissage de descripteurs fonctionne, améliore les scores, et que notre méthode est compétitive avec les méthodes traditionnelles telles que ORB. En particulier, l'invariance bâtie implicitement en formant des paires d'exemples positifs grâce à la construction d'une trajectoire depuis une séquence d'images, ainsi que l'introduction contrôlée d'exemples négatifs ambigus pendant l'entraînement a un réel effet observable sur les scores obtenus.

**Mots clés : SLAM visuel, SfM, mise en correspondance d'images, descripteur, apprentissage profond, apprentissage contrastif**

## ABSTRACT

---

This thesis covers the topic of image matching in a visual SLAM or SfM context. These problems are generally based on a vector representation of the keypoints of one image, called a descriptor, which we seek to map to the keypoints of another, using a similarity measure to compare the descriptors.

However, it remains difficult to perform this matching successfully, especially for challenging scenes where illumination changes, occlusions, motion, textureless and similar features are present, leading to mis-matched points. In this thesis, we develop a self-supervised contrastive deep learning framework for computing robust descriptors, particularly for these challenging situations. We use the TartanAir dataset built explicitly for this task, and in which these difficult scene cases are present.

Our results show that descriptor learning works, improves scores, and that our method is competitive with traditional methods such as ORB. In particular, the invariance built implicitly by training pairs of positive examples through the construction of a trajectory from a sequence of images, as well as the controlled introduction of ambiguous negative examples during training, have a real observable effect on the scores obtained.

**Keywords :** Visual SLAM, SfM, image matching, descriptor, deep learning, contrastive learning

# TABLE DES MATIERES

---

<b>Liste des figures</b>	<b>iv</b>
<b>Liste des tableaux</b>	<b>x</b>
<b>Acronymes et abréviations</b>	<b>xii</b>
<b>Chapitre 1 : Introduction</b>	<b>1</b>
1.1 La mise en correspondance pour le SLAM visuel . . . . .	1
1.2 Détection des points saillants . . . . .	3
1.2.1 Le problème de mise en correspondance . . . . .	3
1.2.2 Les détecteurs de points saillants . . . . .	3
1.3 Descripteurs . . . . .	5
1.3.1 Définition . . . . .	5
1.3.2 Différence entre caractéristiques, points saillants, et descripteurs	6
1.3.3 SIFT (Scale-Invariant Feature Transform) . . . . .	7
1.3.4 ORB (Oriented FAST and Rotated BRIEF) . . . . .	7
1.3.5 Comparaison entre SIFT et ORB . . . . .	9
<b>Chapitre 2 : Apprentissage profond</b>	<b>10</b>
2.1 Apprentissage contrastif . . . . .	10
2.1.1 Deepdesc (Discriminative Learning of Deep Convolutional Feature Point Descriptors) . . . . .	10
2.1.2 SimCLR (Simple Framework for Contrastive Learning of Visual Representations) . . . . .	12

2.2	Apprentissage pour la détection, la description et la mise en correspondance . . . . .	14
2.2.1	SuperPoint . . . . .	14
2.2.2	SuperGlue . . . . .	17
<b>Chapitre 3 :</b>	<b>TartanAir</b>	<b>20</b>
3.1	Présentation du jeu de données . . . . .	20
3.1.1	Conversion des poses . . . . .	21
3.1.2	Mise en correspondance des pixels . . . . .	22
3.1.3	Reconstruction 3D en nuage de points . . . . .	24
3.1.4	Création des LUTs (LookUp Table) . . . . .	25
3.2	Génération des trajectoires . . . . .	26
<b>Chapitre 4 :</b>	<b>Méthode</b>	<b>30</b>
4.1	Hypothèse . . . . .	30
4.2	Architecture du réseau . . . . .	31
4.3	Fonction de coût . . . . .	32
4.4	Détails d'entraînement . . . . .	34
<b>Chapitre 5 :</b>	<b>Expériences</b>	<b>36</b>
5.1	Configuration et hyperparamètres . . . . .	36
5.2	Création des générateurs . . . . .	36
5.3	Dimension des patchs utilisés pour l'entraînement . . . . .	38
5.4	Augmentation de données . . . . .	38
5.5	Variation des hyperparamètres . . . . .	39
<b>Chapitre 6 :</b>	<b>Évaluation</b>	<b>42</b>
6.1	Séparation des classes . . . . .	42

6.2	Tests de mise en correspondance . . . . .	42
6.2.1	KP-RAND : Test de mise en correspondance avec le réseau dans une fenêtre de l'image suivante . . . . .	44
6.2.2	KP-ORB : Test de mise en correspondance avec le réseau et ORB entre images consécutives . . . . .	45
6.2.3	KP-ORB-W : Test de mise en correspondance avec le réseau et ORB entre images non-consécutives . . . . .	46
6.3	Analyse des résultats et discussion . . . . .	47
6.3.1	Différence de classes . . . . .	47
6.3.2	KP-RAND . . . . .	51
6.3.3	KP-ORB . . . . .	68
6.3.4	KP-ORB-W . . . . .	86
6.3.5	Discussion . . . . .	102
<b>Chapitre 7 :</b>	<b>Conclusion</b>	<b>105</b>
<b>Annexe A :</b>		<b>114</b>

## LISTE DES FIGURES

---

1.1	Cadre SLAM visuel . . . . .	2
1.2	Points saillants Harris . . . . .	4
1.3	Points saillants FAST . . . . .	5
1.4	Points saillants SIFT . . . . .	8
2.1	Descripteurs profonds DeepDesc . . . . .	11
2.2	Entraînement SimCLR . . . . .	13
3.1	Échantillon d'images du jeu de données d'entraînement . . . . .	21
3.2	Reconstruction 3D . . . . .	26
3.3	Exemple de LUT . . . . .	27
3.4	Trajectoire . . . . .	29
4.1	Architecture du réseau . . . . .	32
5.1	Sortie du générateur non-augmenté . . . . .	37
5.2	Bruit pour la création d'exemples négatifs . . . . .	39
5.3	Sortie du générateur augmenté . . . . .	40
6.1	Échantillon d'images du jeu de données d'entraînement . . . . .	43
6.2	Différence de classes réseaux non-augmentés . . . . .	49
6.3	Différence de classes réseaux non-augmentés . . . . .	49
6.4	Différence de classes réseaux augmentés . . . . .	50
6.5	KP-RAND 10-155 DELTAS NA . . . . .	51
6.6	KP-RAND 10-155 DELTAS A . . . . .	52

6.7	KP-RAND 10-155 DELTAS NT . . . . .	52
6.8	KP-RAND 10-155 HIST NORMS NA . . . . .	53
6.9	KP-RAND 10-155 HIST NORMS A . . . . .	53
6.10	KP-RAND 10-155 HIST NORMS NT . . . . .	54
6.11	KP-RAND 10-155 MATCHING NA . . . . .	55
6.12	KP-RAND 10-155 MATCHING A . . . . .	56
6.13	KP-RAND 10-155 MATCHING NT . . . . .	57
6.14	KP-RAND 13-34 DELTAS NA . . . . .	59
6.15	KP-RAND 13-34 DELTAS A . . . . .	60
6.16	KP-RAND 13-34 DELTAS NT . . . . .	60
6.17	KP-RAND 13-34 HIST NORMS NA . . . . .	61
6.18	KP-RAND 13-34 HIST NORMS A . . . . .	61
6.19	KP-RAND 13-34 HIST NORMS NT . . . . .	62
6.20	KP-RAND 13-34 MATCHING NA . . . . .	63
6.21	KP-RAND 13-34 MATCHING A . . . . .	64
6.22	KP-RAND 13-34 MATCHING NT . . . . .	65
6.23	KP-ORB 10-155 DELTAS NA . . . . .	69
6.24	KP-ORB 10-155 DELTAS A . . . . .	69
6.25	KP-ORB 10-155 DELTAS NT ORB . . . . .	70
6.26	KP-ORB 10-155 HIST NORMS NA . . . . .	70
6.27	KP-ORB 10-155 HIST NORMS A . . . . .	70
6.28	KP-ORB 10-155 HIST NORMS NT ORB . . . . .	71
6.29	KP-ORB 10-155 MATCHING NA . . . . .	72
6.30	KP-ORB 10-155 MATCHING A . . . . .	73
6.31	KP-ORB 10-155 MATCHING NT ORB . . . . .	74
6.32	KP-ORB 10-155 ROCS . . . . .	75

6.33 KP-ORB 13-34 DELTAS NA . . . . .	77
6.34 KP-ORB 13-34 DELTAS A . . . . .	77
6.35 KP-ORB 13-34 DELTAS NT ORB . . . . .	78
6.36 KP-ORB 13-34 HIST NORMS NA . . . . .	78
6.37 KP-ORB 13-34 HIST NORMS A . . . . .	79
6.38 KP-ORB 13-34 HIST NORMS NT ORB . . . . .	79
6.39 KP-ORB 13-34 MATCHING NA . . . . .	80
6.40 KP-ORB 13-34 MATCHING A . . . . .	81
6.41 KP-ORB 13-34 MATCHING NT ORB . . . . .	82
6.42 KP-ORB 13-34 ROCS . . . . .	83
6.43 KP-ORB-W 1-7 DELTAS NA . . . . .	86
6.44 KP-ORB-W 1-7 DELTAS A . . . . .	87
6.45 KP-ORB-W 1-7 DELTAS NT ORB . . . . .	87
6.46 KP-ORB-W 1-7 HIST NORMS NA . . . . .	88
6.47 KP-ORB-W 1-7 HIST NORMS A . . . . .	88
6.48 KP-ORB-W 1-7 HIST NORMS NT ORB . . . . .	88
6.49 KP-ORB-W 1-7 MATCHING NA . . . . .	89
6.50 KP-ORB-W 1-7 MATCHING A . . . . .	90
6.51 KP-ORB-W 1-7 MATCHING NT ORB . . . . .	91
6.52 KP-ORB-W 1-7 ROCS . . . . .	92
6.53 KP-ORB-W 92-97 DELTAS NA . . . . .	94
6.54 KP-ORB-W 92-97 DELTAS A . . . . .	94
6.55 KP-ORB-W 92-97 DELTAS NT ORB . . . . .	95
6.56 KP-ORB-W 92-97 HIST NORMS NA . . . . .	95
6.57 KP-ORB-W 92-97 HIST NORMS A . . . . .	96
6.58 KP-ORB-W 92-97 HIST NORMS NT ORB . . . . .	96

6.59 KP-ORB-W 92-97 MATCHING NA . . . . .	97
6.60 KP-ORB-W 92-97 MATCHING A . . . . .	98
6.61 KP-ORB-W 92-97 MATCHING NT ORB . . . . .	99
6.62 KP-ORB-W 92-97 ROCS . . . . .	100
A.1 KP-RAND 10-260 DELTAS NA . . . . .	114
A.2 KP-RAND 10-260 DELTAS A . . . . .	115
A.3 KP-RAND 10-260 DELTAS NT . . . . .	115
A.4 KP-RAND 10-260 HIST NORMS NA . . . . .	116
A.5 KP-RAND 10-260 HIST NORMS A . . . . .	116
A.6 KP-RAND 10-260 HIST NORMS NT . . . . .	116
A.7 KP-RAND 10-260 MATCHING NA . . . . .	117
A.8 KP-RAND 10-260 MATCHING A . . . . .	118
A.9 KP-RAND 10-260 MATCHING NT . . . . .	119
A.10 KP-RAND 12-281 DELTAS NA . . . . .	120
A.11 KP-RAND 12-281 DELTAS A . . . . .	120
A.12 KP-RAND 12-281 DELTAS NT . . . . .	121
A.13 KP-RAND 12-281 HIST NORMS NA . . . . .	121
A.14 KP-RAND 12-281 HIST NORMS A . . . . .	122
A.15 KP-RAND 12-281 HIST NORMS NT . . . . .	122
A.16 KP-RAND 12-281 MATCHING NA . . . . .	123
A.17 KP-RAND 12-281 MATCHING A . . . . .	124
A.18 KP-RAND 12-281 MATCHING NT . . . . .	125
A.19 KP-ORB 10-260 DELTAS NA . . . . .	127
A.20 KP-ORB 10-260 DELTAS A . . . . .	127
A.21 KP-ORB 10-260 DELTAS NT ORB . . . . .	128
A.22 KP-ORB 10-260 HIST NORMS NA . . . . .	128

A.23 KP-ORB 10-260 HIST NORMS A . . . . .	129
A.24 KP-ORB 10-260 HIST NORMS NT ORB . . . . .	129
A.25 KP-ORB 10-260 MATCHING NA . . . . .	130
A.26 KP-ORB 10-260 MATCHING A . . . . .	131
A.27 KP-ORB 10-260 MATCHING NT ORB . . . . .	132
A.28 KP-ORB 10-260 ROCS . . . . .	133
A.29 KP-ORB 12-281 DELTAS NA . . . . .	135
A.30 KP-ORB 12-281 DELTAS A . . . . .	135
A.31 KP-ORB 12-281 DELTAS NT ORB . . . . .	136
A.32 KP-ORB 12-281 HIST NORMS NA . . . . .	136
A.33 KP-ORB 12-281 HIST NORMS A . . . . .	137
A.34 KP-ORB 12-281 HIST NORMS NT ORB . . . . .	137
A.35 KP-ORB 12-281 MATCHING NA . . . . .	138
A.36 KP-ORB 12-281 MATCHING A . . . . .	139
A.37 KP-ORB 12-281 MATCHING NT ORB . . . . .	140
A.38 KP-ORB 12-281 ROCS . . . . .	141
A.39 KP-ORB-W 93-113 DELTAS NA . . . . .	143
A.40 KP-ORB-W 93-113 DELTAS A . . . . .	143
A.41 KP-ORB-W 93-113 DELTAS NT ORB . . . . .	144
A.42 KP-ORB-W 93-113 HIST NORMS NA . . . . .	144
A.43 KP-ORB-W 93-113 HIST NORMS A . . . . .	145
A.44 KP-ORB-W 93-113 HIST NORMS NT ORB . . . . .	145
A.45 KP-ORB-W 93-113 MATCHING NA . . . . .	146
A.46 KP-ORB-W 93-113 MATCHING A . . . . .	147
A.47 KP-ORB-W 93-113 MATCHING NT ORB . . . . .	148
A.48 KP-ORB-W 93-113 ROCS . . . . .	149

A.49 KP-ORB-W 244-270 DELTAS NA . . . . .	151
A.50 KP-ORB-W 244-270 DELTAS A . . . . .	151
A.51 KP-ORB-W 244-270 DELTAS NT ORB . . . . .	152
A.52 KP-ORB-W 244-270 HIST NORMS NA . . . . .	152
A.53 KP-ORB-W 244-270 HIST NORMS A . . . . .	153
A.54 KP-ORB-W 244-270 HIST NORMS NT ORB . . . . .	153
A.55 KP-ORB-W 244-270 MATCHING NA . . . . .	154
A.56 KP-ORB-W 244-270 MATCHING A . . . . .	155
A.57 KP-ORB-W 244-270 MATCHING NT ORB . . . . .	156
A.58 KP-ORB-W 244-270 ROCS . . . . .	157

## LISTE DES TABLEAUX

---

5.1	Liste des méthodes expérimentées . . . . .	41
6.1	Résultats quantitatifs regroupant les métriques des différentes méthodes utilisées pour réaliser la mise en correspondance entre les points saillants choisis aléatoirement de deux images consécutives. . . . .	58
6.2	Résultats quantitatifs regroupant les métriques des différentes méthodes utilisées pour réaliser la mise en correspondance entre les points saillants choisis aléatoirement de deux images consécutives. . . . .	66
6.3	Résultats quantitatifs regroupant les métriques des différentes méthodes utilisées pour réaliser la mise en correspondance entre les points saillants ORB de deux images consécutives. . . . .	76
6.4	Résultats quantitatifs regroupant les métriques des différentes méthodes utilisées pour réaliser la mise en correspondance entre les points saillants ORB de deux images consécutives. . . . .	84
6.5	Résultats quantitatifs regroupant les métriques des différentes méthodes utilisées pour réaliser la mise en correspondance entre les points saillants ORB de deux images non-consécutives. . . . .	93
6.6	Résultats quantitatifs regroupant les métriques des différentes méthodes utilisées pour réaliser la mise en correspondance entre les points saillants ORB de deux images non-consécutives. . . . .	101
A.1	Résultats quantitatifs regroupant les métriques des différentes méthodes utilisées pour réaliser la mise en correspondance entre les points saillants choisis aléatoirement de deux images consécutives. . . . .	119

A.2	Résultats quantitatifs regroupant les métriques des différentes méthodes utilisées pour réaliser la mise en correspondance entre les points saillants choisis aléatoirement de deux images consécutives. . . . .	126
A.3	Résultats quantitatifs regroupant les métriques des différentes méthodes utilisées pour réaliser la mise en correspondance entre les points saillants ORB de deux images consécutives. . . . .	134
A.4	Résultats quantitatifs regroupant les métriques des différentes méthodes utilisées pour réaliser la mise en correspondance entre les points saillants ORB de deux images consécutives. . . . .	142
A.5	Résultats quantitatifs regroupant les métriques des différentes méthodes utilisées pour réaliser la mise en correspondance entre les points saillants ORB de deux images non-consécutives. . . . .	150
A.6	Résultats quantitatifs regroupant les métriques des différentes méthodes utilisées pour réaliser la mise en correspondance entre les points saillants ORB de deux images non-consécutives. . . . .	158

## LISTE DES ACRONYMES ET ABRÉVIATIONS

---

<b>ACC</b>	Accuracy
<b>AUC</b>	Area Under the Curve
<b>BA</b>	Bundle Adjustment
<b>BRIEF</b>	Binary Robust Independent Elementary Features
<b>CNN</b>	Convolutional Neural Networks
<b>Deepdesc</b>	Discriminative Learning of Deep Convolutional Feature Point Descriptors
<b>EKF</b>	Extended Kalman Filter
<b>FAST</b>	Features from Accelerated Segment Test
<b>LUT</b>	LookUp Table
<b>ORB</b>	Oriented FAST and Rotated BRIEF
<b>ReLU</b>	Fonction Unité Linéaire Rectifiée
<b>RGB</b>	Red Green Blue
<b>ROC</b>	Receiver Operating Characteristic
<b>SfM</b>	Structure from Motion
<b>SIFT</b>	Scale-Invariant Feature Transform
<b>SimCLR</b>	Simple Framework for Contrastive Learning of Visual Representations
<b>SLAM</b>	Simultaneous Localization And Mapping
<b>VP</b>	Vrai Positif
<b>FP</b>	Faux Positif
<b>VN</b>	Vrai Négatif
<b>FN</b>	Faux Négatif
<b>TPR</b>	True Positive Rate
<b>FPR</b>	False Positive Rate

## REMERCIEMENTS

---

La réalisation du présent mémoire m'a permis de renforcer et développer mes connaissances en vision par ordinateur, aussi bien par l'étude des méthodes traditionnelles que celles par apprentissage profond. Je remercie mes proches : famille, amis, collègues, pour leur soutien et intérêt dans mes travaux. Enfin, je remercie mon directeur de recherche Sébastien Roy ainsi que le corps enseignant de l'Université pour la transmission d'un précieux savoir.

# Chapitre 1

## INTRODUCTION

---

### *1.1 La mise en correspondance pour le SLAM visuel*

Le problème de SLAM (Simultaneous Localization And Mapping) consiste à localiser un agent qui se déplace dans un environnement et en même temps reconstruire cet environnement [9]. Le SLAM se résout comme suit : l'agent qui se déplace à chaque intervalle de temps dans l'environnement va prendre des mesures de cet environnement. C'est l'estimation du déplacement relatif entre intervalles de temps qui va permettre de calculer les positions successives de l'agent ainsi que la localisation des points de l'environnement.

On peut catégoriser le SLAM selon le type de capteur et donc de signal utilisé pour prendre la mesure. Dans ce mémoire, nous nous intéressons à une sous-catégorie du SLAM, appelée SLAM visuel, c'est-à-dire un système qui utilise une caméra comme capteur et des images comme signal, et nous exploitons une seule image par intervalle de temps. Il s'agit donc de SLAM monoculaire.

À partir de SLAM visuel monoculaire, on peut catégoriser le SLAM en fonction de sa méthode de résolution. Par exemple, le premier système de SLAM visuel monoculaire MonoSLAM [9], utilise des EKF (Extended Kalman Filter) pour mettre à jour la position des caméras et la carte de l'environnement. À contrario, ORB-SLAM utilise une méthode de BA (Bundle Adjustment) global, qui va prendre en compte l'ensemble des points appartenant aux images qui apportent suffisamment de nouvelles informations ("keyframes"), plutôt que de faire une estimation incrémentale comme le fait MonoSLAM. La méthode d'optimisation globale de SLAM est aujourd'hui la plus utilisée de

par son efficacité et sa justesse, comme il l'a été démontré dans [41].

De façon générale, et comme nous pouvons le voir dans la Figure 1.1, un système de SLAM visuel monoculaire avec méthode d'optimisation globale comporte les parties suivantes : une initialisation, une estimation du mouvement entre l'image actuelle et la précédente, une optimisation de l'ensemble de la trajectoire et de la carte, et enfin une méthode de fermeture de boucle (loop closure).

Cette structure varie en fonction des méthodes utilisées. Cette variation existe notamment sur la partie d'estimation du mouvement appelée "Odométrie visuelle", qui peut être qualifiée de directe, lorsque l'estimation du mouvement de la caméra est déterminée à partir du déplacement de tous les pixels (dense) de l'image de départ, ou indirecte quand on utilise uniquement un ensemble de points saillants (éparse) de l'image de départ [23].

C'est sur les systèmes utilisant la méthode indirecte que nous avons apporté des améliorations en travaillant sur l'étape de mise en correspondance.

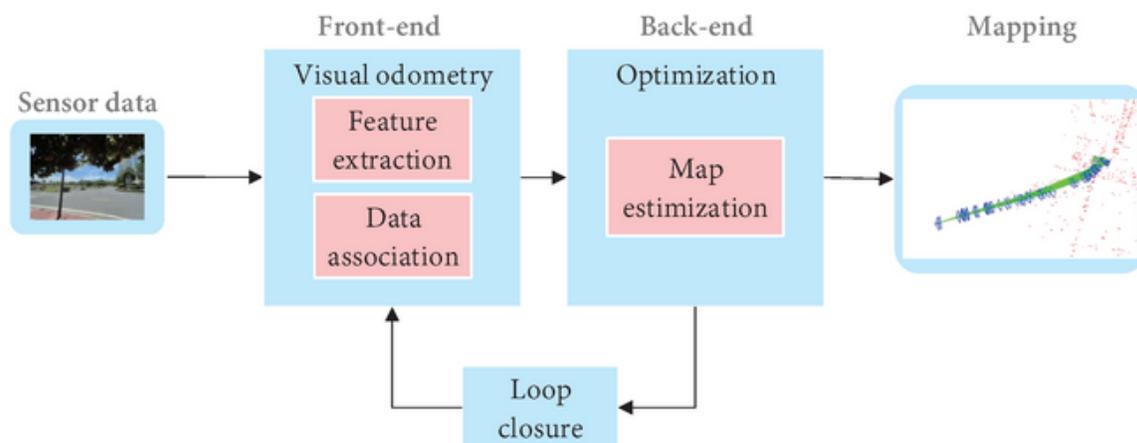


FIGURE 1.1 – Organisation du SLAM visuel en modules. Source : [7].

## **1.2 Détection des points saillants**

### *1.2.1 Le problème de mise en correspondance*

Dans le contexte de tâches de reconstruction 3D comme le SfM (Structure from Motion) ou le SLAM, il faut résoudre le problème de mise en correspondance entre deux images. Ce problème consiste à mettre en correspondance des parties de deux images. Une correspondance correspond à une paire de pixels, où chaque pixel appartient à une des images, et pour lesquels ces pixels représentent la projection du même point du monde sur leur plan image respectif. La mise en correspondance de plusieurs paires de pixels entre les deux images permet d'estimer le déplacement de caméra relatif qui a été effectué entre les deux prises, et donc de retrouver la trajectoire absolue de la caméra en connaissant la position absolue de la première prise.

Comme le SLAM vise une application en temps réel, la mise en correspondance éparse ("sparse") est préférée dans les méthodes par souci de performance [7]. Il faut donc choisir quels points suivre lors d'un parcours de suivi ("tracking"). Pour suivre un point, il faut que ce point soit par nature le plus reconnaissable entre les images, il faut donc qu'il soit particulier. Généralement, c'est un point saillant qui est choisi, c'est-à-dire un point qui représente une coupure entre deux parties de l'image, tel qu'un coin. On appelle ces points des points saillants, et la première étape d'une mise en correspondance est leur détection dans l'image.

### *1.2.2 Les détecteurs de points saillants*

La méthode des coins de Harris développée dans [16], dont l'application sur une image est présentée en Figure 1.2, est l'une des méthodes reconnues pour réaliser cette détection de points saillants. Cette méthode consiste à calculer les dérivées verticales et horizontales de l'image. Mathématiquement, la méthode des coins de Harris définit un point saillant comme un point dont la dérivée varie le plus au sein de l'image,

représentant bien l'idée de coupure.

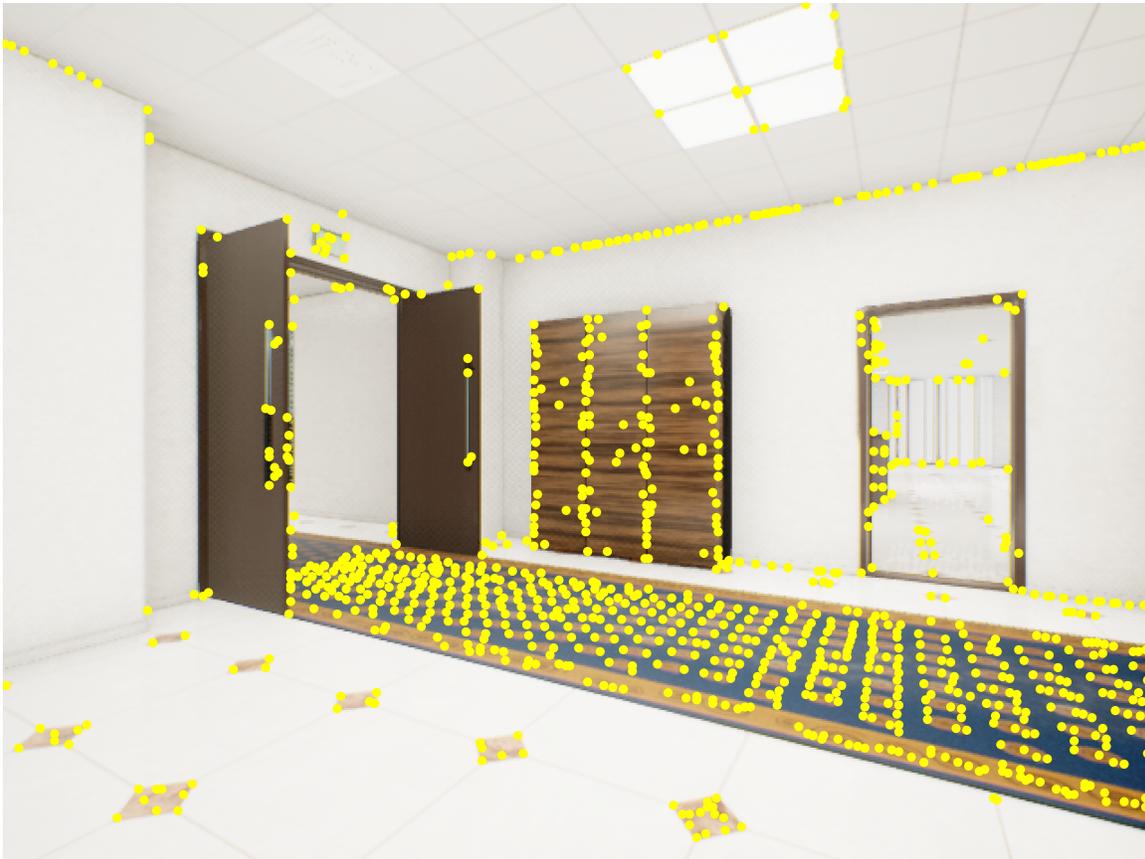


FIGURE 1.2 – Détection des points saillants avec le détecteur des coins de Harris.

Il existe aussi la méthode de détection FAST (Features from Accelerated Segment Test) [32], dont l'application sur une image est présentée en Figure 1.3. Cette méthode est quant à elle plus rapide que la méthode des coins de Harris et va considérer le voisinage circulaire de chaque point et comparer l'intensité du pixel central avec ses voisins pour déterminer s'il est un point saillant.



FIGURE 1.3 – Détection des points saillants avec le détecteur FAST.

### **1.3 Descripteurs**

#### **1.3.1 Définition**

Les points saillants détectés par l'une des méthodes présentées ci-dessus ne suffisent pas à réaliser une mise en correspondance robuste. En effet, pour retrouver quel pixel de la première image correspond à quel pixel de la seconde, il faut comparer ces pixels. Or, il est difficile de comparer des pixels seulement à l'aide de leur intensité. Si on prend le descripteur le plus simple d'un pixel d'une image RGB (Red Green Blue), ce serait le vecteur correspondant aux trois composantes de sa couleur. Dans ce cas, un pixel correspondrait fort à tous les autres pixels uniquement sur la base de sa couleur.

Cependant, un même point peut subir des changements de couleur en fonction d'où l'image a été prise et des paramètres d'éclairages. De plus, des pixels pourtant bien différents pourraient partager la même couleur.

Nous voyons bien à travers ces exemples le raisonnement qui a poussé la communauté scientifique à développer des descripteurs robustes pour réaliser la mise en correspondance. Un descripteur est un vecteur qui représente un point dans une image, mais en y intégrant les informations de son voisinage. Il représente donc une région dont le centre est un point saillant. Il doit avoir la qualité de pouvoir être retrouvé dans différentes images d'une trajectoire filmée, ainsi que de ne pas être confondu avec d'autres pixels. On parle de répétabilité et d'unicité du descripteur.

En effet, un point pris dans une image doit pouvoir être retrouvé dans d'autres, même s'il a subi plusieurs transformations de couleur, floutage, et d'illumination : il doit être invariant à ces changements de conditions. En ce qui concerne l'unicité, il faut que le pixel soit suffisamment unique au sein de son voisinage et dans l'image pour ne pas générer de correspondance faussement positive. Pour atteindre ce double objectif, les méthodes qui calculent les descripteurs utilisent des informations de voisinage du point saillant et certaines prennent aussi en compte son orientation.

### *1.3.2 Différence entre caractéristiques, points saillants, et descripteurs*

Notons aussi que dans la littérature, les termes points saillants, descripteurs et caractéristique ("features") sont souvent utilisés ensemble ou de façon interchangeable, portant à confusion. Pour clarifier, il faut distinguer les différentes étapes du processus de mise en correspondance. La première étape est l'étape de détection des points saillants, pour laquelle on utilise un détecteur de points saillants. La seconde correspond à la description de ces points saillants. Enfin, on parle de caractéristique pour parler du résultat final, le descripteur, mais certaines méthodes utilisent un détecteur et une façon de décrire le point saillant spécifiques. Dès lors, on peut parler de carac-

téristique comme du descripteur résultant de l'application de ces deux étapes p. ex. les caractéristiques SIFT ou les caractéristiques ORB.

### 1.3.3 *SIFT (Scale-Invariant Feature Transform)*

La méthode SIFT est une méthode reconnue en vision par ordinateur développée dans [24]. Son application sur une image est visible sur la Figure 1.4. Comme son nom l'indique, l'objectif de cette méthode est de fournir des descripteurs qui ne varient pas en fonction de l'échelle de l'image. Ces descripteurs ont aussi pour but de fournir une robustesse aux changements d'orientation, d'illumination, et à l'occultation partielle que l'on peut retrouver dans une séquence d'images.

La première étape de l'algorithme SIFT est la détection des points saillants, qui se fait en calculant la différence de gaussiennes sur une pyramide d'images, c'est-à-dire un ensemble de plusieurs images progressivement réduites du même facteur par rapport à l'image d'origine. Ensuite, l'algorithme assigne une orientation à chaque point saillant sélectionné dans l'étape précédente.

Cette orientation est déterminée par le calcul du gradient autour du point permettant de déterminer la direction dominante. C'est cette étape qui permet de bâtir l'invariance à la rotation, car peu importe comment le point sera transformé, l'orientation correspondant à son voisinage changerait en même temps que lui. Enfin, vient l'étape de calcul des descripteurs représentant ces points saillants. Ces derniers sont calculés à l'aide d'histogrammes des orientations du gradient, pondérés avec une fenêtre gaussienne. Cette étape assure l'invariance aux changements d'illumination.

### 1.3.4 *ORB (Oriented FAST and Rotated BRIEF)*

Contrairement aux descripteurs SIFT conçus principalement pour introduire de l'invariance à plusieurs paramètres, le but des descripteurs ORB [34] est de permettre d'améliorer la performance de rapidité pour le calcul et la mise en correspondance.



FIGURE 1.4 – Détection des points saillants avec la méthode SIFT

La méthode ORB combine un détecteur FAST pour détecter les points saillants, avec un calcul spécifique de descripteurs. La particularité des descripteurs ORB est qu'ils sont des vecteurs de 256 bits, choix de conception pour des questions de performance. En effet, les descripteurs ORB sont une variante des descripteurs BRIEF (Binary Robust Independent Elementary Features) [4].

Les descripteurs BRIEF sont des descripteurs qui représentent la différence d'intensité entre les pixels au voisinage d'un point. Tout comme SIFT, pour bâtir l'invariance à l'orientation et à l'échelle, ORB modifie légèrement le calcul de BRIEF. Pour calculer l'orientation, ORB utilise les gradients autour de l'image. Pour bâtir l'invariance à l'échelle, ORB utilise également une pyramide d'images.

ORB est fameusement utilisé dans le système ORB-SLAM [27] qui repose sur le calcul rapide et efficace de ces descripteurs de 256 bits pour réaliser la mise en correspondance et aussi reconnaître le lieu.

### 1.3.5 Comparaison entre SIFT et ORB

Le choix entre les descripteurs ORB et SIFT dépend de l'application. La méthode SIFT sera meilleure que ORB sur la précision, dans le sens où elle sera plus robuste à différents changements grâce à l'invariance bâtie dans le calcul de ses descripteurs. SIFT est donc un descripteur de choix pour des applications de reconstruction 3D telles que le SfM, comme c'est le cas dans la méthode COLMAP [8]. D'autre part, la méthode ORB sera moins précise mais bien plus rapide et donc particulièrement adaptée à des applications requérant un traitement en temps réel telles que le SLAM.

Bien que les descripteurs SIFT et ORB soient les descripteurs définissant l'état de l'art, de nombreuses variantes ont été développées dans le but de les améliorer. Les méthodes SIFT et ORB sont aujourd'hui utilisées dans plusieurs applications et ont aussi servi de base de développement et de comparaison pour d'autres méthodes. On désigne ces descripteurs comme des descripteurs faits main ("handcrafted") dans le sens où ils ont été conçus avec un algorithme précis pour résoudre la tâche de mise en correspondance. Au contraire, parmi les méthodes émergentes, il existe les méthodes apprises, qui ont essayé de rebâtir l'invariance recherchée par apprentissage profond en employant des réseaux de neurones.

## Chapitre 2

### APPRENTISSAGE PROFOND

---

#### 2.1 Apprentissage contrastif

##### 2.1.1 Deepdesc (*Discriminative Learning of Deep Convolutional Feature Point Descriptors*)

Le but de la méthode présentée dans [39] est de fournir une alternative aux descripteurs faits main en entraînant un réseau de neurones pour qu'il apprenne une représentation discriminante de descripteurs. Pour cela, cette méthode utilise l'apprentissage contrastif avec deux réseaux convolutifs (CNN) qui partagent leurs poids, comme on peut le voir en Figure 2.1.

Leur entraînement est directement lié à la tâche finale visée : les données utilisées sont des paires de patches d'images qui correspondent ou non. La fonction de coût optimisée est la distance euclidienne entre les deux descripteurs calculés par les réseaux. Dans cette étude, les auteurs affirment que l'idée clé est d'introduire un échantillonnage négatif en fournissant des exemples négatifs difficiles à discerner pour forcer l'entraînement.

La fonction de coût utilisée est la suivante :

$$l(\mathbf{x}_1, \mathbf{x}_2) = \begin{cases} \|D(\mathbf{x}_1) - D(\mathbf{x}_2)\|_2, & p_1 = p_2 \\ \max(0, C - \|D(\mathbf{x}_1) - D(\mathbf{x}_2)\|_2), & p_1 \neq p_2 \end{cases} \quad (2.1)$$

où  $p_1$  et  $p_2$  sont les indices des points 3D projetés respectivement sur  $x_1$  et  $x_2$ , et où  $D(\mathbf{x}_1)$  et  $D(\mathbf{x}_2)$  représentent leurs descripteurs respectifs produits par le réseau.  $C$  est une constante positive qui pénalise la fonction de coût négative maximale quand les descripteurs sont similaires.

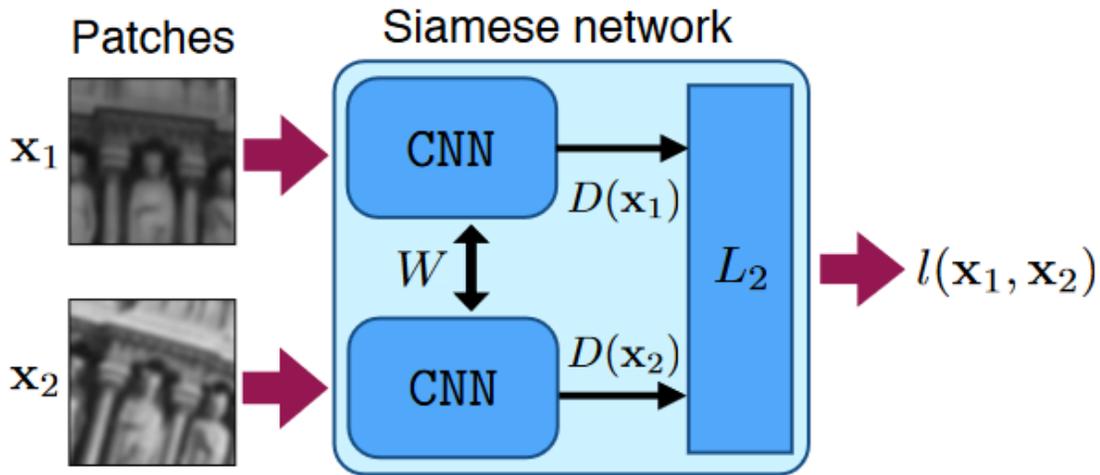


FIGURE 2.1 – Architecture du réseau de neurone entraîné par apprentissage contrastif de la méthode DeepDesc. Source : [39].

Pour créer le jeu de données d'entraînement, les auteurs génèrent  $k$  patches à partir d'un nombre  $m$  de points 3D dans l'espace, tel que  $m \leq k$ . Pour chaque point 3D,  $c_i$  est défini comme le nombre de patches créés à partir de celui-ci. Le nombre de paires positives  $P$  qu'il est possible de créer est défini par :

$$P = \sum_{i=1}^m \frac{c_i(c_i - 1)}{2} \quad (2.2)$$

Et le nombre de paires négatives  $N$  qu'il est possible de créer est défini par :

$$N = \sum_{i=1}^m c_i(k - c_i) \quad (2.3)$$

Il est donc possible ensuite d'échantillonner un nombre  $s_p$  de paires positives et un nombre  $s_n$  de paires négatives pour chaque paquet ("batch"). Pour améliorer l'entraînement, les exemples négatifs et positifs pour lesquels le réseau s'est le plus trompé et qui sont proches dans l'espace des vecteurs de sortie sont conservés, puis une partie de ceux-ci seront réintroduits dans le prochain paquet, forçant le réseau à voir plus de ces exemples difficiles et à y adapter ses poids.

De plus, la méthode est évaluée sur des jeux de données où de la rotation a été introduite et des jeux de données comportant des changements d'illumination, montrant que leurs descripteurs appris sont aussi performants sinon meilleurs pour la tâche de mise en correspondance que les méthodes traditionnelles.

### 2.1.2 *SimCLR (Simple Framework for Contrastive Learning of Visual Representations)*

La méthode SimCLR présentée dans [6] est un autre bon exemple du succès appliqué des réseaux convolutifs sur des images dans un contexte d'entraînement contrastif. L'entraînement contrastif est un paradigme d'entraînement auto-supervisé, ce qui signifie que ce sont les données d'entraînement non-étiquetées qui permettent de superviser l'apprentissage. En effet, pour l'apprentissage contrastif, on va fournir au réseau des paires d'images qui représentent la même chose ou non. Par exemple, deux patches qui représentent le même objet formeront une paire positive, tandis que deux patches différents formeront une paire négative. Le réseau doit apprendre à fournir des vecteurs en sortie tels que deux images similaires auront des représentations tendant à être similaires dans l'espace des vecteurs de sortie tandis que les images formant une paire négative auront des vecteurs très éloignés dans ce même espace.

L'architecture du réseau est décrite en Figure 2.2. À partir d'une image, deux transformations  $t$  et  $t'$  sont échantillonnées puis lui sont appliquées. Ceci produit deux images différentes  $x_i$  et  $x_j$ . Ces deux images vont ensuite être fournies en entrée au sous-réseau  $f$  (un ResNet [18]) pour en obtenir des représentations intermédiaires  $\mathbf{h}_i$  et  $\mathbf{h}_j$ . Le ResNet (Residual neural network) est un réseau très utilisé en apprentissage profond qui comporte plusieurs couches et qui emploie un paradigme d'apprentissage résiduel qui consiste à connecter l'entrée d'un sous-réseau ou bloc résiduel (plusieurs couches) avec sa sortie. Enfin, on applique  $g$  à ces représentation intermédiaires, qui est un MLP (Multilayer perceptron) avec une couche linéaire suivi d'une fonction d'activation ReLU (Fonction Unité Linéaire Rectifiée), pour obtenir les représentations  $z_i$ ,

et  $z_j$ .

Pour réaliser l'entraînement, le réseau va recevoir des paires positives et négatives formées à partir d'images d'un jeu de données non-étiqueté. Pour former une paire positive, la méthode va prendre une image puis lui faire subir ces transformations telles que les images résultantes soient différentes mais appartiennent à la même classe d'image. Lors d'une passe d'entraînement, cette opération est appliquée pour N images ce qui résultera en N paires positives. Les images originales et transformées n'appartenant pas à la même classe sont utilisées pour former des paires négatives.

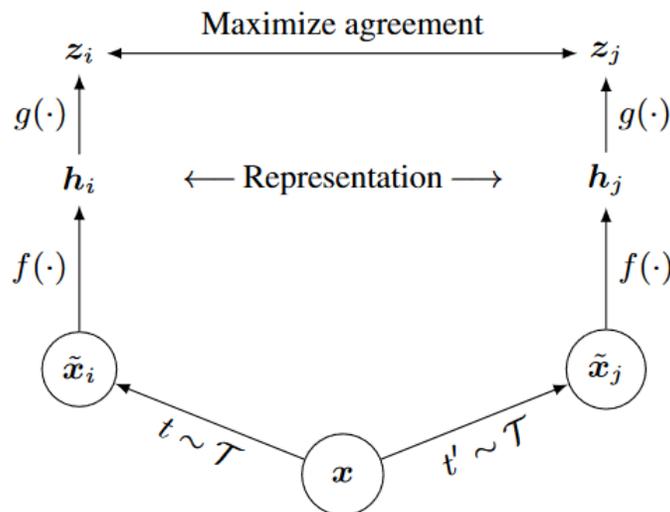


FIGURE 2.2 – Organisation de l'entraînement par apprentissage contrastif de la méthode SimCLR, où  $t$  et  $t'$  sont des transformations appartenant à un ensemble de transformations  $\mathcal{T}$ ,  $x_i$  et  $x_j$  sont les images ayant reçu ces transformations respectives,  $f$  est un sous-réseau à l'architecture ResNet,  $h_i$  et  $h_j$  sont des représentations intermédiaires,  $g$  est un MLP suivi d'une fonction ReLU, et  $z_i$ , et  $z_j$  sont les représentations finales produites par le réseau. Source : [6].

La fonction de coût utilisée dans l'équation 2.4 est une fonction mesurant la similarité ou dissimilarité entre les paires. Ce mode d'entraînement va permettre au réseau

d'apprendre une représentation fidèle de chaque image :

$$l_{i,j} = -\log \frac{\exp\left(\frac{\text{sim}(\mathbf{z}_i, \mathbf{z}_j)}{\tau}\right)}{\sum_{k=1}^{2N} \mathbb{1}_{k \neq j} \exp\left(\frac{\text{sim}(\mathbf{z}_i, \mathbf{z}_k)}{\tau}\right)} \quad (2.4)$$

où  $(\mathbf{z}_i, \mathbf{z}_j)$  représentent les vecteurs de sortie,  $\tau$  représente une constante appelée température, et  $\text{sim}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u}^T \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}$ .

L'objectif d'une telle méthode d'apprentissage contrastif est d'utiliser le réseau entraîné pour des tâches sous-jacentes après l'avoir entraîné de façon auto-supervisée. En effet, une fois entraîné, il est par exemple possible d'enlever la dernière couche de projection linéaire et de la remplacer avec une couche de classification.

## 2.2 Apprentissage pour la détection, la description et la mise en correspondance

### 2.2.1 SuperPoint

Le problème que SuperPoint [12] essaie de résoudre est de générer les points saillants d'une image avec les descripteurs les représentant ainsi que leur région avoisinante. Pour cela, les auteurs utilisent un réseau de neurones qui va apprendre à détecter ces points saillants et produire des descripteurs à partir d'un entraînement auto-supervisé.

Cette méthode est utile pour les problèmes de SLAM et de SfM, où il est nécessaire de détecter des points saillants sur différentes images. La résolution de ces problèmes s'appuie sur la mise en correspondance des points saillants entre les images, en comparant leurs descripteurs. L'objectif est de surpasser les méthodes traditionnelles en espérant trouver de meilleurs points saillants ou de trouver des points saillants qui ne sont pas détectés par d'autres méthodes.

Le but du réseau est d'apprendre à détecter les points saillants et à les exprimer sous forme de descripteurs. L'utilisation du réseau en inférence permettra de réaliser ces deux opérations à partir d'une image fournie en entrée. L'originalité de ce travail vient

du fait qu'il s'agit d'une méthode auto-supervisée, qui ne repose ni sur des annotations humaines ni sur des détecteurs pour apprendre.

En ce qui concerne l'utilisation de détecteurs connus pour la supervision, les auteurs mentionnent que le problème de ce type de supervision est qu'il forcerait certains points saillants à être détectés, mais cela reviendrait à limiter la découverte de points saillants aux résultats produits par les méthodes traditionnelles, ce qui ne permettrait pas au réseau d'apprendre à découvrir par lui-même de nouveaux points. Il faut aussi noter que le réseau utilise des données synthétiques pour réaliser un pré-entraînement qui permettra d'automatiser l'étiquetage des images. L'article suit une procédure d'entraînement qui comporte trois parties : le pré-entraînement, l'auto-étiquetage, et l'apprentissage joint. Le pré-entraînement consiste à entraîner un réseau de neurones de détection de base sur des formes synthétiques étiquetées et augmentées par des homographies. L'auto-étiquetage consiste à créer un jeu de données d'images réelles étiquetées grâce notamment au réseau de détection de base. Enfin, l'entraînement joint permet d'apprendre à générer les points saillants et descripteurs grâce aux deux parties précédentes.

La création du jeu de données est réalisée en appliquant à chaque image non-étiquetée une série d'homographies. Le modèle utilisé considère qu'appliquer une transformation sur un point revient à appliquer cette même transformation sur l'image quand on lui applique une fonction covariante sur ce point et est défini comme suit :

$$\mathbf{x} = f_{\theta}(I) \tag{2.5}$$

$$\mathcal{H}\mathbf{x} = f_{\theta}(\mathcal{H}(I)) \tag{2.6}$$

$$\mathbf{x} = \mathcal{H}^{-1}f_{\theta}(\mathcal{H}(I)) \tag{2.7}$$

où  $I$  est une image dans laquelle on considère un point  $\mathbf{x}$ ,  $\mathcal{H}$  est une homographie

aléatoire, et  $f_\theta$  est une fonction qui doit être covariante avec  $\mathcal{H}$ .

Ensuite, on applique le détecteur de base pré-entraîné à l'étape précédente sur l'image originale et ses augmentations, ce qui permet d'obtenir pour chacune de ces images les points saillants. Enfin, on applique l'homographie inverse sur chacune des augmentations, ce qui donne plusieurs versions de l'image originale, différant par leurs points saillants. On agrège ensuite tous les points saillants pour former une seule image étiquetée.

En appliquant la procédure sur toutes les images du jeu de données, on obtient un jeu de données d'images naturelles étiquetées. Le réseau se compose de plusieurs couches de convolutions et de pooling qui réduisent progressivement la résolution de l'image.

Pour le pré-entraînement, un détecteur de base est utilisé pour apprendre à détecter les points saillants sur un jeu de données synthétiques composé d'images de formes géométriques simples étiquetées au niveau des jonctions, au centre des ellipses, et au niveau des extrémités de lignes. Le jeu de données synthétiques est aussi augmenté à l'aide d'homographies. L'entrée du réseau est donc une image avec sa référence ("ground-truth") de positions des points saillants.

Lors de l'entraînement, la sortie du réseau est un tenseur de dimensions  $(N, C, H, W)$  où  $N$  représente le nombre d'éléments dans un paquet,  $C$  représente le nombre de canaux, et  $H$  et  $W$  représentent la hauteur et la largeur de l'image. En appliquant la fonction Softmax le long de la dimension  $C$ , le but est de faire ressortir un "gagnant". Le pixel gagnant peut donc être situé sur l'un des 65 canaux de cette sortie. Après l'application de la fonction Softmax, la 65ième dimension va ensuite être ignorée : le système transforme la sortie  $(N, 65, H, W)$  en  $(N, 64, H, W)$ . Cela signifie que si le gagnant du Softmax était au 65ième canal, alors les valeurs aux autres 64 canaux tendent vers 0 et donc que ce pixel n'est pas considéré comme un point saillant.

Aussi, le réseau de base est ré-entraîné pendant l'entraînement pour le rendre meilleur sur des images réelles. C'est-à-dire que le même procédé que le pré-

entraînement est utilisé sur les paires d’images réelles.

La fonction de coût de l’entraînement joint prend en compte les résultat des sous-réseaux de détection et de description pour l’image originale et l’image transformée :

$$\mathcal{L}(\mathcal{X}, \mathcal{X}', \mathcal{D}, \mathcal{D}'; \mathcal{Y}, \mathcal{Y}', S) = \mathcal{L}_p(\mathcal{X}, \mathcal{Y}) + \mathcal{L}_p(\mathcal{X}', \mathcal{Y}') + \lambda \mathcal{L}_d(\mathcal{D}, \mathcal{D}', S) \quad (2.8)$$

où  $\mathcal{X}$  représente les vecteurs de sortie du réseau de détection,  $\mathcal{X}'$  représente les vecteurs de sortie du réseau de détection pour l’image transformée,  $\mathcal{Y}$  représente les références de l’image, et  $\mathcal{Y}'$  représente les références de l’image transformée. Enfin,  $\mathcal{D}$  et  $\mathcal{D}'$  représentent respectivement les descripteurs sortis du réseau pour l’image et sa transformation.

### 2.2.2 SuperGlue

Contrairement à [39, 12, 6], le réseau SuperGlue [35] est une méthode d’apprentissage profond qui a pour objectif de réaliser la mise en correspondance entre deux ensembles de points saillants représentés par leur descripteur. SuperGlue se déploie dans la continuité de SuperPoint en utilisant les points saillants et leurs descripteurs associés produits par celui-ci. Le réseau se décompose en deux parties majeures.

La première partie a pour but de produire une matrice de score afin d’alimenter la deuxième étape qui correspond à l’algorithme de Sinkhorn. Pour la première étape, le réseau va prendre en entrée la sortie d’un réseau de type SuperPoint. Cette entrée consiste en deux images qui doivent être mises en correspondance. Pour ce faire, le réseau va lire pour chaque image une matrice contenant les points saillants détectés par le réseau avec leur descripteurs associés, également obtenus avec un réseau de type SuperPoint.

Le réseau va ensuite combiner les informations de cette entrée en passant les positions des points saillants dans un perceptron à plusieurs couches, en y ajoutant leur descripteur respectif associé. Pour la suite, la méthode conceptualise un graphe multi-

plex pour représenter l'information et l'exploiter en utilisant des couches d'apprentissage spécialisées pour les données de type graphe. Ce graphe est formé à partir de la paire d'images. Ses nœuds sont les points saillants des deux images pour lesquels deux types d'arêtes sont définies : les arêtes intra-graphe, entre les points saillants d'une même image, et les arêtes inter-images, liant les points saillants entre les images de la paire. L'étape importante du réseau est la mise à jour de chaque nœud (point saillant) du graphe dont on calcule le message, valeur prenant en compte la contribution des nœuds voisins. Un nœud va en effet être mis à jour en prenant en compte sa valeur actuelle et la valeur du message calculé.

Les couches de mises à jour sont successivement appliquées aux nœuds du graphe. Le réseau de neurones utilise un mécanisme d'attention, c'est-à-dire qu'au fur et à mesure que le réseau progresse, les nœuds vont recevoir des informations de la part des nœuds de leur image ainsi que des nœuds de l'autre image. On parle dans le premier cas d'auto-attention ("self-attention") et dans le second d'attention croisée ("cross-attention"). Le message est calculé avec une moyenne pondérée des valeurs transformées par des couches linéaires appelées "les valeurs". Les valeurs sont pondérées à l'aide d'un coefficient qui est le Softmax du produit scalaire entre la valeur représentant le point saillant et la valeur d'un autre point saillant parmi la liste, chacune de ses valeurs étant elle-même transformée par une couche de projection linéaire. Enfin, le vecteur final est obtenu en réalisant une nouvelle fois une projection linéaire avec une couche d'apprentissage.

La fonction de coût est définie comme suit :

$$L = - \sum_{(i,j) \in \mathcal{M}} \log \bar{\mathbf{P}}_{i,j} - \sum_{i \in \mathcal{I}} \log \bar{\mathbf{P}}_{i,N+1} - \sum_{j \in \mathcal{J}} \log \bar{\mathbf{P}}_{M+1,j} \quad (2.9)$$

où  $\bar{\mathbf{P}}$  représente la matrice de sortie avec les assignations des mises en correspondance,  $\mathcal{M} = (i, j) \subset \mathcal{A} \times \mathcal{B}$  représente les références des mises en correspondance, avec  $\mathcal{A}$  et  $\mathcal{B}$  étant les ensembles respectifs des points de chaque image avec  $|\mathcal{A}| = N$  et  $|\mathcal{B}| = M$ .

De plus, la méthode SuperGlue utilise aussi un système de corbeille permettant de donner une chance au réseau d'éliminer les points saillants n'ayant pas de correspondance, de la même façon que SuperPoint élimine les pixels qui n'ont pas été détectés.

Enfin, l'algorithme de Sinkhorn [40] est utilisé pour réaliser la mise en correspondance entre les deux ensembles de points saillants représentés par une matrice de score. L'algorithme de Sinkhorn va normaliser itérativement les rangées et colonnes de la matrice.

## Chapitre 3

### TARTANAIR

---

#### ***3.1 Présentation du jeu de données***

Le jeu de données TartanAir [47] est décrit par ses auteurs comme un jeu de données comprenant des images photoréalistes correspondant à des séquences prises dans différents environnements générés synthétiquement. L'objectif de ce jeu de données est d'améliorer les méthodes de SLAM visuel, en introduisant notamment une diversité de scènes comportant des situations difficiles pour la mise en correspondance. La diversité de scènes se retrouve par la présence de scènes rurales, urbaines, de nature, domestiques, de lieux publics, et d'environnements de science-fiction. Un échantillon des images utilisées pour l'entraînement de notre réseau de neurones est présent en Figure 3.1.

Les situations difficiles pour la mise en correspondance proviennent de changements d'illumination, d'objets dynamiques, et de changements de conditions météorologiques. L'avantage d'un tel jeu de données, puisque les scènes sont obtenues depuis un environnement contrôlé, est qu'il fournit toutes les informations nécessaires qui serviront de référence ou peuvent aider certaines méthodes de SLAM visuel. Le jeu de données de base correspond à des photos prises en mode stéréoscopique, c'est-à-dire que pour chaque scène, un répertoire de photos gauches et un répertoire de photos droite sont tous deux fournis. Les données disponibles sont celles des positions de caméras à chaque prise, le flux optique entre des images consécutives, et la carte de profondeur de chaque image.

Grâce à ces références, nous disposons de tous les éléments nécessaires pour créer un jeu de données correspondant à notre problème afin d'entraîner notre réseau de

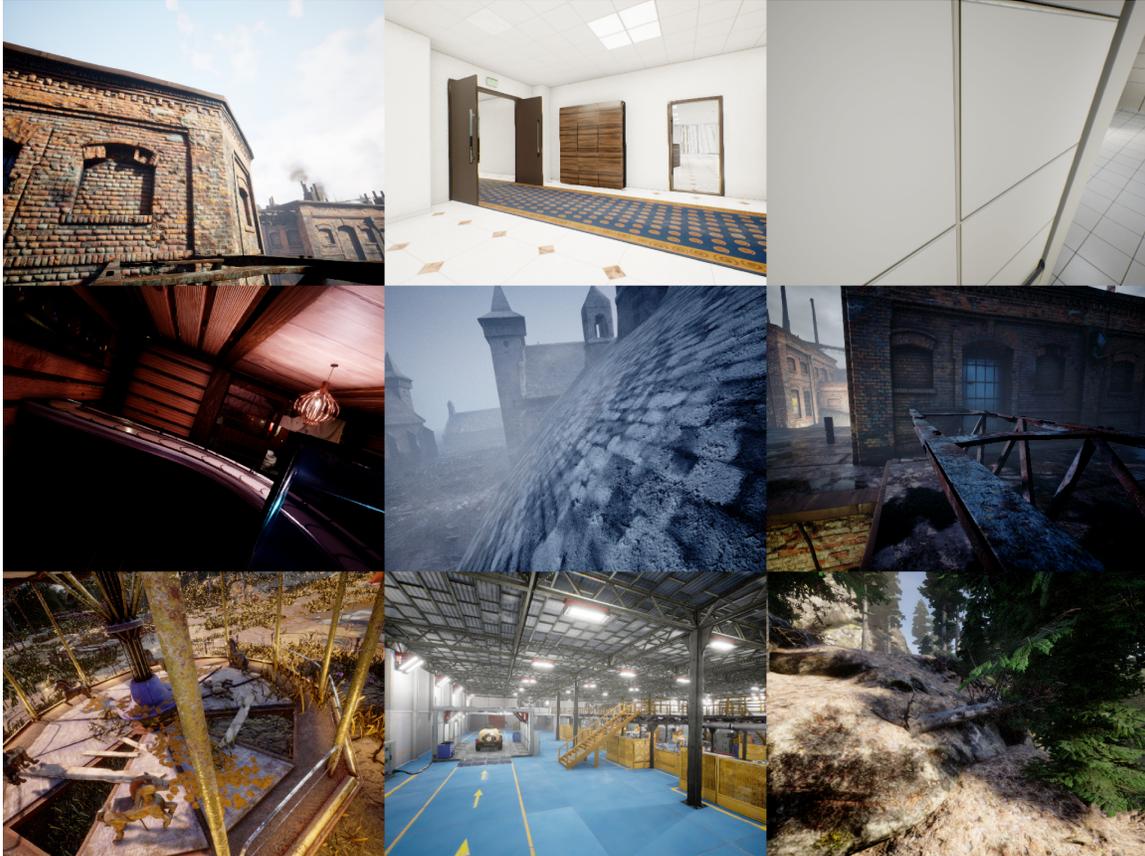


FIGURE 3.1 – Échantillon d’images du jeu de données d’entraînement : chaque image est tirée d’une séquence d’une scène de TartanAir.

neurones.

### 3.1.1 Conversion des poses

Il existe de multiples façons de représenter les rotations 2D et 3D. Pour des raisons de stabilité numérique, la représentation en quaternions est la plus utilisée, et c’est cette représentation qui est utilisée dans TartanAir. Cependant, il est analytiquement plus confortable de manipuler des matrices de position qui incluent la rotation et la translation définissant une position de caméra dans l’espace sous forme d’une seule matrice. Les quaternions correspondent à une extension des nombres complexes. Les

nombres complexes peuvent être visualisés sur un plan 2D pour représenter des points dans ce plan. L'idée derrière les quaternions est d'ajouter une dimension pour que cela fonctionne en 3D. Un quaternion s'exprime de la façon suivante :

$$\mathbf{q} = w + xi + yj + zk \quad (3.1)$$

Et nous pouvons utiliser la notation suivante pour exprimer un quaternion :

$$\mathbf{q} = [s, \mathbf{v}]^T, \quad s = w \in \mathbb{R}, \quad \mathbf{v} = [x, y, z] \in \mathbb{R}^3 \quad (3.2)$$

Grâce à la représentation en quaternion unitaire on peut réaliser la rotation d'un vecteur autour d'un axe. Pour retrouver la matrice de rotation, nous pouvons utiliser la conversion suivante :

$$\mathbf{R} = \begin{bmatrix} w^2 + x^2 - y^2 - z^2 & 2xy + 2wz & 2xz - 2wy \\ 2xy - 2wz & w^2 - x^2 + y^2 - z^2 & 2yz + 2wx \\ 2xz + 2wy & 2yz - 2wx & w^2 - x^2 - y^2 + z^2 \end{bmatrix} \quad (3.3)$$

### 3.1.2 Mise en correspondance des pixels

Grâce aux positions des caméras à chaque prise disponible du jeu de données, nous pouvons créer une fonction permettant de retrouver la position d'un point de départ dans l'image suivante ou dans l'image précédente.

Pour cela, nous devons utiliser les équations de projection. Dans les équations suivantes, nous travaillons en coordonnées homogènes comme il est souvent de rigueur en vision par ordinateur pour des raisons pratiques. Travailler en coordonnées homogènes signifie qu'un point 3D  $\mathbf{p} = [x, y, z] \in \mathbb{R}^3$  en coordonnées non-homogènes s'exprime  $\tilde{\mathbf{p}} = [x, y, z, w] = w[x/w, y/w, 1] \in \mathbb{P}^3$  en coordonnées homogènes.

Un point d'une image correspond à la projection d'un point du monde  $\mathbf{p}_w$  sur un plan image. Cette projection se déroule en plusieurs étapes. Dans un premier temps, le point dans le monde va être ramené dans le système de coordonnées de la caméra

en lui appliquant la matrice  $\mathbf{M}_{4 \times 4}$  qui comprend la rotation et la translation (Équation 3.5).

$$\mathbf{p}_c = \mathbf{M}_{3 \times 4} \mathbf{M}_{4 \times 4} \mathbf{p}_w = \mathbf{P} \mathbf{p}_w \quad (3.4)$$

Puis, ce point va être plaqué en lui appliquant la matrice  $\mathbf{M}_{3 \times 4}$  (Équation 3.6), c'est-à-dire que sa dimension de profondeur va disparaître, pour n'être représenté plus qu'en 2D dans le système de la caméra. La transformation pour obtenir le point dans le système de la caméra est donné par l'équation 3.4.

$$\mathbf{M}_{4 \times 4} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

$$\mathbf{M}_{3 \times 4} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (3.6)$$

Enfin, le point va être converti en coordonnées pixels (Équation 3.7) suivant les propriétés de la caméras ayant réalisé la prise.

$$\mathbf{p} = \begin{bmatrix} f & s & c_x \\ 0 & \alpha f & c_y \\ 0 & 0 & 1 \end{bmatrix} \mathbf{p}_c = \mathbf{K} \mathbf{p}_c \quad (3.7)$$

où  $f$  est la distance focale de la camera,  $s$  permet de modéliser un plan image non rectangulaire,  $\alpha$  est le ratio d'aspect, et  $(c_x, c_y)$  représente l'intersection de l'axe optique avec le plan image.

Pour trouver le pixel correspondant, on doit produire la transformation inverse. Originellement, l'objectif de la vision par ordinateur est de retrouver la position de la caméra ayant pris une image et de retrouver la donnée de profondeur grâce à la prise

du même point par plusieurs caméras [42]. Dans notre cas, comme on possède ces informations dans le jeu de données, nous pouvons aisément faire la transformation inverse.

Prenons le cas simple d'un point 3D dont la projection est présente dans deux images consécutives. Alors il existe bien une correspondance entre l'image du point dans la première image et l'image du point dans la seconde. Quand on connaît l'image du point dans la première image, on va commencer par ramener le point dans le système du monde.

Pour cela, on inverse les transformations pour retrouver un point dans le système de la caméra en 2D, puis dans le système de la caméra en 3D grâce à la carte de profondeur, et enfin dans le système du monde grâce à la position de la caméra. Maintenant que l'on a opéré cette rétroprojection et que l'on connaît où le point se situe dans le monde, on peut le projeter cette fois dans le plan image de la seconde caméra. On va donc amener ce point dans le système de la caméra 2, le projeter, puis convertir ses coordonnées 2D en pixel. L'équation qui décrit cette étape est la suivante :

$$\mathbf{p}_2 = \mathbf{P}_2 \mathbf{P}_1^{-1} \mathbf{p}_1 \quad (3.8)$$

où  $\mathbf{P}_1$  et  $\mathbf{P}_2$  représentent les projections respectives de la première et de la seconde caméra.

### 3.1.3 Reconstruction 3D en nuage de points

Afin d'opérer une vérification visuelle du fonctionnement des différents programmes de conversion des poses, nous recréons les scènes en 3D de TartanAir. Pour cela, nous allons parcourir chaque image d'une scène et rétroprojeter leurs points.

Pour une image donnée, nous allons parcourir les pixels de cette image pour retrouver la coordonnée du point dans le monde. La première étape est de ramener le pixel vers le plan caméra en multipliant par l'inverse de la matrice des paramètres internes.

Ensuite, on va multiplier le point obtenu par la profondeur que nous connaissons grâce à la carte de profondeur mise à disposition dans TartanAir. Enfin, nous multiplions ce nouveau point par la matrice de transformation de la caméra vers le monde, pour retrouver le point dans le système du monde. Nous utilisons l'équation suivante pour réaliser cette étape :

$$\mathbf{p}_w = \mathbf{P}^{-1}\mathbf{p} \quad (3.9)$$

où  $\mathbf{P}$  comprend la transformation qui permet de projeter un point 3D sur le plan image.

Comme on connaît les positions des centres de caméras dans le monde et la position des points mis en correspondance, nous pouvons aussi effectuer une triangulation pour calculer l'intersection de deux rayons qui donnent la position du point 3D :

$$\mathbf{p}_w = \frac{1}{2}(\mathbf{c}_1 + \mathbf{v}_1 \frac{\|(\mathbf{c}_1 - \mathbf{c}_2) \times \mathbf{v}_2\|}{\|\mathbf{v}_1 \times \mathbf{v}_2\|} + \mathbf{c}_2 + \mathbf{v}_2 \frac{\|(\mathbf{c}_2 - \mathbf{c}_1) \times \mathbf{v}_1\|}{\|\mathbf{v}_2 \times \mathbf{v}_1\|}) \quad (3.10)$$

où  $\mathbf{c}_1$  et  $\mathbf{c}_2$  sont les positions des centres de caméra et  $\mathbf{v}_1$  et  $\mathbf{v}_2$  sont les directions entre le centre de la caméra et le point projeté.

Finalement, nous exportons un fichier du nuage de points où chaque point est représenté par ses coordonnées et sa couleur RGB (voir Figure 3.2).

#### 3.1.4 Création des LUTs (LookUp Table)

Les LUTs sont des matrices permettant de stocker des résultats de calcul pour toutes les positions de l'image, dans le but d'y avoir accès rapidement quand nécessaire. Dans notre cas, nous formons des LUTs permettant de connaître pour un point donné d'une image la position du point correspondant dans l'image suivante ou précédente. Cela permet de parcourir rapidement une trajectoire en sachant où un point d'une image de départ va se retrouver dans les images successives. On peut créer un LUT entre deux images consécutives ou non. Nous stockons ces LUTs sous forme d'images dont

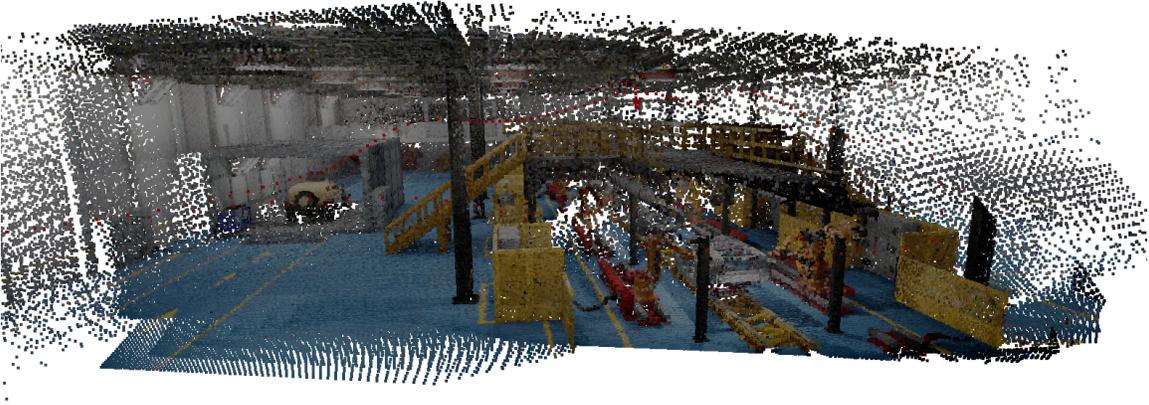


FIGURE 3.2 – Nuage de points obtenu à partir des images d’une séquence de TartanAir et de leurs cartes de profondeur associées.

les canaux représentent les nouvelles positions, ce qui permet d’avoir une idée visuelle du déplacement ayant lieu, comme le montre la Figure 3.3.

Pour créer ces LUTs, nous trouvons le pixel correspondant entre deux images tel que décrit dans la partie précédente. Les coordonnées du pixel correspondant seront stockées dans les canaux rouge et vert de l’image. Le canal bleu stockera un masque booléen indiquant la validité de la mise en correspondance. En effet, si le pixel correspondant tombe hors des limites de l’image, le pixel ne sera pas valide et donc la valeur à cette position dans le masque du canal bleu sera nulle.

### ***3.2 Génération des trajectoires***

Pour générer les trajectoires, nous utilisons les LUTs que nous avons calculé pour connaître où un pixel d’une image  $i$  se situe dans l’image  $i+1$  ou dans l’image  $i-1$ . Dans



FIGURE 3.3 – LUT avant représentant les positions des points d'une image de départ vers une image d'arrivée.

le premier cas on parle de LUT "avant" et dans le second de LUT "arrière". Grâce à cela, nous établissons la procédure suivante. Pour n'importe quelle scène, on commence par choisir une image au hasard, qui sera notre image de départ. Ensuite nous sélectionnons une position aléatoire dans cette image.

Puis nous allons utiliser les LUTs avant et arrière pour aller chercher les positions suivantes et les positions précédentes depuis la position initiale. Si on prend le chemin avant, on commence par obtenir la position du point initial dans l'image  $i + 1$ , puis ensuite nous regardons où ce nouveau point va se situer dans l'image  $i + 2$ , et ainsi de suite. Nous procédons de la même façon pour le chemin arrière.

Cela nous permet d'obtenir une séquence de positions d'un même point à travers plusieurs images consécutives de la séquence, formant une trajectoire, dont on peut voir un exemple en Figure 3.4. Notons que la taille des différentes trajectoires pouvant être générées varie car certains points apparaissent et disparaissent lors de la séquence. En effet, un point peut ne plus être visible à partir d'un certain nombre de prises comme d'autres peuvent apparaître, ou encore apparaître, disparaître puis réapparaître.

Nous ne gardons pas toutes les trajectoires générées. En effet, il existe des points dont la profondeur varie énormément, dans le cas notamment d'occultations. Nous éliminons donc les trajectoires qui se trouvent dans ce cas, car l'objectif est ici d'avoir le même point pris dans différentes conditions dépendamment de la position de la caméra ayant effectué la prise.

Grâce à cette méthode nous générons 20k trajectoires pour chaque scène dans lesquelles nous pourrions aller piocher pour constituer notre jeu de données d'entraînement. Ces trajectoires permettent d'obtenir un niveau de variance désiré, c'est-à-dire qui est celui rencontré lors d'une prise d'image séquentielle, et non issue de transformations appliquées manuellement comme c'est le cas dans les travaux précédents. Nous retrouvons dans ces trajectoires des changements d'illumination, d'échelle, et parfois même de mouvement, pour les projections d'un point du monde sur les images.

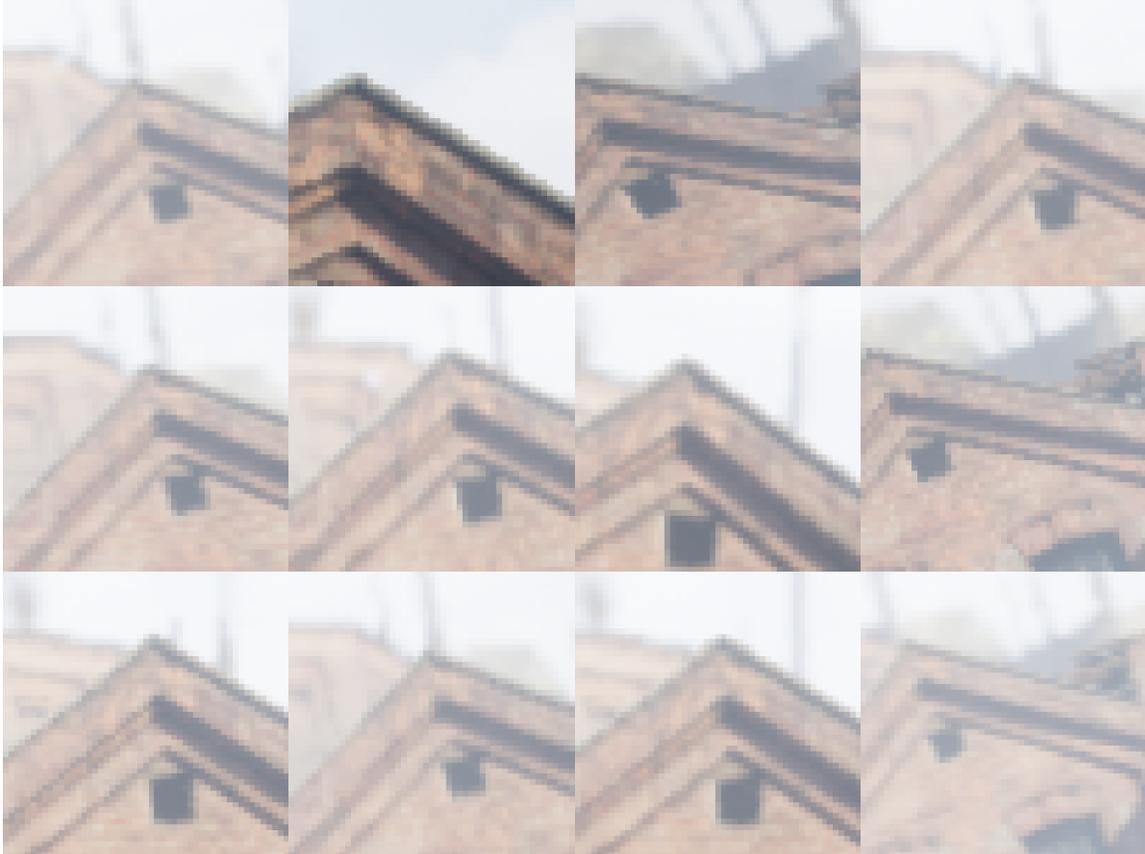


FIGURE 3.4 – Exemples de patches  $51 \times 51$  appartenant à la même trajectoire d'une scène de TartanAir. Le point suivi est au centre de chaque patch.

## Chapitre 4

### MÉTHODE

---

#### 4.1 Hypothèse

Notre idée est la suivante : les descripteurs actuellement utilisés sont calculés manuellement avec les méthodes classiques telles que ORB et souffrent encore sur des scènes difficiles au niveau de la mise en correspondance. Nous pensons que la mise en correspondance pourrait bénéficier des dernières avancées en apprentissage automatique pour la génération de descripteurs robustes, en particulier sur ces scènes difficiles. En effet, la diversité des exemples présents au sein d'un jeu de données tel que TartanAir, associée à un objectif d'apprentissage adapté, permet d'établir une situation d'apprentissage employant un réseau de neurones pour répondre à ce problème.

L'avantage d'utiliser l'apprentissage profond pour améliorer la résolution de ce problème est qu'il va rencontrer des exemples réalistes et apprendre une représentation d'un point et de son voisinage qui aura été optimisée pour la tâche de mise en correspondance. Nous allons donc forcer le réseau à produire des descripteurs dans cette optique. Pour cela, il faut que deux points similaires, c'est-à-dire étant des projections d'un même point du monde sur leur plan image, aient des descripteurs les plus similaires possibles. Dans l'autre cas, des descripteurs différents doivent être produits pour des points différents. Mais à quel point similaire et à quel point différent, c'est là que toute la force d'un réseau de neurones apparaît : c'est le réseau qui va lui même apprendre cette nuance.

Pour se faire, nous allons utiliser le paradigme d'apprentissage contrastif auto-supervisé comme dans [39, 6]. Comme expliqué dans le chapitre 2, ce paradigme permet d'économiser l'étiquetage des données et assure une cohérence dans l'apprentis-

sage. Au sujet de l'architecture du réseau, nous allons utiliser un réseau entièrement convolutif. L'avantage d'un tel réseau est de pouvoir s'adapter à différentes dimensions d'entrée pour fournir une carte de descripteurs dense.

## **4.2 Architecture du réseau**

L'architecture du réseau et de ses sous-réseaux est décrite en Figure 4.1. Pour réaliser notre apprentissage, nous utilisons la même architecture que [39] et [6] en suivant le paradigme d'apprentissage contrastif. Notre réseau prend en entrée une paire de patches où chaque patch passe dans un sous-réseau. Les deux sous-réseaux sont identiques : ils ont la même architecture et partagent leurs poids. Chaque patch est donc transformé en un vecteur par le sous-réseau, et les deux vecteurs de sortie sont comparés dans la fonction de coût  $\mathcal{L}$  (voir équation 4.3).

Le sous-réseau utilisé est un réseau entièrement convolutif qui va produire des cartes de caractéristiques de 64, 32, 64, et enfin 128 canaux, faisant en quelque sorte office d'une réduction de dimension par l'encodage de l'entrée en 32 canaux, puis d'un décodage de cette représentation. En ce qui concerne les fonctions d'activation, nous utilisons  $\tanh$  (fonction tangente hyperbolique) en sortie des couches de convolutions intermédiaires et ReLU (fonction Unité Linéaire Rectifiée) en sortie de la couche finale. Cette couche finale est une convolution de 128 filtres de la dimension du patch reçu de la couche précédente, sans rembourrage ("padding"), et qui produit un vecteur de 128 caractéristiques que nous pouvons utiliser comme descripteur du patch fourni en entrée dans un contexte de mise en correspondance.

Comme le sous-réseau n'a pas de couches entièrement connectées, il peut donc s'adapter à différentes tailles d'images d'entrée. Cette propriété d'être entièrement convolutif est très intéressante pour entraîner sur des patches et ensuite utiliser le sous-réseau sur une image au complet pour générer une carte dense de descripteurs.

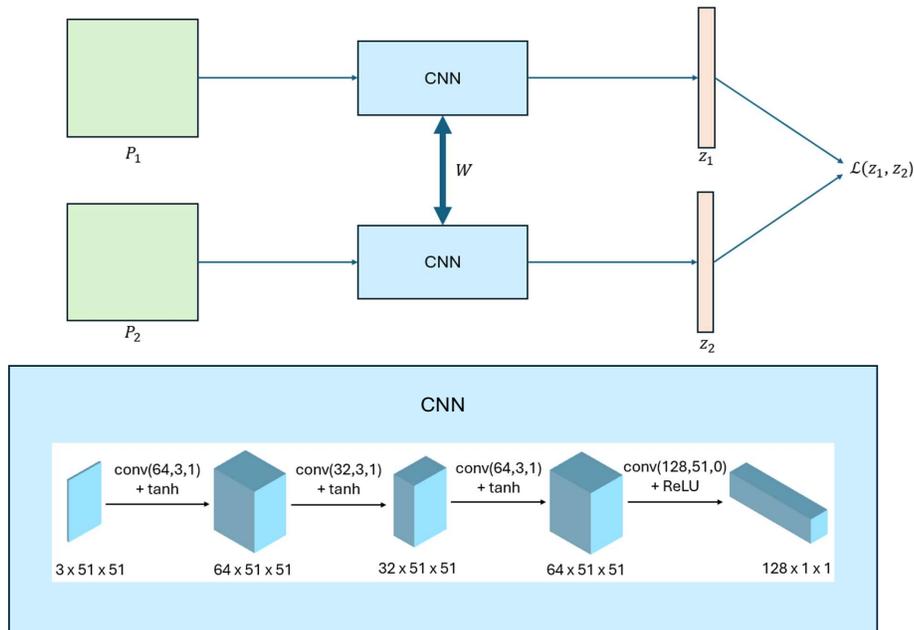


FIGURE 4.1 – Haut : Architecture du réseau de neurones entraîné par apprentissage contrastif. Le réseau prend en entrée deux patches  $P_1$  et  $P_2$  fournis aux sous-réseaux partageant les poids  $W$ , et produit deux vecteurs de 128 caractéristiques  $z_1$  et  $z_2$ , comparés dans la fonction de coût  $\mathcal{L}$ . Bas : Architecture du sous-réseau entièrement convolutif, où  $\text{conv}(c, k, p)$  correspond à une convolution produisant une carte de caractéristiques de  $c$  canaux, en utilisant des filtres de dimensions  $k \times k$ , et avec un rembourrage  $p$  de zéros. Les opérations de convolutions sont suivies de l'application d'une fonction d'activation,  $\tanh$  pour les couches intermédiaires, et  $\text{ReLU}$  pour la couche finale.

### 4.3 Fonction de coût

La similarité entre deux vecteurs peut être calculée au moyen de différentes méthodes. Traditionnellement, c'est la norme euclidienne qui a été utilisée. Depuis quelques

années et l'émergence de l'apprentissage automatique, le produit scalaire est de plus en plus utilisé pour estimer cette similarité. Le produit scalaire correspond au cosinus de l'angle entre les deux vecteurs et donne une bonne quantification de la proximité entre ceux-ci. En effet, si les vecteurs sont très proches et donc que l'angle est situé entre 0 et 90 degrés, alors le cosinus sera élevé et prendra sa valeur maximale quand les vecteurs sont colinéaires, puis descendra ensuite jusqu'à 0 quand l'angle est de 90 degrés. Dans le cas où les vecteurs sont orthogonaux, le cosinus est égal à 0 et indique clairement une dissimilarité entre les vecteurs. Quand l'angle dépasse 90 degrés, le cosinus devient négatif et ne cesse de descendre jusqu'à atteindre une valeur minimale de -1 à 180 degrés. Si l'angle continue d'augmenter, alors le cosinus recommencera à augmenter jusqu'à ce que le vecteur retrouve son point de départ et prend sa valeur maximale de 1 quand l'angle est de 360 degrés. En résumé, dans le cas où le cosinus est positif, alors les vecteurs pointent dans la même direction et sont corrélés. Dans le cas où les vecteurs sont orthogonaux, alors ils ne sont pas corrélés. Dans le troisième et dernier cas, les vecteurs sont décorrélés.

En conclusion, le produit scalaire est une bonne mesure de la similarité entre deux vecteurs normalisés sortant d'un réseau de neurones car il prend en compte l'angle entre les vecteurs dans l'espace appris.

La fonction de coût que nous utilisons prend en entrée les vecteurs produits par le réseau A et les vecteurs produits par le réseau B. L'objectif de la fonction de coût est de calculer le niveau de similarité entre les différents vecteurs, tel que les vecteurs issus d'images de la même trajectoire soient similaires et ceux issus d'images n'appartenant pas aux mêmes trajectoires soient différents. Comme expliqué précédemment, nous utilisons le produit scalaire pour mesurer le niveau de similarité entre les vecteurs. Les vecteurs produits par le réseau sont normalisés. Nous utilisons deux termes pour mesurer la contribution de chacune des fonctions de coût. Le premier terme correspond à la fonction de coût pour une paire de patches positive, c'est-à-dire deux patches appartenant à la même trajectoire :

$$\mathcal{L}_{pos} = \log \frac{1}{1 - \mathbf{z}_1 \mathbf{z}_2} \quad (4.1)$$

La seconde correspond à la contribution des patchs négatives, c'est-à-dire n'appartenant pas à la même trajectoire :

$$\mathcal{L}_{neg} = \log \frac{1}{\mathbf{z}_1 \mathbf{z}_2} \quad (4.2)$$

Enfin, les contributions des deux termes précédents sont additionnées pour donner la fonction de coût total :

$$\mathcal{L} = \mathcal{L}_{pos} + \mathcal{L}_{neg} \quad (4.3)$$

où  $\mathbf{z}_1$  et  $\mathbf{z}_2$  représentent les descripteurs respectifs fournis par le réseau pour chacun des patchs.

#### 4.4 Détails d'entraînement

Pour réaliser l'entraînement, nous utilisons l'optimiseur Adam avec une taille de paquet de 16 exemples, et nous fournissons 7000 paquets au réseau. L'optimiseur Adam [21] est un algorithme permettant de réaliser la descente stochastique de gradient qui est l'étape d'optimisation de la fonction de coût pour mettre à jour les poids du réseau. Il inclut une composante de taux d'apprentissage adaptatif, présentée dans [15], et de moment [29]. Cet algorithme est très utilisé dans la recherche en apprentissage profond et est reconnu pour sa stabilité numérique. Comme on a 8 patchs par exemple, un réseau entraîné dans ces conditions aura vu  $7000 \times 16 \times 8 = 896\text{k}$  patchs pendant l'entraînement.

Les exemples d'entraînement sont des patchs échantillonnés parmi 9 scènes comportant au total 4425 images de dimensions  $480 \times 640$ . Au sein d'une image, on peut sélectionner  $480 \times 640$  patchs différents. Les patchs qui excèdent les limites de

l'image sont rembourrés avec des zéros. On échantillonne donc chaque patch parmi  $4425 \times 480 \times 640 = 1\,359\,360\,000$  patchs possibles. Au sein d'une scène il y a un déplacement de la caméra qui capture différentes parties de la scène donnant une diversité d'images différentes. Le tout donne un jeu d'entraînement diversifié de patchs représentant des points et leur voisinage dans tous types d'environnement, offrant au modèle l'opportunité de généraliser.

## Chapitre 5

### EXPÉRIENCES

---

#### ***5.1 Configuration et hyperparamètres***

Nous réalisons différentes expériences dans lesquelles nous faisons varier deux hyperparamètres : le ratio de paires positives et négatives générées par le générateur, et la part de paires négatives difficiles à distinguer.

Nous pensons que le nombre de paires positives et négatives que le réseau ingère pendant l'entraînement pourrait avoir une influence sur son comportement. Nous entraînons donc trois réseaux pour lesquels nous faisons varier uniquement cet hyperparamètre.

Dans la même lignée que [39] qui va retenir les exemples négatifs sur lesquels le réseau s'est le plus trompé pendant l'entraînement, nous souhaitons bâtir cette forme de robustesse durant l'apprentissage. Pour cela, nous allons appliquer une transformation aléatoire sur des patchs appartenant à la même trajectoire, faisant que ces patchs bien qu'assez similaires visuellement, ne représentent pas la projection du même point mais d'un point voisin. Ces exemples sont difficiles pour le réseau à distinguer en termes de descripteurs et le forceront à affiner sa représentation de descripteurs pour des patchs dont la négativité est ambiguë.

#### ***5.2 Création des générateurs***

Pour réaliser notre entraînement, nous ne créons pas un jeu de données fixe mais utilisons plutôt un entraînement "à la volée". Ce type d'entraînement consiste à fournir au réseau des patchs d'images pendant l'entraînement obtenus grâce à un générateur. Le générateur fonctionne comme suit : il va aller chercher une scène au hasard parmi les 9

scènes de TartanAir retenues pour l'entraînement (Figure 3.1) pour générer des patchs de dimension  $51 \times 51$ . Ces patchs seront ensuite ingérés par le réseau pour produire des descripteurs de patchs, qui seront ensuite comparés dans la fonction de coût. L'avantage de ce type d'entraînement est que le réseau ne verra jamais le même patch deux fois et donc augmentera sa capacité de généralisation. Grâce aux 20k trajectoires générées par scène depuis les images de TartanAir, le générateur peut choisir une trajectoire aléatoire au sein d'une scène et piocher des indices également au hasard au sein de cette trajectoire. Les points retenus permettent d'obtenir plusieurs points appartenant à la même trajectoire, qui ne seront pas forcément consécutifs, et donc qui auront subi une certaine transformation. L'ensemble des paires de patchs formées à partir de ces patchs appartenant à une même trajectoire constitue les exemples positifs. Pour générer des exemples négatifs, nous choisissons aléatoirement une trajectoire ainsi qu'un indice de cette trajectoire. Si nous répétons l'opération plusieurs fois, on obtient un ensemble de patchs n'appartenant pas à la même trajectoire, et dont les paires formées à partir de ceux-ci constituent les exemples négatifs. Un exemple de génération de patchs positifs est présent en Figure 5.1.

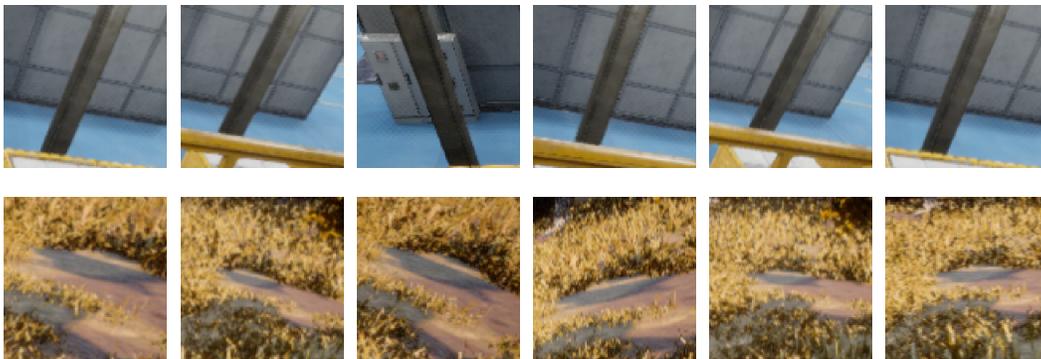


FIGURE 5.1 – Patchs générés par le générateur. Ligne du haut : patchs appartenant à une trajectoire. Ligne du bas : patchs appartenant chacun à une trajectoire différente.

### **5.3 Dimension des patches utilisés pour l'entraînement**

Pour choisir la taille du patch, nous avons considéré les travaux similaires tels que [39] qui utilisaient des patches 64 x 64. Le point de la trajectoire à suivre est au centre du patch. Nous avons voulu tester avec une taille volontairement plus petite que dans la littérature. Aussi, nous voulions montrer que nous ne sommes pas limités par la taille des patches et que le réseau peut être entraîné avec différentes tailles de patches et utilisé en inférence sur différentes tailles également. Pour le choix d'une taille de patch, il aurait été intéressant de tester une variété de tailles. Par contre, on sait qu'à mesure qu'on augmente la taille du patch, les performances vont augmenter avec une amélioration de plus en plus faible. Considérant que le réseau doit rester de taille raisonnable à entraîner, il a été décidé de prendre des patches de tailles comparables (et/ou plus petites) que ce qui est commun dans la littérature.

### **5.4 Augmentation de données**

Afin de rendre la tâche d'apprentissage plus compliquée pour le réseau et d'être capable de fournir des descripteurs robustes dans des situations ambiguës, à savoir deux patches qui se ressemblent mais ne représentant pas en réalité la projection du même point, nous décidons d'effectuer une augmentation de données. Cette augmentation consiste à appliquer une transformation aléatoire qui est une simple translation qui va déplacer la prise du patch dans une zone décrite par une fonction de bruit présentée en Figure 5.2, tel que le centre du patch ne sera pas le point suivi.

Durant l'entraînement, on peut alors déterminer à quelle fréquence le générateur doit générer ces exemples difficiles à distinguer (Figure 5.3).

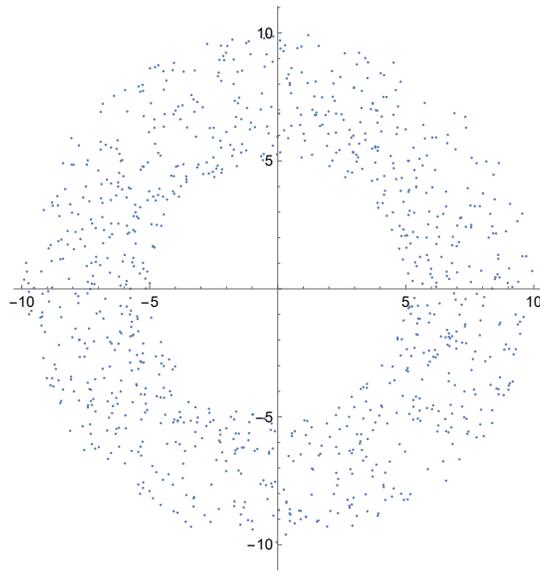


FIGURE 5.2 – Fonction du bruit ajouté pour la création d'exemples négatifs difficiles. L'unité des axes est en nombre de pixels.

### ***5.5 Variation des hyperparamètres***

Les hyperparamètres du réseau et de l'entraînement ont un réel impact sur la façon dont il se comporte en inférence. Pour l'observer, nous allons entraîner plusieurs réseaux en faisant varier ces hyperparamètres. Nous pouvons distinguer trois façons d'entraîner le réseau pour ensuite observer les effets provenant de la variation de ces hyperparamètres. Premièrement, nous pouvons tester un réseau qui n'a pas été entraîné, et dont la valeur des poids est aléatoire. On teste le réseau non-entraîné au même titre que les réseaux entraînés pour observer l'influence de l'apprentissage sur les résultats. Cette vérification est peu faite dans la littérature mais nous paraissait cruciale pour comprendre comment le problème réagit dans son intégralité à l'apprentissage et quelles étaient les dispositions du réseau non-entraîné comparées aux réels progrès observés après apprentissage. Grâce à ces comparaisons, on a pu remarquer que l'apprentissage, notamment sans augmentation de données, n'est pas toujours suffisant et qu'il arrive



FIGURE 5.3 – Patches générés par le générateur. Ligne du haut : patches appartenant à la même trajectoire. Ligne du bas : patches de la ligne du haut ayant subi une transformation.

que sur certains exemples le réseau non-entraîné soit meilleur que les réseaux entraînés. On remarque aussi que c'est dans la réduction des FP (Faux Positif) que les réseaux entraînés sont meilleurs que le réseau non-entraîné. Cela conforte l'une de nos hypothèses de départ : nous nous attendions à ce que le réseau soit bon sur des patches similaires mais mauvais sur des patches différents, indépendamment de leur relation de correspondance. L'augmentation des données combinée aux résultats du réseau non-entraîné nous donne donc une idée du fonctionnement du réseau et nous montre ce qu'a réellement apporté l'apprentissage.

Deuxièmement, nous pouvons faire varier la part d'exemples de paires positives  $N_A$  et de paires négatives  $N_B$  produites par le générateur. Enfin, nous pouvons faire varier la fréquence à laquelle nous générons des exemples négatifs difficiles  $\tau_D$ , qui sont fournis par l'augmentation de données négatives produites par le générateur adapté.

Chaque exemple fourni au réseau est constitué à partir de  $(N_A + N_B)$  patches. Ces patches sont utilisés pour former des paires de patches. Avec les hyperparamètres  $N_A$  et  $N_B$ , on aura donc  $(N_A \times N_A) + (N_A \times N_B)$  paires de patches fournies par paquet. Si  $N_A = 4$  et  $N_B = 4$ , alors le réseau verra 16 paires positives et 16 paires négatives par paquet. Si  $N_A = 2$  et  $N_B = 6$ , alors le réseau verra 4 paires positives et 12 paires négatives par paquet. Les hyperparamètres  $N_A$  et  $N_B$  permettent donc de moduler le nombre de paires positives et négatives fournies au réseau pendant l'entraînement. En ce qui concerne  $\tau_D$ , il s'agit de la proportion d'exemples négatifs difficiles fournis au réseau. Si  $\tau_D = 0.75$  et  $N_B = 4$ , alors le réseau verra  $0.75 \times 4$  exemples négatifs difficiles et  $0.25 \times 4$  exemples négatifs classiques.

La liste des différents réseaux et la valeur de leurs hyperparamètres sont données par le tableau 5.1.

TABLEAU 5.1 – Liste des méthodes expérimentées

Méthode	$N_A$	$N_B$	$\tau_D$
R0	/	/	/
R1	4	4	0
R2	6	2	0
R3	2	6	0
R4	4	4	0.25
R5	4	4	0.50
R6	4	4	0.75

## Chapitre 6

### ÉVALUATION

---

Nous réalisons quatre tests pour évaluer notre méthode et comparer les différents réseaux entraînés : la séparation de classes, la mise en correspondance de points saillants pris au hasard, la mise en correspondance de points saillants ORB entre deux images consécutives, et la mise en correspondance de points saillants ORB entre deux images non-consécutives.

Pour les tests ci-dessous, nous appelons les réseaux entraînés sans augmentation de données (sans ajout d'exemples négatifs difficiles) les réseaux "non-augmentés" (NA), et les réseaux entraînés avec augmentation de données (ajout d'exemples négatifs difficiles) les réseaux "augmentés" (A).

#### ***6.1 Séparation des classes***

Afin de vérifier rapidement le pouvoir discriminant de nos réseaux de neurones pour la production de descripteurs utilisables pour une tâche de mise en correspondance, nous procédons à un test consistant à calculer le produit scalaire entre descripteurs de milliers de paires positives ainsi que de paires négatives. Nous visualisons le résultat et calculons la surface de superposition entre la masse de produits scalaires de paires positives et négatives afin de vérifier que les réseaux soient bien capables de fournir la distinction attendue (ces résultats sont présentés en section 6.3.1).

#### ***6.2 Tests de mise en correspondance***

Nous testons la capacité du réseau à définir des descripteurs robustes pour réaliser la mise en correspondance entre deux images du jeu de données TartanAir. Comme

c'est le cas dans [27], nous commençons par obtenir 1000 points saillants d'une image de départ et allons chercher pour chacun de ces points saillants le point correspondant dans l'autre image. Notons que les images utilisées lors de la réalisation de ces tests appartiennent à des scènes différentes (Figure 6.1) que celles utilisées par le réseau pendant l'entraînement.

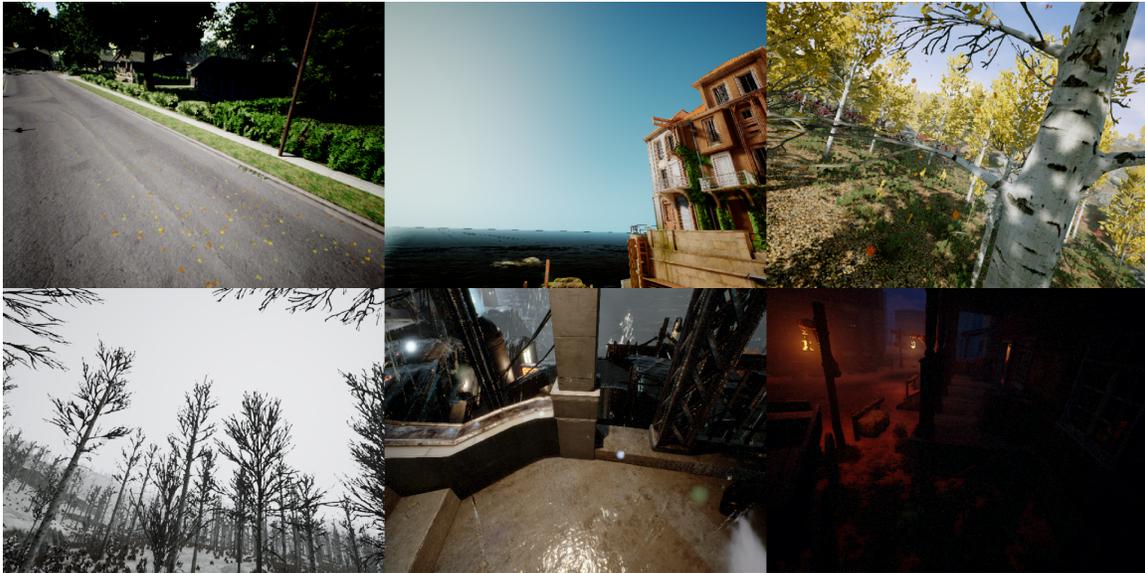


FIGURE 6.1 – Échantillon d'images du jeu de données de test : chaque image est tirée d'une séquence d'une scène de TartanAIR.

Compte tenu du nombre important de tests réalisés pour produire une évaluation exhaustive de notre méthode, nous employons une nomenclature cohérente pour la dénomination des sous-sections, et des figures et tableaux des résultats obtenus. Ainsi, les trois tests présentés dans les sections suivantes sont référés respectivement par KP-RAND, KP-ORB, et KP-ORB-W. Les figures sont nommées de la façon suivante : <nom du test> <paire d'images> <type de visualisation> <méthode(s)> dans la liste des figures et leur description complète est présente dans leur légende. Nous rappelons également que NA (R1, R2, R3), A (R4, R5, R6), et NT (R0), font respectivement références aux réseaux non-augmentés, aux réseaux augmentés, et au réseau non-entraîné,

pour lesquels les hyperparamètres sont fournis dans le Tableau 5.1. Aussi, étant donné l'exhaustivité des résultats produits, nous avons sélectionné les plus représentatifs pour présentation dans les sections suivantes. Les autres résultats sont présents en annexe. Les observations générales portent cependant sur l'ensemble des résultats, y compris ceux en annexe.

### 6.2.1 *KP-RAND : Test de mise en correspondance avec le réseau dans une fenêtre de l'image suivante*

Comme notre réseau a la capacité de fournir une carte de descripteurs dense pour n'importe quel patch ou image fournie en entrée, nous pouvons également le tester pour des positions de points saillants choisies aléatoirement dans l'image de départ, et nous ne sommes pas limités aux points fournis par les détecteurs. Cela permet de ne pas être dépendant du détecteur et de mesurer fiablement la capacité des descripteurs à être retrouvés dans l'image d'arrivée. Nous réalisons la recherche du point saillant correspondant au sein d'une fenêtre de  $40 \times 40$  pixels autour de la position du point de départ dans l'autre image. Nous aurions pu réaliser la recherche du point saillant correspondant parmi l'ensemble des points de l'image suivante. Cependant, pour des raisons de performance et pour conserver une flexibilité dans la recherche du point correspondant, nous nous sommes limités à une recherche dans une fenêtre  $40 \times 40$ . Pour chaque point saillant pris au hasard dans l'image de départ, la mise en correspondance est réalisée en calculant les produits scalaires entre le descripteur de ce point saillant et les descripteurs des points dans la fenêtre de l'image suivante. Pour déterminer le point saillant correspondant, on choisit le point dans la fenêtre de l'image suivante qui a donné le produit scalaire maximal. Grâce aux LUTs avant, nous pouvons estimer l'erreur d'estimation entre la position mise en correspondance et la référence.

### 6.2.2 *KP-ORB : Test de mise en correspondance avec le réseau et ORB entre images consécutives*

Pour évaluer la performance de notre méthode et la comparer avec les méthodes traditionnelles telles que ORB, nous réalisons un test de mise en correspondance en utilisant uniquement les points saillants détectés par ORB, aussi bien pour la détection que la recherche dans l'autre image.

Similairement à ORB-SLAM, on choisit 1000 points saillants et nous cherchons pour chacun d'entre eux le point saillant correspondant. Pour chacun des points saillants de l'image de départ, on va calculer son descripteur et le comparer avec tous les descripteurs des points saillants de l'image suivante dans la séquence. On va choisir le descripteur le plus similaire de l'image suivante, et retenir sa position. Cette position est la position prédite par la mise en correspondance. Comme pour le test précédent, on peut finalement calculer l'erreur d'estimation entre la position du point saillant et celle de la référence accessible par le LUT avant entre l'image de départ et l'image suivante.

Pour que deux points soient considérés comme correspondants, il faut que leur produit scalaire soit  $\geq 0.6$ . Après avoir observé l'influence de ce seuil lors du développement de la méthode, nous avons constaté qu'il était difficile de déterminer un seuil optimal (seuil donnant le moins de Faux Positif, ci-après FP et le plus de Vrai Positif, ci-après VP) car celui-ci varie entre les images pour un même test et entre les tests, et nous souhaitons conserver une cohérence dans la présentation des résultats. Nous avons donc choisi 0.6 comme valeur minimale d'un produit scalaire entre deux descripteurs appris pour l'acceptation d'une mise en correspondance, qui est un seuil plutôt restrictif compte tenu de nos observations pendant le développement. Pour qu'une position prédite soit considérée comme juste (VP), nous fixons le seuil d'une distance euclidienne  $\leq 0.8$  pixel. En ce qui concerne la détermination du seuil de 0.8, qui est la distance maximale au point de référence, cela dépend du degré de précision requis par l'application. Pour notre évaluation, nous avons choisi arbitrairement une valeur

assez restrictive (inférieure à 1 pixel). Par exemple, à titre de comparaison, dans [19] un seuil de 2.5 pixels est utilisé. Ces deux seuils permettent de calculer des scores de performance, tels que l'ACC (Accuracy) et l'AUC (Area Under the Curve), de notre méthode et de la mettre en comparaison avec ORB.

Pour le calcul des métriques, nous suivons la formulation employée dans [42] :

$$TPR = \frac{VP}{VP + FN} = \frac{VP}{P} \quad (6.1)$$

$$FPR = \frac{FP}{FP + VN} = \frac{FP}{N} \quad (6.2)$$

$$ACC = \frac{VP + VN}{P + N} \quad (6.3)$$

Pour ce test, nous réalisons la mise en correspondance avec les descripteurs générés par les réseaux entraînés ainsi qu'avec les descripteurs ORB, pour établir une comparaison entre notre méthode apprise et la méthode traditionnelle.

### 6.2.3 KP-ORB-W : Test de mise en correspondance avec le réseau et ORB entre images non-consécutives

Nous testons également la capacité de nos réseaux à mettre en correspondance des points saillants ORB sur des images éloignées dans le temps. Il est important de tester la robustesse de ces descripteurs pour ce cas, où les transformations subies par les points sont importantes. Pour réaliser ces tests, nous avons choisi des paires d'images au sein de plusieurs séquences qui comportent des éléments communs avec des transformations et des changements d'illumination.

### **6.3 Analyse des résultats et discussion**

#### *6.3.1 Différence de classes*

Les résultats des produits scalaires des descripteurs produits par le réseau non-entraîné sont visibles en Figure 6.2. On remarque que les paires positives sont assez correctement représentées puisque leur produit scalaire a un sommet autour de 1, bien qu'il produise quelques faux positifs. Cependant, on peut constater que la distribution des produits scalaires des paires négatives n'est pas centrée sur 0 mais ressemble plus à une distribution bimodale avec un mode autour de 0.25 et un mode qui tend vers 1. On constate que le réseau non-entraîné représente correctement certains patchs de paires positives mais qu'il aura tendance à proposer trop de faux positifs, avec une différence entre la distribution des produits scalaires des paires positives et des paires négatives pas assez marquée, avec notamment pour les paires négatives, une distribution qui n'est pas centrée sur 0 contrairement aux réseaux entraînés (Figures 6.3 et 6.4).

Pour les réseaux non-augmentés, on remarque en observant les résultats deux distributions distinctes pour chacun des trois réseaux. Les résultats des produits scalaires des descripteurs appris pour les paires négatives et les résultats des produits scalaires des descripteurs appris pour les paires positives sont visibles en Figure 6.3.

Ces distributions sont cohérentes avec ce que nous attendions, à savoir que les descripteurs des paires négatives aient un produit scalaire tendant vers 0 car ils sont bien différents, étant pris dans des trajectoires différentes, et que les paires positives aient un produit scalaire qui tend vers 1 car ils représentent le même point présent dans une trajectoire.

Ces distributions permettent d'avoir un premier aperçu de la justesse du modèle et de relever la part de produits scalaires qui représentent des faux positifs. Si l'on prend un seuil d'acceptabilité à 0.6, on remarque que c'est le réseau R3 qui comporte le moins de faux positifs. En revanche R3, comporte le plus de faux négatifs. Les réseaux R1 et R2 ont des distributions très similaires. En ce qui concerne la superposition, aucun

réseau ne se distingue particulièrement mais elle permet de donner un bon indicateur de la séparation entre les paires positives et négatives.

Si l'on observe la Figure 6.4, on remarque un comportement différent par rapport aux réseaux non-augmentés. En effet, les erreurs sont ici des faux négatifs, et il n'y a pratiquement plus de faux positifs. Ces réseaux produisent des descripteurs dont les produits scalaires sont plus restrictifs que les réseaux non-augmentés. Cela s'explique par le fait que ces réseaux ont vu des patches en apparence similaires mais appartenant en réalité à des trajectoires différentes, donc ont tendance à classer plus facilement en négatif, même quand les paires se ressemblent. L'avantage de ceci est la quasi-absence de faux positifs avec des paires positives qui sont dans l'ensemble bien décrites, particulièrement pour le réseau R6, pour lequel la superposition est presque 1.6 fois moins importante que pour les réseaux non-augmentés et R5. Notons aussi l'évolution favorable de la distribution des paires négatives à mesure que l'on augmente la part d'exemples négatifs difficiles pendant l'entraînement : la part de faux négatifs se réduit progressivement.

Pour les tests suivants, nous allons évaluer les réseaux entraînés sur plusieurs paires d'images. Pour les deux premiers tests, nous choisissons les quatre mêmes paires d'images consécutives au sein d'une scène pour lesquelles nous allons réaliser la mise en correspondance.

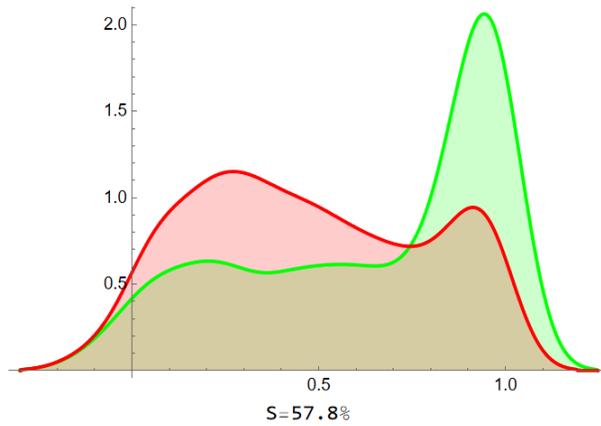


FIGURE 6.2 – Distributions des produits scalaires entre les descripteurs de patchs 51 x 51 fournis par le réseau non-entraîné, où S correspond au pourcentage de superposition entre les 2 distributions. Rouge : produit scalaire entre 2 descripteurs d’une paire négative. Vert : produit scalaire entre 2 descripteurs d’une paire positive.

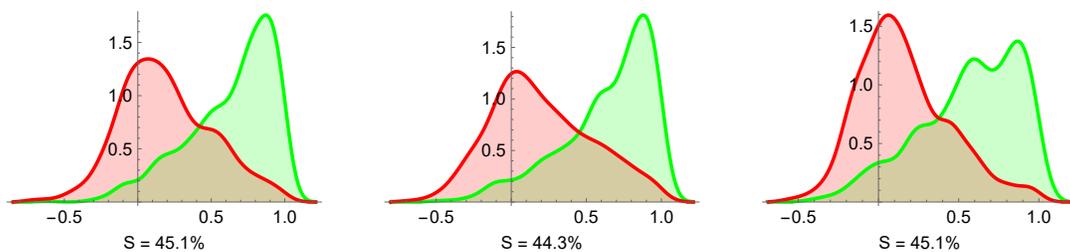


FIGURE 6.3 – Distributions des produits scalaires entre les descripteurs de patchs 51 x 51 fournis par les réseaux non-augmentés, où S correspond au pourcentage de superposition entre les 2 distributions. Rouge : produit scalaire entre 2 descripteurs d’une paire négative. Vert : produit scalaire entre 2 descripteurs d’une paire positive. Gauche : R1. Milieu : R2. Droite : R3.

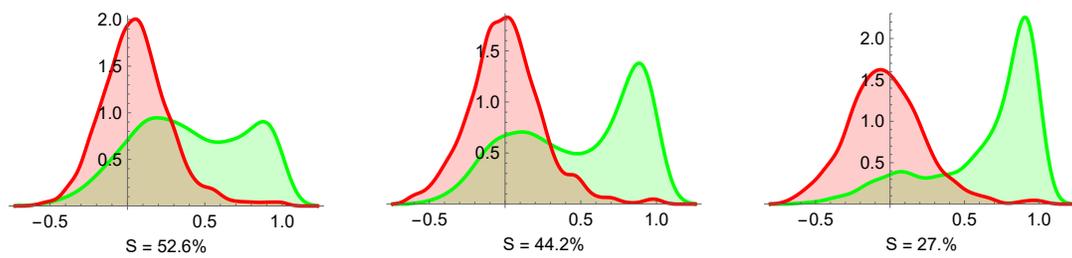


FIGURE 6.4 – Distributions des produits scalaires entre les descripteurs de patches 51 x 51 fournis par les réseaux augmentés, où S correspond au pourcentage de superposition entre les 2 distributions. Rouge : produit scalaire entre 2 descripteurs d’une paire négative. Vert : produit scalaire entre 2 descripteurs d’une paire positive. Gauche : R4. Milieu : R5. Droite : R6.

### 6.3.2 KP-RAND

Dans la présente section, nous présentons les résultats du test KP-RAND pour les paires d'images 10-155, et 13-34. Les résultats pour les autres paires d'images sont disponibles en annexe.

#### KP-RAND 10-155

Pour cette première paire d'images, nous remarquons en observant les Figures 6.5 à 6.13 ainsi que le tableau des résultats 6.1 que l'erreur moyenne est plus basse pour les réseaux R4 et R5, montrant que les réseaux augmentés ont ici une meilleure précision que les réseaux non-augmentés. Les réseaux sont tous meilleurs que le réseau non-entraîné. On remarque au sein de cette paire une zone particulièrement difficile à mettre en correspondance, pour laquelle de nombreux points saillants n'ont pas été correctement mis en correspondance donnant des FP. Il s'agit de la zone d'herbe dans le coin inférieur gauche des images, visible sur la Figure 6.11.

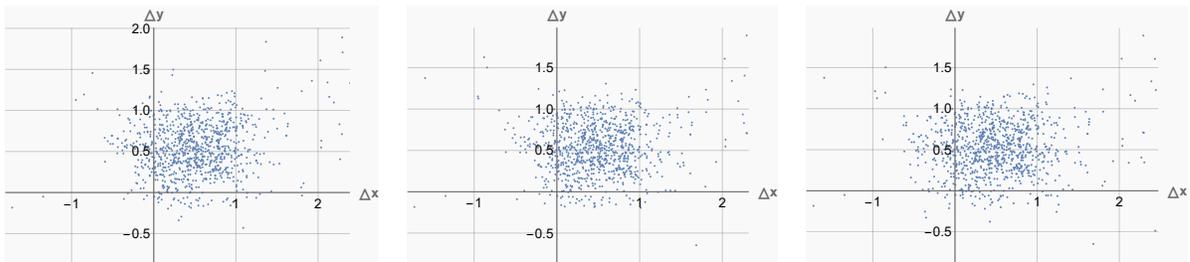


FIGURE 6.5 – Visualisation des différences entre les points mis en correspondance avec les descripteurs fournis par les réseaux non-augmentés, et leur référence. Gauche : R1. Milieu : R2. Droite : R3.

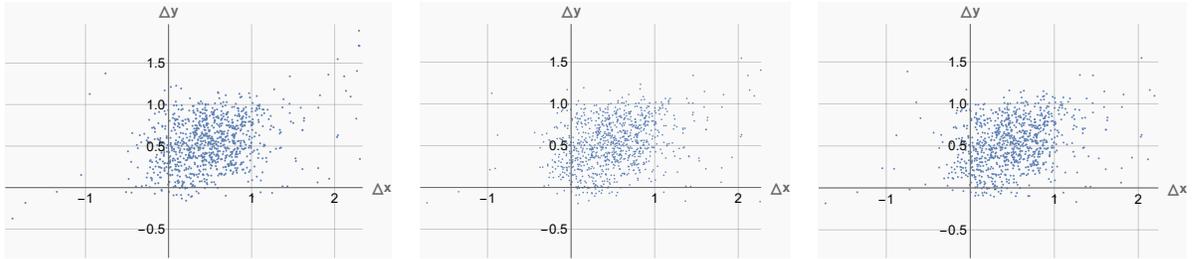


FIGURE 6.6 – Visualisation des différences entre les points mis en correspondance avec les descripteurs fournis par les réseaux augmentés, et leur référence. Gauche : R4. Milieu : R5. Droite : R6.

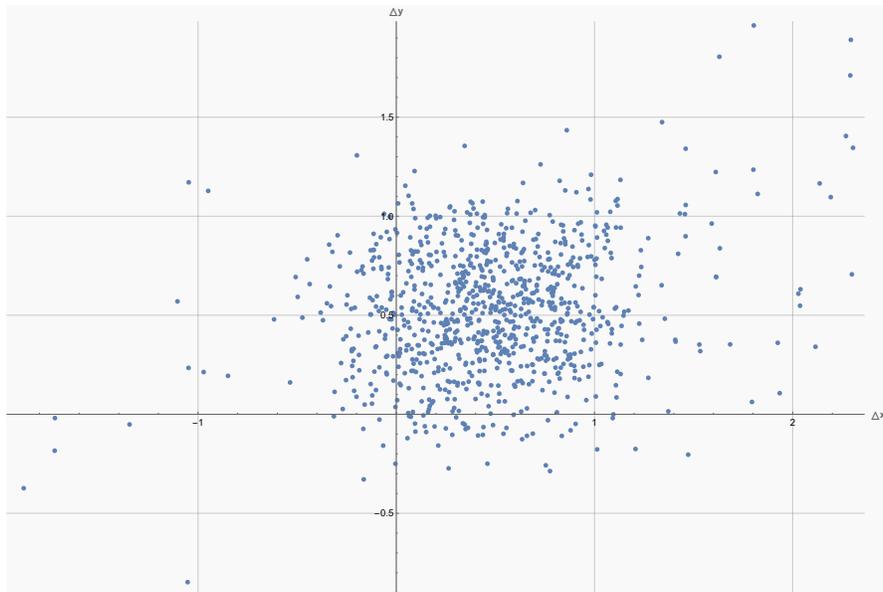


FIGURE 6.7 – Visualisation des différences entre les points mis en correspondance avec les descripteurs fournis par le réseau non-entraîné R0, et leur référence.

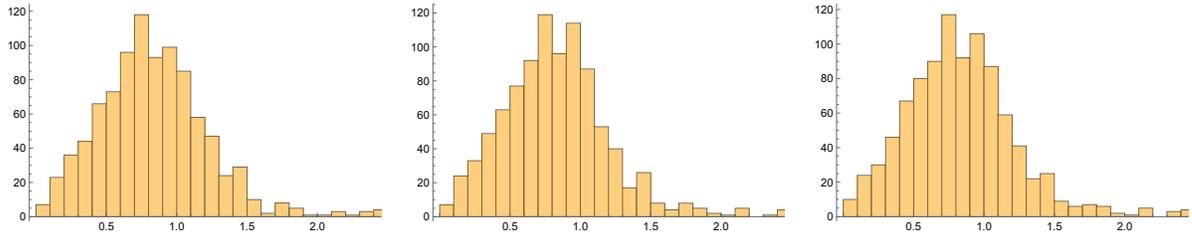


FIGURE 6.8 – Histogramme de la distance euclidienne (en nombre de pixels) entre les points mis en correspondance avec les descripteurs fournis par les réseaux non-augmentés, et leur référence. Gauche : R1. Milieu : R2. Droite : R3.

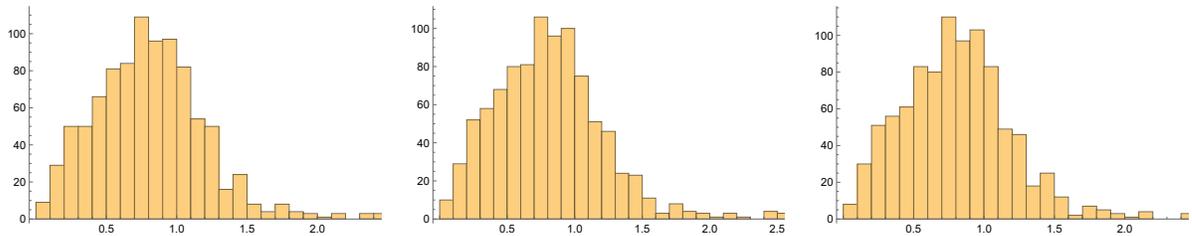


FIGURE 6.9 – Histogramme de la distance euclidienne (en nombre de pixels) entre les points mis en correspondance avec les descripteurs fournis par les réseaux non-augmentés, et leur référence. Gauche : R4. Milieu : R5. Droite : R6.

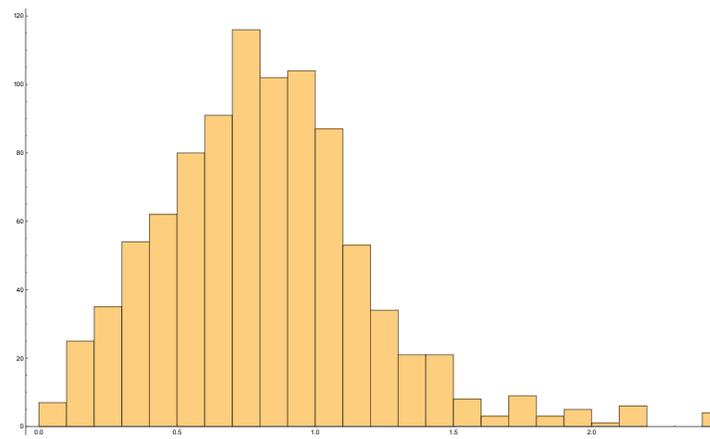


FIGURE 6.10 – Histogramme de la distance euclidienne (en nombre de pixels) entre les points mis en correspondance avec les descripteurs fournis par le réseau non-entraîné R0, et leur référence.

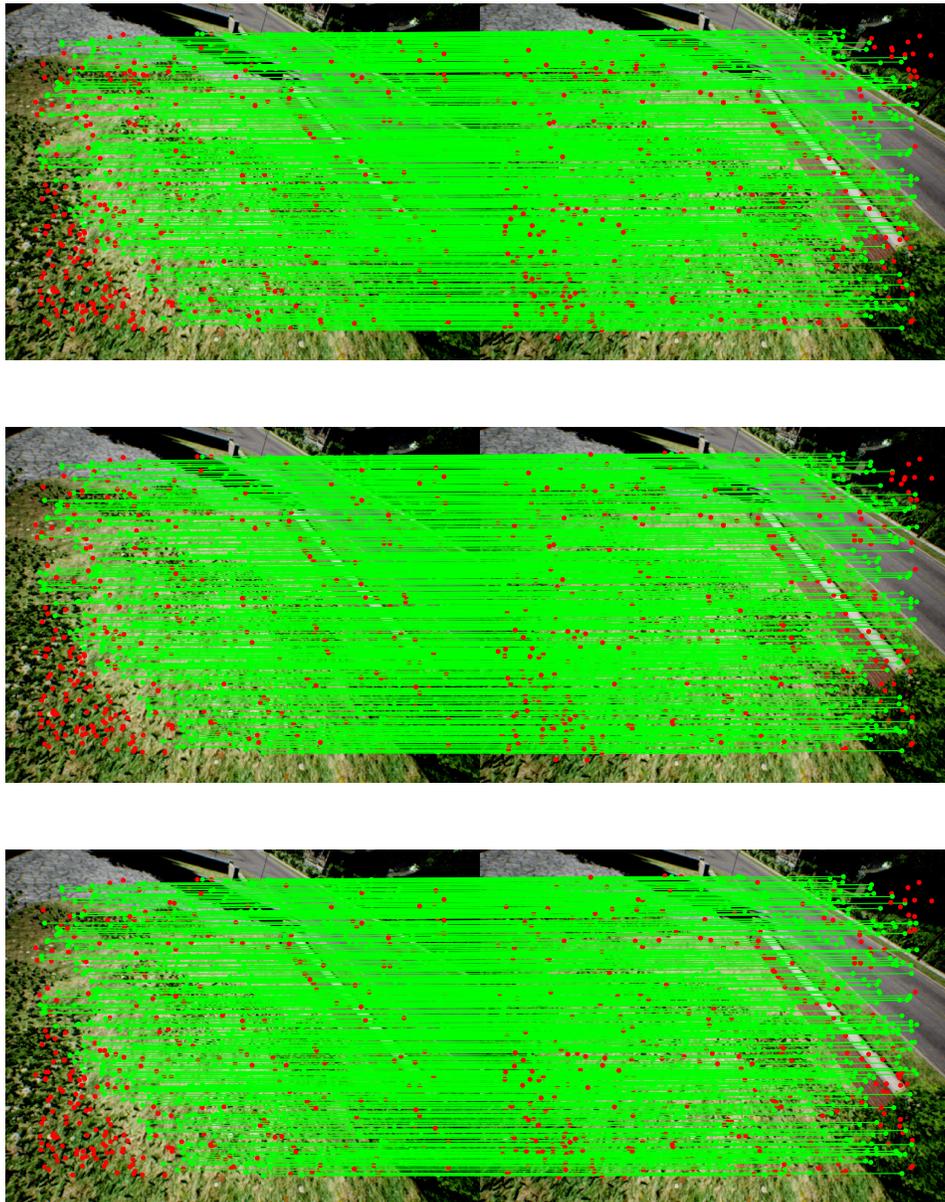


FIGURE 6.11 – Résultat qualitatif de la mise en correspondance entre les points saillants choisis aléatoirement de deux images consécutives réalisée avec les descripteurs fournis par les réseaux non-augmentés. Les lignes vertes représentent des mises en correspondance correctes. Les points rouges représentent des points qui n'ont pas été correctement mis en correspondance. Haut : R1. Milieu : R2. Bas : R3.

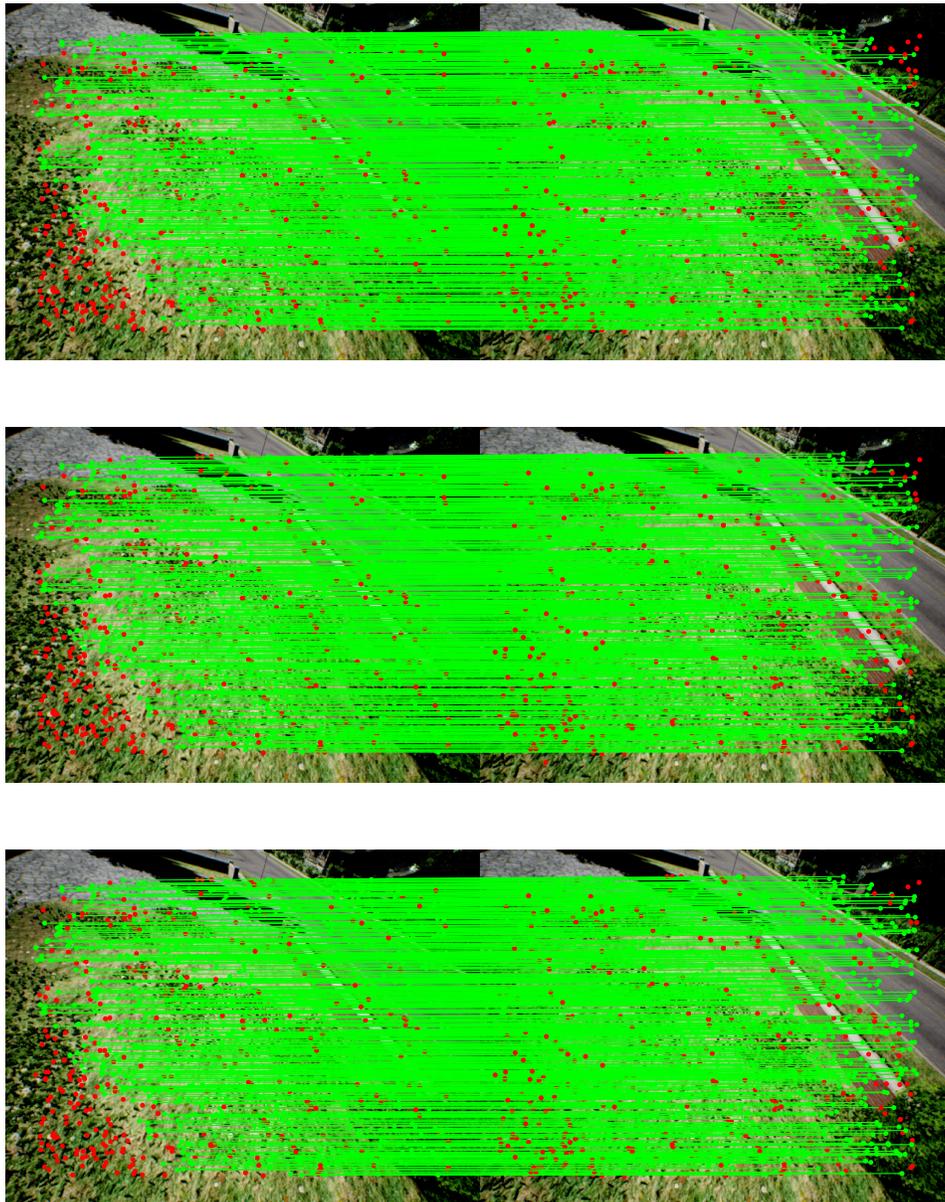


FIGURE 6.12 – Résultat qualitatif de la mise en correspondance entre les points saillants choisis aléatoirement de deux images consécutives réalisée avec les descripteurs fournis par les réseaux augmentés. Les lignes vertes représentent des mises en correspondance correctes. Les points rouges représentent des points qui n'ont pas été correctement mis en correspondance. Haut : R4. Milieu : R5. Bas : R6.

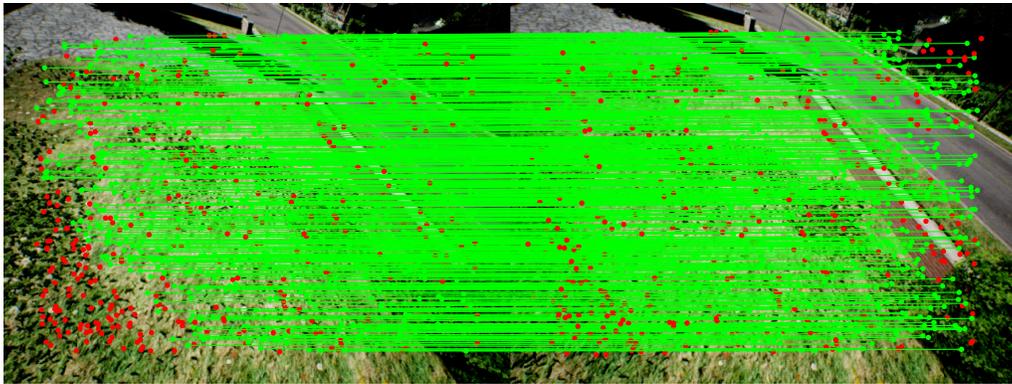


FIGURE 6.13 – Résultat qualitatif de la mise en correspondance entre les points saillants choisis aléatoirement de deux images consécutives réalisée avec les descripteurs fournis par le réseau non-entraîné R0. Les lignes vertes représentent des mises en correspondance correctes. Les points rouges représentent des points qui n'ont pas été correctement mis en correspondance.

TABLEAU 6.1 – Résultats quantitatifs regroupant les métriques des différentes méthodes utilisées pour réaliser la mise en correspondance entre les points saillants choisis aléatoirement de deux images consécutives.

Méthode	Erreur moyenne	Erreur médiane	Écart – type erreur
R0	2.37	0.83	7.15
R1	1.51	0.84	4.08
R2	1.48	0.84	4.03
R3	1.53	0.84	4.08
R4	1.24	0.82	2.88
R5	1.17	0.82	2.63
R6	1.17	0.82	2.57

Cette paire d'images est beaucoup plus difficile à mettre en correspondance comme en témoignent les histogrammes qui montrent une étendue plus importante des normes (erreur moyenne jusqu'à 30 fois plus importante que pour les paires précédentes). Cela tient au fait que les feuilles des arbres sont très similaires entre elles et donnent beaucoup de difficulté aux descripteurs pour se mettre correctement en correspondance. De façon générale, en regardant les Figures 6.14 à 6.22, ainsi que le tableau des résultats 6.2, on remarque que ce sont les réseaux non-augmentés qui sont les meilleurs, en ajoutant que plus les réseaux ont été augmentés, moins ils sont performants pour cette paire.

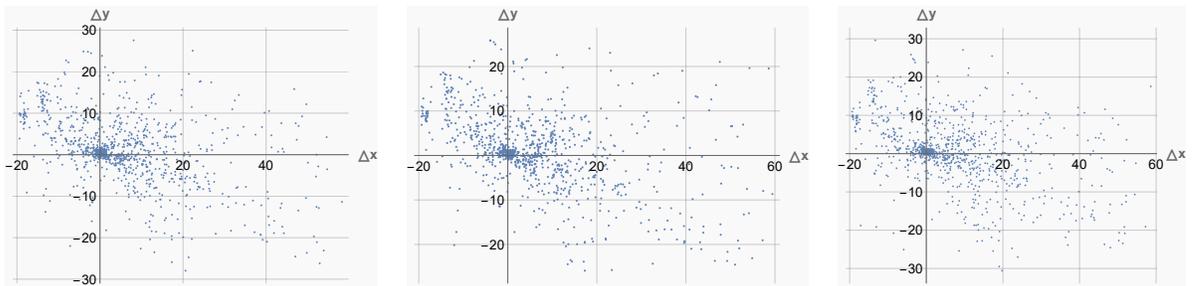


FIGURE 6.14 – Visualisation des différences entre les points mis en correspondance avec les descripteurs fournis par les réseaux non-augmentés, et leur référence. Gauche : R1. Milieu : R2. Droite : R3.

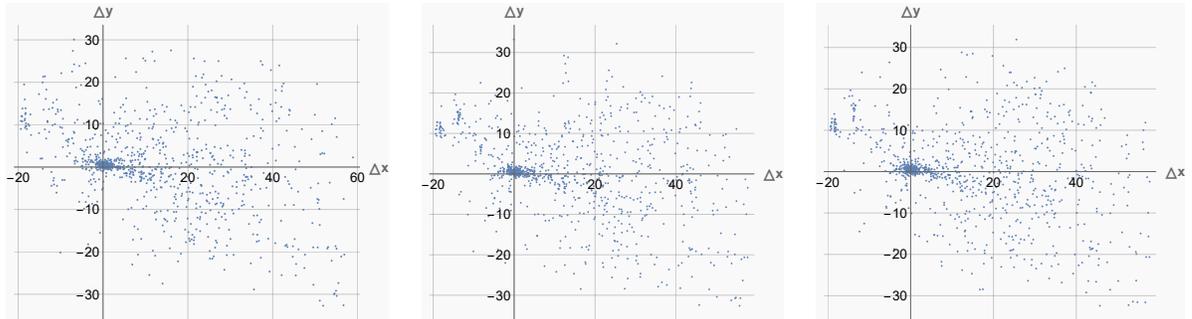


FIGURE 6.15 – Visualisation des différences entre les points mis en correspondance avec les descripteurs fournis par les réseaux augmentés, et leur référence. Gauche : R4. Milieu : R5. Droite : R6.

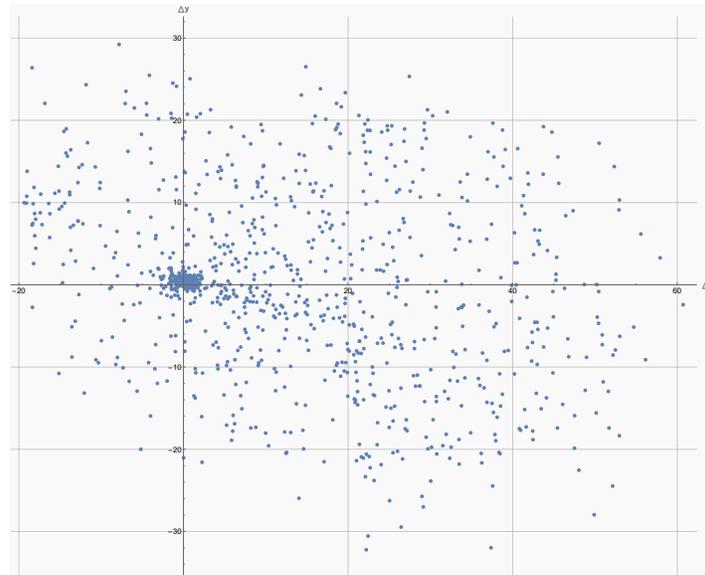


FIGURE 6.16 – Visualisation des différences entre les points mis en correspondance avec les descripteurs fournis par le réseau non-entraîné R0, et leur référence.

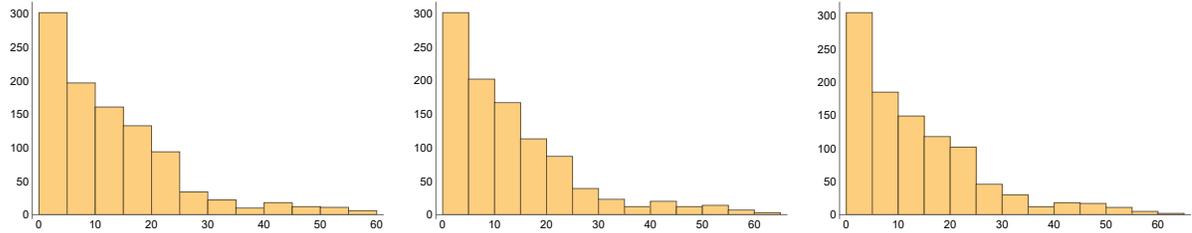


FIGURE 6.17 – Histogramme de la distance euclidienne (en nombre de pixels) entre les points mis en correspondance avec les descripteurs fournis par les réseaux non-augmentés, et leur référence. Gauche : R1. Milieu : R2. Droite : R3

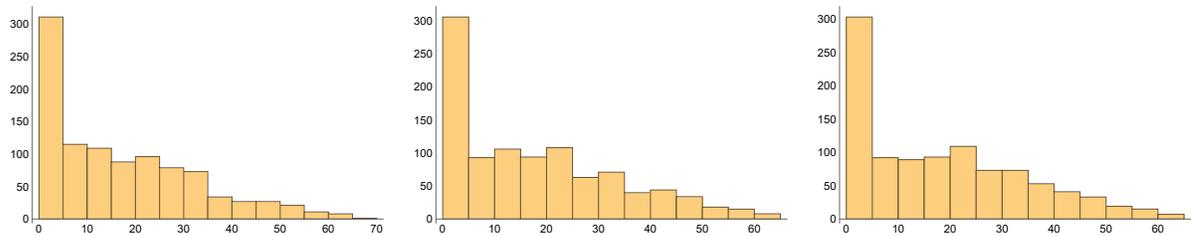


FIGURE 6.18 – Histogramme de la distance euclidienne (en nombre de pixels) entre les points mis en correspondance avec les descripteurs fournis par les réseaux non-augmentés, et leur référence. Gauche : R4. Milieu : R5. Droite : R6.

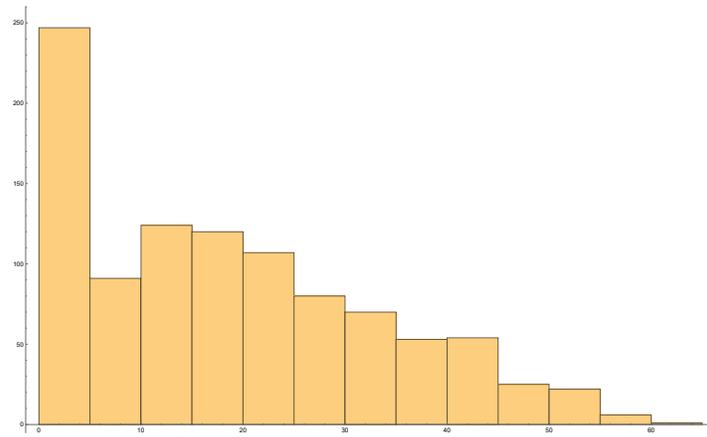


FIGURE 6.19 – Histogramme de la distance euclidienne (en nombre de pixels) entre les points mis en correspondance avec les descripteurs fournis par le réseau non-entraîné R0, et leur référence.

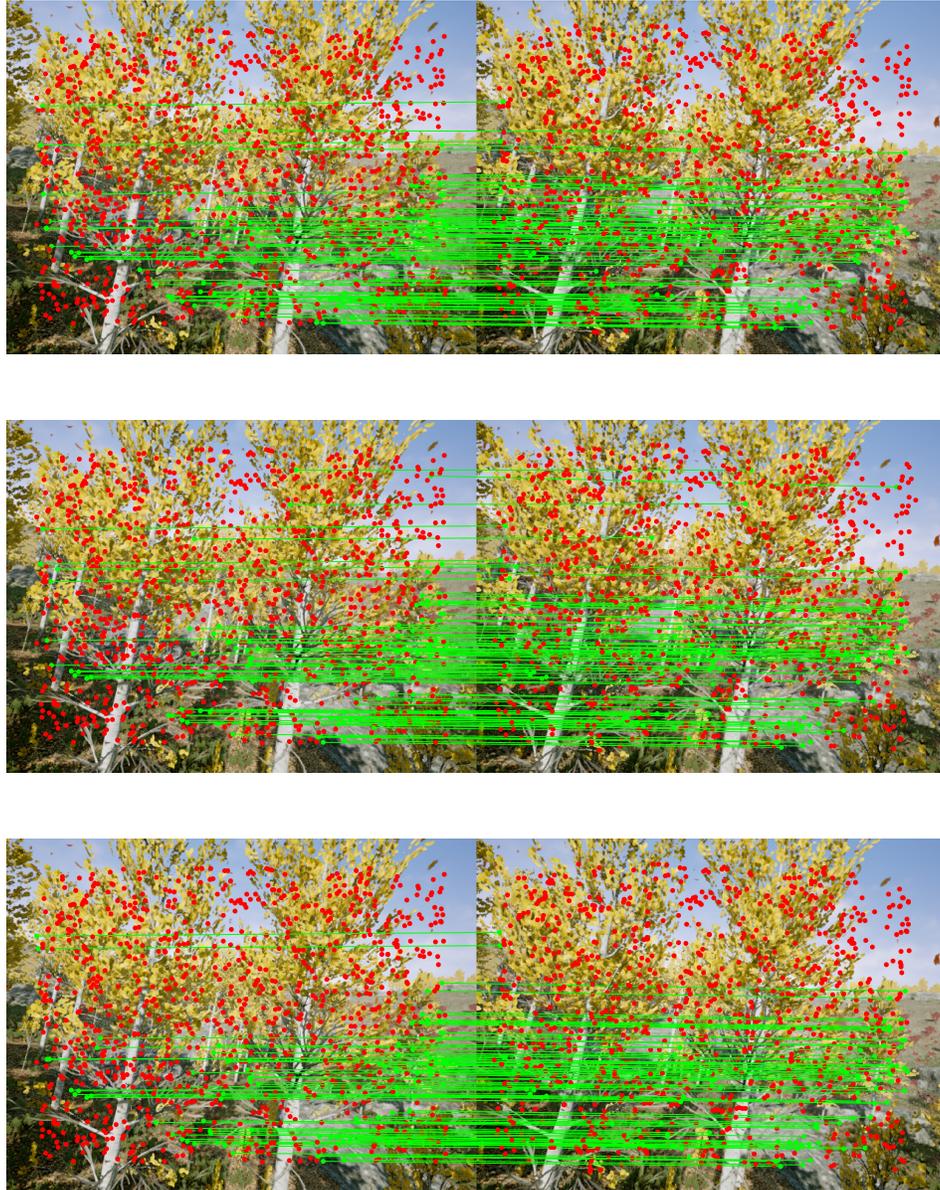


FIGURE 6.20 – Résultat qualitatif de la mise en correspondance entre les points saillants choisis aléatoirement de deux images consécutives réalisée avec les descripteurs fournis par les réseaux non-augmentés. Les lignes vertes représentent des mises en correspondance correctes. Les points rouges représentent des points qui n'ont pas été correctement mis en correspondance. Haut : R1. Milieu : R2. Bas : R3.

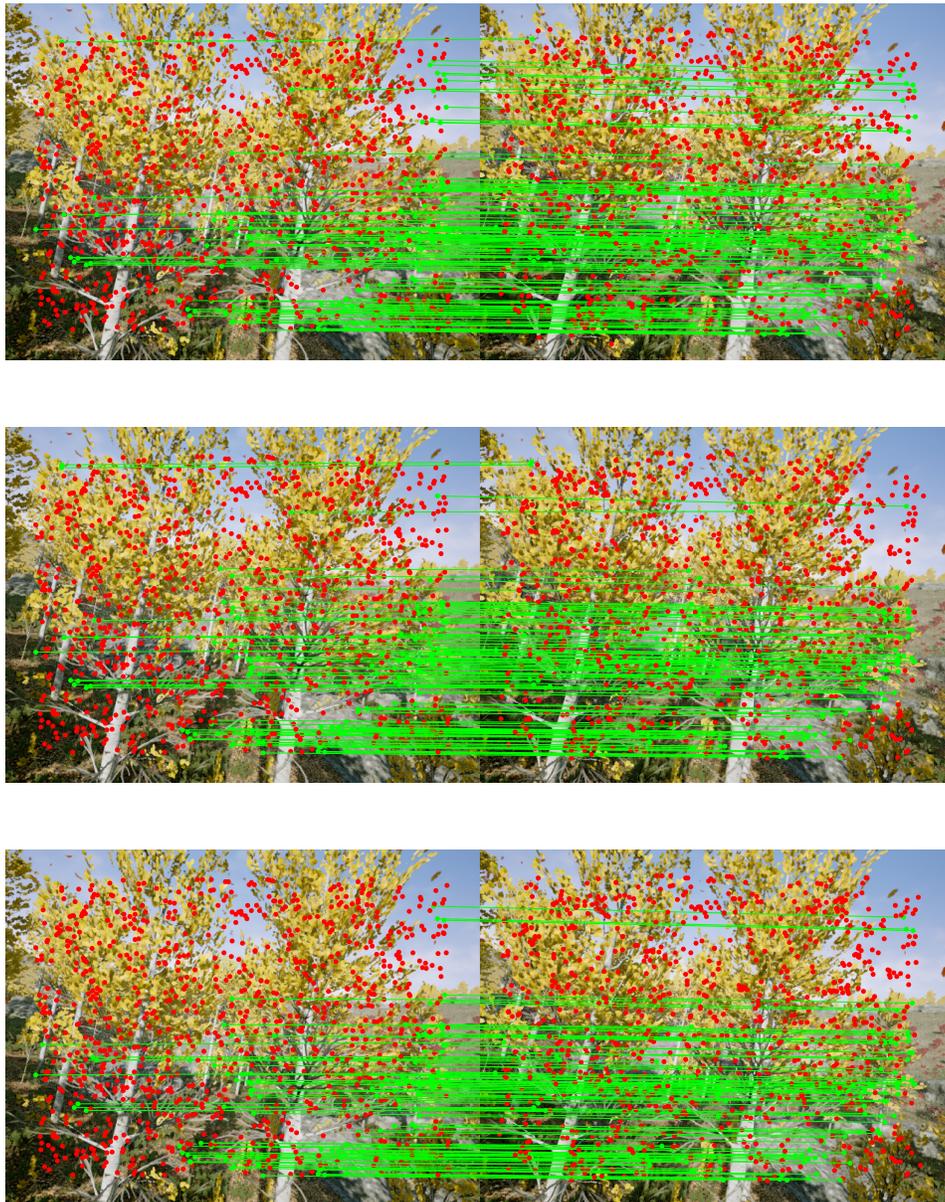


FIGURE 6.21 – Résultat qualitatif de la mise en correspondance entre les points saillants choisis aléatoirement de deux images consécutives réalisée avec les descripteurs fournis par les réseaux augmentés. Les lignes vertes représentent des mises en correspondance correctes. Les points rouges représentent des points qui n’ont pas été correctement mis en correspondance. Haut : R4. Milieu : R5. Bas : R6.

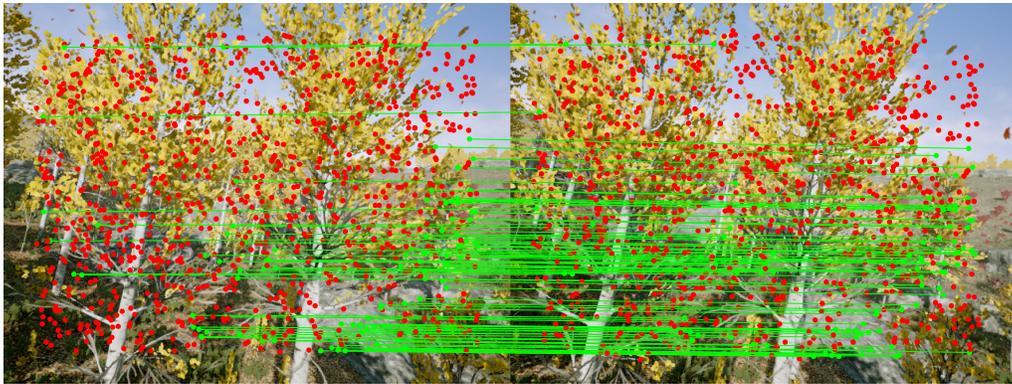


FIGURE 6.22 – Résultat qualitatif de la mise en correspondance entre les points saillants choisis aléatoirement de deux images consécutives réalisée avec les descripteurs fournis par le réseau non-entraîné R0. Les lignes vertes représentent des mises en correspondance correctes. Les points rouges représentent des points qui n'ont pas été correctement mis en correspondance.

TABLEAU 6.2 – Résultats quantitatifs regroupant les métriques des différentes méthodes utilisées pour réaliser la mise en correspondance entre les points saillants choisis aléatoirement de deux images consécutives.

Méthode	Erreur moyenne	Erreur médiane	Écart – type erreur
R0	18.6	16.92	14.71
R1	12.72	10.06	11.64
R2	12.97	9.69	12.35
R3	13.34	10.33	12.27
R4	16.95	13.5	15.33
R5	17.85	14.89	15.81
R6	18.23	15.8	15.8

### *Observations générales*

Au niveau de ce test, on peut faire les constatations suivantes : les images ne réagissent pas toutes de la même façon aux réseaux. En effet sur les 3 paires d'images (10-155, 10-260, 12-281) ce sont les réseaux augmentés qui ont la meilleure précision. En revanche, sur la dernière paire d'images (13-34 : Figures 6.20, 6.21, 6.22) les réseaux augmentés sont moins performants. De plus, on peut aussi remarquer qu'ils ne donnent pas plus de faux positifs mais qu'ils ne sont pas capables de produire autant de vrais positifs que les réseaux non-augmentés. Nous remarquons que les réseaux sont toujours meilleurs que le réseau non-entraîné, montrant que l'apprentissage fonctionne.

### 6.3.3 KP-ORB

Dans la présente section, nous présentons les résultats du test KP-ORB pour les paires d'images 10-155, et 13-34. Les résultats pour les autres paires d'images sont disponibles en annexe.

#### *KP-ORB 10-155*

Pour la première paire d'images, on remarque en observant les Figures 6.23 à 6.32 ainsi que le tableau des résultats 6.3, que ORB a le moins de VP mais le plus de VN (Vrai Négatif). Les réseaux augmentés ont moins de FP que les réseaux non-augmentés, avec des points correctement identifiés comme VN. En ce qui concerne l'ACC et l'AUC, les réseaux augmentés sont meilleurs que les réseaux non-augmentés. On voit clairement la supériorité des réseaux non-augmentés sur ORB quand on regarde les courbes ROC (Receiver Operating Characteristic). Les courbes ROC sont générées comme suit : pour une paire d'image  $(I_i, I_j)$ , contenant chacune des points saillants et leurs descripteurs associés, la courbe ROC est calculée à partir de l'ensemble des produits scalaires de paires de descripteurs respectivement constituées d'un descripteur  $d_i$  de l'image  $I_i$ , et d'un descripteur  $d_k$  de l'image  $I_j$ , « gagnant » de la mise en correspondance, c'est-à-dire celui pour lequel le produit scalaire avec  $d_i$  est le plus élevé. Les valeurs VP (Vrai Positif), FP (Faux Positif), FN (Faux Négatif), VN (Vrai Négatif) sont générées à partir de seuils appliqués sur ces produits scalaires. Nous prenons un seuil allant de 1 à -1 avec un pas de 0.01, où 1 est le seuil le moins permissif et -1 le seuil le plus permissif. Pour chacun de ces pas, les valeurs VP, FP, FN, VN permettent de calculer le FPR (False Positive Rate) et TPR (True Positive Rate) représentant les coordonnées d'un point de la courbe ROC pour ce pas. En revanche, ce sont les réseaux non-augmentés qui produisent l'erreur moyenne la plus faible, et pour laquelle ORB est la méthode la moins précise. On peut également noter que les réseaux entraînés ont en général la même capacité de détection de VP pour cette paire avec plus de 300 points saillants correctement mis en

correspondance contre 170 pour ORB.

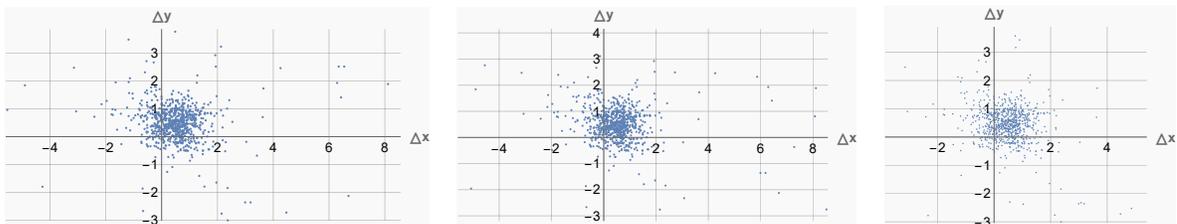


FIGURE 6.23 – Visualisation des différences entre les points saillants mis en correspondance avec les descripteurs fournis par les réseaux non-augmentés, et leur référence.  
Gauche : R1. Milieu : R2. Droite : R3.

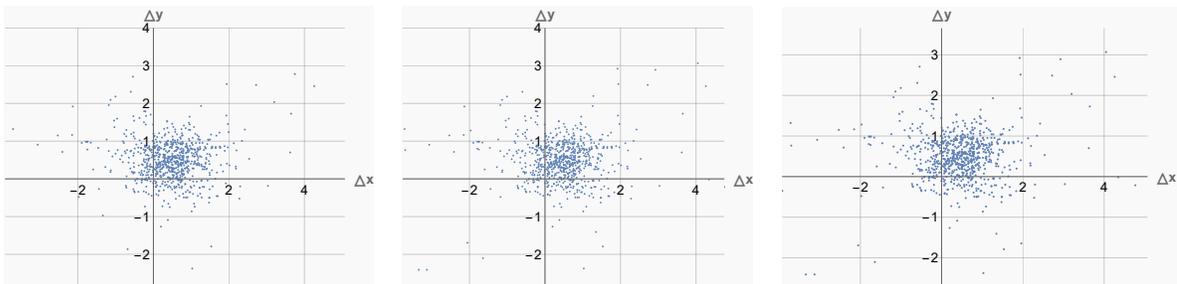


FIGURE 6.24 – Visualisation des différences entre les points saillants mis en correspondance avec les descripteurs fournis par les réseaux augmentés, et leur référence.  
Gauche : R4. Milieu : R5. Droite : R6.

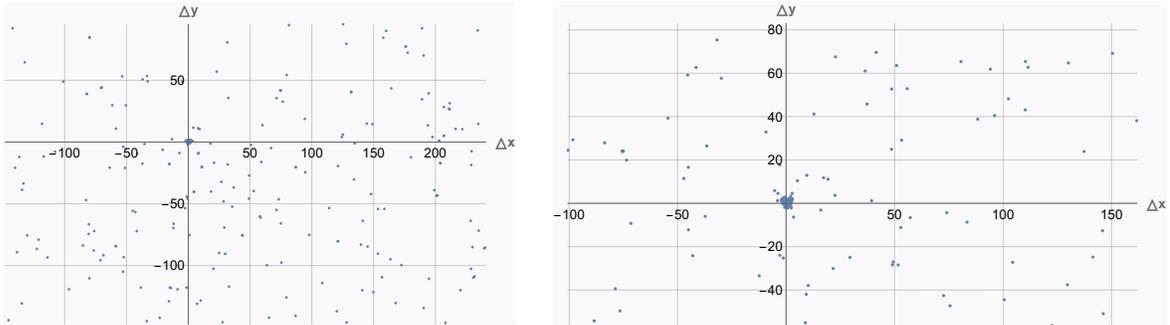


FIGURE 6.25 – Visualisation des différences entre les points saillants mis en correspondance avec les descripteurs fournis par le réseau non-entraîné et les descripteurs ORB, et leur référence. Gauche : NT. Droite : ORB.

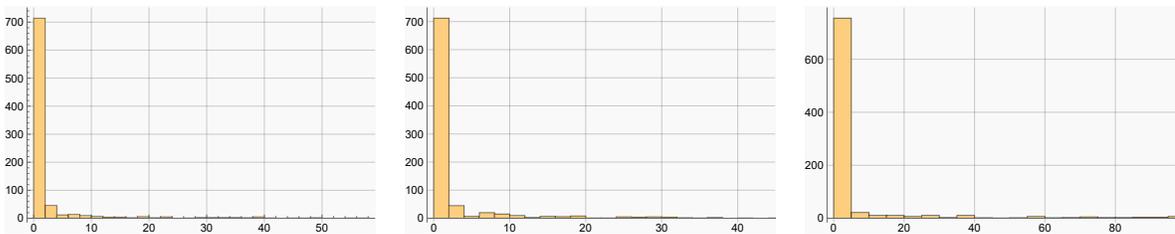


FIGURE 6.26 – Histogramme de la distance euclidienne (en nombre de pixels) entre les points saillants mis en correspondance avec les descripteurs fournis par les réseaux non-augmentés, et leur référence. Gauche : R1. Milieu : R2. Droite : R3.

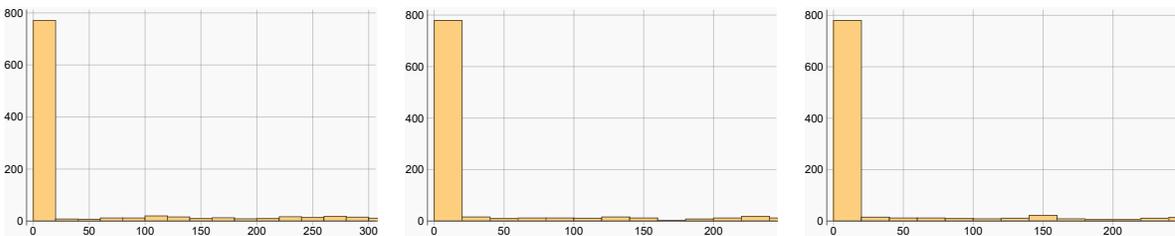


FIGURE 6.27 – Histogramme de la distance euclidienne (en nombre de pixels) entre les points saillants mis en correspondance avec les descripteurs fournis par les réseaux augmentés, et leur référence. Gauche : R4. Milieu : R5. Droite : R6.

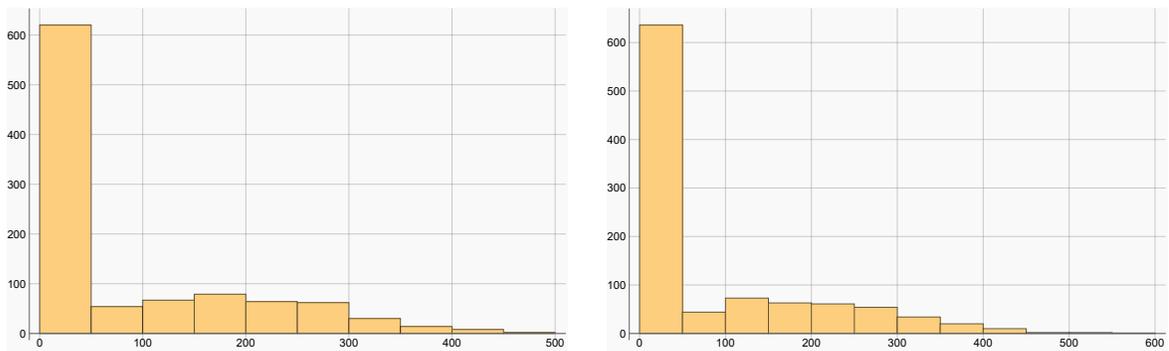


FIGURE 6.28 – Histogramme de la distance euclidienne (en nombre de pixels) entre les points saillants mis en correspondance avec les descripteurs fournis par le réseau non-entraîné et les descripteurs ORB, et leur référence. Gauche : NT. Droite : ORB.

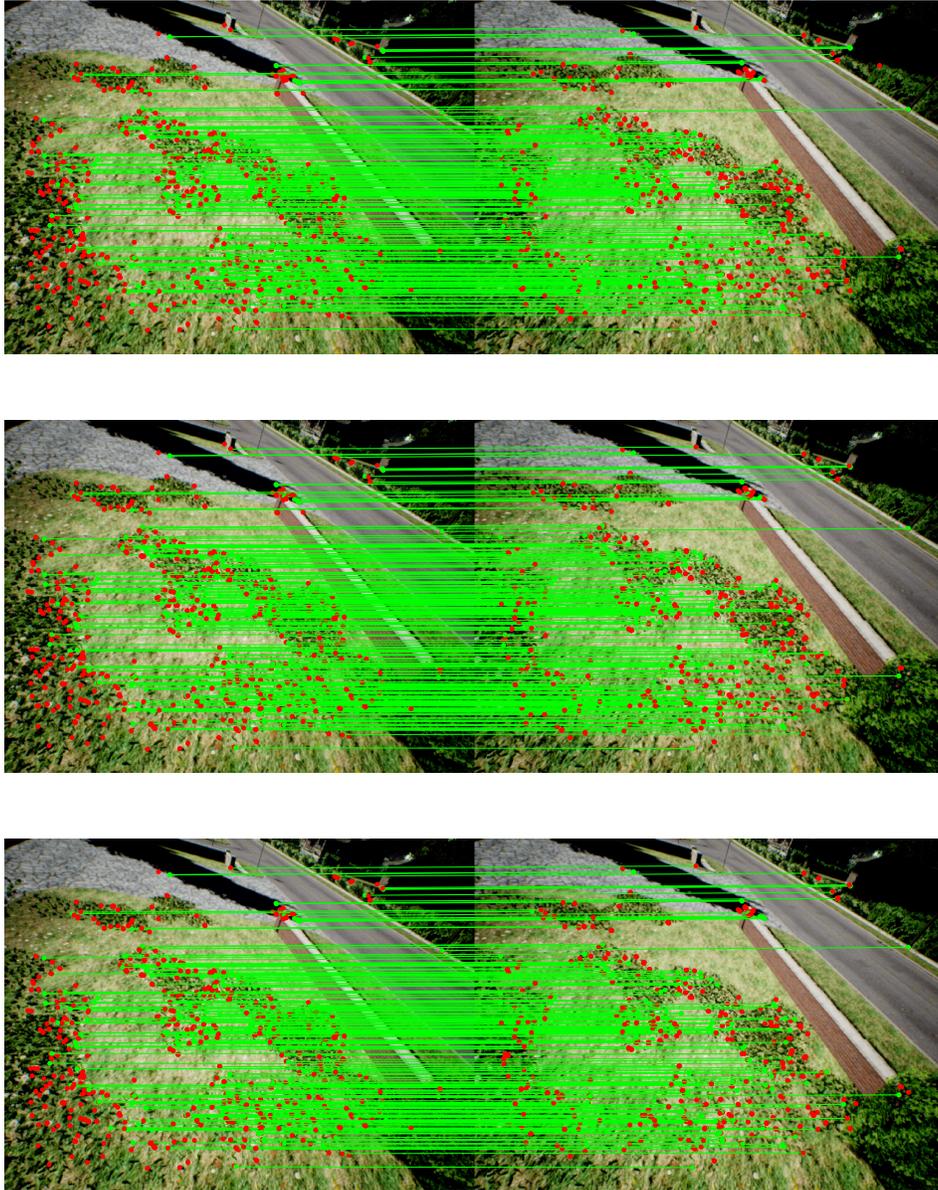


FIGURE 6.29 – Résultat qualitatif de la mise en correspondance entre les points saillants ORB de deux images consécutives réalisée avec les descripteurs fournis par les réseaux non-augmentés. Les lignes vertes représentent des mises en correspondance correctes. Les points rouges représentent des points qui n'ont pas été correctement mis en correspondance. Haut : R1. Milieu : R2. Bas : R3.

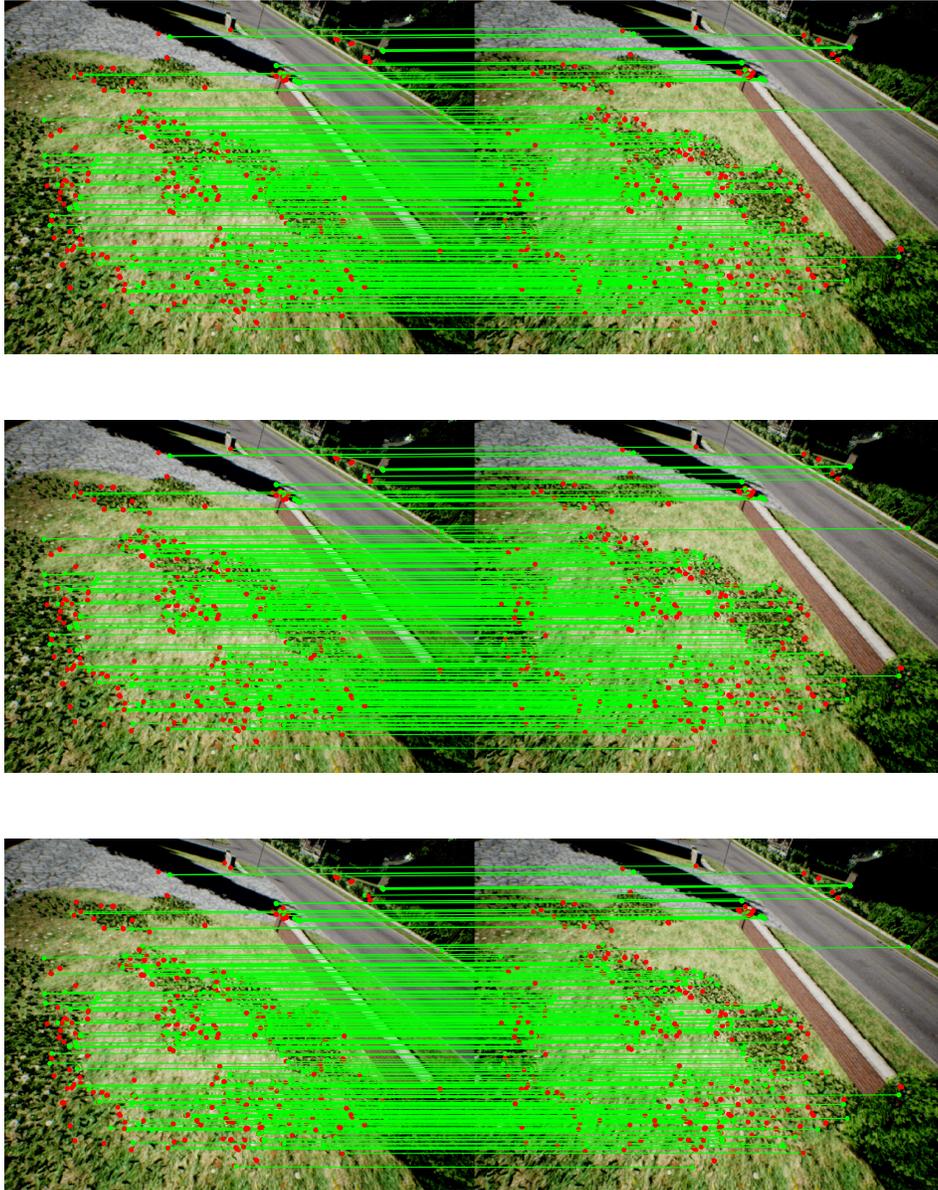


FIGURE 6.30 – Résultat qualitatif de la mise en correspondance entre les points saillants ORB de deux images consécutives réalisée avec les descripteurs fournis par les réseaux augmentés. Les lignes vertes représentent des mises en correspondance correctes. Les points rouges représentent des points qui n'ont pas été correctement mis en correspondance. Haut : R4. Milieu : R5. Bas : R6.

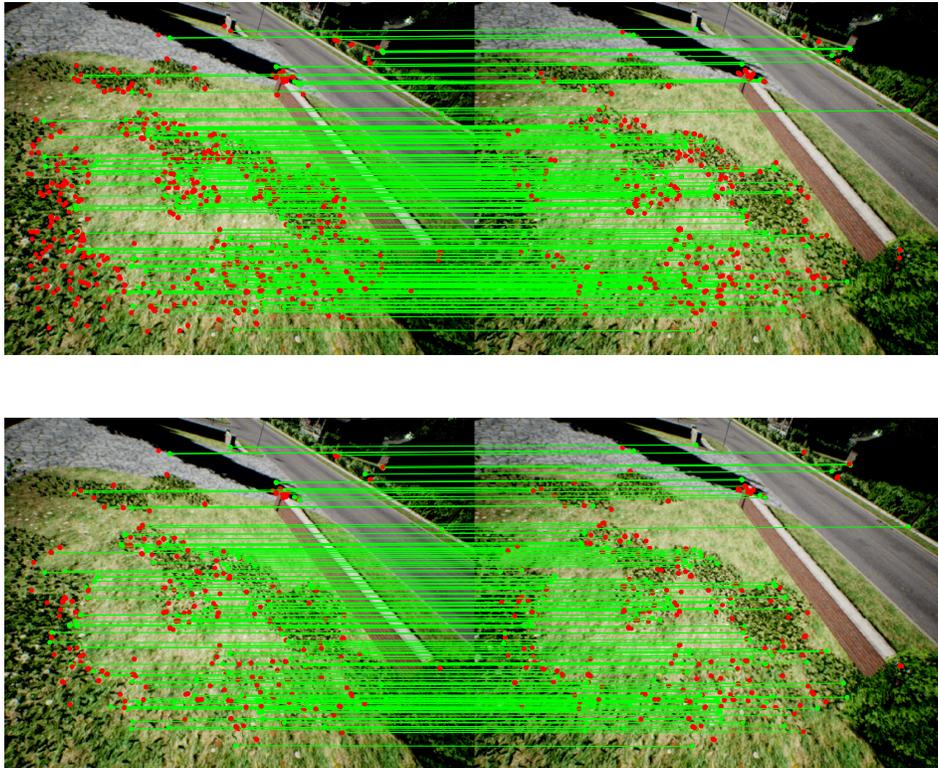


FIGURE 6.31 – Résultat qualitatif de la mise en correspondance entre les points saillants ORB de deux images consécutives réalisée avec les descripteurs fournis par le réseau non-entraîné R0, et les descripteurs ORB. Les lignes vertes représentent des mises en correspondance correctes. Les points rouges représentent des points qui n'ont pas été correctement mis en correspondance. Haut : R0. Bas : ORB.

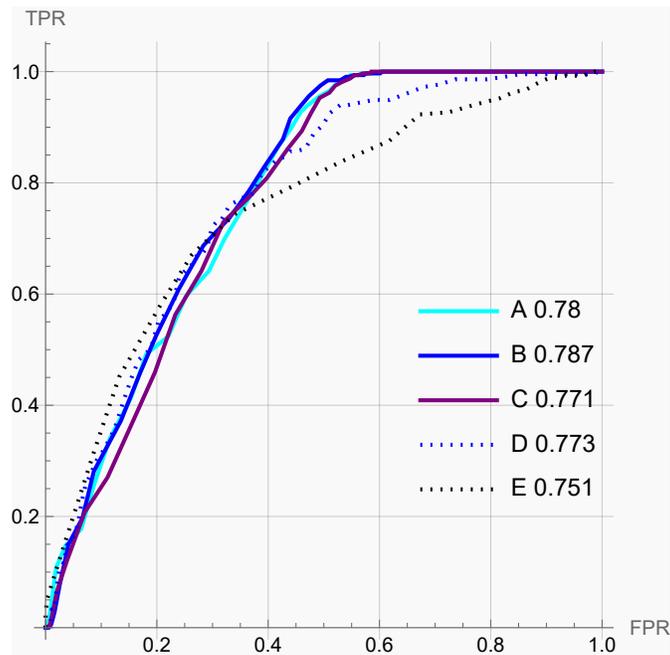
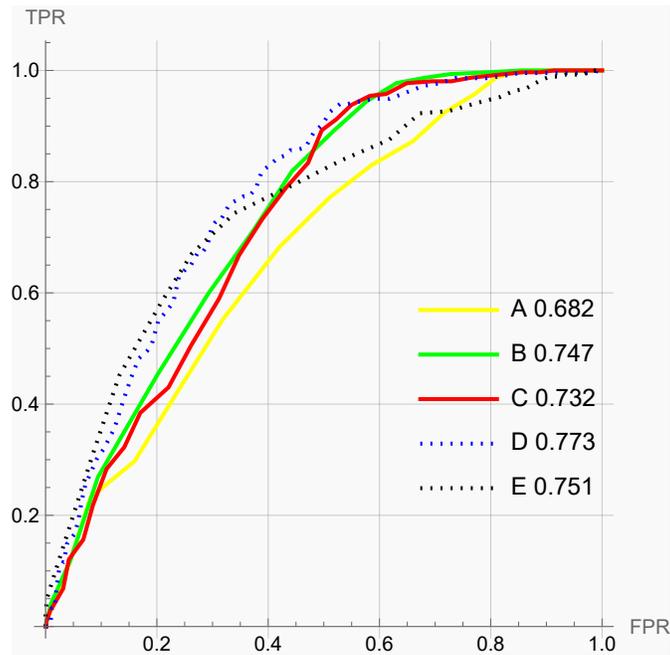


FIGURE 6.32 – Courbes ROC des réseaux augmentés et non-augmentés. Haut : Réseaux non-augmentés avec A : R1, B : R2, C : R3, D : ORB, E : R0. Bas : Réseaux augmentés avec A : R4, B : R5, C : R6, D : ORB, E : R0.

TABLEAU 6.3 – Résultats quantitatifs regroupant les métriques des différentes méthodes utilisées pour réaliser la mise en correspondance entre les points saillants ORB de deux images consécutives.

Méthode	VP	FP	FN	VN	Erreur moyenne	Erreur médiane	Écart-type erreur	ACC	AUC
R0	273	727	0	0	78.12	1.32	112.09	0.27	0.75
R1	306	694	0	0	38.13	1.15	98.97	0.31	0.68
R2	309	691	0	0	32.72	1.15	88.05	0.31	0.75
R3	307	675	0	18	37.2	1.12	92.41	0.32	0.73
R4	314	407	1	278	49.72	1.14	104.38	0.59	0.78
R5	320	418	0	262	46.83	1.12	101.53	0.58	0.79
R6	318	430	0	252	47.35	1.12	103.32	0.57	0.77
ORB	170	296	46	488	78.4	1.6	116.69	0.66	0.77

KP-ORB 13-34

Pour cette paire d'images, on voit clairement en observant les Figures 6.33 à 6.42 ainsi que le tableau des résultats 6.4, que les observations du test précédent sont cohérentes. On a beaucoup moins de FP et un peu moins de VP pour les réseaux augmentés que les réseaux non-augmentés, ce qui donne une meilleure ACC. On remarque que les réseaux augmentés font plus de FN et plus de VN, ce qui confirme encore une fois nos premières observations. ORB est la meilleure méthode mais toujours en étant très restrictif et avec une grosse erreur moyenne. Le réseau non-entraîné a de moins bons résultats encore une fois ce qui montre l'importance de l'apprentissage.

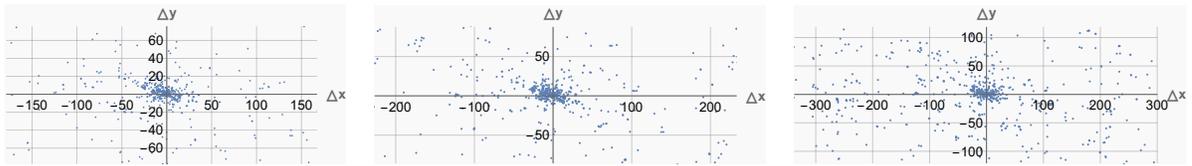


FIGURE 6.33 – Visualisation des différences entre les points saillants mis en correspondance avec les descripteurs fournis par les réseaux non-augmentés, et leur référence. Gauche : R1. Milieu : R2. Droite : R3.

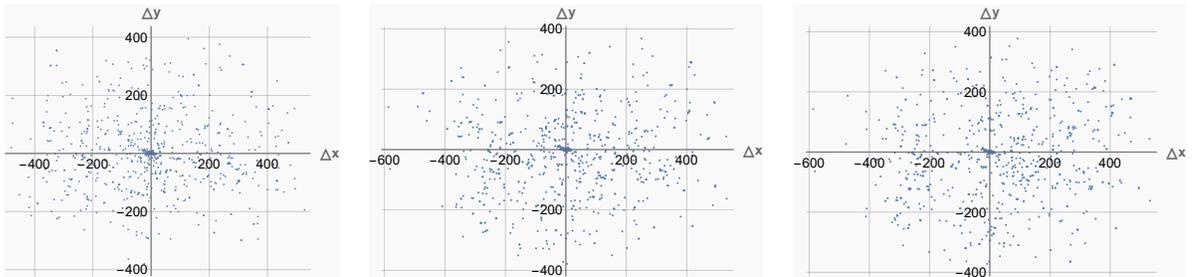


FIGURE 6.34 – Visualisation des différences entre les points saillants mis en correspondance avec les descripteurs fournis par les réseaux augmentés, et leur référence. Gauche : R4. Milieu : R5. Droite : R6.

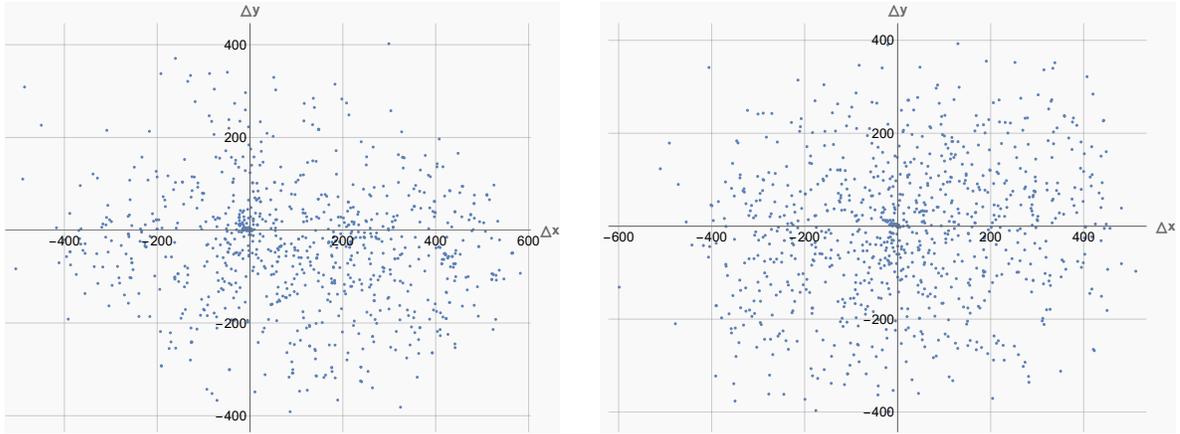


FIGURE 6.35 – Visualisation des différences entre les points saillants mis en correspondance avec les descripteurs fournis par le réseau non-entraîné et les descripteurs ORB, et leur référence. Gauche : NT. Droite : ORB.

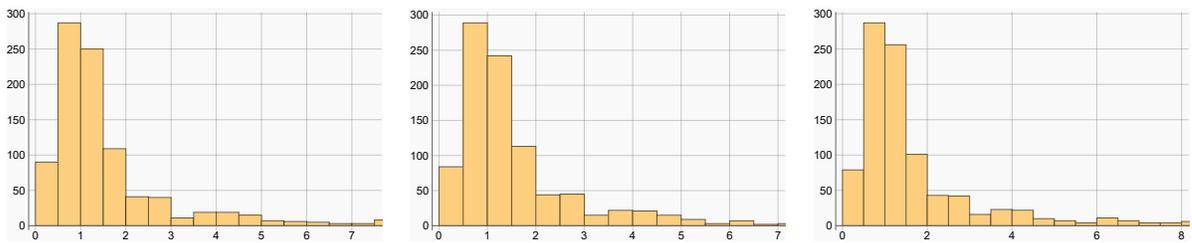


FIGURE 6.36 – Histogramme de la distance euclidienne (en nombre de pixels) entre les points saillants mis en correspondance avec les descripteurs fournis par les réseaux non-augmentés, et leur référence. Gauche : R1. Milieu : R2. Droite : R3.

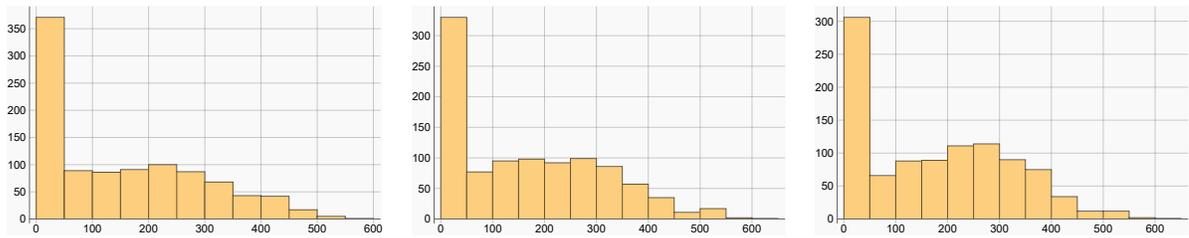


FIGURE 6.37 – Histogramme de la distance euclidienne (en nombre de pixels) entre les points saillants mis en correspondance avec les descripteurs fournis par les réseaux augmentés, et leur référence. Gauche : R4. Milieu : R5. Droite : R6.

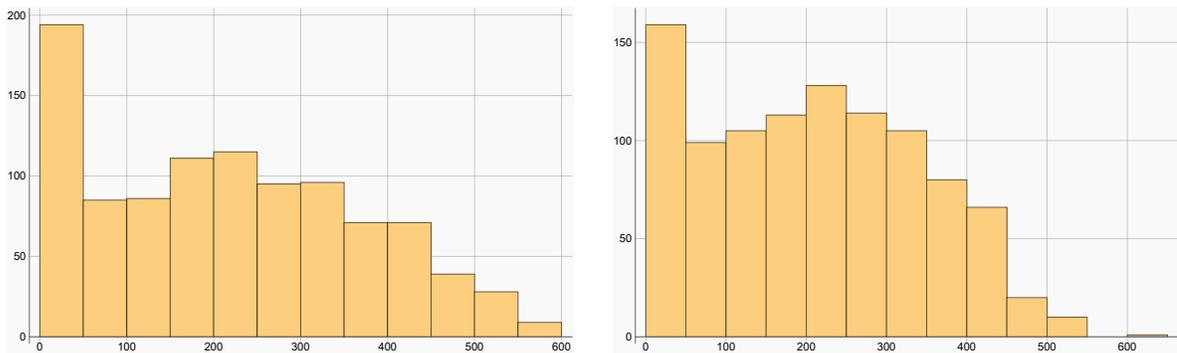


FIGURE 6.38 – Histogramme de la distance euclidienne (en nombre de pixels) entre les points saillants mis en correspondance avec les descripteurs fournis par le réseau non-entraîné et les descripteurs ORB, et leur référence. Gauche : NT. Droite : ORB.

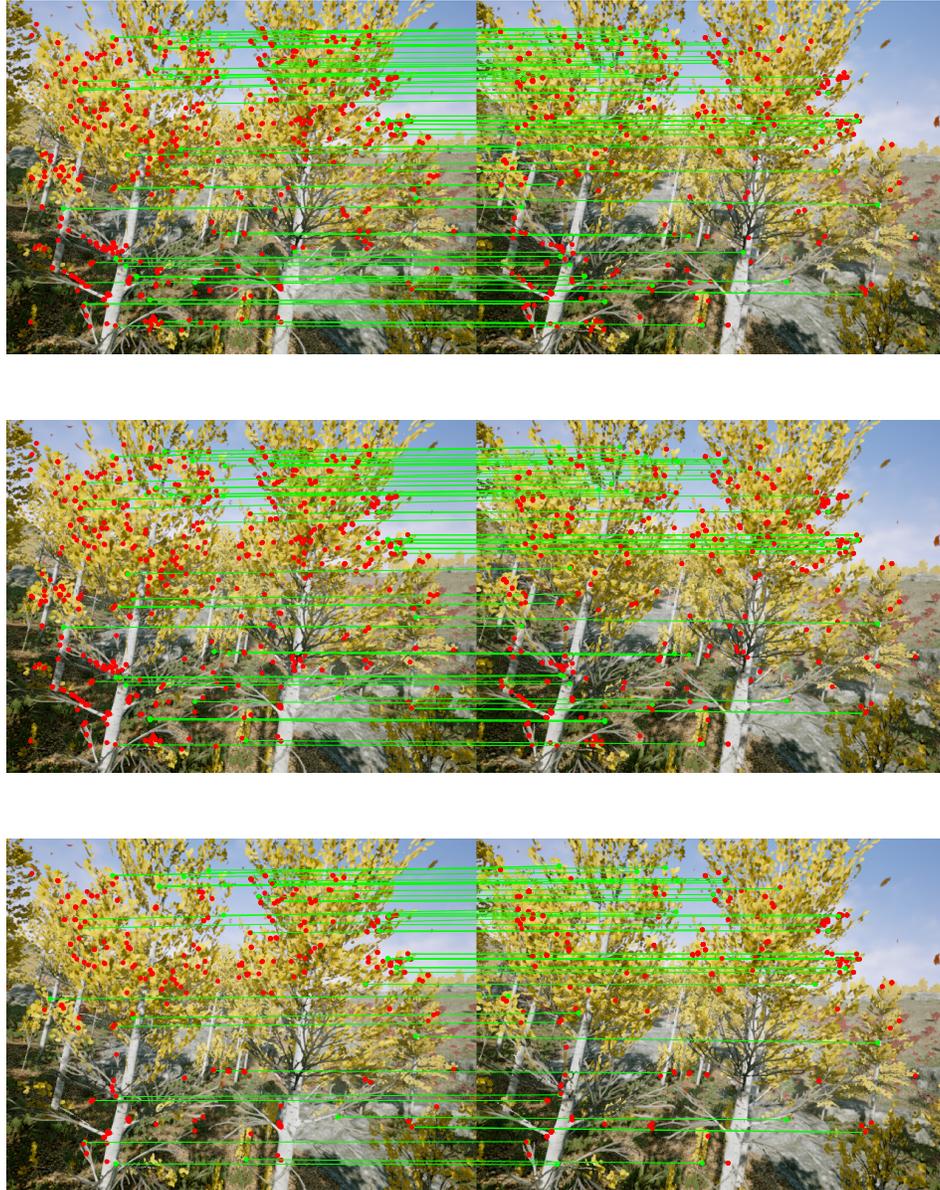


FIGURE 6.39 – Résultat qualitatif de la mise en correspondance entre les points saillants ORB de deux images consécutives réalisée avec les descripteurs fournis par les réseaux non-augmentés. Les lignes vertes représentent des mises en correspondance correctes. Les points rouges représentent des points qui n'ont pas été correctement mis en correspondance. Haut : R1. Milieu : R2. Bas : R3.

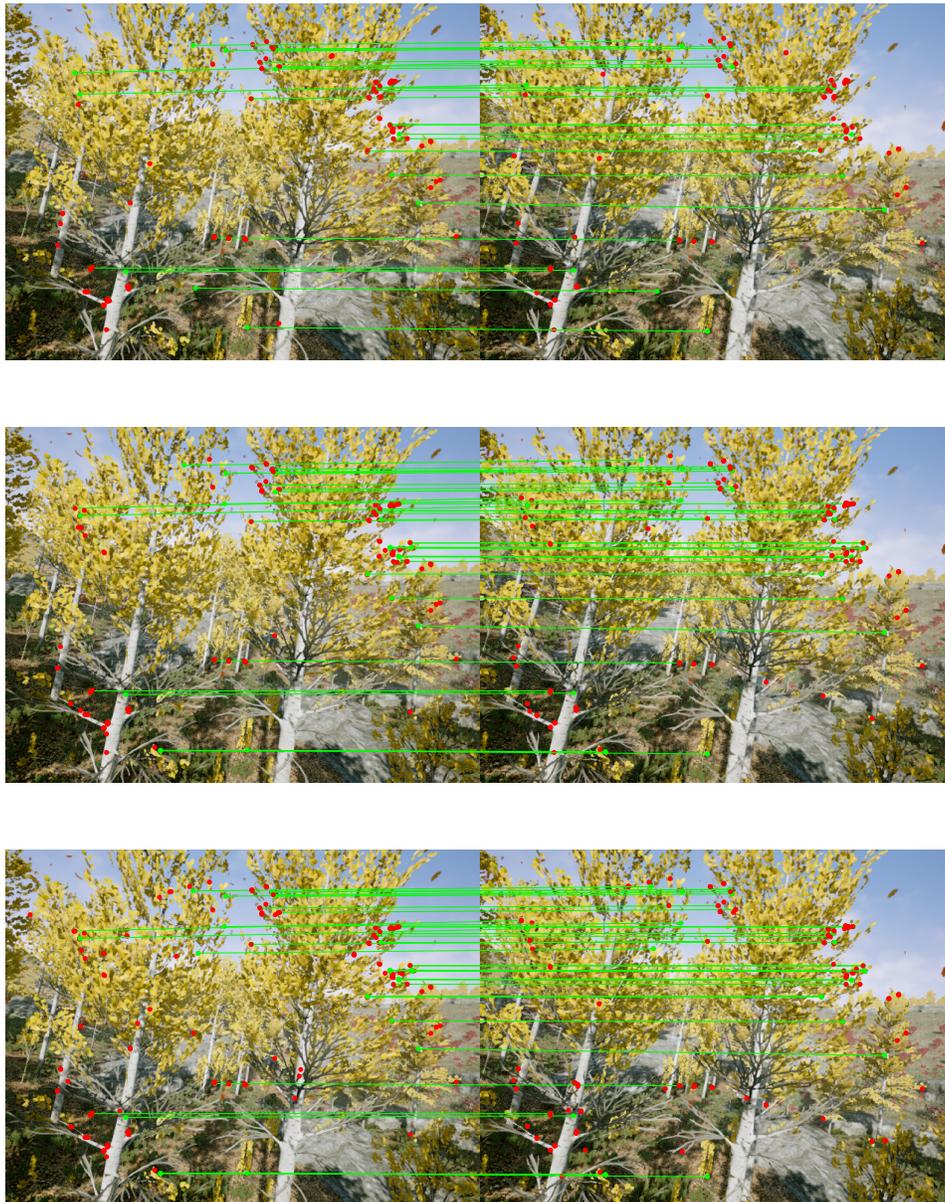


FIGURE 6.40 – Résultat qualitatif de la mise en correspondance entre les points saillants ORB de deux images consécutives réalisée avec les descripteurs fournis par les réseaux augmentés. Les lignes vertes représentent des mises en correspondance correctes. Les points rouges représentent des points qui n'ont pas été correctement mis en correspondance. Haut : R4. Milieu : R5. Bas : R6.

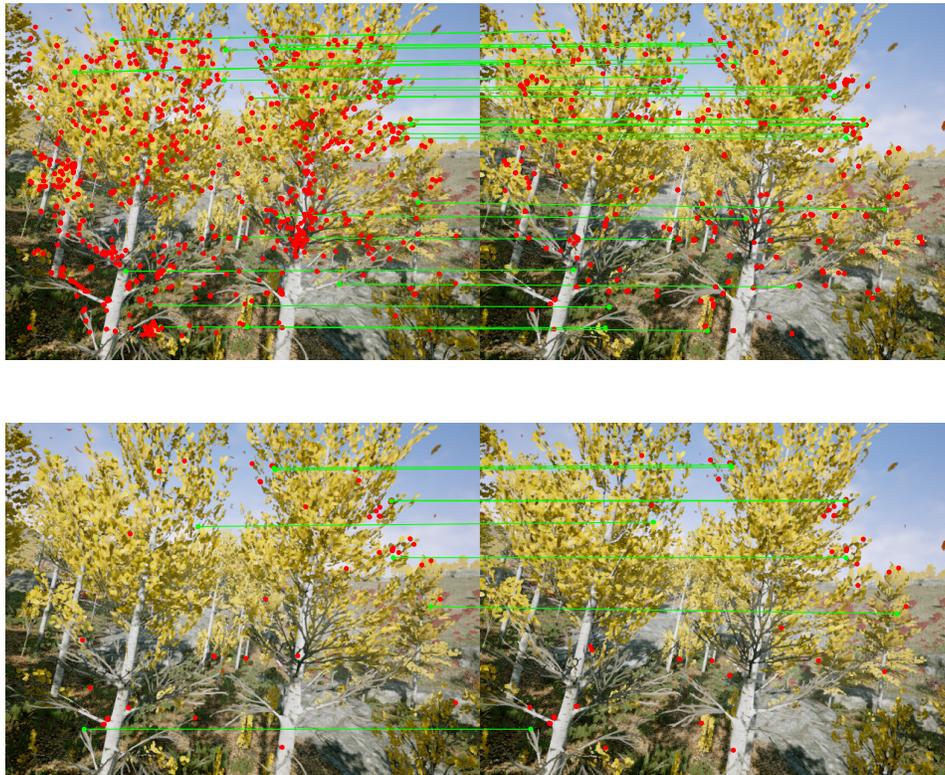


FIGURE 6.41 – Résultat qualitatif de la mise en correspondance entre les points saillants ORB de deux images consécutives réalisée avec les descripteurs fournis par le réseau non-entraîné R0, et les descripteurs ORB. Les lignes vertes représentent des mises en correspondance correctes. Les points rouges représentent des points qui n’ont pas été correctement mis en correspondance. Haut : R0. Bas : ORB.

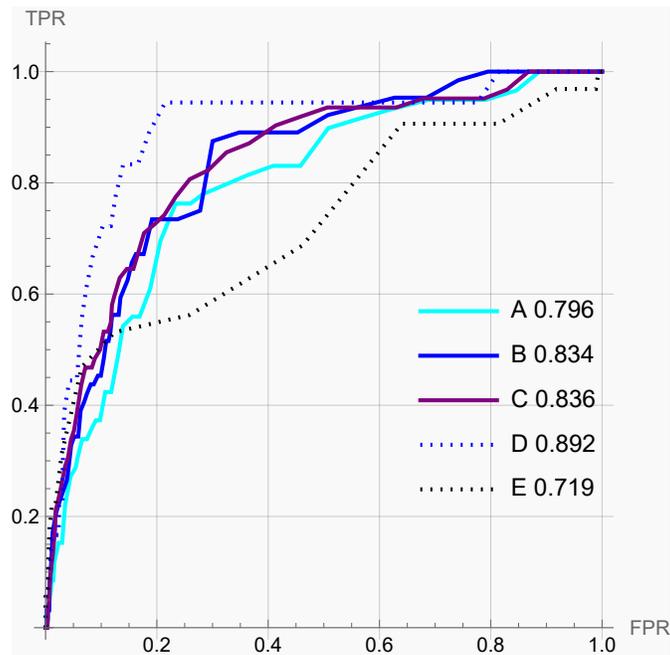
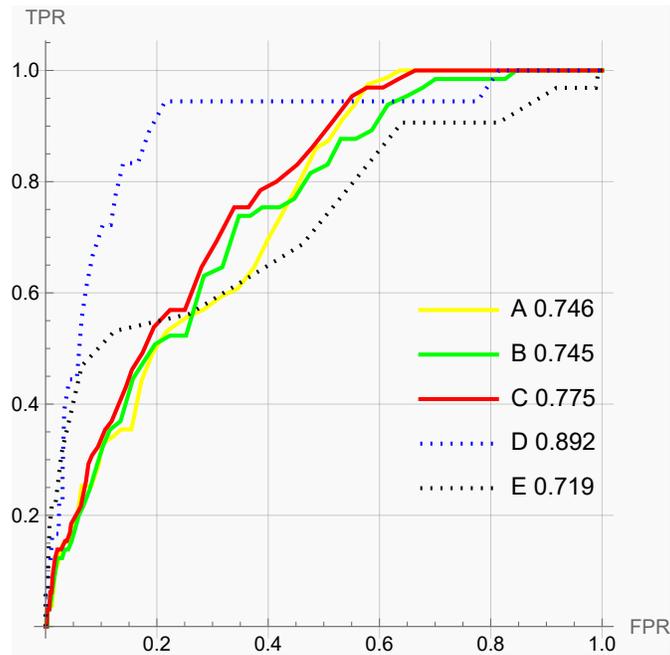


FIGURE 6.42 – Courbes ROC des réseaux augmentés et non-augmentés. Haut : Réseaux non-augmentés avec A : R1, B : R2, C : R3, D : ORB, E : R0. Bas : Réseaux augmentés avec A : R4, B : R5, C : R6, D : ORB, E : R0.

TABLEAU 6.4 – Résultats quantitatifs regroupant les métriques des différentes méthodes utilisées pour réaliser la mise en correspondance entre les points saillants ORB de deux images consécutives.

Méthode	VP	FP	FN	VN	Erreur moyenne	Erreur médiane	Écart-type erreur	ACC	AUC
R0	32	968	0	0	216.29	212.02	153.12	0.03	0.72
R1	79	586	0	335	84.88	18.68	117.77	0.41	0.75
R2	62	607	3	328	90.41	21.56	120.73	0.39	0.74
R3	49	317	16	618	96.76	30.13	118.65	0.67	0.77
R4	25	101	34	840	150.29	129.08	141.84	0.86	0.8
R5	38	126	26	810	165.13	149.33	146.33	0.85	0.83
R6	40	148	22	790	174.97	167.52	145.38	0.83	0.84
ORB	8	49	10	933	209.54	207.84	135.94	0.94	0.89

### *Observations générales*

Pour la paire d'images 10-260 (Figures A.25, A.26, A.27 disponibles en annexe) et la paire d'images 12-281 (Figures A.35, A.36, A.37 disponibles en annexe) on observe la même tendance que pour le précédent test, avec ORB qui est très restrictif et les réseaux qui le sont moins, avec plus de VP mais aussi plus de FP que ORB. En effet, les réseaux augmentés sont encore meilleurs sur l'ACC et l'AUC mais moins performants sur la précision. On voit donc que les réseaux non-augmentés se trompent plus souvent avec plus de FP, mais quand il se trompent leur erreur n'est pas aussi importante que les réseaux augmentés. Les réseaux augmentés se trompent moins mais quand ils se trompent, l'erreur est plus importante.

Pour la paire d'images 13-34 (Figures 6.39, 6.40, 6.41) nos observations du test précédent se confirment. On a beaucoup moins de FP et un peu moins de VP pour les réseaux réseaux non-augmentés que pour les réseaux augmentés, ce qui donne une meilleure ACC. On remarque que les réseaux augmentés font plus de FN et plus de VN ce qui confirme encore une fois nos premières observations. ORB est le meilleur réseau mais toujours en étant très restrictif et avec une grosse erreur moyenne. Le réseau non-entraîné a de moins bons résultats encore une fois ce qui montre l'importance de l'apprentissage.

### 6.3.4 KP-ORB-W

Pour ce dernier test on constitue 4 paires d'images non-consécutives prises dans des scènes différentes, pour lesquelles nous observons des changements d'orientation et de couleur importants.

Nous présentons ci-dessous les résultats du test KP-ORB-W pour les paires d'images 1-7, et 92-97. Les résultats pour les autres paires d'images sont disponibles en annexe.

#### KP-ORB-W 1-7

Pour la première image, nous remarquons en observant les Figures 6.43 à 6.52 ainsi que le tableau des résultats 6.5, que les réseaux augmentés sont incapables de produire des VP, contrairement aux réseaux augmentés, dont le meilleur est R1, et à ORB. Le nombre de FP est en revanche beaucoup plus important pour les réseaux non-augmentés, avec une erreur moyenne plus basse que pour les réseaux augmentés, comme nous l'avons déjà observé. L'ACC des réseaux augmentés est bien meilleure car ils sont assez restrictifs. Cependant leur AUC est dominée par ORB et les réseaux non-augmentés.

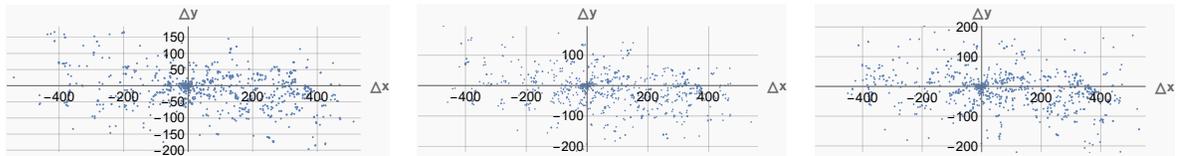


FIGURE 6.43 – Visualisation des différences entre les points saillants mis en correspondance avec les descripteurs fournis par les réseaux non-augmentés, et leur référence. Gauche : R1. Milieu : R2. Droite : R3.

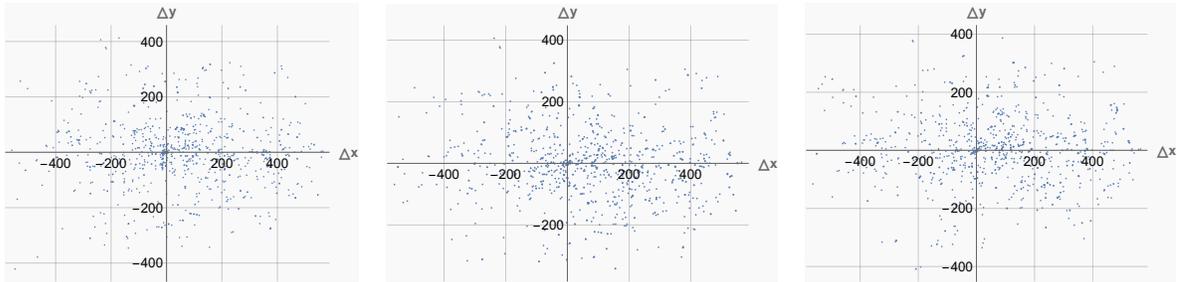


FIGURE 6.44 – Visualisation des différences entre les points saillants mis en correspondance avec les descripteurs fournis par les réseaux augmentés, et leur référence. Gauche : R4. Milieu : R5. Droite : R6.

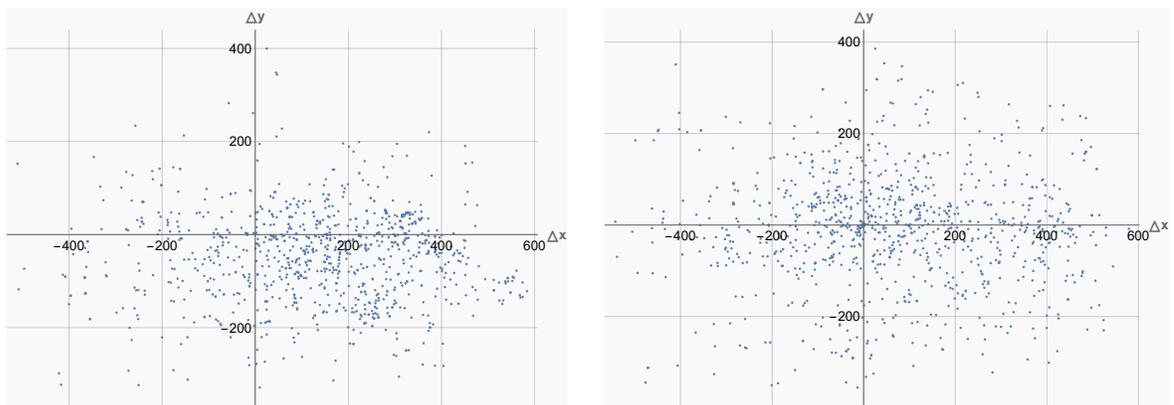


FIGURE 6.45 – Visualisation des différences entre les points saillants mis en correspondance avec les descripteurs fournis par le réseau non-entraîné et les descripteurs ORB, et leur référence. Gauche : NT. Droite : ORB.

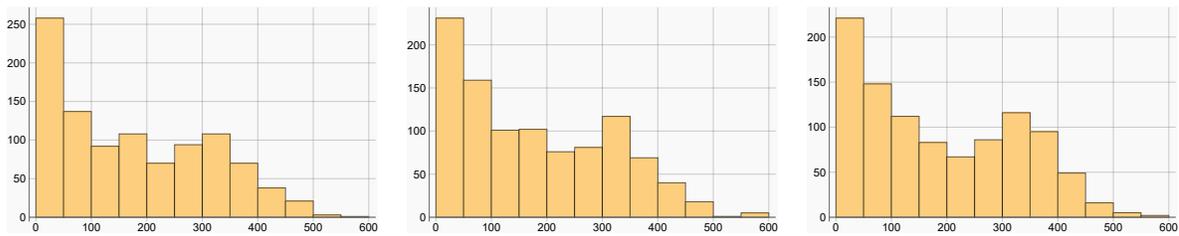


FIGURE 6.46 – Histogramme de la distance euclidienne (en nombre de pixels) entre les points saillants mis en correspondance avec les descripteurs fournis par les réseaux non-augmentés, et leur référence. Gauche : R1. Milieu : R2. Droite : R3.

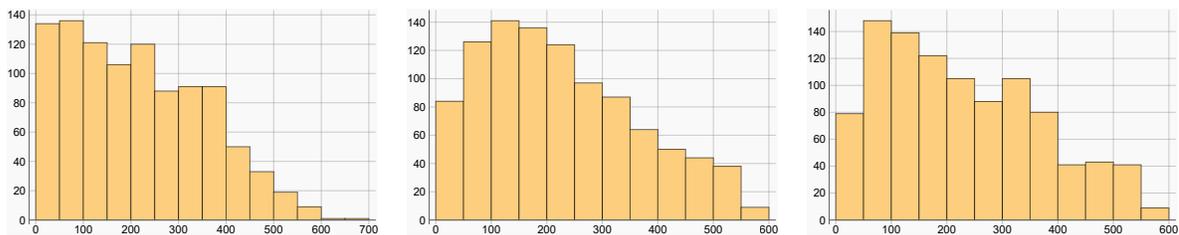


FIGURE 6.47 – Histogramme de la distance euclidienne (en nombre de pixels) entre les points saillants mis en correspondance avec les descripteurs fournis par les réseaux augmentés, et leur référence. Gauche : R4. Milieu : R5. Droite : R6.

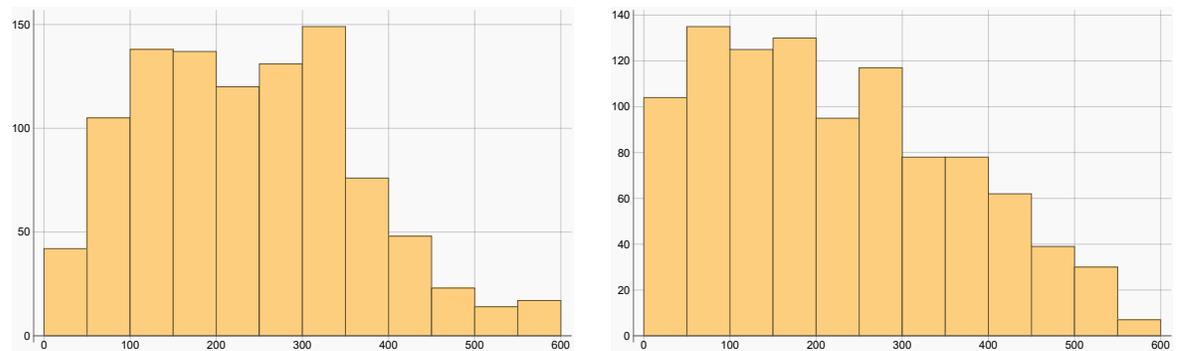


FIGURE 6.48 – Histogramme de la distance euclidienne (en nombre de pixels) entre les points saillants mis en correspondance avec les descripteurs fournis par le réseau non-entraîné et les descripteurs ORB, et leur référence. Gauche : NT. Droite : ORB.



FIGURE 6.49 – Résultat qualitatif de la mise en correspondance entre les points saillants ORB de deux images non-consécutives réalisée avec les descripteurs fournis par les réseaux non-augmentés. Les lignes vertes représentent des mises en correspondance correctes. Les lignes rouges représentent les mises en correspondance incorrectes. Haut : R1. Milieu : R2. Bas : R3.

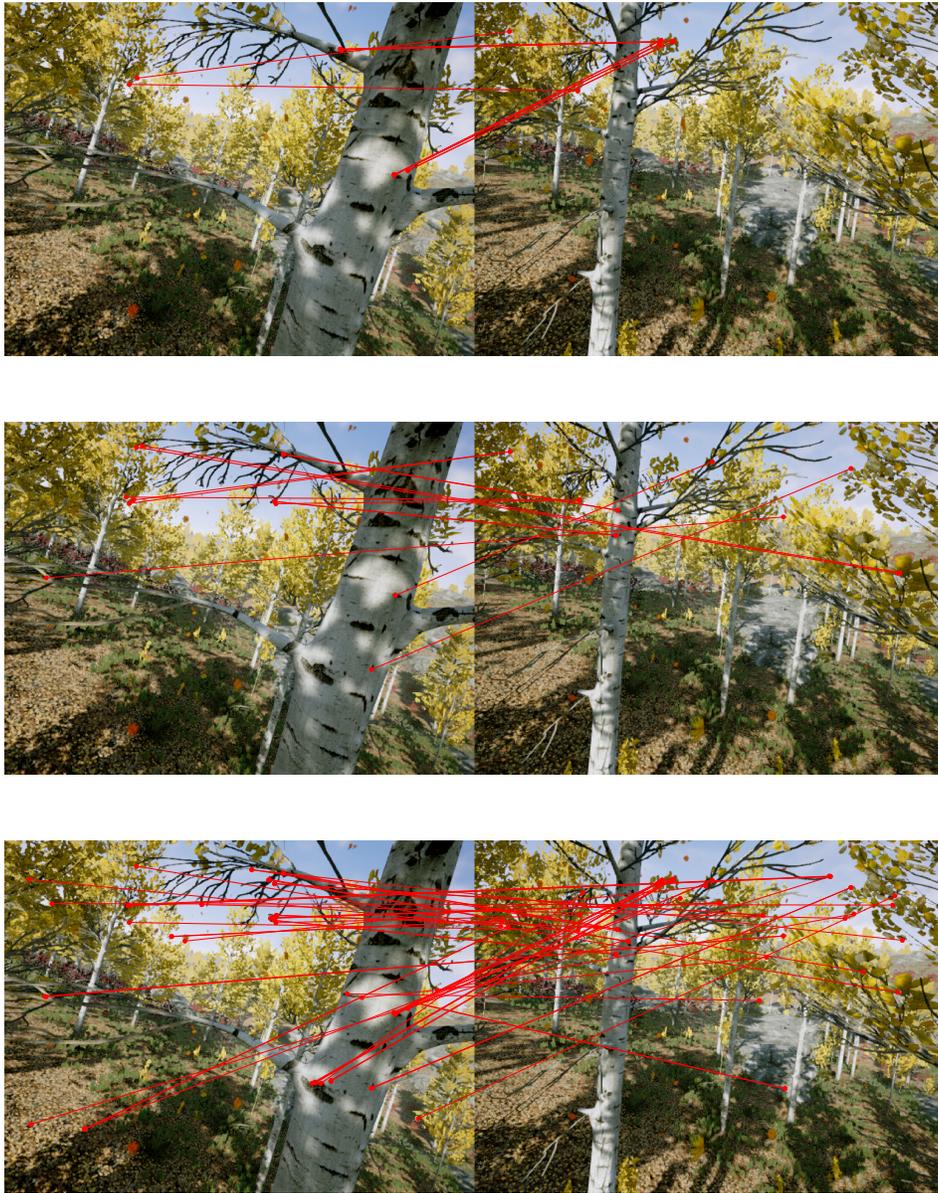


FIGURE 6.50 – Résultat qualitatif de la mise en correspondance entre les points saillants ORB de deux images non-consécutives réalisée avec les descripteurs fournis par les réseaux augmentés. Les lignes rouges représentent les mises en correspondance incorrectes. Haut : R4. Milieu : R5. Bas : R6.

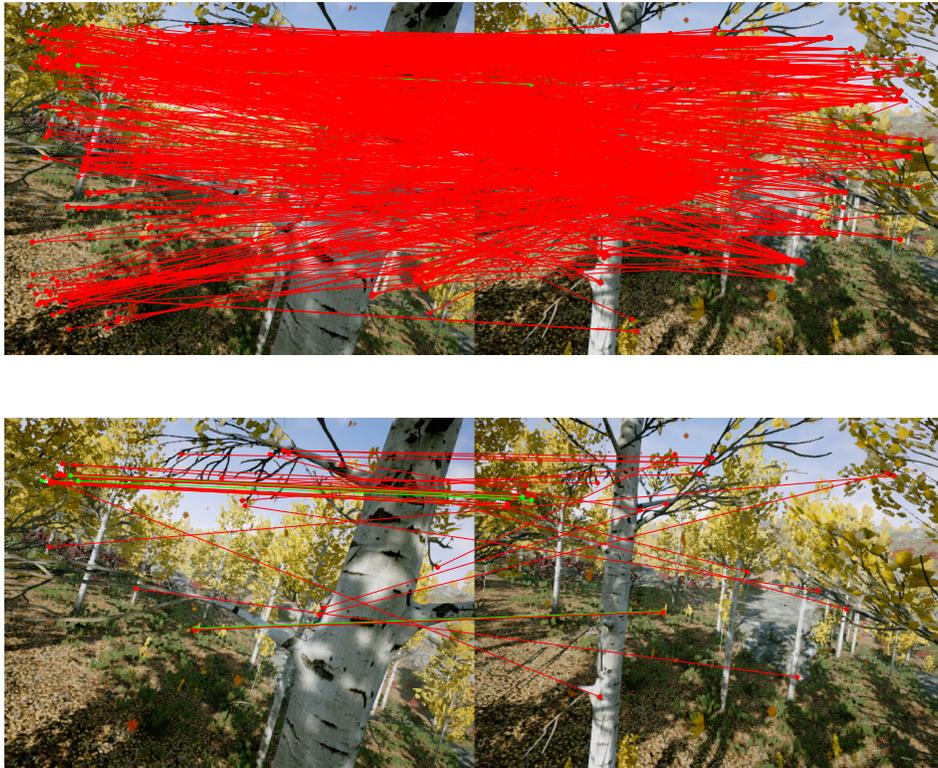


FIGURE 6.51 – Résultat qualitatif de la mise en correspondance entre les points saillants ORB de deux images non-consécutives réalisée avec les descripteurs fournis par le réseau non-entraîné R0, et les descripteurs ORB. Les lignes vertes représentent des mises en correspondance correctes. Les lignes rouges représentent les mises en correspondance incorrectes. Haut : R0. Bas : ORB.

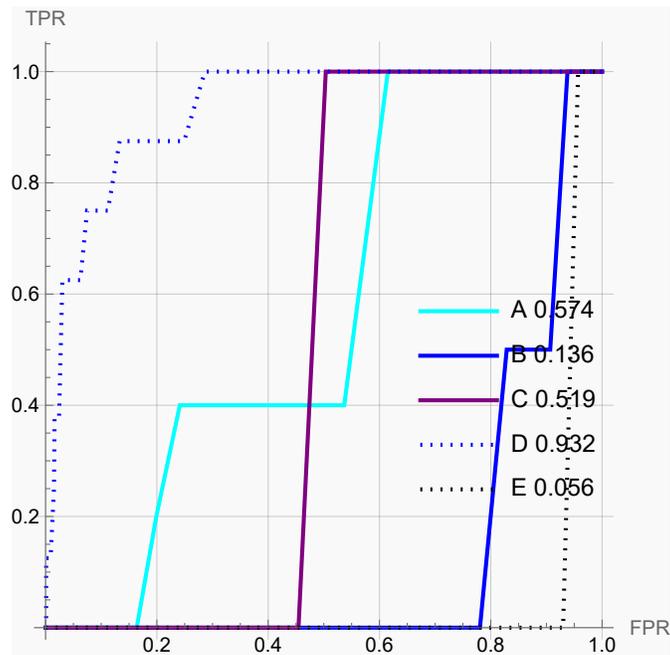
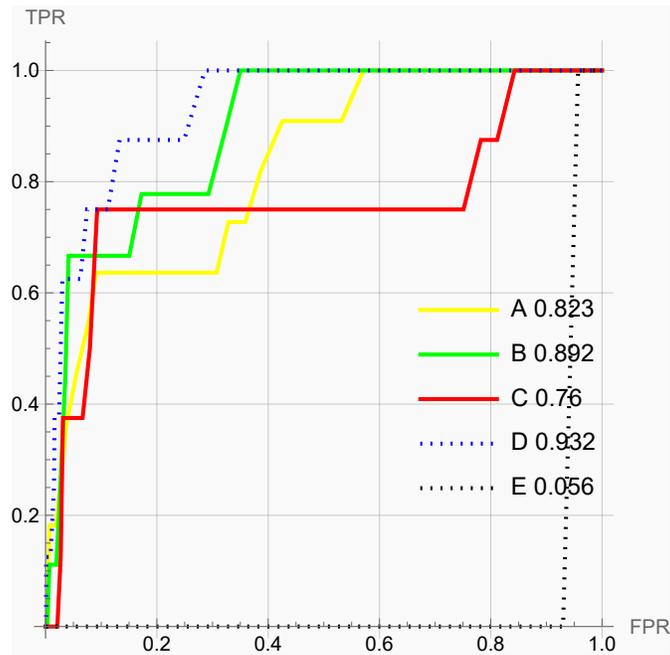


FIGURE 6.52 – Courbes ROC des réseaux augmentés et non-augmentés. Haut : Réseaux non-augmentés avec A : R1, B : R2, C : R3, D : ORB, E : R0. Bas : Réseaux augmentés avec A : R4, B : R5, C : R6, D : ORB, E : R0.

TABLEAU 6.5 – Résultats quantitatifs regroupant les métriques des différentes méthodes utilisées pour réaliser la mise en correspondance entre les points saillants ORB de deux images non-consécutives.

Méthode	VP	FP	FN	VN	Erreur moyenne	Erreur médiane	Écart-type erreur	ACC	AUC
R0	1	999	0	0	238.74	231.18	125.02	0.	0.06
R1	11	593	0	396	174.12	153.9	139.38	0.41	0.82
R2	9	543	0	448	176.13	154.67	138.13	0.46	0.89
R3	6	221	2	771	186.18	159.24	141.86	0.78	0.76
R4	0	11	5	984	214.	202.88	142.88	0.98	0.57
R5	0	13	2	985	227.34	205.	140.	0.98	0.14
R6	0	57	1	942	228.02	204.9	142.2	0.94	0.52
ORB	5	36	3	956	223.59	202.56	142.02	0.96	0.93

Pour cette paire d'images, nous notons sur les Figures 6.53 à 6.62 ainsi que sur le tableau des résultats 6.6, de plus faibles différences entre les réseaux augmentés et non-augmentés, avec les réseaux augmentés qui produisent plus de VP. Le réseau R4 se distingue particulièrement, avec le plus de VP parmi les méthodes. C'est le réseau R6 qui a la meilleure AUC. ORB conserve l'avantage en ce qui concerne l'ACC mais est toujours le moins performant pour l'erreur moyenne.

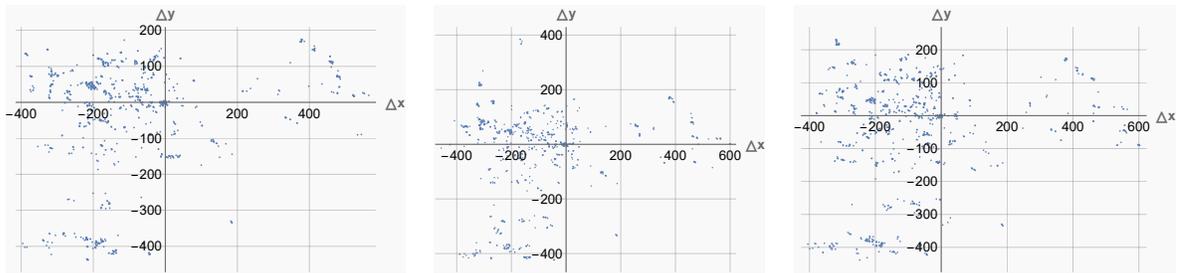


FIGURE 6.53 – Visualisation des différences entre les points saillants mis en correspondance avec les descripteurs fournis par les réseaux non-augmentés, et leur référence. Gauche : R1. Milieu : R2. Droite : R3.

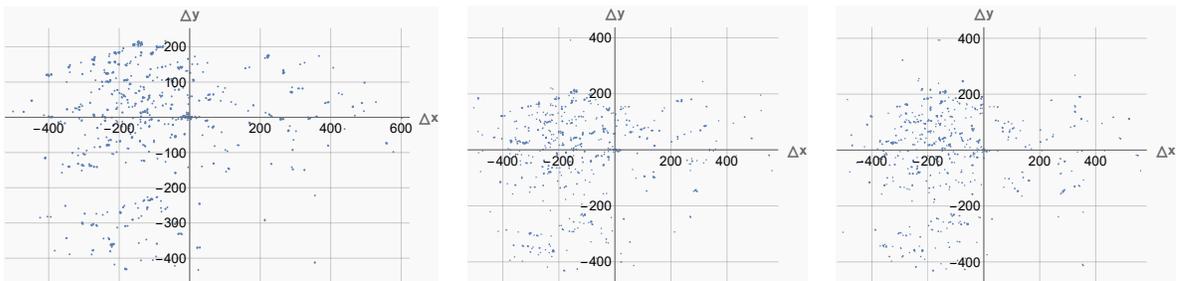


FIGURE 6.54 – Visualisation des différences entre les points saillants mis en correspondance avec les descripteurs fournis par les réseaux augmentés, et leur référence. Gauche : R4. Milieu : R5. Droite : R6.

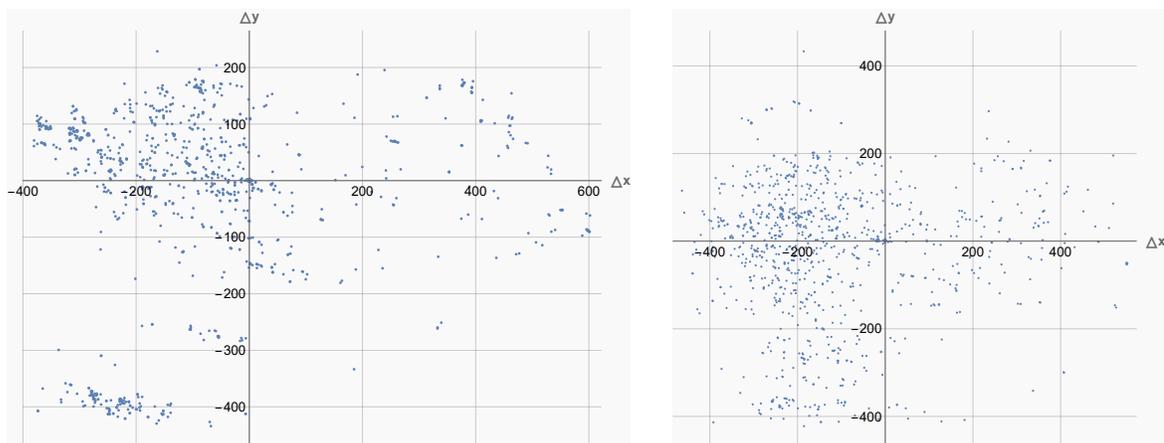


FIGURE 6.55 – Visualisation des différences entre les points saillants mis en correspondance avec les descripteurs fournis par le réseau non-entraîné et les descripteurs ORB, et leur référence. Gauche : NT. Droite : ORB.

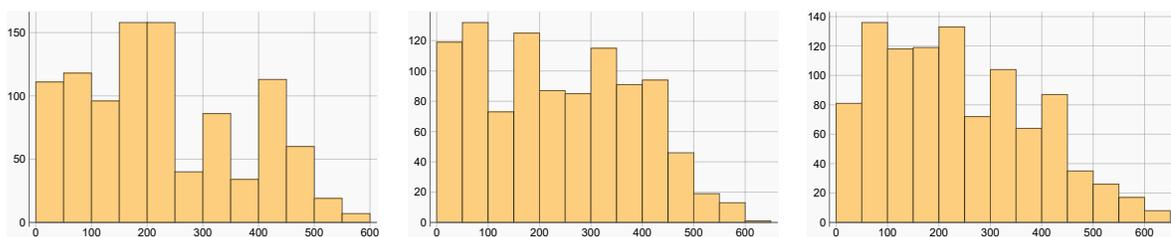


FIGURE 6.56 – Histogramme de la distance euclidienne (en nombre de pixels) entre les points saillants mis en correspondance avec les descripteurs fournis par les réseaux non-augmentés, et leur référence. Gauche : R1. Milieu : R2. Droite : R3.

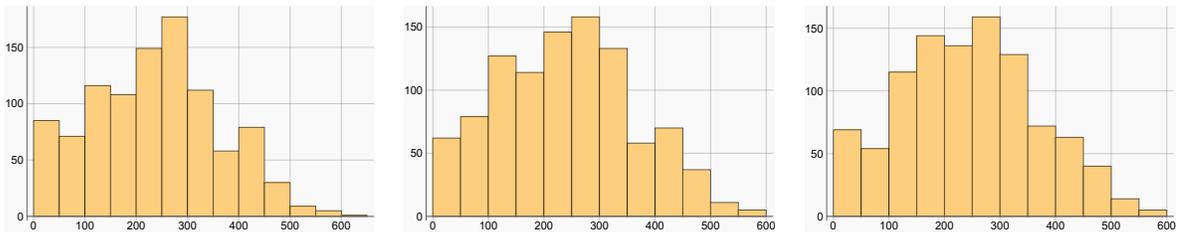


FIGURE 6.57 – Histogramme de la distance euclidienne (en nombre de pixels) entre les points saillants mis en correspondance avec les descripteurs fournis par les réseaux augmentés, et leur référence. Gauche : R4. Milieu : R5. Droite : R6.

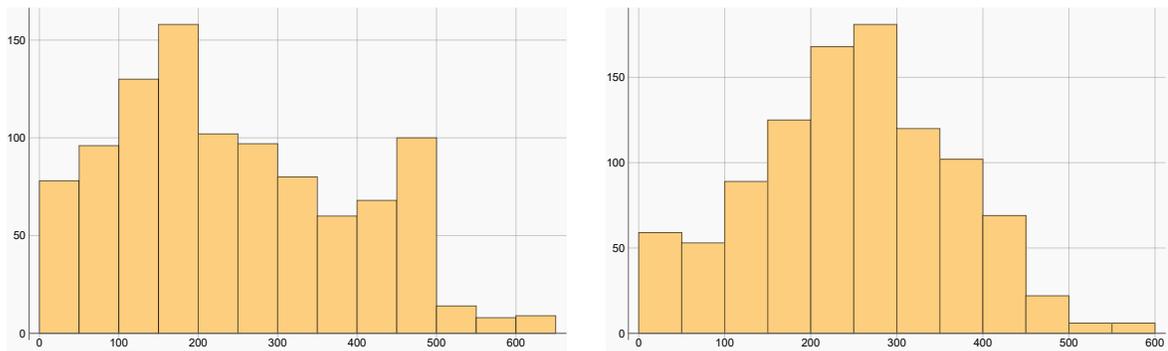


FIGURE 6.58 – Histogramme de la distance euclidienne (en nombre de pixels) entre les points saillants mis en correspondance avec les descripteurs fournis par le réseau non-entraîné et les descripteurs ORB, et leur référence. Gauche : NT. Droite : ORB.



FIGURE 6.59 – Résultat qualitatif de la mise en correspondance entre les points saillants ORB de deux images non-consécutives réalisée avec les descripteurs fournis par les réseaux non-augmentés. Les lignes vertes représentent des mises en correspondance correctes. Les lignes rouges représentent les mises en correspondance incorrectes. Haut : R1. Milieu : R2. Bas : R3.

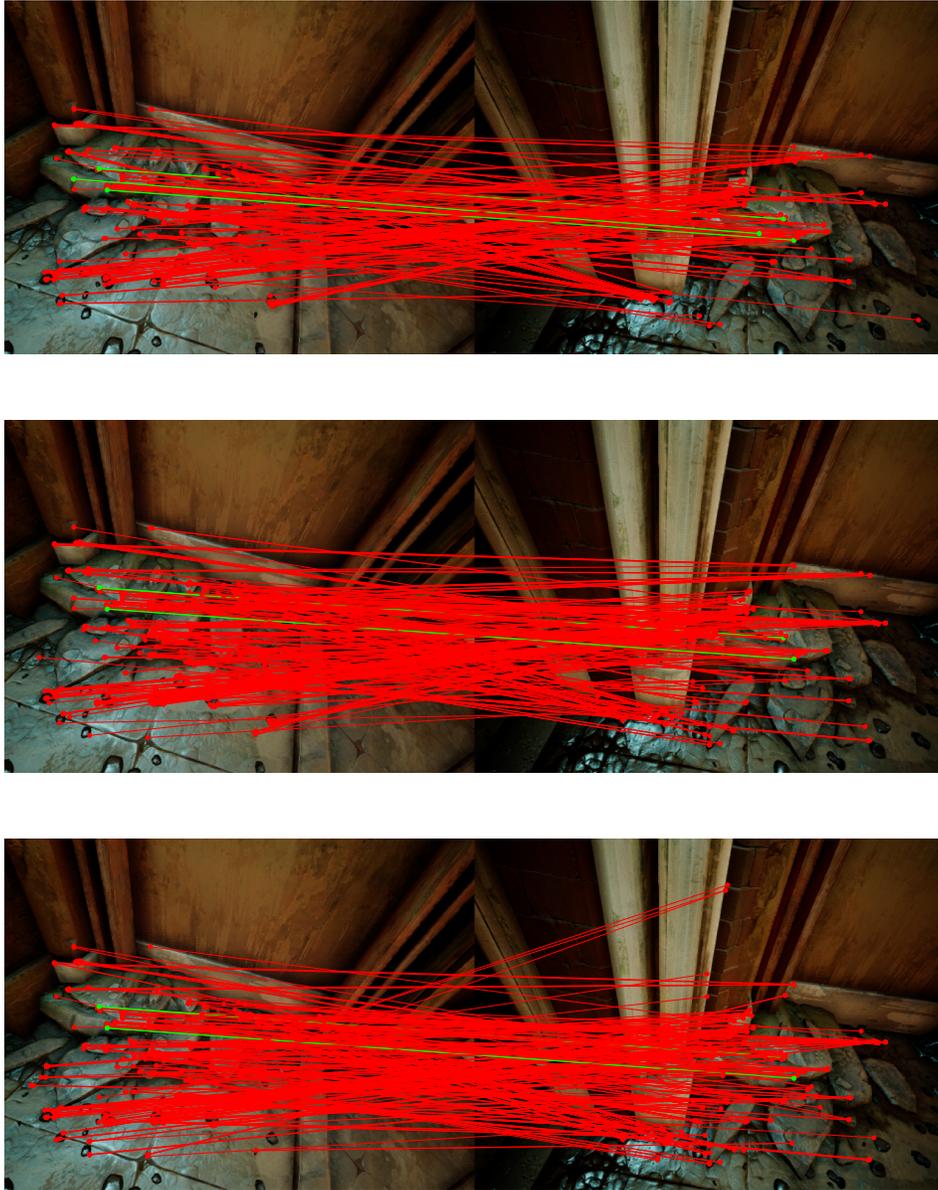


FIGURE 6.60 – Résultat qualitatif de la mise en correspondance entre les points saillants ORB de deux images non-consécutives réalisée avec les descripteurs fournis par les réseaux augmentés. Les lignes vertes représentent des mises en correspondance correctes. Les lignes rouges représentent les mises en correspondance incorrectes. Haut : R4. Milieu : R5. Bas : R6.



FIGURE 6.61 – Résultat qualitatif de la mise en correspondance entre les points saillants ORB de deux images non-consécutives réalisée avec les descripteurs fournis par le réseau non-entraîné R0, et les descripteurs ORB. Les lignes vertes représentent des mises en correspondance correctes. Les lignes rouges représentent les mises en correspondance incorrectes. Haut : R0. Bas : ORB.

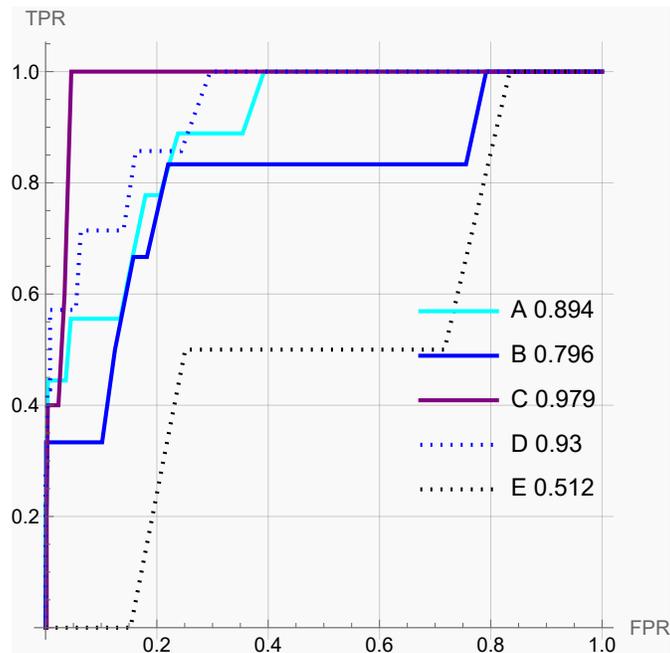
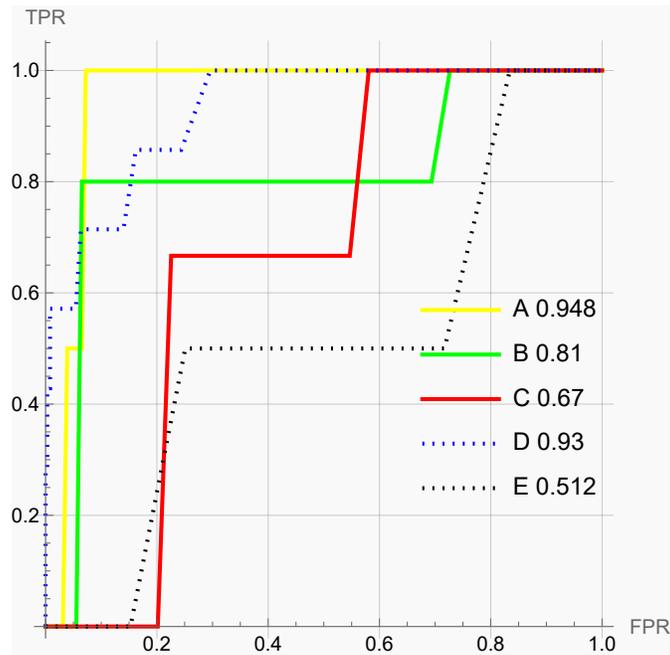


FIGURE 6.62 – Courbes ROC des réseaux augmentés et non-augmentés. Haut : Réseaux non-augmentés avec A : R1, B : R2, C : R3, D : ORB, E : R0. Bas : Réseaux augmentés avec A : R4, B : R5, C : R6, D : ORB, E : R0.

TABLEAU 6.6 – Résultats quantitatifs regroupant les métriques des différentes méthodes utilisées pour réaliser la mise en correspondance entre les points saillants ORB de deux images non-consécutives.

Méthode	VP	FP	FN	VN	Erreur moyenne	Erreur médiane	Écart-type erreur	ACC	AUC
R0	4	996	0	0	244.78	221.84	146.09	0.	0.51
R1	4	343	0	653	227.5	203.57	147.22	0.66	0.95
R2	4	377	1	618	236.75	223.93	148.63	0.62	0.81
R3	0	201	3	796	236.66	210.87	146.66	0.8	0.67
R4	8	282	1	709	234.52	244.55	126.88	0.72	0.89
R5	5	405	1	589	240.42	246.95	123.72	0.59	0.8
R6	5	383	0	612	244.62	247.44	124.24	0.62	0.98
ORB	5	118	2	875	250.09	251.05	116.77	0.88	0.93

### *Observations générales*

En général, nous pouvons voir que l'apprentissage a un réel effet sur les résultats obtenus. De plus, les réseaux ont toujours le plus grand taux de VP. Les réseaux non-augmentés ont clairement un impact sur la réduction des FP. Les réseaux ont des AUC qui rivalisent avec ORB. La mise en correspondance est particulièrement difficile sur les images non-consécutives.

#### *6.3.5 Discussion*

De façon générale, on peut constater que les réseaux sont meilleurs que ORB en erreur moyenne de la distance entre le point mis en correspondance et la référence. On peut supposer que leurs descripteurs sont beaucoup plus précis grâce notamment à l'invariance bâtie par les paires positives créées à partir de points d'une même trajectoire par le générateur. Cependant, le réseau entraîné ne bat pas systématiquement ORB pour l'ACC et l'AUC, bien que son AUC soit dans certains cas légèrement supérieure ou égale à ORB, démontrant le potentiel de notre méthode. Ce qui est entendu ici par "beaucoup plus précis", c'est que sur l'ensemble des expériences KP-ORB et KP-ORB-W, les réseaux entraînés sont systématiquement meilleurs que ORB sur l'erreur (la distance euclidienne en nombre de pixels entre la prédiction et le point de référence) moyenne et médiane. Nous remarquons aussi que la variation de l'hyperparamètre du ratio entre paires positives et négatives (allant jusqu'à 3 dans nos expériences) n'a pas d'effet visible sur les résultats. Nous pensons que l'ordre de grandeur de ce ratio devrait être bien plus important pour avoir un réel effet.

Nous observons également que les réseaux, et notamment augmentés, sont moins performants en erreur moyenne sur le test KP-ORB que sur le test KP-RAND, alors qu'ils font beaucoup moins de FP. Nous pensons que cela est dû au fait que KP-RAND utilise une recherche dans une fenêtre pour trouver le point correspondant, et qu'il est donc plus facile de trouver le bon point. La mise en correspondance est réalisée en calculant

les produits scalaires entre le descripteur du point saillant de l'image de départ et les descripteurs des points dans la fenêtre de l'image suivante. Pour la correspondance on choisit le point dans la fenêtre de l'image suivante dont le descripteur donne le produit scalaire maximal avec le descripteur du point saillant de l'image de départ. De plus, cette recherche bornée limite la possibilité de faire de grosses erreurs, tandis que pour KP-ORB la recherche se fait parmi tous les points saillants et il arrive que des points totalement incorrects soient mis en correspondance, ce qui augmente cette erreur moyenne. Nous pensons que les réseaux augmentés font beaucoup moins de FP car ils ont vu des exemples qui les empêchent de se tromper sur les pièges, notamment les exemples négatifs difficiles, qui se ressemblent mais ne représentent pas le même point.

D'autre part, les réseaux non-augmentés produisent plus de VP : ils mettent en correspondance de façon plus permissive, donc les zones qui se ressemblent sont correctement mises en correspondance.

La méthode ORB produit moins de VP mais au bénéfice de produire beaucoup moins de FP : elle est plus restrictive car a été conçue pour faire de la recherche d'images ("image retrieval"), là où les points varient beaucoup. Cette méthode prend en compte toutes les régions de l'image et rend donc la mise en correspondance restrictive. Cependant, quand bien même ORB réalise moins de FP, son erreur moyenne est élevée.

Malgré que ses poids soient aléatoires, le réseau non-entraîné trouve des correspondances. Ceci est dû à l'architecture même du réseau qui comporte des convolutions prenant en compte les voisinages des pixels et lui donnant de bonnes prédispositions pour la mise en correspondance. L'entraînement permet d'apprendre au réseau à générer des descripteurs plus robustes face aux FP, notamment lorsque ces réseaux ont été entraînés avec des exemples négatifs ambigus. Les résultats du réseau non-entraîné démontrent d'ailleurs que la tâche de mise en correspondance sur des patchs correspondants qui se ressemblent est facile. C'est la distinction juste entre des patchs qui ne se ressemblent pas qui est difficile et nécessite un apprentissage.

Pour améliorer les résultats de notre méthode, il faudrait idéalement transformer les FP en VP. On a déjà des exemples négatifs plus difficiles, ce qui permet d'apprendre à ne pas mettre en correspondance trop facilement et à bien voir que deux points qui se ressemblent peuvent en fait être différents. Il faudrait montrer que deux patchs qui ne se ressemblent pas peuvent en réalité appartenir au même point. En effet, un exemple positif facile correspond à deux patchs qui se ressemblent et correspondent, tandis qu'un exemple positif difficile correspond à deux patchs qui ne se ressemblent pas et correspondent.

En conclusion, il faudrait augmenter le nombre de paires positives que le réseau voit, ou du moins augmenter le nombre de paires difficiles, en prenant des patchs au sein d'une trajectoire qui soient les plus différents possibles. On pourrait contrôler le taux de ces exemple positifs difficiles, et voir son influence sur les résultats comme c'est le cas pour les exemples négatifs.

## Chapitre 7

### CONCLUSION

---

Comme le montrent les résultats obtenus sur les tests de mise en correspondance pour plusieurs paires d'images consécutives ou non-consécutives, nous remarquons que malgré les prédispositions du réseau non-entraîné à obtenir des résultats acceptables, de par son architecture convolutive qui prend en considération le voisinage des pixels, les réseaux entraînés sont en général meilleurs, ce qui démontre que les réseaux ont appris à produire des descripteurs locaux représentatifs, robustes, et suffisamment invariants. Le réseau non-entraîné est naturellement bon pour mettre en correspondance des patches qui se ressemblent car il prend en compte le voisinage des pixels. Le but de l'entraînement est de limiter le nombre de FP. Pour le test de KP-RAND, où on effectue la recherche dans une fenêtre  $40 \times 40$ , alors les descripteurs sont très similaires pour des régions très similaires et il y a ici moins d'occasions de rencontrer des FP, surtout sur des images consécutives. En revanche, dès lors qu'une image contient des motifs qui se répètent, comme des feuilles, les réseaux se trompent et R0 a de très mauvais résultats sur l'erreur moyenne. R0 a aussi des résultats catastrophiques, notamment pour les images non-consécutives. En conclusion, nous pourrions dire que R0 est presque comparable aux réseaux entraînés sur des patches correspondants qui se ressemblent, mais il a de grandes chances d'échouer dans les autres situations, car le manque d'apprentissage ne lui a pas appris à correctement éliminer les FP.

Le mode d'apprentissage contrastif auto-supervisé a l'avantage d'utiliser les données telles quelles, et d'apprendre une représentation satisfaisante pour la tâche sous-jacente de mise en correspondance sans nécessiter d'étiquetage. Le réseau peut être utilisé sur des patches d'image ou sur l'image au complet grâce à son architecture entièrement convolutive. De plus, le réseau a la capacité de fournir des descripteurs de façon dense,

de ne pas être limité par un détecteur, et peut s'adapter et se comparer à d'autres méthodes utilisant leurs propres points saillants.

Contrairement aux travaux précédents qui utilisent des augmentations de données manuelles pour bâtir l'invariance en appliquant des transformations aléatoires aux points, nous fournissons une construction d'invariance particulièrement adaptée à la tâche de mise en correspondance pour des données séquentielles. Le niveau d'invariance construit est spécifiquement adapté à la tâche et aux transformations ayant réellement lieu lors d'un parcours de SLAM ou de SfM. De plus, la simplicité du réseau qui est constitué de quelques convolutions permet de montrer qu'un tel réseau peut être intégré au sein d'applications requérant un traitement en temps réel.

De façon générale, les résultats montrent une vraie différence avec la méthode ORB, qui a tendance à être plus restrictive mais de ce fait produit moins de VP. Les résultats de l'AUC soulignent l'importance du seuillage et démontrent le potentiel de notre méthode.

Comme nous l'avons décrit dans les résultats, l'augmentation des données introduisant des exemples négatifs ambigus, c'est-à-dire des patchs représentant des points différents mais en apparence similaire, a un effet bénéfique sur l'apprentissage, et réduit dans certains cas drastiquement les FP.

Dans cette direction, comme nous l'avons réalisé avec des exemples négatifs difficiles, de futurs travaux pourraient s'intéresser à l'introduction d'exemples positifs ambigus, pour lesquels deux patchs qui ne se ressemblent pas appartiendraient au même point. C'est ce que nous avons réalisé dans une certaine mesure avec la création des paires positives, mais il serait intéressant de contrôler ce niveau d'ambiguïté, en prenant, dans un cas extrême, les patchs les plus différents possibles au sein d'une même trajectoire. Nous pensons qu'une telle augmentation de données pourrait augmenter le nombre de VP.

## Bibliographie

---

- [1] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. HPatches : A Benchmark and Evaluation of Handcrafted and Learned Local Descriptors. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3852–3861, Honolulu, HI, July 2017. IEEE.
- [2] Hudson Martins Silva Bruno and Esther Luna Colombini. LIFT-SLAM : A deep-learning feature-based monocular visual SLAM method. *Neurocomputing*, 455 : 97–110, September 2021. ISSN 09252312.
- [3] Mihai Bujanca, Xuesong Shi, Matthew Spear, Pengpeng Zhao, Barry Lennox, and Mikel Luján. Robust SLAM Systems : Are We There Yet? In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5320–5327, September 2021. ISSN : 2153-0866.
- [4] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. BRIEF : Binary Robust Independent Elementary Features. In *Computer Vision – ECCV 2010*, volume 6314 of *Lecture Notes in Computer Science*, pages 778–792, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-15560-4 978-3-642-15561-1.
- [5] Phototourism Challenge. Image Matching Workshop CVPR 2019. URL <https://image-matching-workshop.github.io/challenge/>.
- [6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A Simple Framework for Contrastive Learning of Visual Representations. *ArXiv*, February 2020.

- [7] Weifeng Chen, Guangtao Shang, Aihong Ji, Chengjun Zhou, Xiyang Wang, Chonghui Xu, Zhenxiong Li, and Kai Hu. An Overview on Visual SLAM : From Tradition to Semantic. *Remote Sensing*, 14(13) :3010, January 2022. ISSN 2072-4292.
- [8] COLMAP Structure-From-Motion and Multi-View Stereo. URL <https://demuc.de/colmap/>.
- [9] Andrew J. Davison, Ian D. Reid, Nicholas D. Molton, and Olivier Stasse. Mono-SLAM : Real-Time Single Camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6) :1052–1067, June 2007. ISSN 1939-3539.
- [10] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Deep Image Homography Estimation, June 2016. arXiv :1606.03798 [cs].
- [11] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Self-Improving Visual Odometry, December 2018. arXiv :1812.03245 [cs].
- [12] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint : Self-Supervised Interest Point Detection and Description. pages 224–236, 2018.
- [13] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-Net : A Trainable CNN for Joint Detection and Description of Local Features. In *CVPR 2019 - IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, United States, June 2019.
- [14] Patrick Ebel, Eduard Trulls, Kwang Moo Yi, Pascal Fua, and Anastasiia Mishchuk. Beyond Cartesian Representations for Local Descriptors. pages 253–262. IEEE Computer Society, October 2019. ISBN 978-1-72814-803-8.
- [15] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

- [16] Christopher G. Harris and M. J. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, 1988.
- [17] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. ISBN 978-0-521-54051-3.
- [18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.
- [19] Jared Heinly, Enrique Dunn, and Jan-Michael Frahm. Comparative evaluation of binary features. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, pages 759–773, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-33709-3.
- [20] Jianbo Shi and Tomasi. Good features to track. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition CVPR-94*, pages 593–600, Seattle, WA, USA, 1994. IEEE Comput. Soc. Press. ISBN 978-0-8186-5825-9.
- [21] Diederik P. Kingma and Jimmy Ba. Adam : A method for stochastic optimization, 2017.
- [22] Georg Klein and David Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 225–234, November 2007.
- [23] Tin Lai. A Review on Visual-SLAM : Advancements from Geometric Modelling to Learning-Based Semantic Scene Understanding Using Multi-Modal Sensor Fusion. *Sensors*, 22(19) :7265, September 2022. ISSN 1424-8220.
- [24] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2) :91–110, November 2004. ISSN 1573-1405.

- [25] Jiayi Ma, Xingyu Jiang, Aoxiang Fan, Junjun Jiang, and Junchi Yan. Image Matching from Handcrafted to Deep Features : A Survey. *International Journal of Computer Vision*, 129(1) :23–79, January 2021. ISSN 1573-1405.
- [26] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10) :1615–1630, October 2005. ISSN 1939-3539. Conference Name : IEEE Transactions on Pattern Analysis and Machine Intelligence.
- [27] Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardos. ORB-SLAM : a Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5) :1147–1163, October 2015. ISSN 1552-3098, 1941-0468. arXiv :1502.00956 [cs].
- [28] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. LF-Net : Learning Local Features from Images. May 2018.
- [29] Boris Polyak. Some methods of speeding up the convergence of iteration methods. *Ussr Computational Mathematics and Mathematical Physics*, 4 :1–17, 1964.
- [30] Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. R2D2 : Reliable and Repeatable Detector and Descriptor. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.
- [31] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional Neural Network Architecture for Geometric Matching. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 39–48, Honolulu, HI, July 2017. IEEE. ISBN 978-1-5386-0457-1.
- [32] Edward Rosten and Tom Drummond. Machine Learning for High-Speed Corner Detection. In Aleš Leonardis, Horst Bischof, and Axel Pinz, editors, *Computer Vision – ECCV 2006*, Lecture Notes in Computer Science, pages 430–443, Berlin, Heidelberg, 2006. Springer. ISBN 978-3-540-33833-8.

- [33] Xiaogang Ruan, Fei Wang, and Jing Huang. Relative Pose Estimation of Visual SLAM Based on Convolutional Neural Networks. In *2019 Chinese Control Conference (CCC)*, pages 8827–8832, July 2019. ISSN : 1934-1768.
- [34] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB : An efficient alternative to SIFT or SURF. In *2011 International Conference on Computer Vision*, pages 2564–2571, Barcelona, Spain, November 2011. IEEE. ISBN 978-1-4577-1102-2 978-1-4577-1101-5 978-1-4577-1100-8.
- [35] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabovich. SuperGlue : Learning Feature Matching With Graph Neural Networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4937–4946, Seattle, WA, USA, June 2020. IEEE. ISBN 978-1-72817-168-5.
- [36] Cordelia Schmid, Roger Mohr, and Christian Bauckhage. Evaluation of Interest Point Detectors. *International Journal of Computer Vision*, 37 :151–172, 2000.
- [37] Johannes L. Schonberger and Jan-Michael Frahm. Structure-From-Motion Revisited. pages 4104–4113, 2016.
- [38] Eli Shechtman and Michal Irani. Matching Local Self-Similarities across Images and Videos. Minneapolis, MN, USA, 2007.
- [39] Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, P. Fua, and Francesc Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 118–126, 2015.
- [40] Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21 :343–348, 1967.

- [41] Hauke Strasdat, J. M. M. Montiel, and Andrew J. Davison. Real-time monocular SLAM : Why filter? In *2010 IEEE International Conference on Robotics and Automation*, pages 2657–2664, May 2010. ISSN : 1050-4729.
- [42] Richard Szeliski. Computer vision - algorithms and applications. In *Texts in Computer Science*, 2011.
- [43] Zachary Teed and Jia Deng. DROID-SLAM : Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. August 2021.
- [44] Lorenzo Torresani, Vladimir Kolmogorov, and Carsten Rother. Feature Correspondence Via Graph Matching : Models and Global Optimization. In David Forsyth, Philip Torr, and Andrew Zisserman, editors, *Computer Vision – ECCV 2008*, Lecture Notes in Computer Science, pages 596–609, Berlin, Heidelberg, 2008. Springer. ISBN 978-3-540-88688-4.
- [45] Tomasz Trzcinski, Jacek Komorowski, Lukasz Dabala, Konrad Czarnota, Grzegorz Kurzejamski, and Simon Lynen. SConE : Siamese Constellation Embedding Descriptor for Image Matching. In Laura Leal-Taixé and Stefan Roth, editors, *Computer Vision – ECCV 2018 Workshops*, volume 11129 of *Lecture Notes in Computer Science*, pages 401–413, Cham, 2019. Springer International Publishing. ISBN 978-3-030-11008-6 978-3-030-11009-3.
- [46] Tinne Tuytelaars and Luc J Van Gool. Wide Baseline Stereo Matching based on Local, Affinely Invariant Regions. In *Proceedings of the British Machine Vision Conference 2000*, pages 38.1–38.14, Bristol, 2000. British Machine Vision Association. ISBN 978-1-901725-13-1.
- [47] Wenshan Wang, DeLong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. TartanAir : A Dataset to Push the Limits of Visual SLAM. In *2020 IEEE/RSJ International Conference on Intelligent*

*Robots and Systems (IROS)*, pages 4909–4916, Las Vegas, NV, USA, October 2020. IEEE. ISBN 978-1-72816-212-6.

- [48] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. LIFT : Learned Invariant Feature Transform. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, Lecture Notes in Computer Science, pages 467–483, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46466-4.

## Annexe A

---

Les figures et tableaux présentés en annexe correspondent aux résultats des tests d'évaluation KP-RAND, KP-ORB et KP-ORB-W effectués pour d'autres paires d'images non présentées dans le chapitre 6.

### *KP-RAND 10-260*

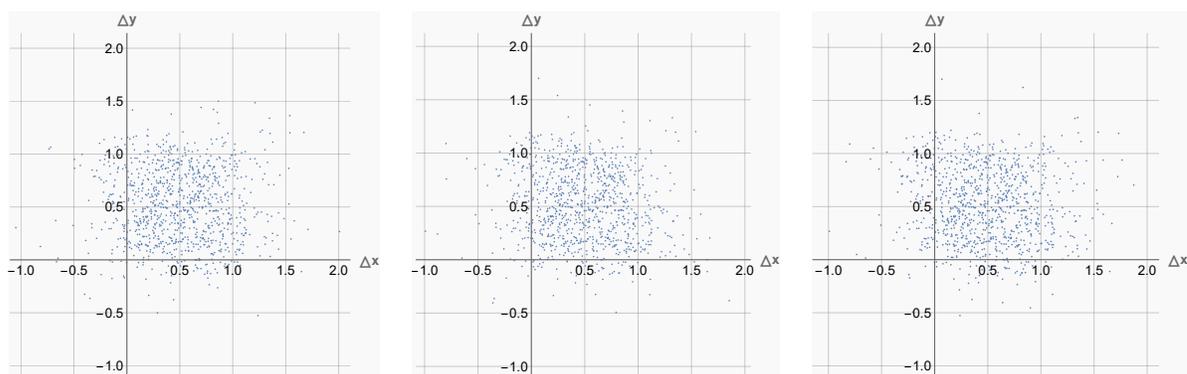


FIGURE A.1 – Visualisation des différences entre les points mis en correspondance avec les descripteurs fournis par les réseaux non-augmentés, et leur référence. Gauche : R1. Milieu : R2. Droite : R3.

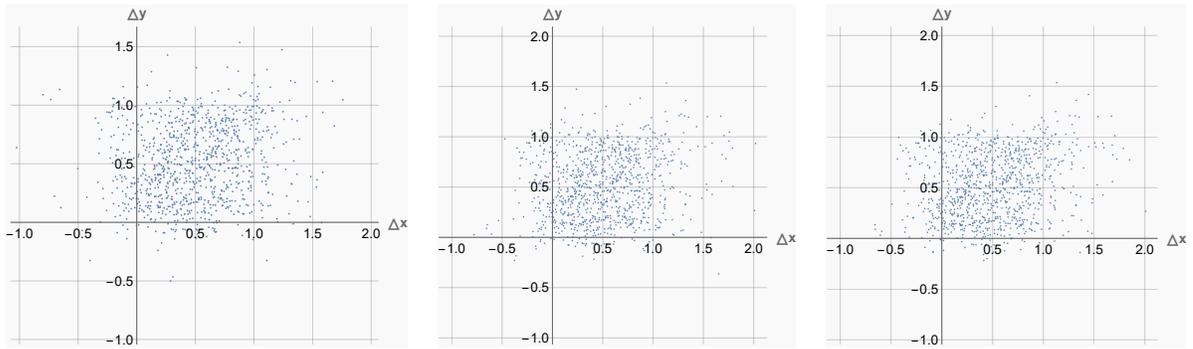


FIGURE A.2 – Visualisation des différences entre les points mis en correspondance avec les descripteurs fournis par les réseaux augmentés, et leur référence. Gauche : R4. Milieu : R5. Droite : R6.

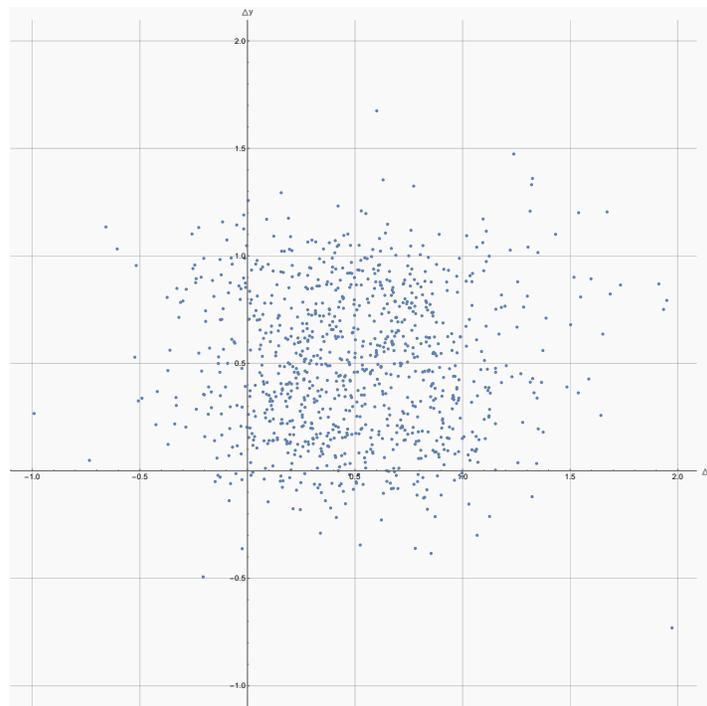


FIGURE A.3 – Visualisation des différences entre les points mis en correspondance avec les descripteurs fournis par le réseau non-entraîné R0, et leur référence.

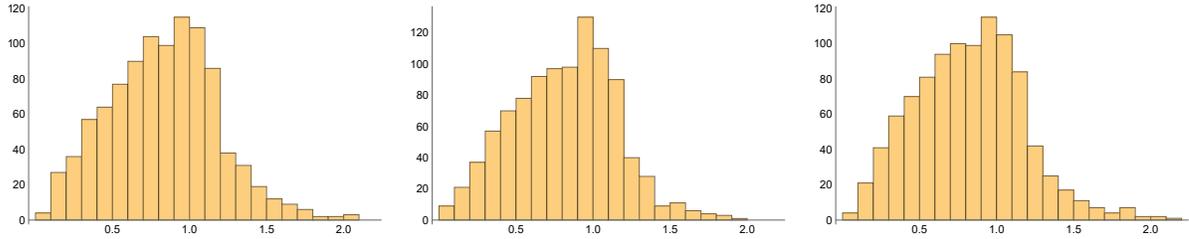


FIGURE A.4 – Histogramme de la distance euclidienne entre les points mis en correspondance avec les descripteurs fournis par les réseaux non-augmentés, et leur référence.  
Gauche : R1. Milieu : R2. Droite : R3.

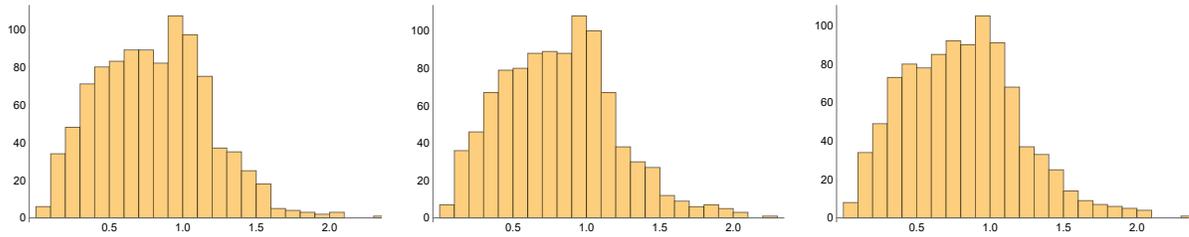


FIGURE A.5 – Histogramme de la distance euclidienne entre les points mis en correspondance avec les descripteurs fournis par les réseaux non-augmentés, et leur référence.  
Gauche : R4. Milieu : R5. Droite : R6.

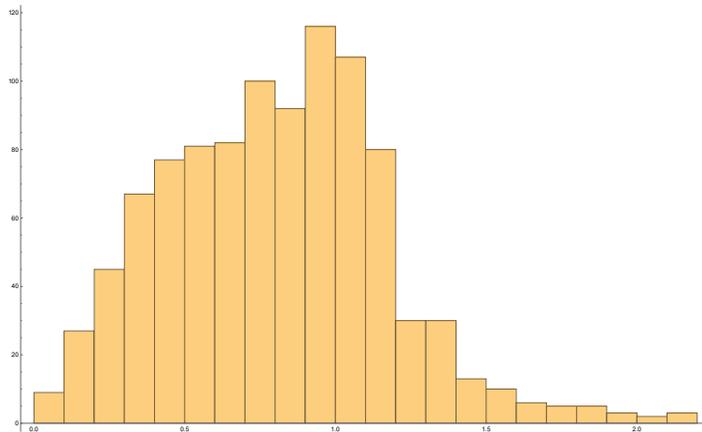


FIGURE A.6 – Histogramme de la distance euclidienne entre les points mis en correspondance avec les descripteurs fournis par le réseau non-entraîné R0, et leur référence.

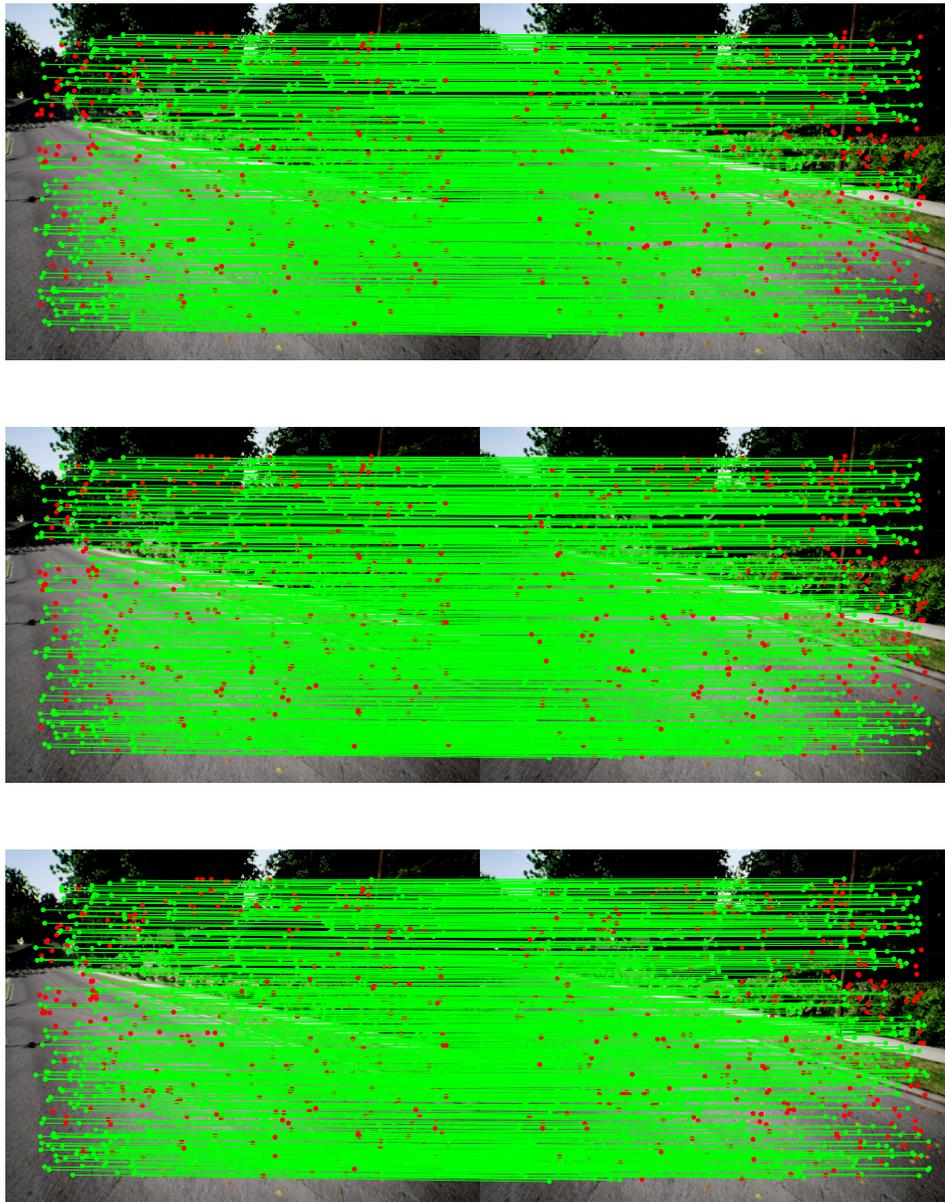


FIGURE A.7 – Résultat qualitatif de la mise en correspondance entre les points saillants choisis aléatoirement de deux images consécutives réalisée avec les descripteurs fournis par les réseaux non-augmentés. Les lignes vertes représentent des mises en correspondance correctes. Les points rouges représentent des points qui n'ont pas été correctement mis en correspondance. Haut : R1. Milieu : R2. Bas : R3.

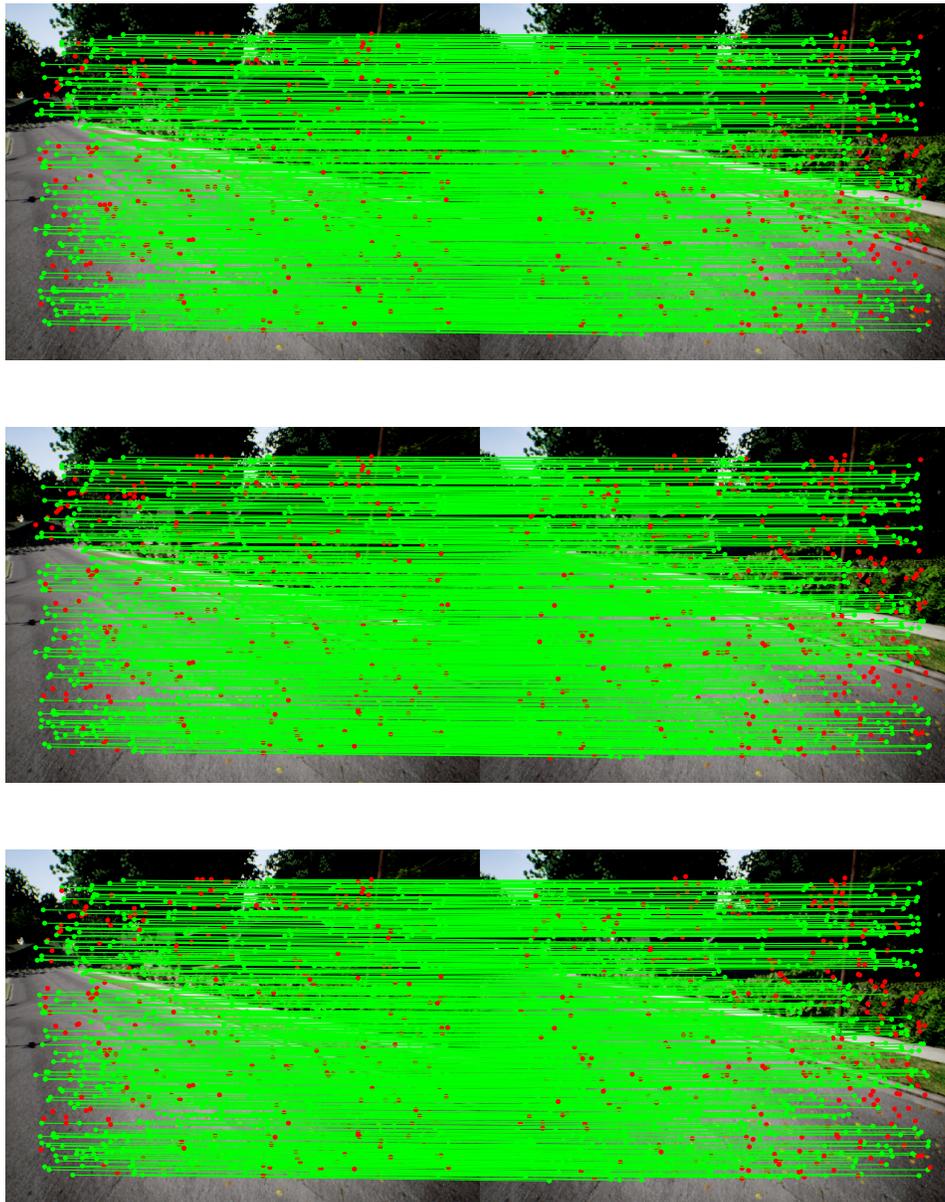


FIGURE A.8 – Résultat qualitatif de la mise en correspondance entre les points saillants choisis aléatoirement de deux images consécutives réalisée avec les descripteurs fournis par les réseaux augmentés. Les lignes vertes représentent des mises en correspondance correctes. Les points rouges représentent des points qui n’ont pas été correctement mis en correspondance. Haut : R4. Milieu : R5. Bas : R6.

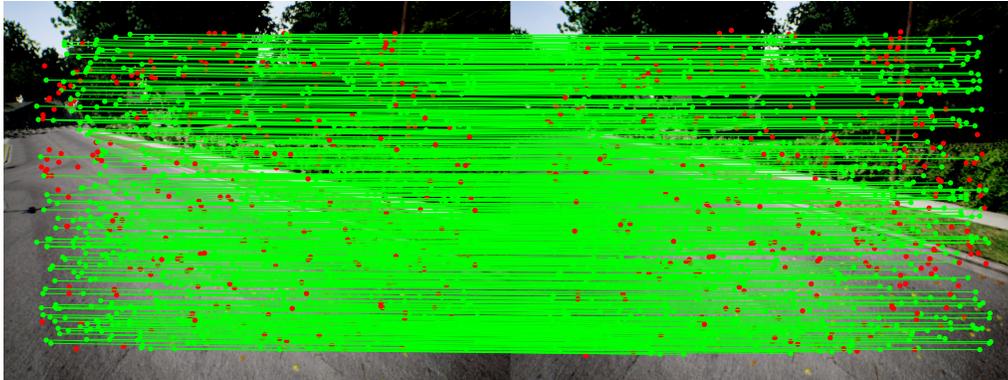


FIGURE A.9 – Résultat qualitatif de la mise en correspondance entre les points saillants choisis aléatoirement de deux images consécutives réalisée avec les descripteurs fournis par le réseau non-entraîné R0. Les lignes vertes représentent des mises en correspondance correctes. Les points rouges représentent des points qui n'ont pas été correctement mis en correspondance.

TABLEAU A.1 – Résultats quantitatifs regroupant les métriques des différentes méthodes utilisées pour réaliser la mise en correspondance entre les points saillants choisis aléatoirement de deux images consécutives.

Méthode	Erreur moyenne	Erreur médiane	Écart – type erreur
R0	0.91	0.81	1.31
R1	0.92	0.84	1.22
R2	0.9	0.84	1.19
R3	0.92	0.83	1.35
R4	0.87	0.8	1.07
R5	0.92	0.81	1.54
R6	0.91	0.8	1.4

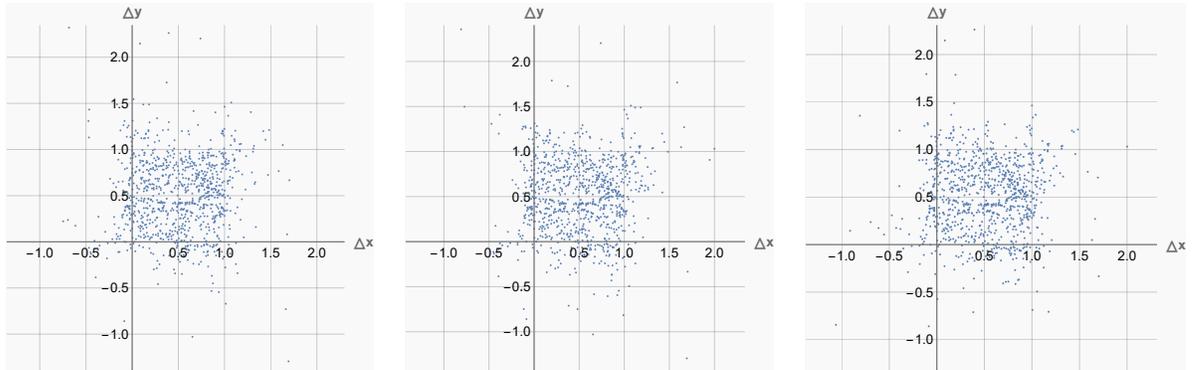


FIGURE A.10 – Visualisation des différences entre les points mis en correspondance avec les descripteurs fournis par les réseaux non-augmentés, et leur référence. Gauche : R1. Milieu : R2. Droite : R3.

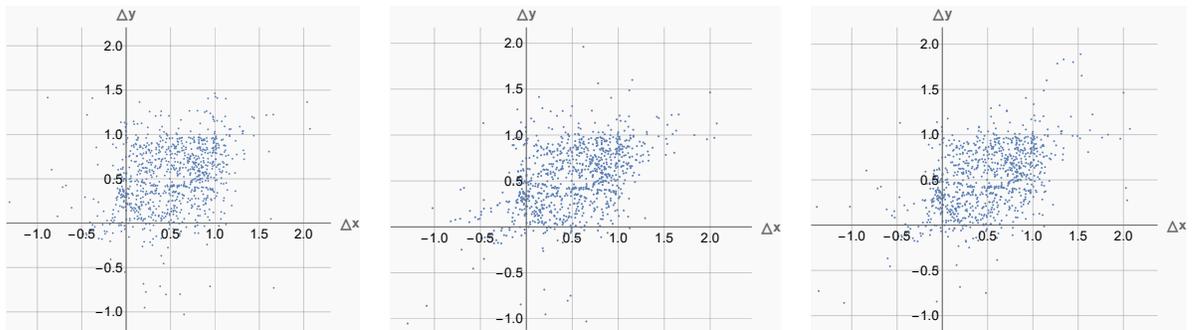


FIGURE A.11 – Visualisation des différences entre les points mis en correspondance avec les descripteurs fournis par les réseaux augmentés, et leur référence. Gauche : R4. Milieu : R5. Droite : R6.

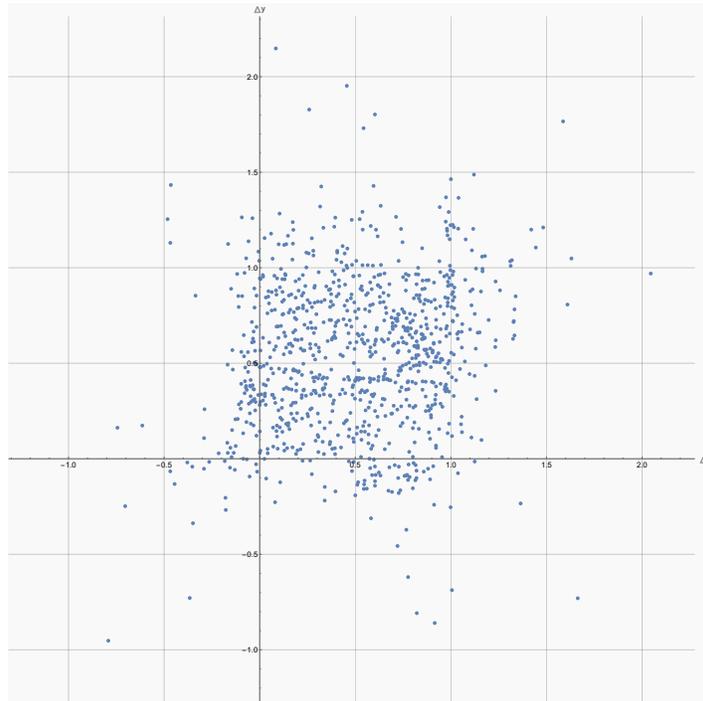


FIGURE A.12 – Visualisation des différences entre les points mis en correspondance avec les descripteurs fournis par le réseau non-entraîné R0, et leur référence.

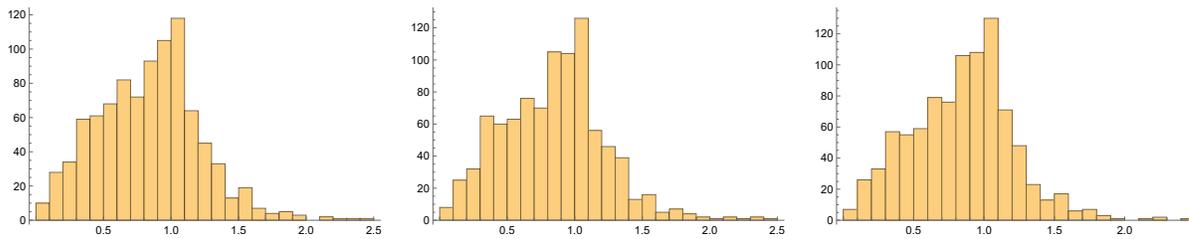


FIGURE A.13 – Histogramme de la distance euclidienne entre les points mis en correspondance avec les descripteurs fournis par les réseaux non-augmentés, et leur référence. Gauche : R1. Milieu : R2. Droite : R3.

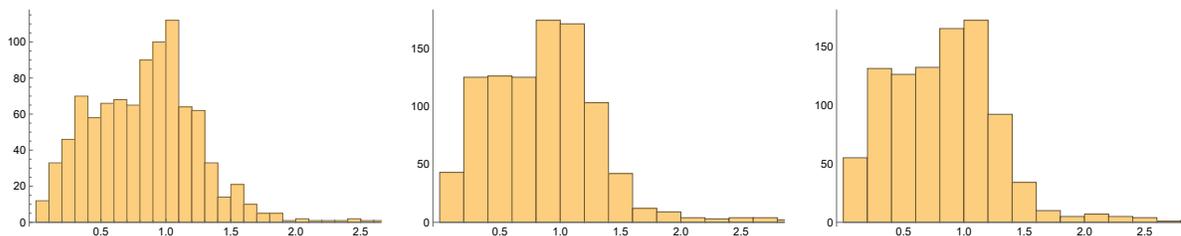


FIGURE A.14 – Histogramme de la distance euclidienne entre les points mis en correspondance avec les descripteurs fournis par les réseaux non-augmentés, et leur référence. Gauche : R4. Milieu : R5. Droite : R6.

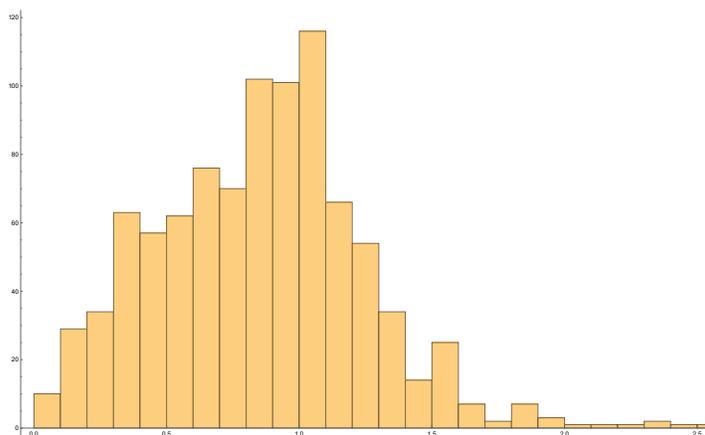


FIGURE A.15 – Histogramme de la distance euclidienne entre les points mis en correspondance avec les descripteurs fournis par le réseau non-entraîné R0, et leur référence.

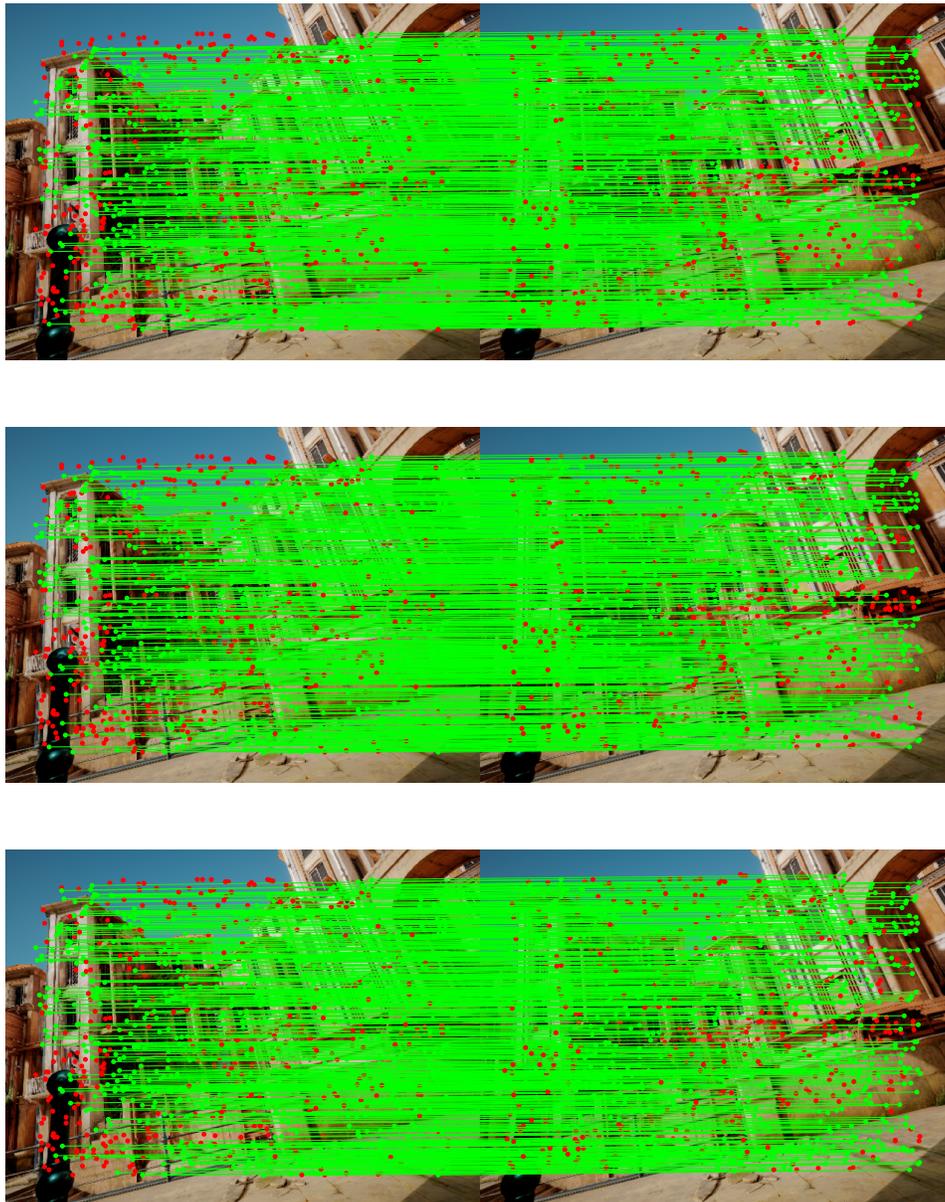


FIGURE A.16 – Résultat qualitatif de la mise en correspondance entre les points saillants choisis aléatoirement de deux images consécutives réalisée avec les descripteurs fournis par les réseaux non-augmentés. Les lignes vertes représentent des mises en correspondance correctes. Les points rouges représentent des points qui n’ont pas été correctement mis en correspondance. Haut : R1. Milieu : R2. Bas : R3.

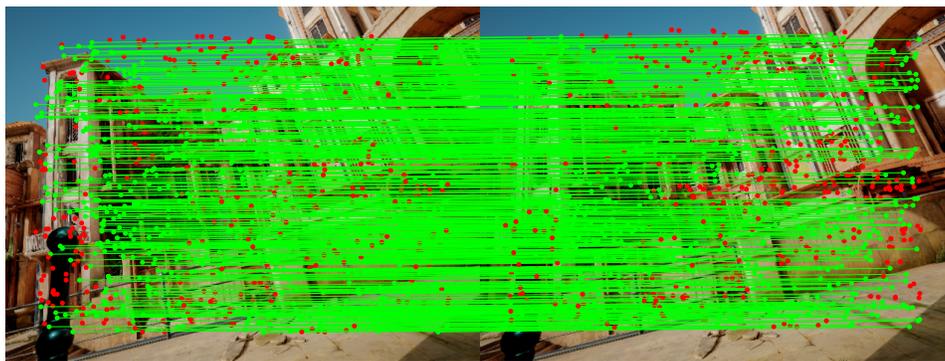
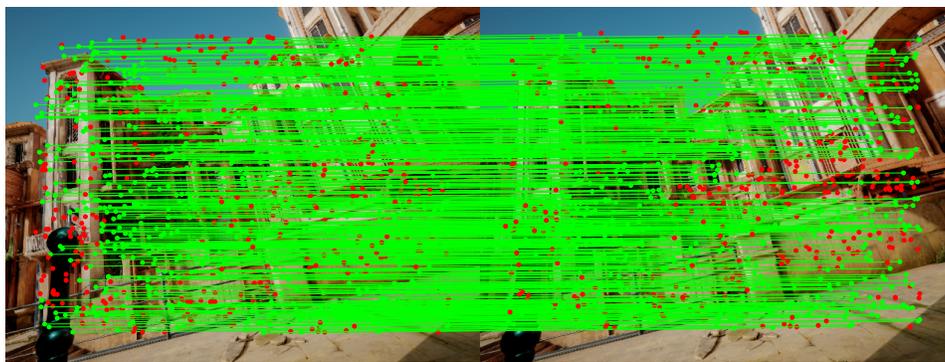
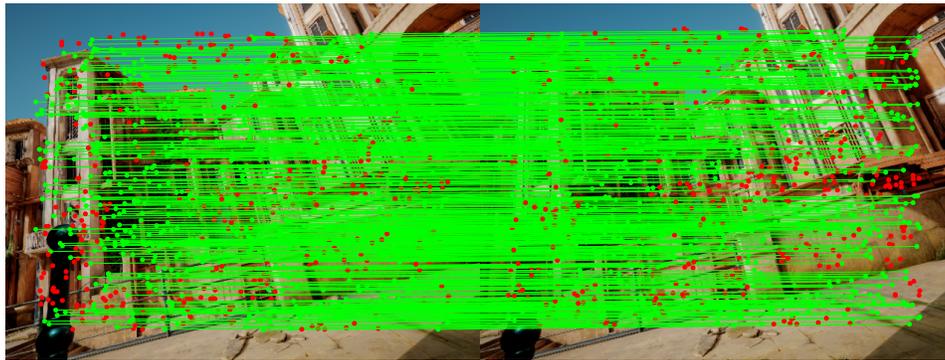


FIGURE A.17 – Résultat qualitatif de la mise en correspondance entre les points saillants choisis aléatoirement de deux images consécutives réalisée avec les descripteurs fournis par les réseaux augmentés. Les lignes vertes représentent des mises en correspondance correctes. Les points rouges représentent des points qui n'ont pas été correctement mis en correspondance. Haut : R4. Milieu : R5. Bas : R6.

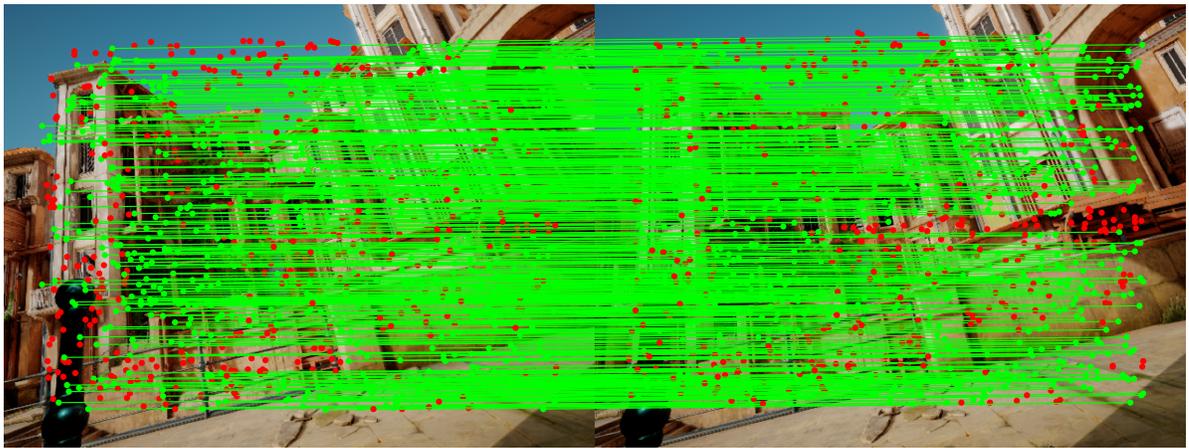


FIGURE A.18 – Résultat qualitatif de la mise en correspondance entre les points saillants choisis aléatoirement de deux images consécutives réalisée avec les descripteurs fournis par le réseau non-entraîné R0. Les lignes vertes représentent des mises en correspondance correctes. Les points rouges représentent des points qui n'ont pas été correctement mis en correspondance.

TABLEAU A.2 – Résultats quantitatifs regroupant les métriques des différentes méthodes utilisées pour réaliser la mise en correspondance entre les points saillants choisis aléatoirement de deux images consécutives.

Méthode	Erreur moyenne	Erreur médiane	Écart – type erreur
R0	1.31	0.9	2.19
R1	1.27	0.89	1.77
R2	1.35	0.89	2.26
R3	1.28	0.9	1.89
R4	1.23	0.89	1.9
R5	1.35	0.9	2.58
R6	1.28	0.87	2.34

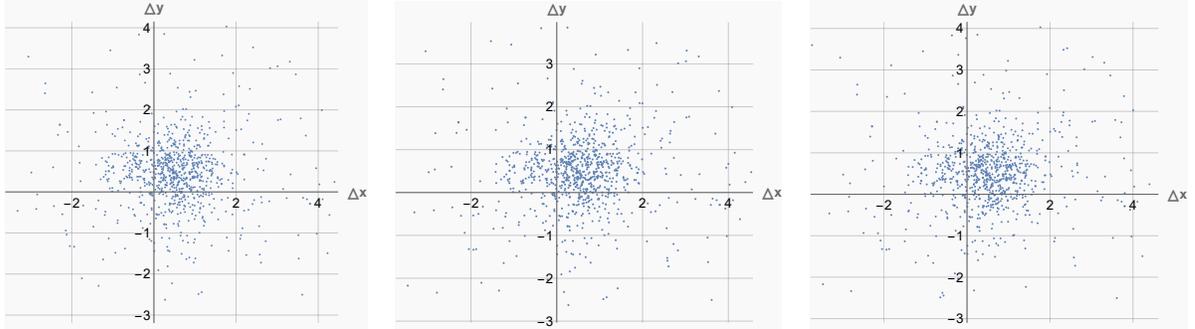


FIGURE A.19 – Visualisation des différences entre les points saillants mis en correspondance avec les descripteurs fournis par les réseaux non-augmentés, et leur référence.  
Gauche : R1. Milieu : R2. Droite : R3.

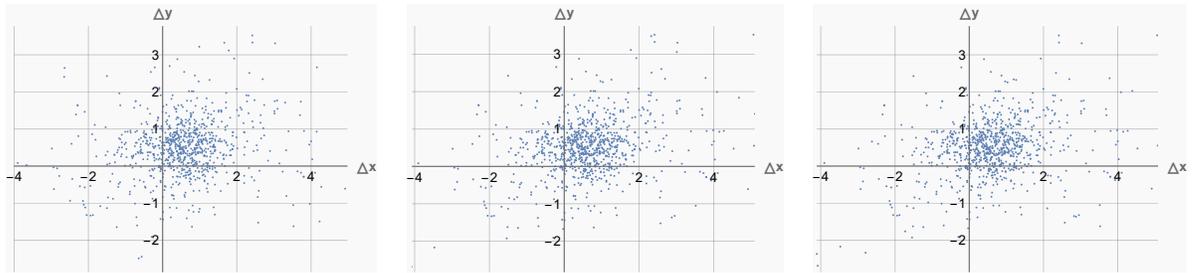


FIGURE A.20 – Visualisation des différences entre les points saillants mis en correspondance avec les descripteurs fournis par les réseaux augmentés, et leur référence.  
Gauche : R4. Milieu : R5. Droite : R6.

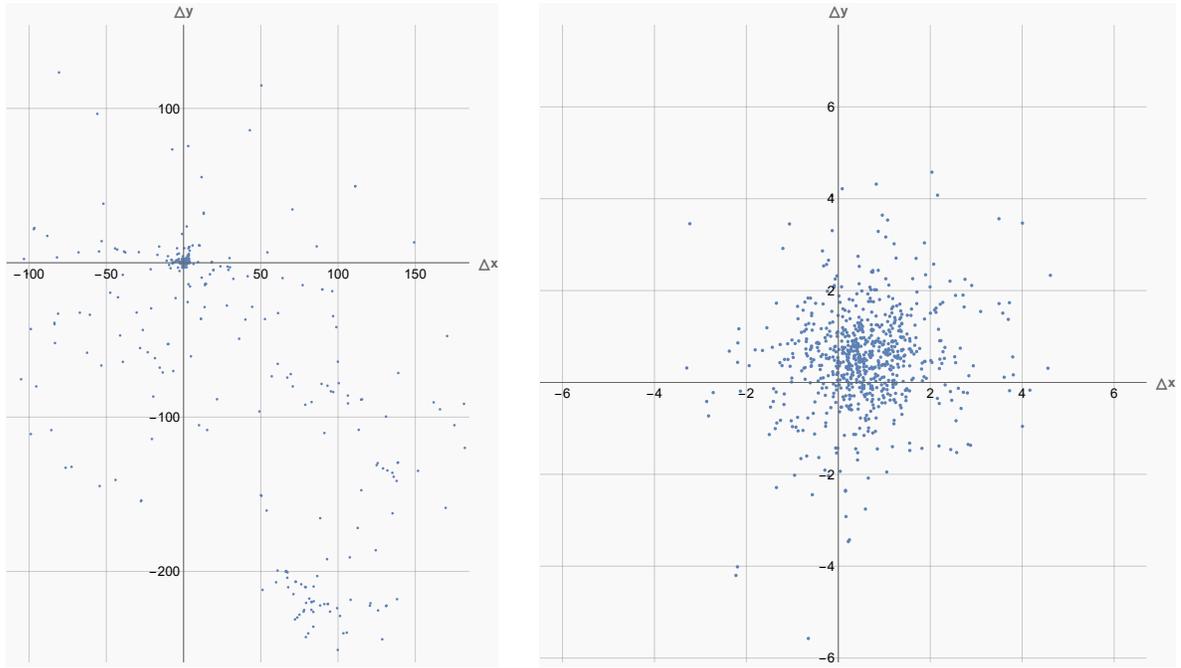


FIGURE A.21 – Visualisation des différences entre les points saillants mis en correspondance avec les descripteurs fournis par le réseau non-entraîné et les descripteurs ORB, et leur référence. Gauche : NT. Droite : ORB.

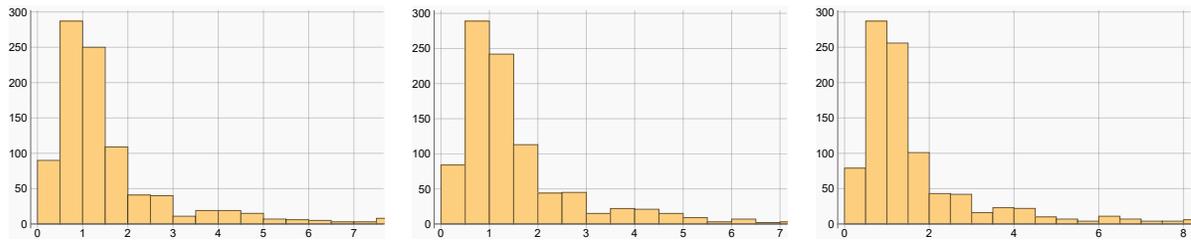


FIGURE A.22 – Histogramme de la distance euclidienne entre les points saillants mis en correspondance avec les descripteurs fournis par les réseaux non-augmentés, et leur référence. Gauche : R1. Milieu : R2. Droite : R3.

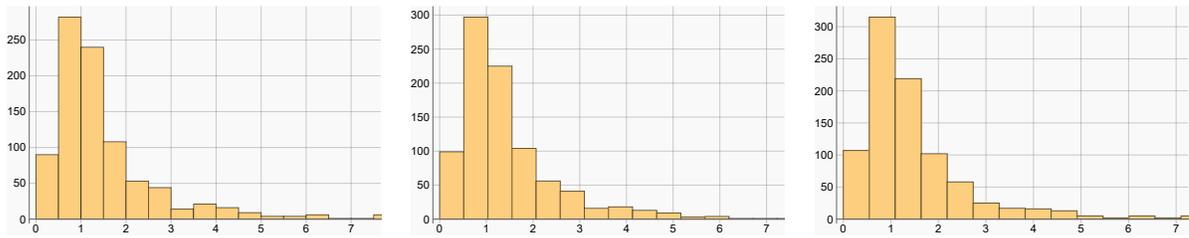


FIGURE A.23 – Histogramme de la distance euclidienne entre les points saillants mis en correspondance avec les descripteurs fournis par les réseaux augmentés, et leur référence. Gauche : R4. Milieu : R5. Droite : R6.

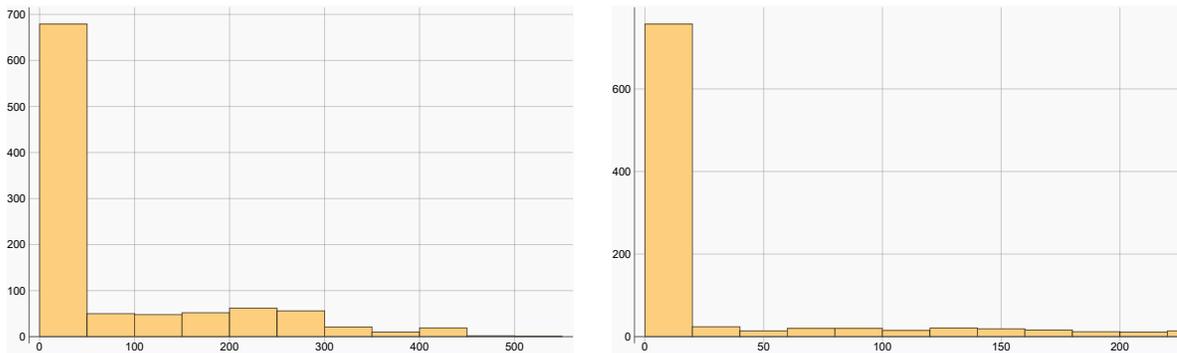


FIGURE A.24 – Histogramme de la distance euclidienne entre les points saillants mis en correspondance avec les descripteurs fournis par le réseau non-entraîné et les descripteurs ORB, et leur référence. Gauche : NT. Droite : ORB.

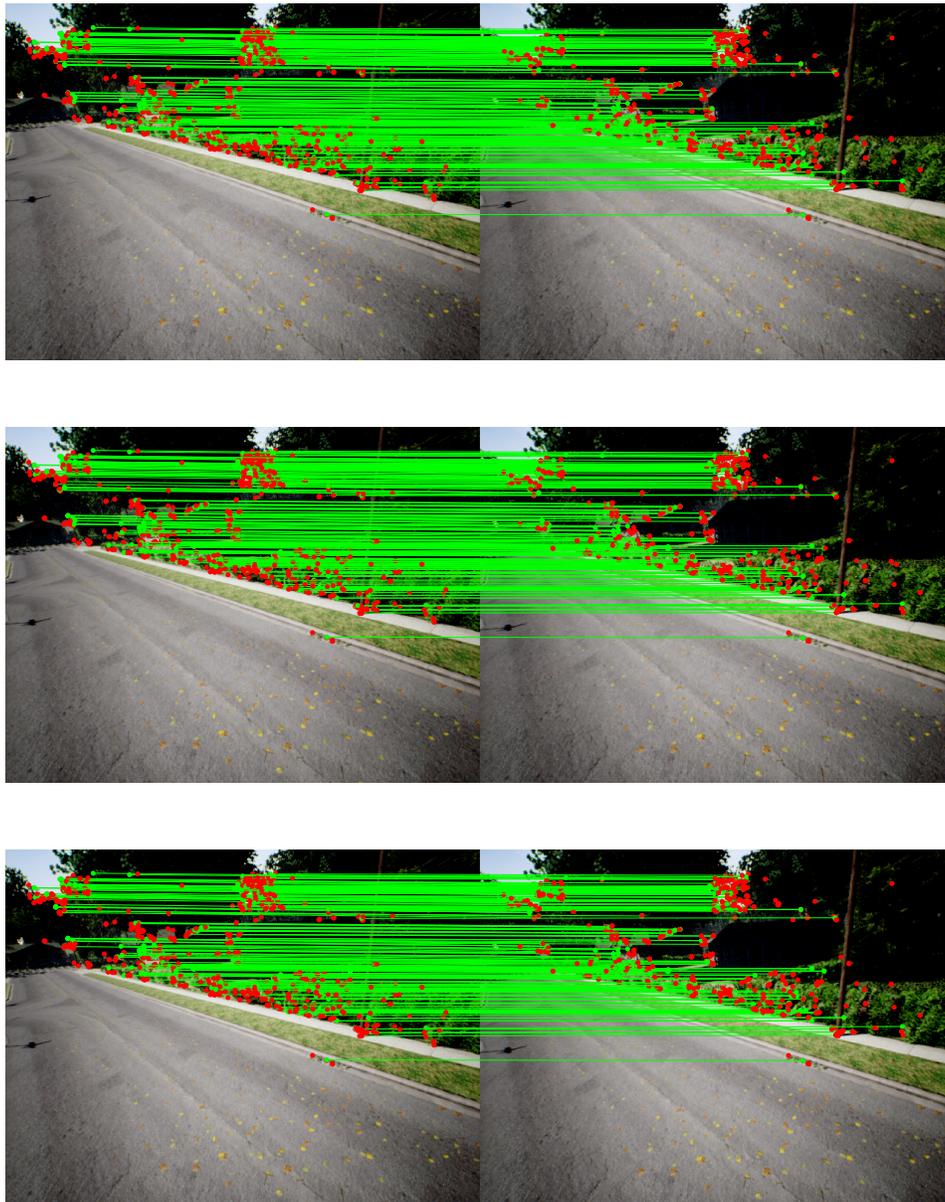


FIGURE A.25 – Résultat qualitatif de la mise en correspondance entre les points saillants ORB de deux images consécutives réalisée avec les descripteurs fournis par les réseaux non-augmentés. Les lignes vertes représentent des mises en correspondance correctes. Les points rouges représentent des points qui n'ont pas été correctement mis en correspondance. Haut : R1. Milieu : R2. Bas : R3.

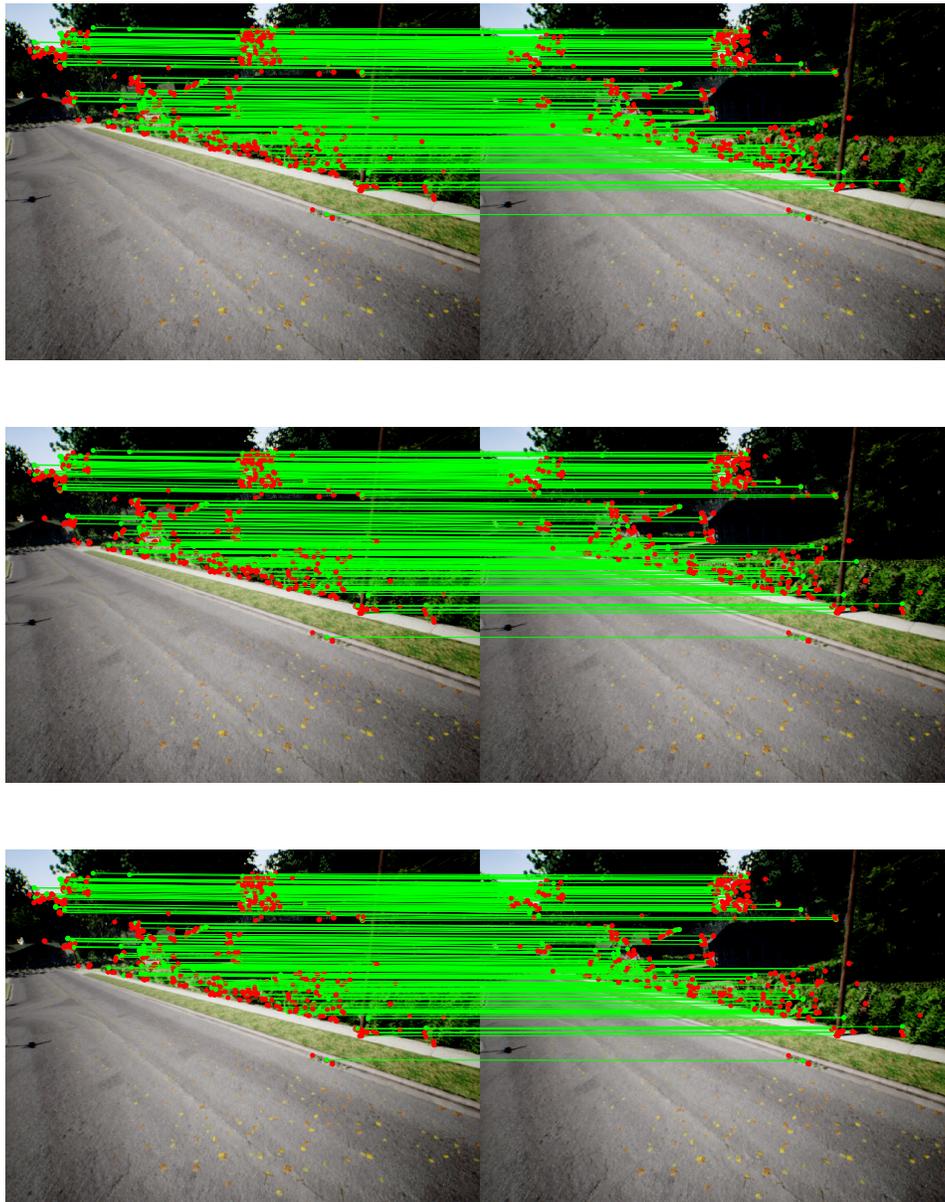


FIGURE A.26 – Résultat qualitatif de la mise en correspondance entre les points saillants ORB de deux images consécutives réalisée avec les descripteurs fournis par les réseaux augmentés. Les lignes vertes représentent des mises en correspondance correctes. Les points rouges représentent des points qui n'ont pas été correctement mis en correspondance. Haut : R4. Milieu : R5. Bas : R6.

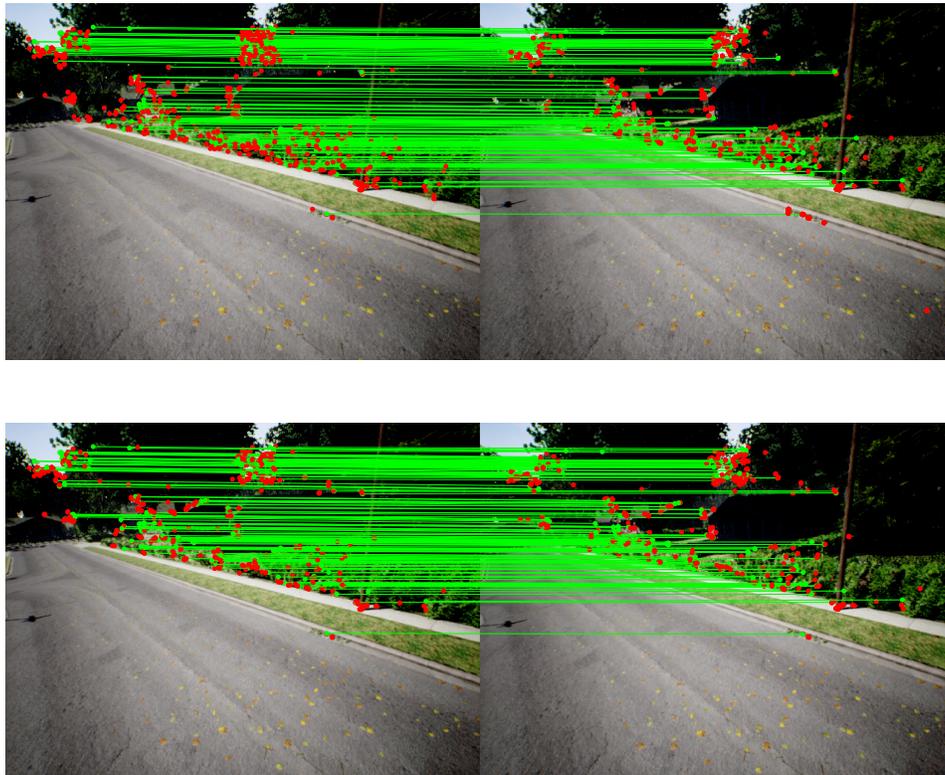


FIGURE A.27 – Résultat qualitatif de la mise en correspondance entre les points saillants ORB de deux images consécutives réalisée avec les descripteurs fournis par le réseau non-entraîné R0, et les descripteurs ORB. Les lignes vertes représentent des mises en correspondance correctes. Les points rouges représentent des points qui n’ont pas été correctement mis en correspondance. Haut : R0. Bas : ORB.

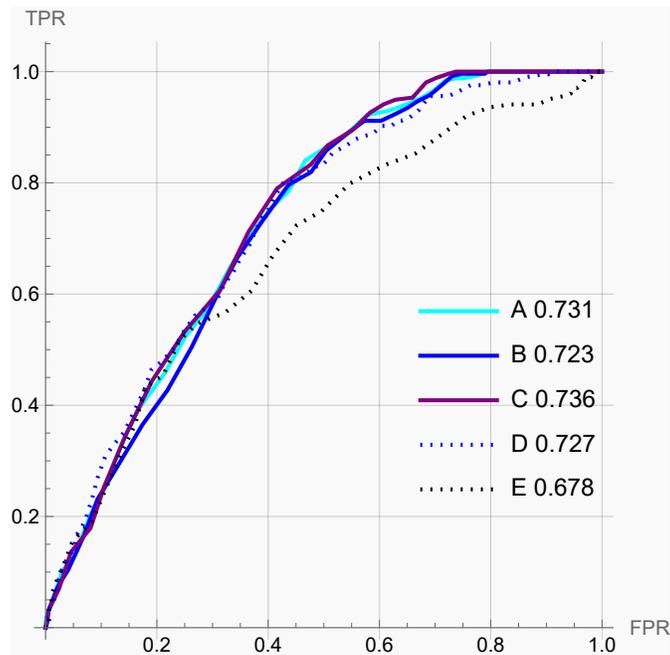
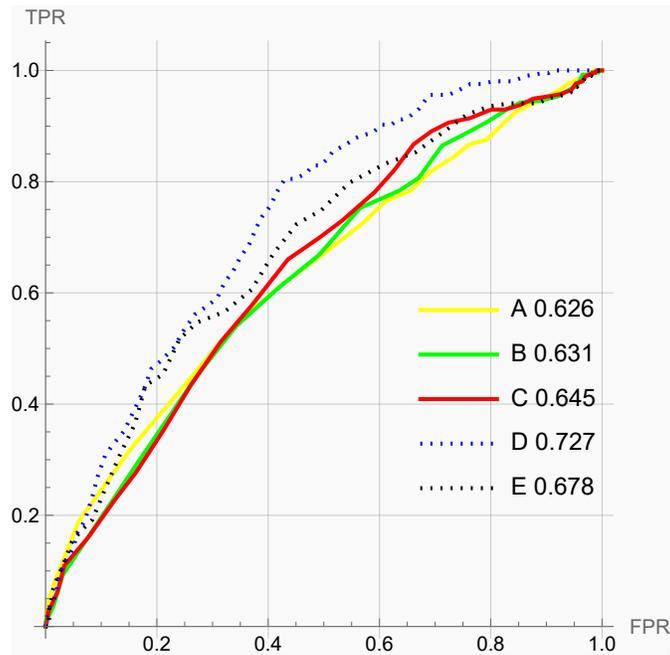


FIGURE A.28 – Courbes ROC des réseaux augmentés et non-augmentés. Haut : Réseaux non-augmentés avec A : R1, B : R2, C : R3, D : ORB, E : R0. Bas : Réseaux augmentés avec A : R4, B : R5, C : R6, D : ORB, E : R0.

TABLEAU A.3 – Résultats quantitatifs regroupant les métriques des différentes méthodes utilisées pour réaliser la mise en correspondance entre les points saillants ORB de deux images consécutives.

Méthode	VP	FP	FN	VN	Erreur moyenne	Erreur médiane	Écart-type erreur	ACC	AUC
R0	191	705	12	92	69.75	1.58	112.29	0.28	0.68
R1	264	735	0	1	7.8	1.21	39.37	0.26	0.63
R2	259	741	0	0	6.22	1.24	27.13	0.26	0.63
R3	256	741	0	3	7.18	1.25	27.5	0.26	0.65
R4	256	643	0	101	12.94	1.25	48.55	0.36	0.73
R5	260	643	0	97	19.65	1.25	66.19	0.36	0.72
R6	257	654	0	89	19.81	1.26	67.89	0.35	0.74
ORB	185	494	20	301	41.38	1.56	86.96	0.49	0.73

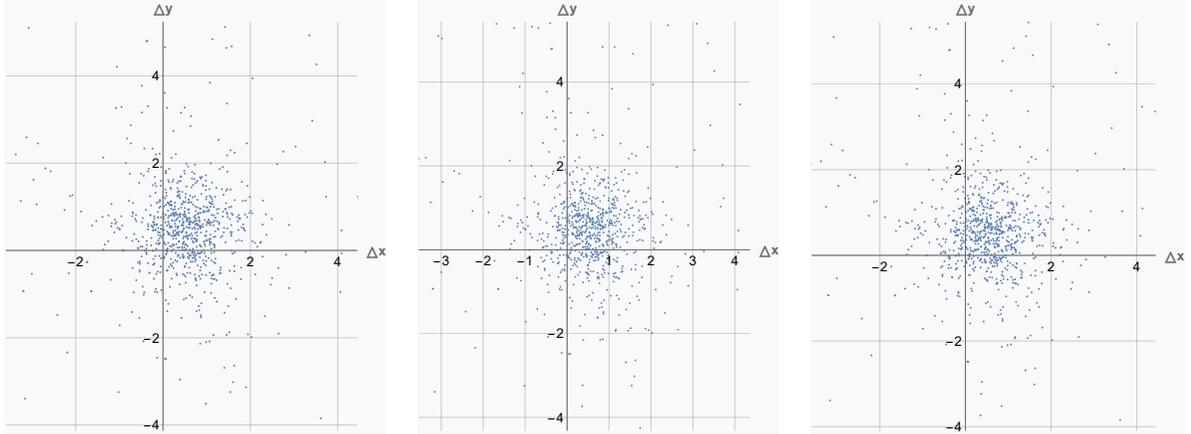


FIGURE A.29 – Visualisation des différences entre les points saillants mis en correspondance avec les descripteurs fournis par les réseaux non-augmentés, et leur référence.  
Gauche : R1. Milieu : R2. Droite : R3.

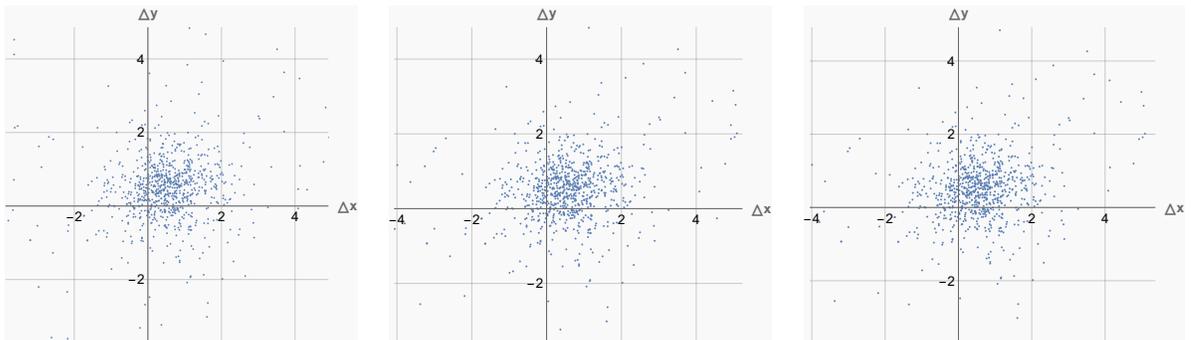


FIGURE A.30 – Visualisation des différences entre les points saillants mis en correspondance avec les descripteurs fournis par les réseaux augmentés, et leur référence.  
Gauche : R4. Milieu : R5. Droite : R6.

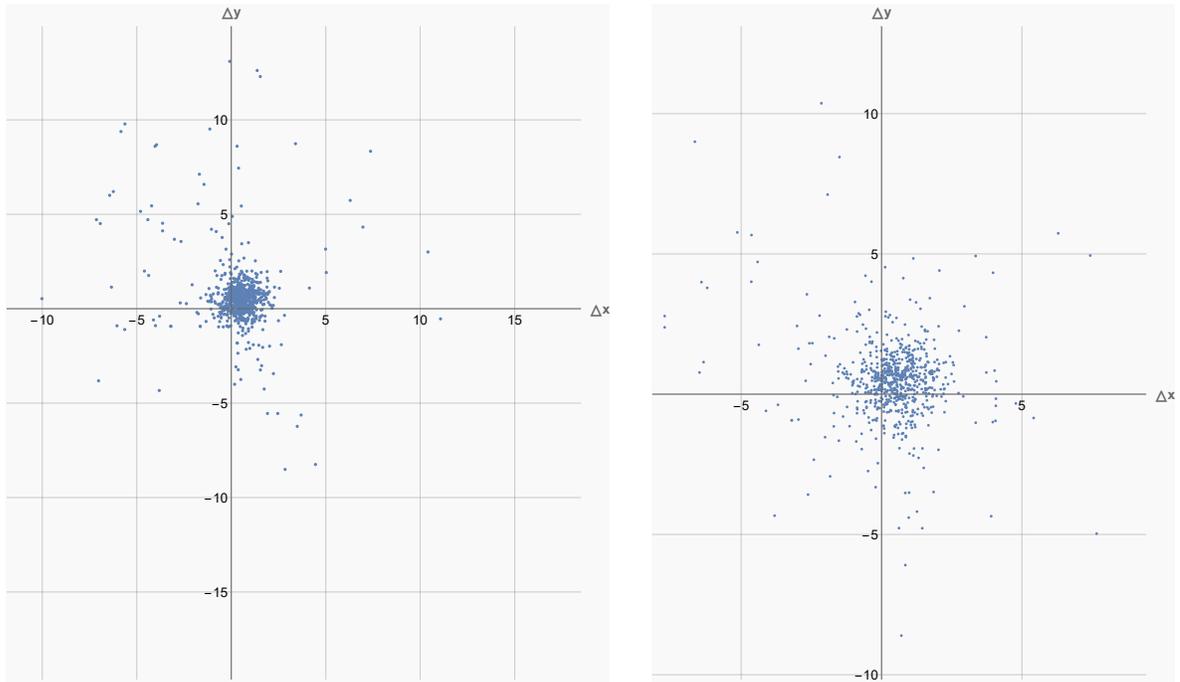


FIGURE A.31 – Visualisation des différences entre les points saillants mis en correspondance avec les descripteurs fournis par le réseau non-entraîné et les descripteurs ORB, et leur référence. Gauche : NT. Droite : ORB.

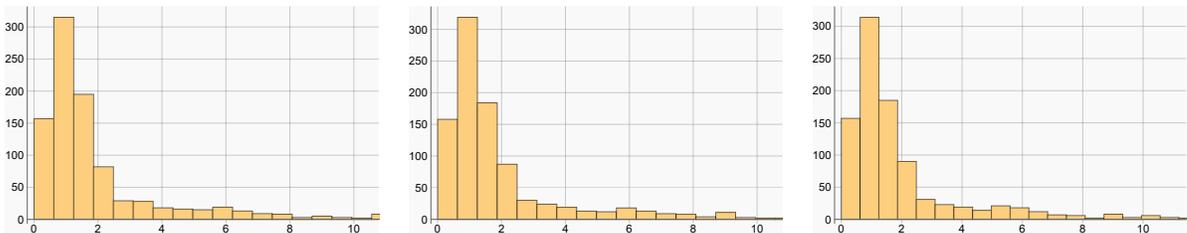


FIGURE A.32 – Histogramme de la distance euclidienne entre les points saillants mis en correspondance avec les descripteurs fournis par les réseaux non-augmentés, et leur référence. Gauche : R1. Milieu : R2. Droite : R3.

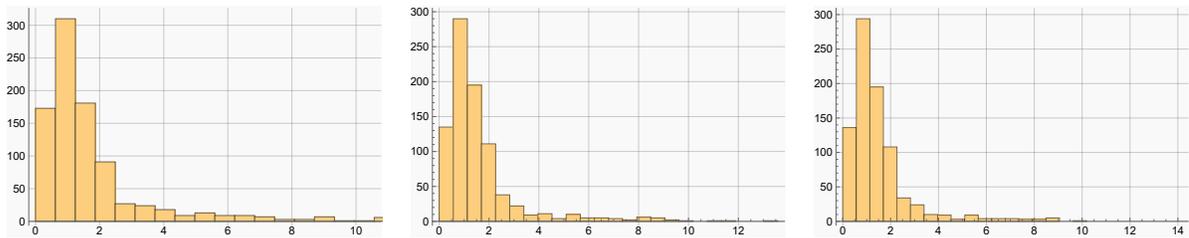


FIGURE A.33 – Histogramme de la distance euclidienne entre les points saillants mis en correspondance avec les descripteurs fournis par les réseaux augmentés, et leur référence. Gauche : R4. Milieu : R5. Droite : R6.

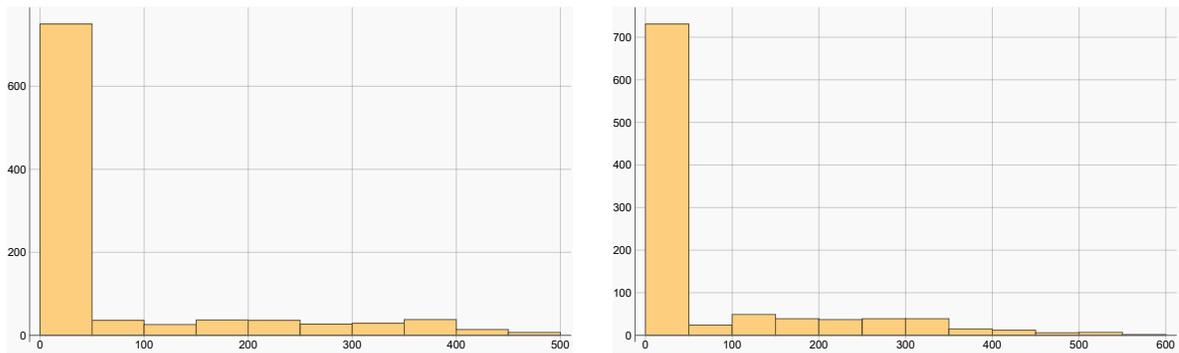


FIGURE A.34 – Histogramme de la distance euclidienne entre les points saillants mis en correspondance avec les descripteurs fournis par le réseau non-entraîné et les descripteurs ORB, et leur référence. Gauche : NT. Droite : ORB.

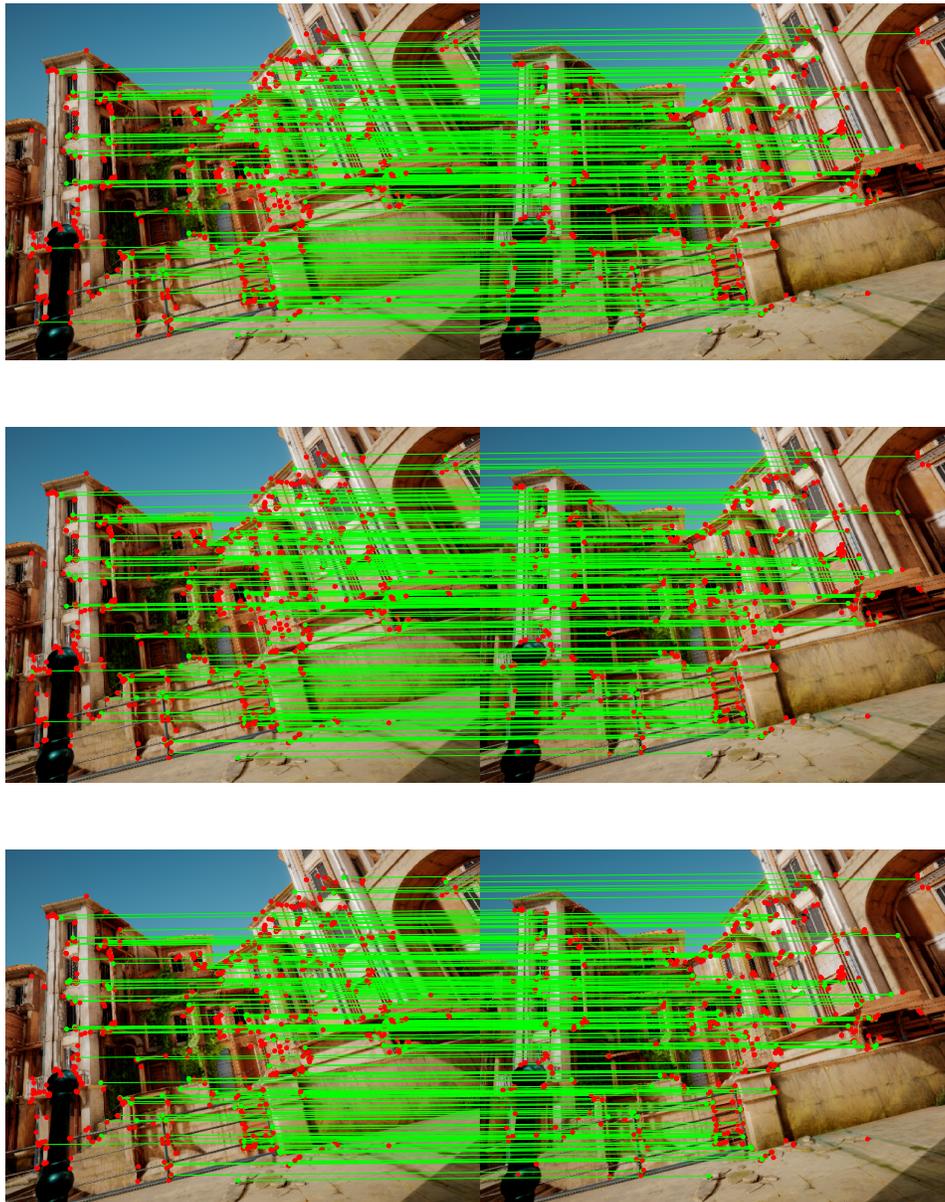


FIGURE A.35 – Résultat qualitatif de la mise en correspondance entre les points saillants ORB de deux images consécutives réalisée avec les descripteurs fournis par les réseaux non-augmentés. Les lignes vertes représentent des mises en correspondance correctes. Les points rouges représentent des points qui n'ont pas été correctement mis en correspondance. Haut : R1. Milieu : R2. Bas : R3.

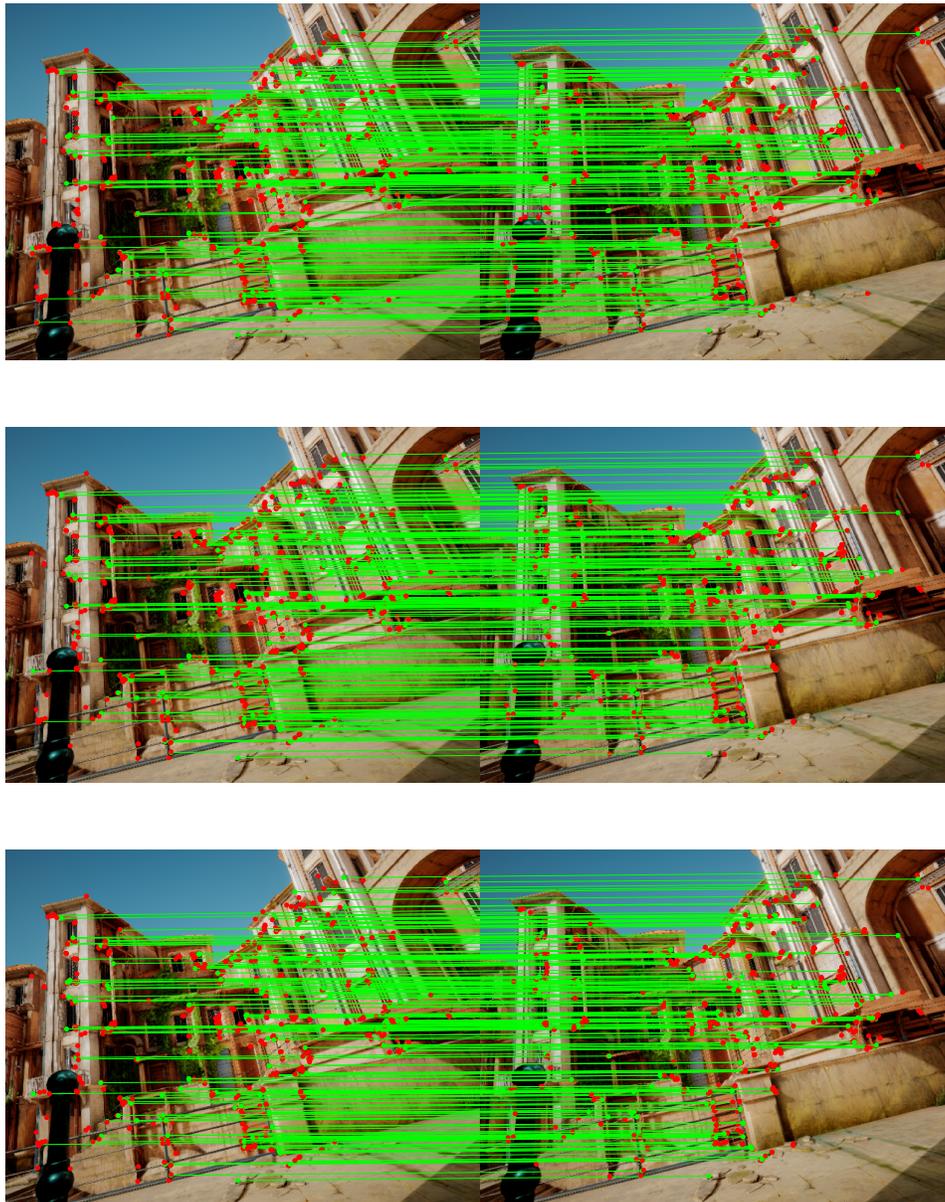


FIGURE A.36 – Résultat qualitatif de la mise en correspondance entre les points saillants ORB de deux images consécutives réalisée avec les descripteurs fournis par les réseaux augmentés. Les lignes vertes représentent des mises en correspondance correctes. Les points rouges représentent des points qui n'ont pas été correctement mis en correspondance. Haut : R4. Milieu : R5. Bas : R6.

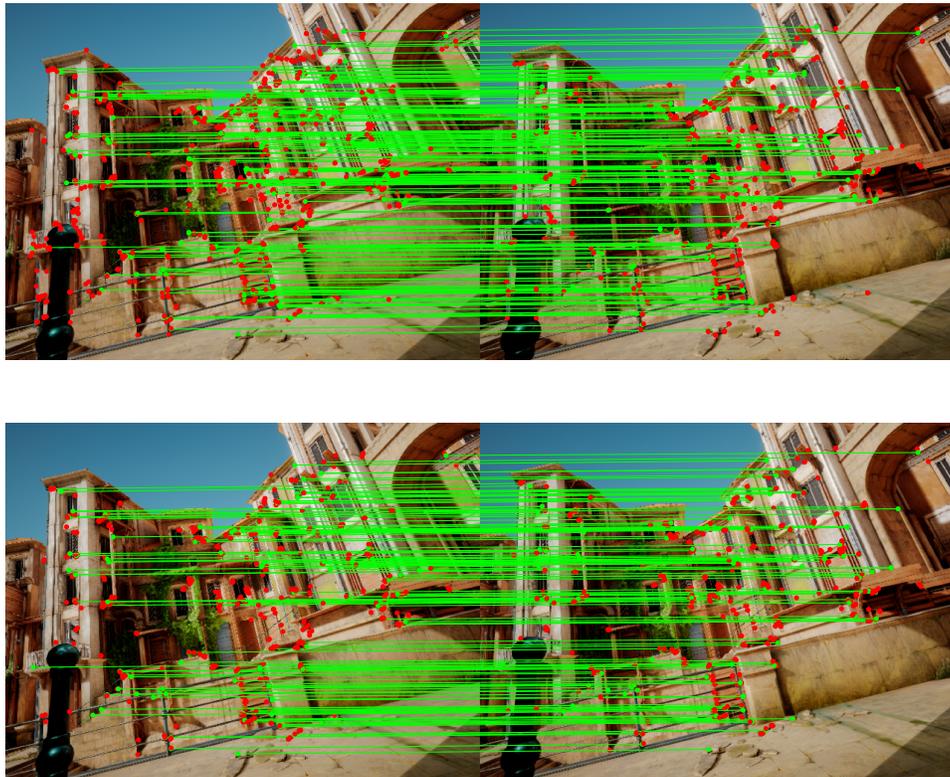


FIGURE A.37 – Résultat qualitatif de la mise en correspondance entre les points saillants ORB de deux images consécutives réalisée avec les descripteurs fournis par le réseau non-entraîné R0, et les descripteurs ORB. Les lignes vertes représentent des mises en correspondance correctes. Les points rouges représentent des points qui n’ont pas été correctement mis en correspondance. Haut : R0. Bas : ORB.

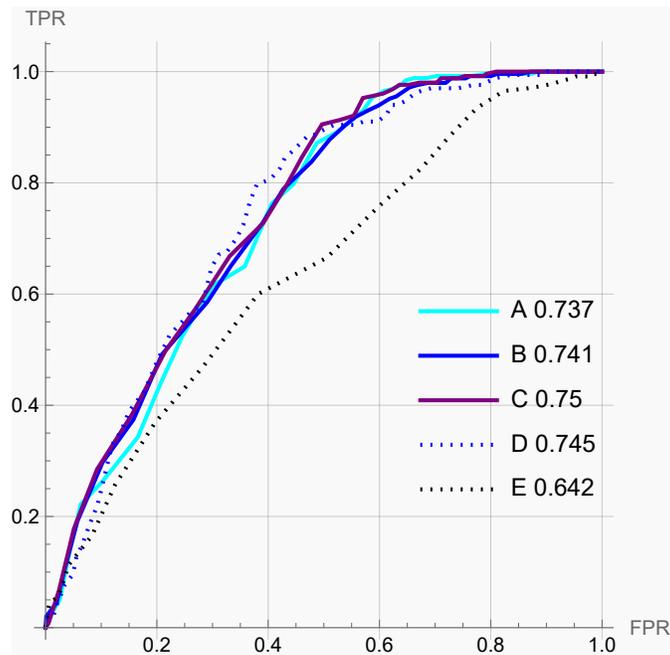
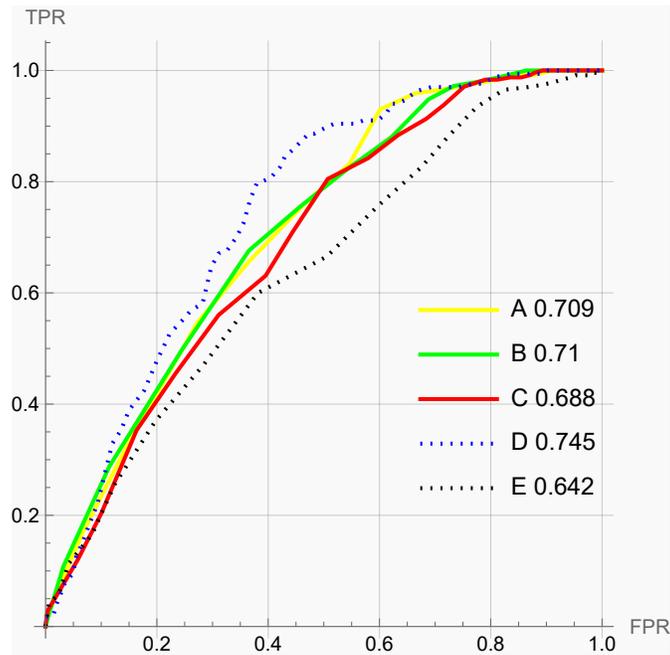


FIGURE A.38 – Courbes ROC des réseaux augmentés et non-augmentés. Haut : Réseaux non-augmentés avec A : R1, B : R2, C : R3, D : ORB, E : R0. Bas : Réseaux augmentés avec A : R4, B : R5, C : R6, D : ORB, E : R0.

TABLEAU A.4 – Résultats quantitatifs regroupant les métriques des différentes méthodes utilisées pour réaliser la mise en correspondance entre les points saillants ORB de deux images consécutives.

Méthode	VP	FP	FN	VN	Erreur moyenne	Erreur médiane	Écart-type erreur	ACC	AUC
R0	230	770	0	0	62.35	1.4	118.39	0.23	0.64
R1	243	755	0	2	13.27	1.3	61.11	0.24	0.71
R2	250	748	0	2	14.66	1.29	64.34	0.25	0.71
R3	241	754	0	5	15.11	1.3	63.28	0.25	0.69
R4	256	636	1	107	19.73	1.27	66.41	0.36	0.74
R5	245	631	1	123	31.49	1.29	89.53	0.37	0.74
R6	253	628	0	119	35.6	1.29	95.19	0.37	0.75
ORB	157	518	10	315	66.72	1.77	121.76	0.47	0.75

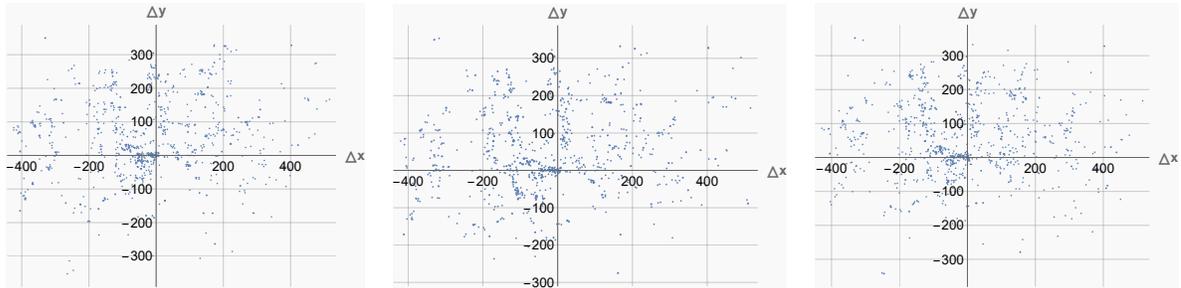


FIGURE A.39 – Visualisation des différences entre les points saillants mis en correspondance avec les descripteurs fournis par les réseaux non-augmentés, et leur référence.  
Gauche : R1. Milieu : R2. Droite : R3.

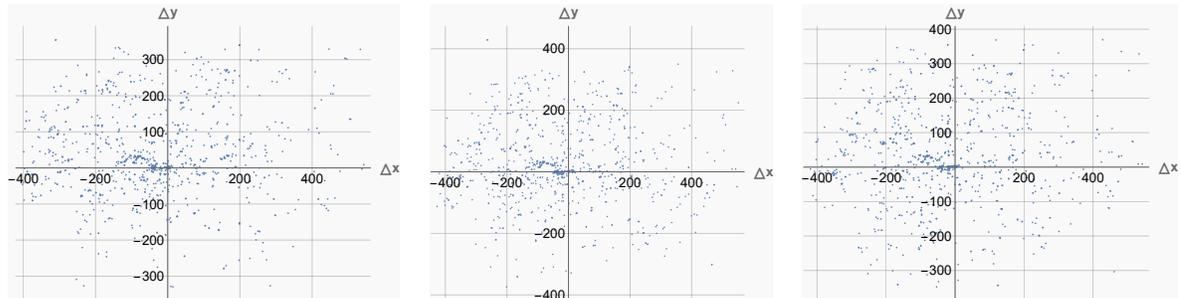


FIGURE A.40 – Visualisation des différences entre les points saillants mis en correspondance avec les descripteurs fournis par les réseaux augmentés, et leur référence.  
Gauche : R4. Milieu : R5. Droite : R6.

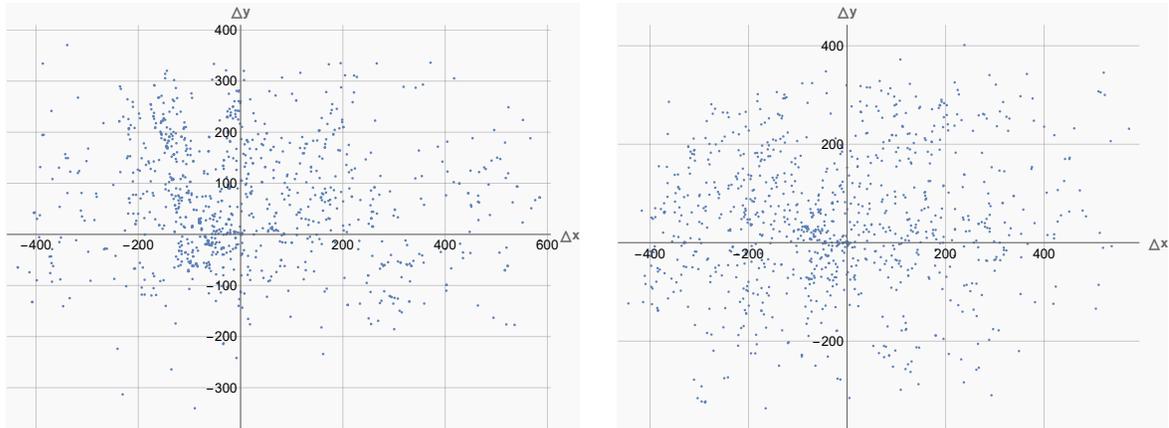


FIGURE A.41 – Visualisation des différences entre les points saillants mis en correspondance avec les descripteurs fournis par le réseau non-entraîné et les descripteurs ORB, et leur référence. Gauche : NT. Droite : ORB.

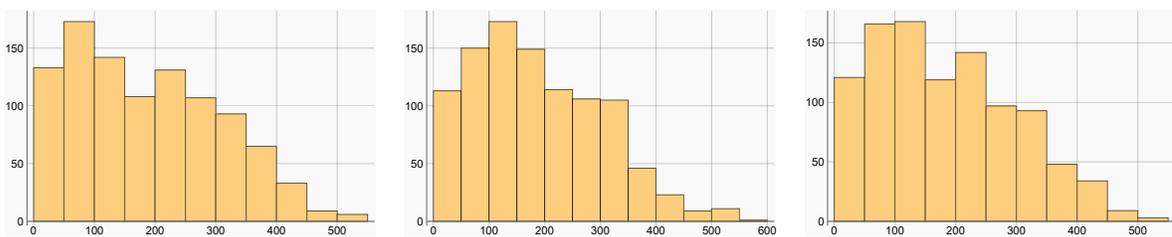


FIGURE A.42 – Histogramme de la distance euclidienne entre les points saillants mis en correspondance avec les descripteurs fournis par les réseaux non-augmentés, et leur référence. Gauche : R1. Milieu : R2. Droite : R3.

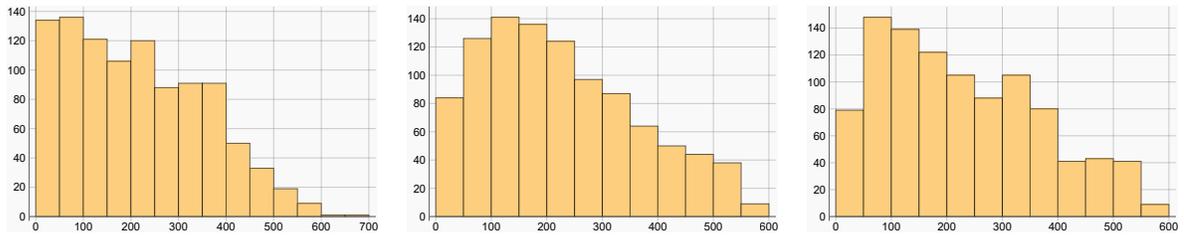


FIGURE A.43 – Histogramme de la distance euclidienne entre les points saillants mis en correspondance avec les descripteurs fournis par les réseaux augmentés, et leur référence. Gauche : R4. Milieu : R5. Droite : R6.

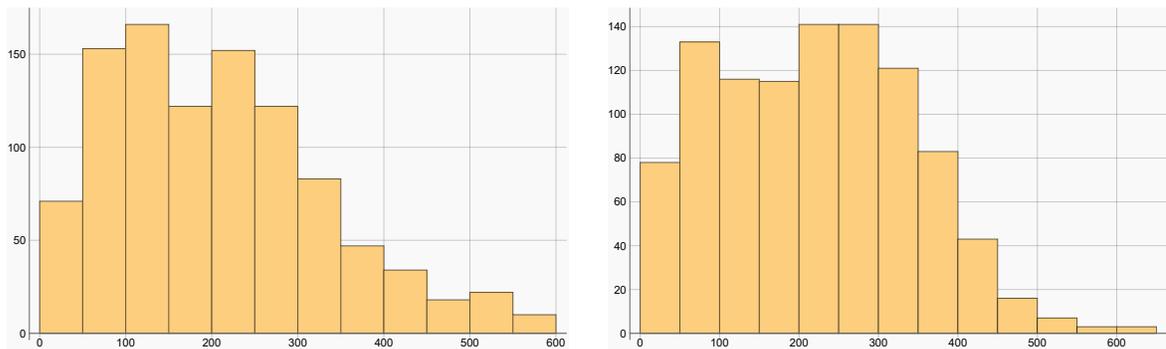


FIGURE A.44 – Histogramme de la distance euclidienne entre les points saillants mis en correspondance avec les descripteurs fournis par le réseau non-entraîné et les descripteurs ORB, et leur référence. Gauche : NT. Droite : ORB.

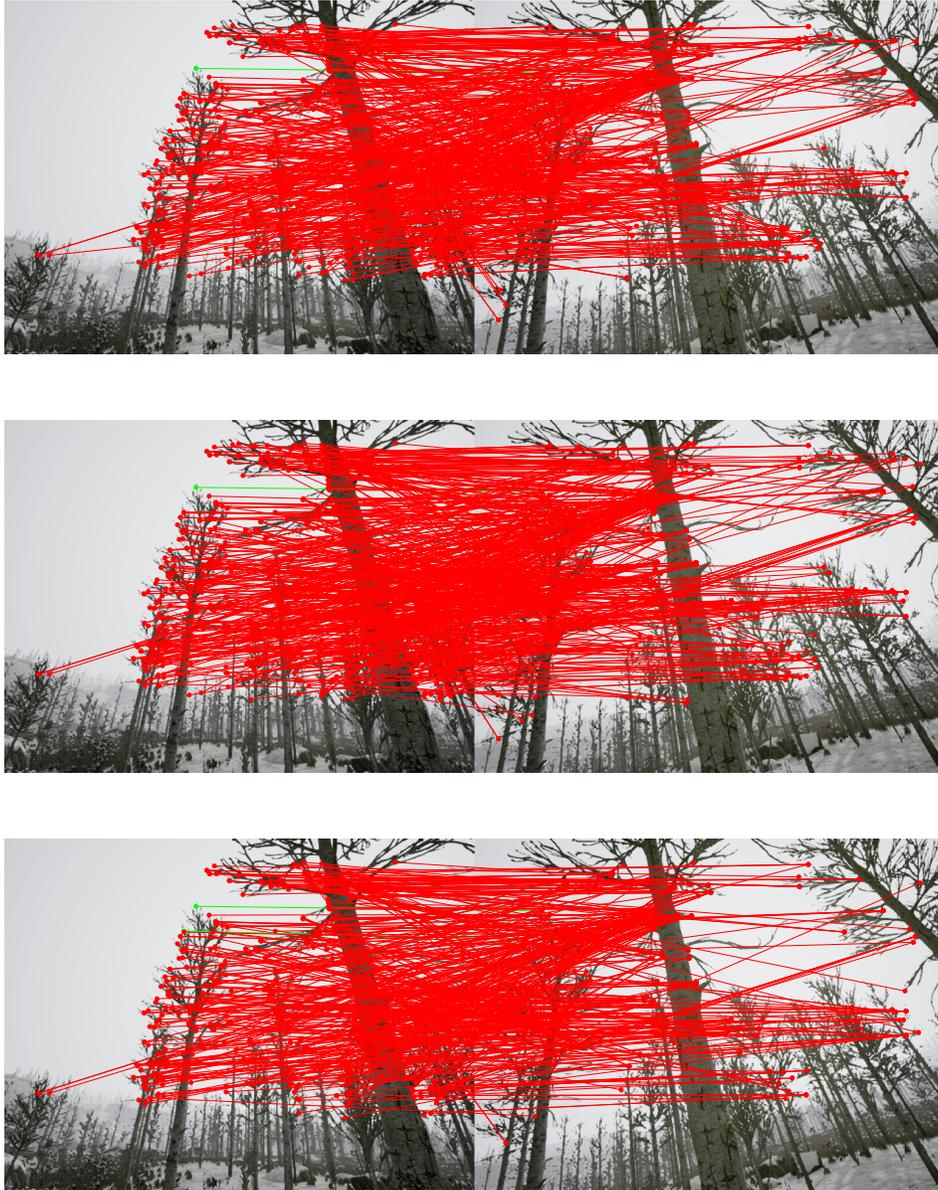


FIGURE A.45 – Résultat qualitatif de la mise en correspondance entre les points saillants ORB de deux images non-consécutives réalisée avec les descripteurs fournis par les réseaux non-augmentés. Les lignes vertes représentent des mises en correspondance correctes. Les lignes rouges représentent les mises en correspondance incorrectes. Haut : R1. Milieu : R2. Bas : R3.



FIGURE A.46 – Résultat qualitatif de la mise en correspondance entre les points saillants ORB de deux images non-consécutives réalisée avec les descripteurs fournis par les réseaux augmentés. Les lignes vertes représentent des mises en correspondance correctes. Les lignes rouges représentent les mises en correspondance incorrectes. Haut : R4. Milieu : R5. Bas : R6.



FIGURE A.47 – Résultat qualitatif de la mise en correspondance entre les points saillants ORB de deux images non-consécutives réalisée avec les descripteurs fournis par le réseau non-entraîné R0, et les descripteurs ORB. Les lignes vertes représentent des mises en correspondance correctes. Les lignes rouges représentent les mises en correspondance incorrectes. Haut : R0. Bas : ORB.

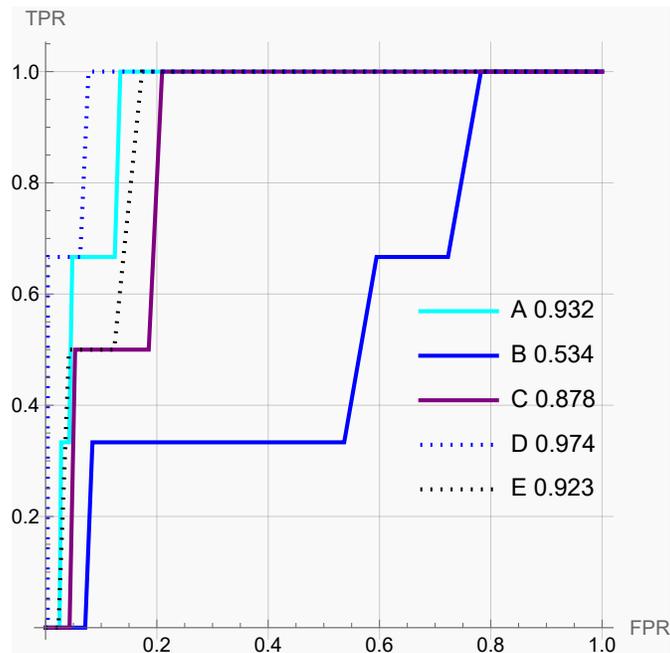
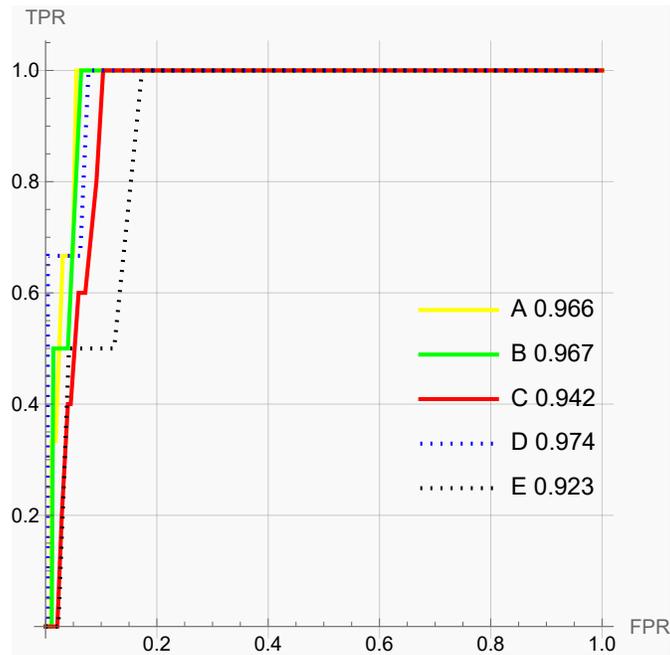


FIGURE A.48 – Courbes ROC des réseaux augmentés et non-augmentés. Haut : Réseaux non-augmentés avec A : R1, B : R2, C : R3, D : ORB, E : R0. Bas : Réseaux augmentés avec A : R4, B : R5, C : R6, D : ORB, E : R0.

TABLEAU A.5 – Résultats quantitatifs regroupant les métriques des différentes méthodes utilisées pour réaliser la mise en correspondance entre les points saillants ORB de deux images non-consécutives.

Méthode	VP	FP	FN	VN	Erreur moyenne	Erreur médiane	Écart-type erreur	ACC	AUC
R0	2	998	0	0	207.55	193.98	125.53	0.	0.92
R1	3	675	0	322	188.06	178.98	121.67	0.32	0.97
R2	2	739	0	259	188.78	173.06	117.09	0.26	0.97
R3	5	561	0	434	185.03	170.68	115.87	0.44	0.94
R4	2	84	1	913	195.92	188.05	123.41	0.92	0.93
R5	0	44	3	953	204.98	200.02	125.04	0.95	0.53
R6	1	117	1	881	208.12	205.58	120.8	0.88	0.88
ORB	3	77	0	920	220.6	222.54	122.55	0.92	0.97

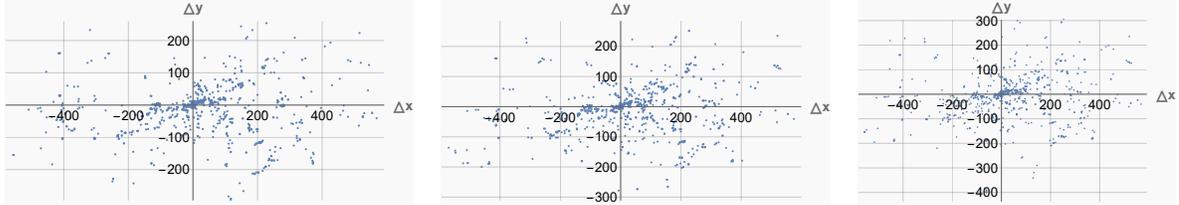


FIGURE A.49 – Visualisation des différences entre les points saillants mis en correspondance avec les descripteurs fournis par les réseaux non-augmentés, et leur référence.  
Gauche : R1. Milieu : R2. Droite : R3.

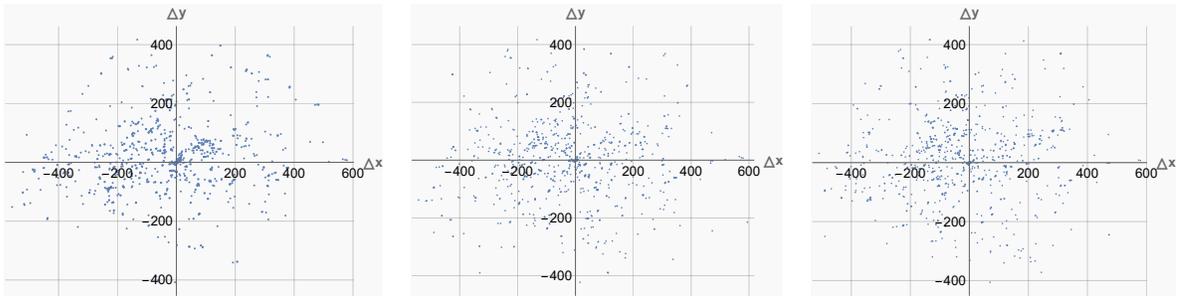


FIGURE A.50 – Visualisation des différences entre les points saillants mis en correspondance avec les descripteurs fournis par les réseaux augmentés, et leur référence.  
Gauche : R4. Milieu : R5. Droite : R6.

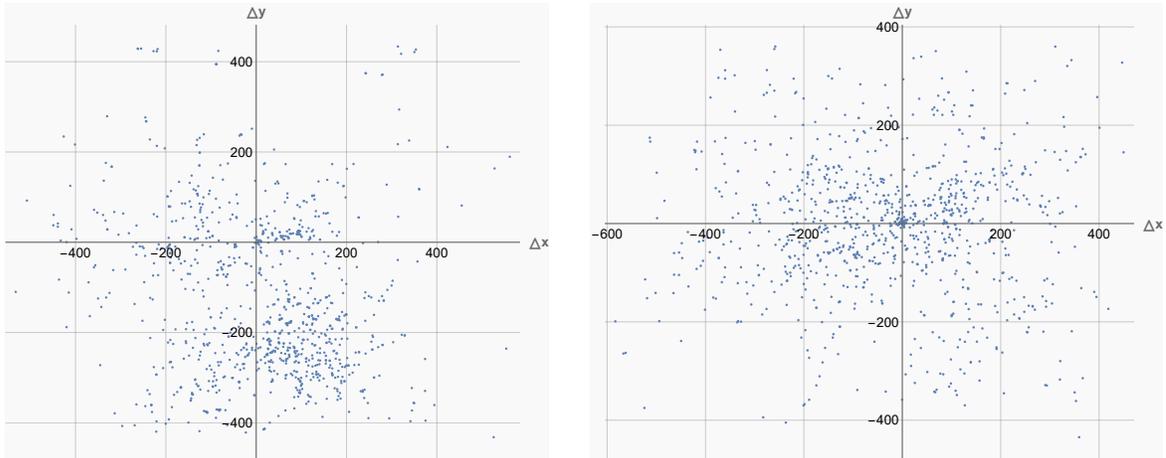


FIGURE A.51 – Visualisation des différences entre les points saillants mis en correspondance avec les descripteurs fournis par le réseau non-entraîné et les descripteurs ORB, et leur référence. Gauche : NT. Droite : ORB.

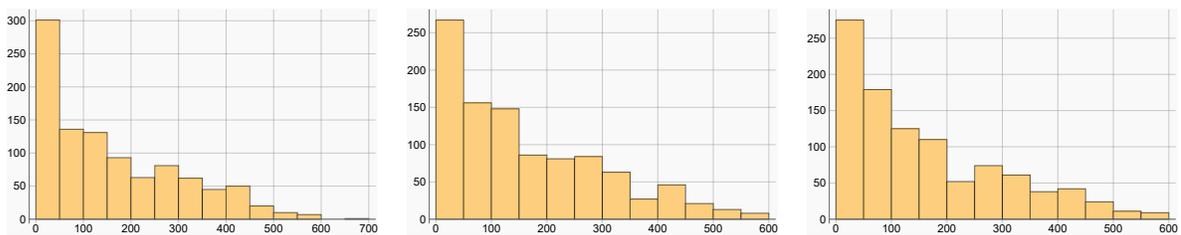


FIGURE A.52 – Histogramme de la distance euclidienne entre les points saillants mis en correspondance avec les descripteurs fournis par les réseaux non-augmentés, et leur référence. Gauche : R1. Milieu : R2. Droite : R3.

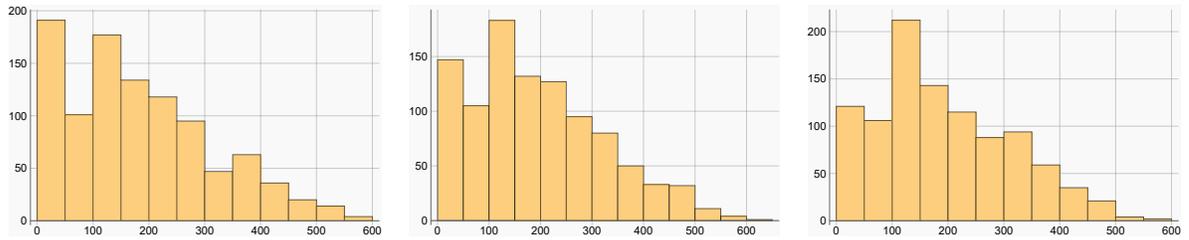


FIGURE A.53 – Histogramme de la distance euclidienne entre les points saillants mis en correspondance avec les descripteurs fournis par les réseaux augmentés, et leur référence. Gauche : R4. Milieu : R5. Droite : R6.

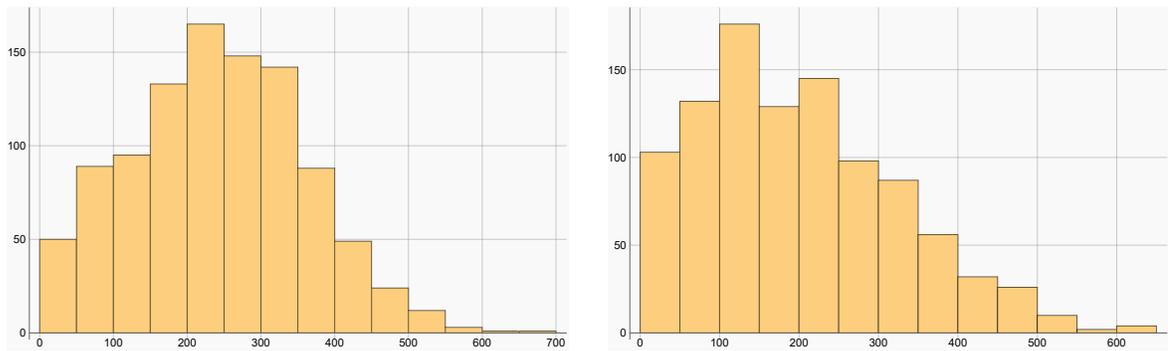


FIGURE A.54 – Histogramme de la distance euclidienne entre les points saillants mis en correspondance avec les descripteurs fournis par le réseau non-entraîné et les descripteurs ORB, et leur référence. Gauche : NT. Droite : ORB.

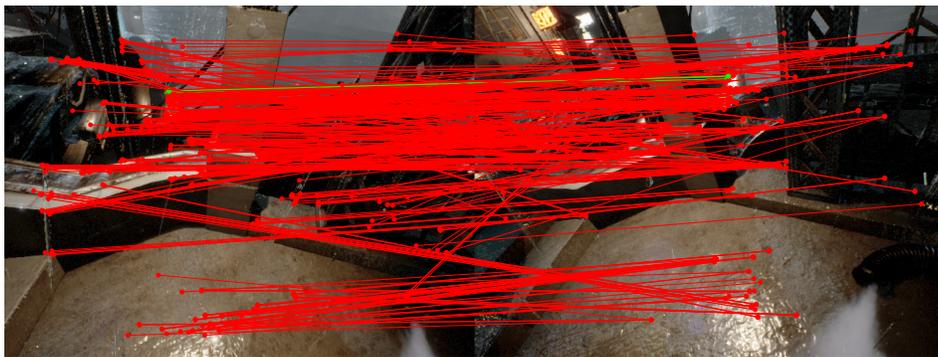
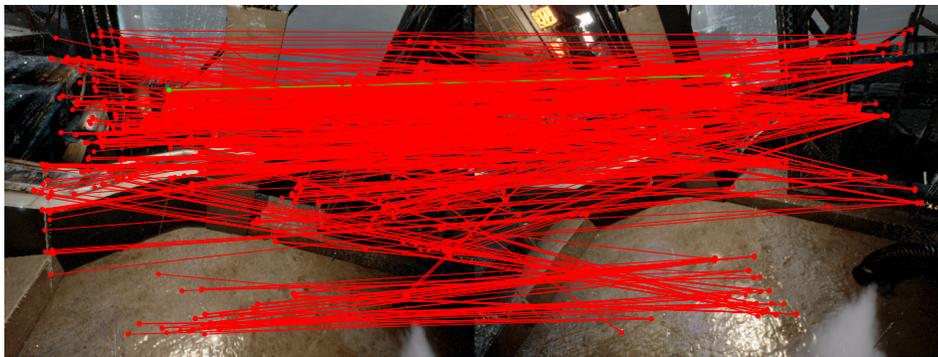
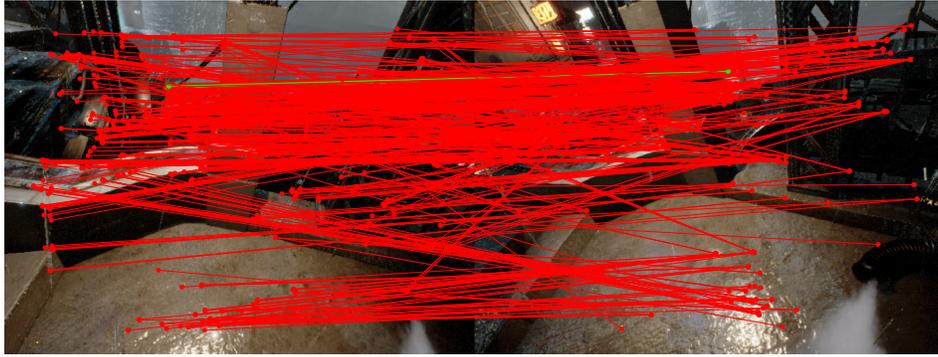


FIGURE A.55 – Résultat qualitatif de la mise en correspondance entre les points saillants ORB de deux images non-consécutives réalisée avec les descripteurs fournis par les réseaux non-augmentés. Les lignes vertes représentent des mises en correspondance correctes. Les lignes rouges représentent les mises en correspondance incorrectes. Haut : R1. Milieu : R2. Bas : R3.



FIGURE A.56 – Résultat qualitatif de la mise en correspondance entre les points saillants ORB de deux images non-consécutives réalisée avec les descripteurs fournis par les réseaux augmentés. Les lignes vertes représentent des mises en correspondance correctes. Les lignes rouges représentent les mises en correspondance incorrectes. Haut : R4. Milieu : R5. Bas : R6.

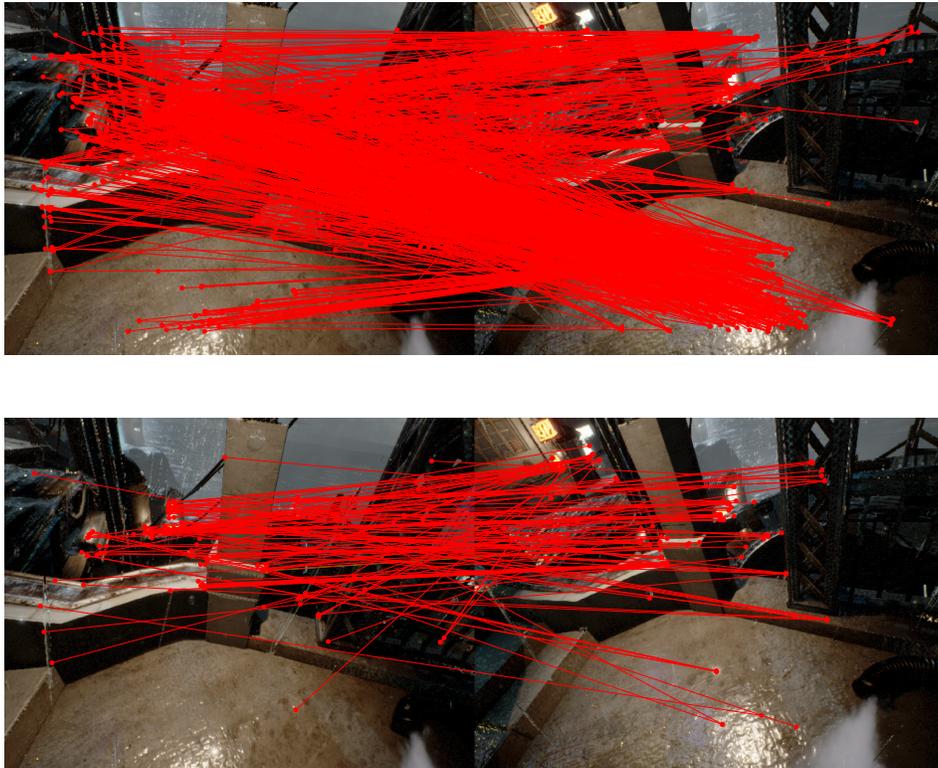


FIGURE A.57 – Résultat qualitatif de la mise en correspondance entre les points saillants ORB de deux images non-consécutives réalisée avec les descripteurs fournis par le réseau non-entraîné R0, et les descripteurs ORB. Les lignes vertes représentent des mises en correspondance correctes. Les lignes rouges représentent les mises en correspondance incorrectes. Haut : R0. Bas : ORB.

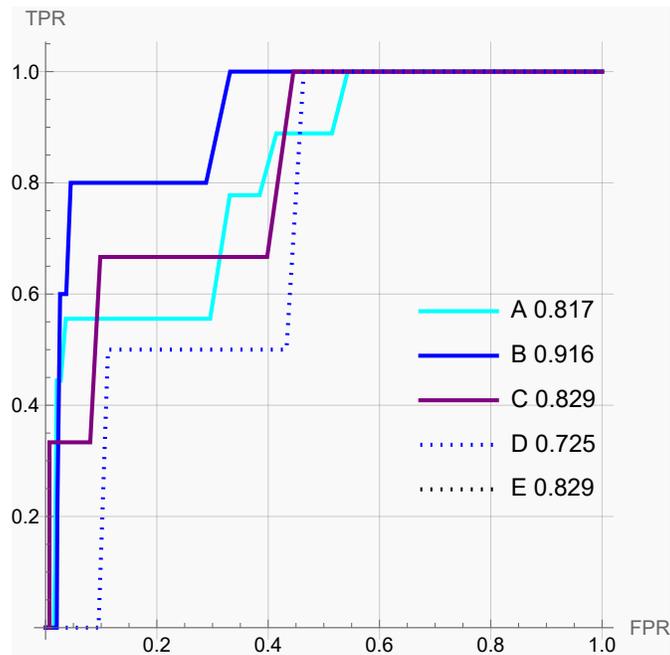
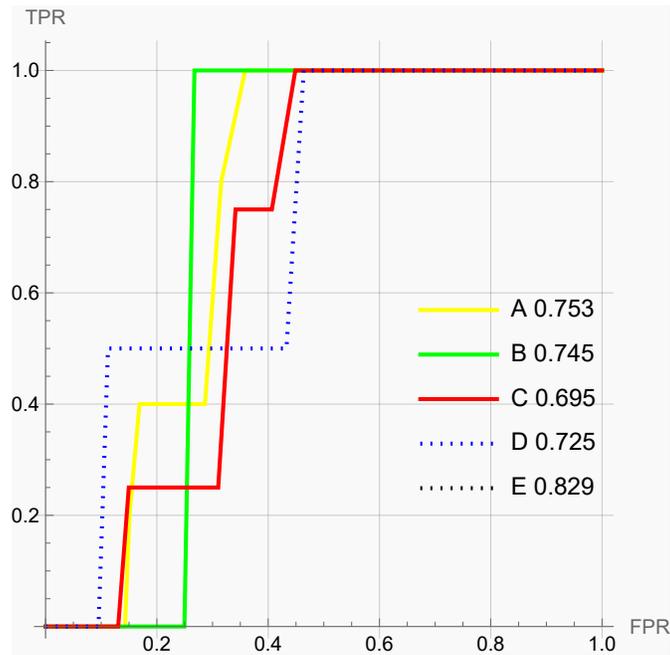


FIGURE A.58 – Courbes ROC des réseaux augmentés et non-augmentés. Haut : Réseaux non-augmentés avec A : R1, B : R2, C : R3, D : ORB, E : R0. Bas : Réseaux augmentés avec A : R4, B : R5, C : R6, D : ORB, E : R0.

TABLEAU A.6 – Résultats quantitatifs regroupant les métriques des différentes méthodes utilisées pour réaliser la mise en correspondance entre les points saillants ORB de deux images non-consécutives.

Méthode	VP	FP	FN	VN	Erreur moyenne	Erreur médiane	Écart-type erreur	ACC	AUC
R0	0	981	0	19	241.49	243.83	117.74	0.02	0.83
R1	5	723	0	272	161.11	121.9	144.65	0.28	0.75
R2	2	858	0	140	162.04	120.09	140.37	0.14	0.74
R3	4	549	0	447	158.44	110.34	142.23	0.45	0.7
R4	5	182	4	809	182.33	161.39	132.08	0.81	0.82
R5	4	193	1	802	192.73	176.96	130.93	0.81	0.92
R6	2	278	1	719	191.57	171.17	122.31	0.72	0.83
ORB	1	140	1	858	200.16	182.63	125.96	0.86	0.72