Patch-based models for image post-production

Aurélie Bugeau

université de BORDEAUX

HABILITATION À DIRIGER LES RECHERCHES

Spécialité Informatique

Patch-based models for image post-production

Aurélie BUGEAU

Soutenue le 30 mai 2018

Membres du jury :

GILBOA Guy	Maître de Conférences, Technion - Israel Inst. of Technology (rapporteur)
LEZORAY Olivier	Professeur, Université de Caen Normandie (rapporteur)
MASNOU Simon	Professeur, Université Lyon 1 (rapporteur)
DOMENGER Jean-Philippe	Professeur, Université de Bordeaux (Examinateur)
KERVRANN Charles	Directeur de recherche, INRIA Rennes (Examinateur)
LEPETIT Vincent	Professeur, Université de Bordeaux (Examinateur)
THOLLOT Joëlle	Professeur, Grenoble INP (Examinatrice)

À Nicolas, Esteban, Anaé

Contents

Résumé des activités de recherche

Ce chapitre décrit l'ensemble de mes activités de recherche depuis l'obtention de mon doctorat en 2007. Ces activités s'articulent autour de trois axes: l'estimation de données manquantes, la segmentation et le suivi d'objets, le traitement et l'analyse de vidéos pour les sciences humaines et sociales. Les chapitres suivants détailleront plus en détail le premier axe qui constitue ma thématique de recherche principale.

I - Estimation de données manquantes

Ma thématique de recherche principale concerne l'estimation de données manquantes à des fins de restauration d'images et vidéos ou de fusion d'informations multi-modales.

I.1 - Inpainting d'images et de vidéos

Ces recherches ont débuté lors de mon post-doctorat au sein de la fondation Barcelona Media de novembre 2007 à juin 2010 dans le cadre du projet espagnol i3media. Ce projet regroupant de nombreux partenaires universitaires et industriels, avait pour objectif la création et la gestion automatique de contenus audiovisuels intelligents. Plus précisément mon post-doctorat, encadré par M. Bertalmìo et V. Caselles, Pr Université Pompeu Fabra, Barcelone, traitait du problème de l'inpainting: étant donnée une image dégradée, il s'agit de reconstruire les zones abîmées de cette image de telle sorte que l'image finale ait l'air naturelle.

Inpainting d'images

Les méthodes de l'état de l'art peuvent être décomposées en deux catégories principales. Les méthodes de la première catégorie, communément appelées méthodes basées-patchs ou à patchs, reposent sur la comparaison de petites zones carrés de l'images. Plus précisément, elles recherchent dans les zones non dégradées de l'image la couleur de substitution des pixels dégradés, en fonction de leur voisinage (section 1.3). Ces méthodes s'adaptent très bien à des régions texturées (eau, herbe, sable ...) mais ne permettent pas

de bien reconstruire les structures d'une image, c'est-à-dire les forts contours. La deuxième catégorie, qui a des propriétés inverses, regroupe les méthodes dites géométriques. Leur principe est de diffuser les couleurs connues sur le bord de la région à reconstruire vers l'intérieur de cette région en suivant les contours extérieurs (section 2.1.1). Aucune des méthodes existantes ne permet cependant de reconstruire correctement tout type d'image. Inspirés par (Bertalmio et al., 2003), nous avons tout d'abord proposer de combiner ces deux types d'approches en décomposant l'image à restaurer en la somme d'une image de texture et d'une image de structure. L'image de texture est tout d'abord reconstruite par une méthode à patchs tandis que l'image de structure est reconstruite par une méthode géométrique. L'algorithme mis en place donne de bons résultats sur une grande variété d'images mais reste trop dépendant de la méthode géométrique utilisée (Bugeau et al., 2009).
Nous avons par la suite combiné les deux types d'approches de manière plus directe. Ceci est fait en considérant une fonction d'énergie contenant trois termes principaux, un pour chaque type d'approche et un augmentant la cohérence entre pixels voisins. Cette énergie est minimisée de manière itérative sous un schéma multi-résolution. Cette méthode a permis d'obtenir des résultats encourageants sur un grand nombre d'images (Bugeau et al., 2010a). Elle a été étendue à l'inpainting d'images stéréo dans (Hervieu et al., 2010; Hervieu et al., 2011).

Inpainting de vidéos

J'ai par la suite étudié le problème d'inpainting de vidéos. La difficulté réside ici dans l'ajout de cohérence temporelle entre les reconstructions successives d'images. Pour que cette reconstruction ait l'air naturelle, il est indispensable de prendre en compte des informations sur le mouvement. Nous avons proposé une méthode basée sur un lissage de Kalman (Bugeau et al., 2010b). Les observations sont les images de la vidéo complétées une à une par une méthode d'inpainting d'image. Les prédictions sont obtenues en appliquant une méthode d'inpainting géométrique sur le flot optique entre deux frames successives .

I.2 - Colorisation d'images et de vidéos

Depuis 2012, je m'intéresse également à la colorisation d'images, *i.e.* à l'estimation des canaux de couleur de chaque pixel d'une image en niveaux de gris.

Méthodes à patchs pour la colorisation d'images

Il existe dans la littérature deux approches principales. La première se base sur la diffusion des couleurs : l'utilisateur choisit manuellement la couleur d'un certain nombre de pixels de l'image puis ces couleurs sont automatiquement diffusées aux pixels voisins (*e.g.* Levin et al., 2004, section 3.1). Cette méthode est limitée à des images simples contenant peu d'objets. Le second type d'approches concerne la colorisation dite par l'exemple. Elle est issue des méthodes de synthèse de texture à patch (Efros et al., 1999). Il s'agit ainsi de copier dans l'image en niveau de gris les teintes ou chrominances d'une image source en couleur (*e.g.* Welsh et al., 2002, section 1.4). L'avantage est que, mis à part la sélection de l'image couleur de référence, ces méthodes sont entièrement automatiques. Ces approches par l'exemple reposent sur la comparaison entre les luminances de l'image en niveau de gris et celles de l'image couleur. Cette comparaison est classiquement effectuée en calculant la (dis-)similarité entre le voisinage d'un pixel à coloriser avec l'ensemble ou un sous-ensemble des patchs de l'image couleur. De nombreuses métriques et descripteurs existent pour comparer des patchs mais aucune étude exhaustive portant sur l'influence de chaque métrique sur les résultats de colorisation n'a été publiée. Une des raisons est qu'il n'existe pas de mesure quantitative adaptée permmettant de valider les résultats de colorisation. Dans ce contexte, en collaboration avec V.-T. Ta, MCF LaBRI/Bordeaux INP, nous avons proposé une méthode permettant de calculer pour chaque pixel la meilleure couleur à lui transférer pour différents descripteurs d'un patch. Chaque pixel a ainsi plusieurs couleurs candidates possibles, une pour chacune des métriques. Dans (Bugeau et al., 2012), la sélection de la couleur finale se fait tout simplement en prenant la couleur médiane de l'ensemble des candidats possibles.

Modèles variationnels pour la colorisation d'images et de vidéos basée patch

Nous avons par la suite, en collaboration avec N. Papadakis, CR CNRS/IMB, proposé la sélection automatique du meilleur candidat par une méthode variationnelle permettant l'ajout d'une contrainte spatiale sur la sélection (Bugeau et al., 2014). Ces travaux ont été poursuivis dans le cadre de la thèse de Fabien Pierre, dirigée par J.-F. Aujol, Pr Université de Bordeaux/IMB et co-encadrée par V.-T. Ta et moi-même (section 1.4.3). Une étude sur le calcul d'une combinaison linéaire optimale de métriques entre patchs pour la colorisation d'images a tout d'abord été menée (Pierre et al., 2015b). Il apparait, avec cette méthode, que le choix de la métrique optimale est trop dépendant de l'image à coloriser et qu'une sélection au sein d'un modèle variationnel (Bugeau et al., 2014) apporte de meilleurs résultats. Nous avons donc poursuivi dans cette direction (chapitre 3).
Un défaut majeur des résultats de colorisation existants est qu'ils sont globalement ternes.

Nous avons analysé les raisons et proposés des solutions applicables à l'ensemble des méthodes de l'état-de-l'art. Lors de la colorisation d'images, l'image originale en niveaux de gris est considérée comme un canal de luminance et les couleurs sont créées en modifiant les deux canaux de chrominance. Le résultat luminance/chrominance obtenu est ensuite converti en RGB pour pouvoir être observé. Néanmoins les conversions des espaces de luminance/chrominance vers RGB ne permettent pas toujours de conserver correctement la teinte de l'image colorisée. La première solution que nous avons proposé consiste à travailler directement dans l'image RGB. Le modèle variationnel (Bugeau et al., 2014) a été modifié de façon à estimer une image RGB tout en respectant la contrainte de luminance (luminance de l'image finale égale aux niveaux de gris de l'image initiale) (Pierre et al., 2014b). Dans cette méthode, la projection de la couleur RGB estimée sous contrainte de luminance est réalisée avec une projection orthogonale. Malheureusement cette projection ne conserve pas correctement la teinte de la couleur RGB initiale. Une étude approfondie des espaces de couleur et de leurs propriétés géométriques nous a amené à proposer une projection oblique menant à des résultats de colorisation faisant maintenant parti de l'état de l'art (Pierre et al., 2015c; Pierre et al., 2015a).
Similairement aux travaux sur l'inpainting, nous avons également travaillé sur la combinaison des deux grandes catégories d'approches de colorisation d'images. Nous avons ainsi unifié dans un seul modèle les approches par l'exemple et les approches basées sur la diffusion de couleurs marquées par un utilisateur (Pierre et al., 2014a; Pierre et al., 2015a). Ces travaux ont fait l'objet d'un dépôt logiciel appelé "Colociel". L'extension de ces travaux (section 3.2.1) à la colorisation de vidéos, obtenue avec l'ajout dans nos fonctionnelles d'un terme encourageant la cohérence temporelle sur les couleurs candidates, a enfin été proposée (Pierre et al., 2017b).

Application au rehaussement de contraste

Les travaux sur les espaces couleurs et sur la préservation de la teinte ont été étendus au problème de réhaussement de contraste suite à une collaboration avec G. Steidl, Pr. université de Kaiserslautern, Allemagne. La mesure du contraste est ici définie comme une moyenne pondérée des différences locales entre pixels voisins. Le réhaussement est alors obtenu par minimisation d'un modèle variationnel augmentant cette mesure de contraste par un facteur donné tout en préservant la teinte de l'image initiale (Pierre et al., 2017a; Pierre et al., 2016) (figure 1).

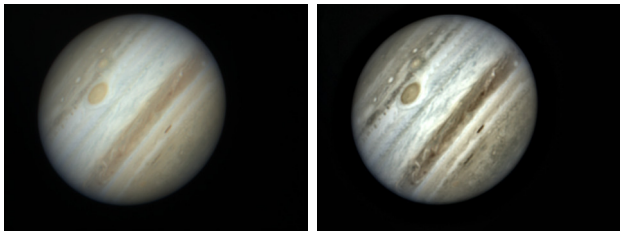


Image initiale Image rehaussée

Figure 1: Résultats de rehaussement de contraste (Pierre et al., 2017a).

I.3 - Enrichissement et complétion de nuages de points LIDAR

La dernière application entamée dans le cadre de mes recherches sur l'estimation de données manquantes concerne la densification de nuages de points LiDAR. Ces travaux sont réalisés en collaboration avec J.-F. Aujol et M. Brédif, chercheur à l'IGN. L'équipe MATIS de l'IGN a développé un système de cartographie mobile (MMS), équipé de détecteur laser (LiDAR), de capteurs optiques, d'un GPS et d'une centrale inertielle. L'objectif à terme est de construire des cartes 3D extrêmement précises des zones urbaines. Les acquisitions LiDAR fournissent des informations de profondeur et de réflectance permettant la reconstruction d'un nuage de points 3D éparse de l'environnement, contenant des millions de points par minute d'acquisition. En complément, les capteurs optiques apportent de nombreuses images HD avec différents angles de vue.

Enrichissement des images optiques

Dans le cadre du post-doctorat de Marco Bevilacqua, nous nous sommes intéressés au couplage des nuages de points LiDAR avec les images optiques. Plus précisément, nous avons travaillé sur l'enrichissement des images optiques en leur ajoutant un canal de profondeur u et de réflectance r provenant des points LiDAR préalablement projetés sur les coordonnées des pixels de l'image optique (figure 2). Les nuages de points LiDAR (r_S et u_S) n'étant pas denses et ayant des coordonnées sous pixelliques dans l'image, l'estimation de la profondeur et de la réflectance en chaque pixel n'est pas directe. Nous avons donc proposé un modèle variationnel encourageant les contours de la reconstruction LiDAR à correspondre aux contours de l'image et s'assurant de la cohérence spatiale des valeurs de profondeurs (Bevilacqua et al., 2016) et de réflectances (Bevilacqua et al., 2017). Pour cela une carte de visibilité dans l'image v de chacun des points projetés est estimée.

Complétion de nuages de points LiDAR

Dans le cadre de la thèse de Pierre Biasutti débutée en septembre 2016, nous nous sommes intéressés à la suppression d'objets non permanents de la scène tels que des objets en

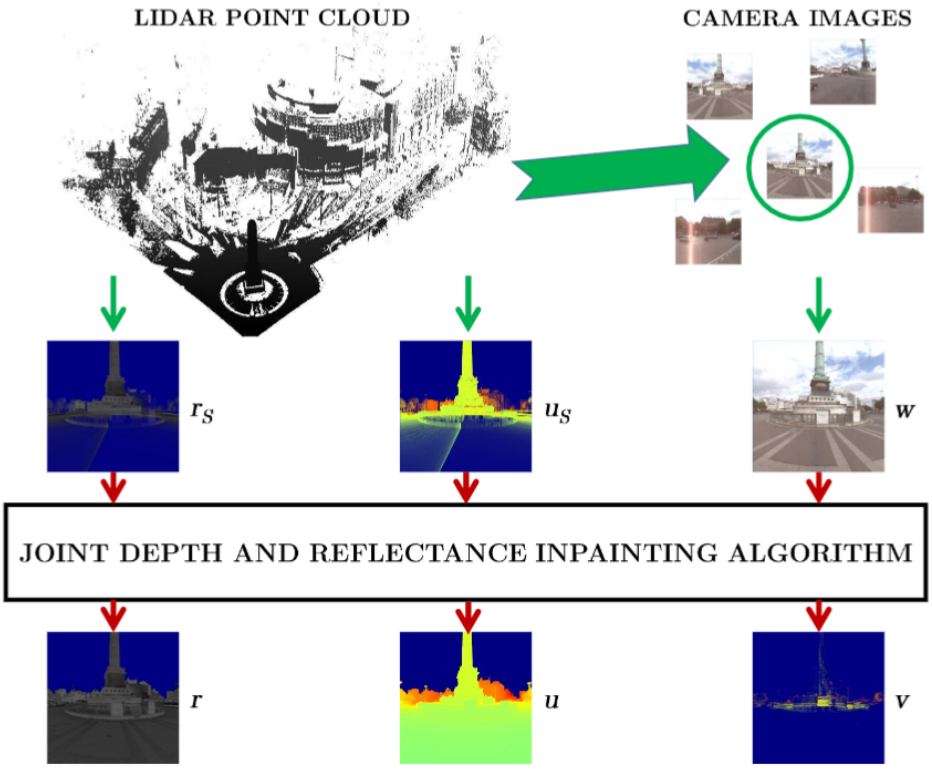


Figure 2: Méthode d'enrichissement des images optiques à partir des données de réflectance et de profondeur issues du capteur LiDAR (Bevilacqua et al., 2017).

mouvement ou des voitures garées. Pour cela nous avons proposé une représentation 2D en topologie capteur du nuage de points 3D. La segmentation de l'histogramme de cette nouvelle image permet ensuite d'extraire semi-automatiquement le masque des objets à supprimer. Une méthode d'inpainting géométrique a alors été proposée pour estimer la profondeur des zones occultées (Biasutti et al., 2017b; Biasutti et al., 2018, section 2.1.2).

Création d'orthoimages à partir de points LiDAR

La complétion de nuages de points LiDAR peut également s'appliquer à la création d'orthoimages. Une orthoimage est une image de la surface terrestre exempte de toute distorsion, généralement acquise par des vecteurs satellites ou aériens. Dans le cadre du stage de Master 2 de Pierre Biasutti, nous avons proposé une méthode générant une orthoimage de réflectance et de hauteur à très grande résolution (1cm) à partir des points LiDAR fournis par un système de cartographie mobile. La projection verticale (sur le sol) du nuage de points LiDAR sur une grille de pixels 2D étant éparse, des algorithmes d'inpainting sont appliqués afin d'obtenir une reconstruction dense des cartes de hauteur et de réflectance (Biasutti et al., 2016; Biasutti et al., 2017a, section 2.1.3).

II - Segmentation et suivi d'objets

Depuis mon doctorat, je m'intéresse également à la segmentation et au suivi d'objets. Dans l'ensemble de ces travaux, ce problème est vue comme un problème d'étiquetage, l'objectif étant d'attribuer à chaque pixel une étiquette "fond" ou "objet".

II.1 - Segmentation et suivi par coupe de graphes

Lors de ma thèse je me suis intéressée à la segmentation et au suivi temporel d'objets en mouvement. Dans ce cadre, une méthode reposant sur la minimisation d'une énergie par un algorithme de coupe minimale/flot maximal dans un graphe a été mise en place. Elle permet de suivre indépendamment chaque objet et regroupe les avantages d'une grande partie des méthodes de suivi existantes. La méthode ainsi créée est capable d'initialiser automatiquement les cibles et gère les entrées et sorties du champ de la caméra. Ceci est possible de part la prise en compte des détections externes d'objets dans l'énergie. D'autre part, la méthode est robuste aux changements d'illumination et aux changements de topologie des objets. Comme les méthodes de suivi par filtrage, la cohérence temporelle est basée sur l'utilisation dans la fonction d'énergie de la prédiction des objets. Enfin, une deuxième énergie multi-objets a été mise en place afin de séparer les objets ayant pu fusionner au cours de la première minimisation et permettre d'appréhender les problèmes d'occultation partielle (Bugeau et al., 2007; Bugeau et al., 2008a; Bugeau et al., 2008b). Cette méthode a néanmoins le défaut de nécessiter deux étapes. J'ai poursuivi mes recherches sur ce sujet et mis en place, en collaboration avec V. Caselles et N. Papadakis, un algorithme de suivi ne requérant qu'une seule minimisation pour tous les objet et étant capable de bien gérer les occultations partielles et globales. Pour cela un objet est divisé en deux parties : une région visible et une région occultée. Elles sont toutes deux suivies conjointement de part la minimisation d'une seule fonction d'énergie par un algorithme de coupe minimale/flot maximal (Papadakis et al., 2011, figure 3).

Application à l'indexation d'images

Ces recherches sur la segmentation par coupe de graphes ont été appliquées à l'indexation d'images avec une méthode reposant sur les sacs de descripteurs et les sacs de mots (Bag of Features and Bag of Words). Comme dans le cadre de la recherche textuelle, le principe des sacs de mots consiste à représenter une image par une distribution discrète des "mots visuels" qui la composent. Pour cela, des descripteurs visuels sont extraits localement des images puis quantifiés à l'aide d'un dictionnaire visuel obtenu par classification K-means. Ce dernier permet d'associer à chaque image un histogramme de ses descripteurs.

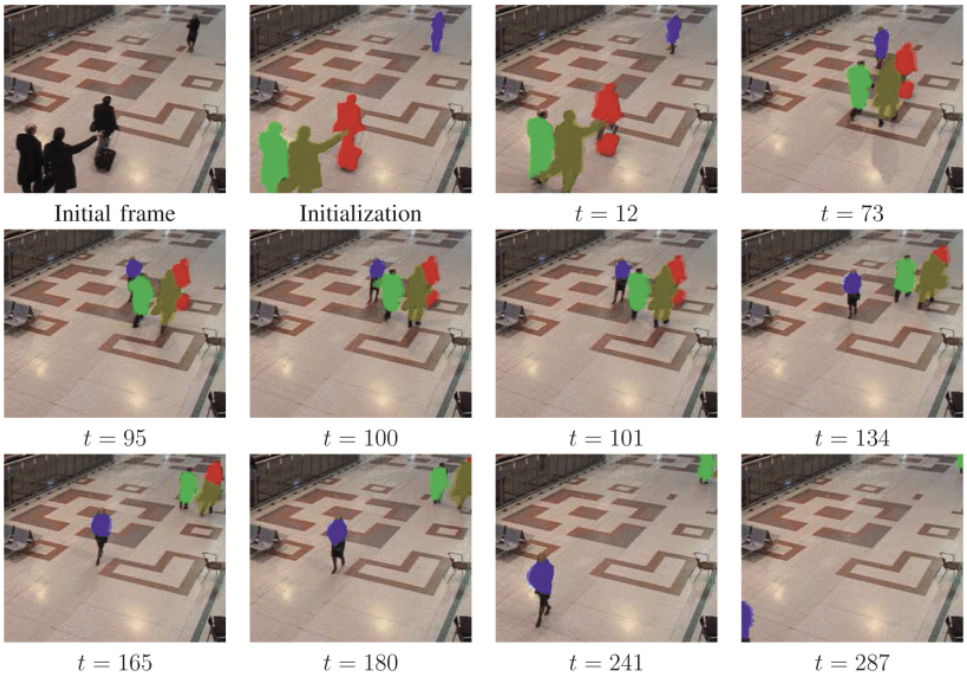


Figure 3: Résultat du suivi d'objets (Papadakis et al., 2011) sur une séquence de la base de données PETS 2001. Les couleurs sombres indiquent les parties visibles des objets suivis, les couleurs clairs indiquant les parties occultées.

Ces méthodes ont le défaut de ne pas tenir compte de l'information spatiale des images, ou plus précisément de la position dans l'image de chacun des descripteurs intervenant dans la construction de l'histogramme. Dans (Karaman et al., 2012; Benois-Pineau et al., 2012), nous avons donc proposé d'intégrer l'information spatiale explicitement au sein des descripteurs, c'est-à-dire avant la création de l'histogramme. Ces descripteurs sont des graphes multi-échelle obtenus par triangulation de Delaunay appliquées aux points SURF (Bay et al., 2006) de l'image.

Afin d'ajouter une information spatiale au moment de la construction de l'histogramme de mots visuels, l'utilisation de pyramides spatiales a été proposé dans la littérature (Lazebnik et al., 2006). Cette méthode, qui a largement contribué aux avancées dans le domaine de la recherche d'images, n'est néanmoins pas invariante aux transformations affines des images. Dans le cadre de la thèse de Yi Ren, co-encadrée par J. Benois-Pineau, Pr LaBRI/Université de Bordeaux, nous avons travaillé sur le découpage de l'image en régions cohérentes, chaque région étant ensuite représentée par son propre histogramme de mots visuels. L'histogramme global d'une image, appelé "Bag-of-Bags of Words", est alors obtenu par concaténation des histogrammes de chaque région. Pour segmenter les images en régions cohérentes, nous avons proposés des énergies ensuite minimisées par un algorithme de coupe minimale/flot maximal (Y. Ren et al., 2014b; Y. Ren et al., 2014a,

figure 4).

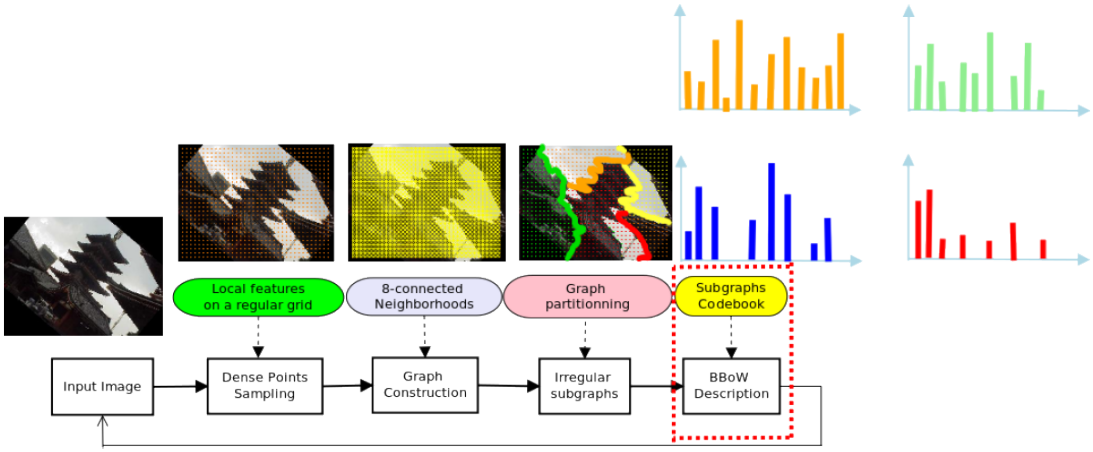


Figure 4: Méthode pour la création des "Bag-of-Bags of Words" (Y. Ren et al., 2014b; Y. Ren et al., 2014a).

II.2 - Segmentation par étiquetage de superpixels

Un autre travail de recherche sur la segmentation d'une image à partir d'une base d'images étiquetées à été proposé en collaboration avec R. Giraud, doctorant LaBRI, V.-T. Ta, N. Papadakis et P. Coupé, CR CNRS/LaBRI. Ce travail repose sur la décomposition de l'image en superpixels (petites zones connexes et de couleur homogène de l'image) avec la méthode SLIC (Achanta et al., 2012). En introduisant un algorithme rapide et robuste de mise en correspondances de "patch de superpixels" (appelés superpatch), nous sommes capables de classifier chacun des superpixels d'une image à segmenter (Giraud et al., 2017, figure 5).

III - Traitement et analyse de vidéos pour les sciences humaines et sociales

Ma troisième thématique de recherche est multidisciplinaire et repose sur plusieurs collaborations avec des chercheurs de sciences cognitiques et de sciences humaines et sociales.

III.1 - Anonymisation fine de visages

J'ai débuté en 2013 une collaboration avec Maria-Caterina Manes Gallo, Pr en Sciences de l'Information et de la Communication à l'université Bordeaux Montaigne et rattachée au laboratoire MICA (Mediation, Information, Communication, Art). L'objectif était

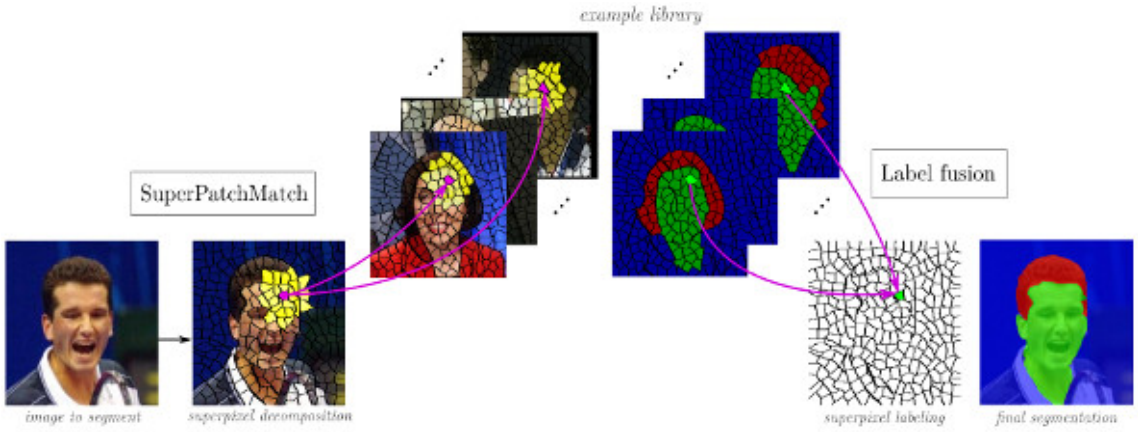


Figure 5: Utilisation de la méthode SuperPatchMatch pour la segmentation d'images (Giraud et al., 2017).

de réaliser une dé-identification fine des visages présents dans une vidéo. Il s'agit de préserver l'anonymat des personnes tout en conservant au maximum les expressions du visage et de manière plus générale les signes de communications non verbaux. Pour réaliser ce travail, j'ai co-encadré G. Letournel, ingénieur d'étude, avec J.-P. Domenger, Pr. LaBRI/Université de Bordeaux, et V.-T. Ta. Dans un premier temps, nous avons étudié les méthodes de détection automatique de visages ainsi que de points caractéristiques sur les visages. Nous avons ensuite proposé une méthode de dé-identification fine sur image fixe reposant sur un lissage adaptatif du visage (figure 6). Les yeux et la bouche sont conservés car indispensables à la reconnaissance des expressions. Le nez est supprimé car il aide à la reconnaissance faciale mais pas à celle des expressions. Le reste du visage est lissé avec une force proportionnelle à la distance aux yeux et à la bouche. L'obtention du résultat est obtenu par minimisation d'un modèle variationnel.

La validation des résultats est une partie très importante de cette thématique de recherche. En effet, il faut s'assurer que les personnes sont effectivement correctement anonymisées mais également que leurs expressions sont préservées. Cette validation doit être faite par des humains mais aussi par des systèmes automatiques. Nous avons ainsi appliqué des algorithmes de reconnaissance de visages de l'état-de-l'art pour valider nos résultats. Il apparaît que le taux de reconnaissance passe de 86% avant anonymisation à 51% après (soit proche du hasard). Pour la validation humaine, nous nous sommes appuyés sur les connaissances en tests de perception de chercheurs en sciences cognitiques. Le co-encadrement d'un stagiaire avec V. Lespinet Najib, MCF IMS/ENSC a mené à la mise en place d'un protocole de test de perception pour la validation des résultats d'anonymisation de visages. Le test proposé évalue la reconnaissance de visage puis la reconnaissance des émotions. Il a été testé sur un panel de 22 sujets avec des résultats très prometteurs.

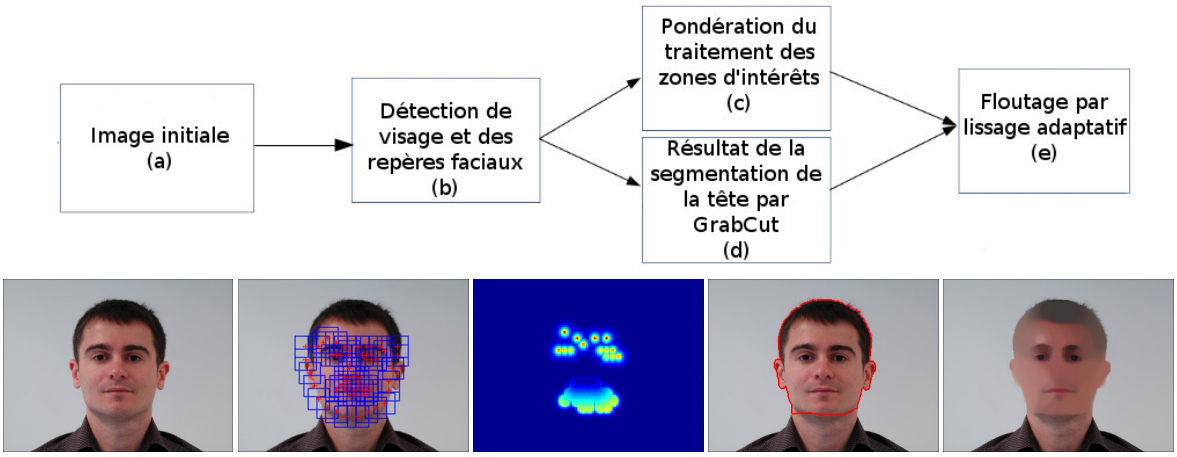


Figure 6: Chaîne de traitements pour l'anonymisation fine de visages (Letournel et al., 2015): (a) image initiale, (b) détection de visages et localisation de repères faciaux, (c) pondération du traitement des zones d'intérêts, (d) segmentation de la tête, (e) résultat du lissage adaptatif.

L'ensemble de ces travaux sur l'anonymisation fine de visages dans les image fixes ont été publiés dans (Letournel et al., 2015).

III.2 - Caractérisation visuelle des affects sociaux

En parallèle de ces travaux, j'ai travaillé sur la caractérisation visuelle des affects sociaux. Ce dernier projet a été réalisé de part l'encadrement du post-doctorant Zuheng Ming et en collaboration avec T. Shochi, MCF CLLE-ERSSaB/Université Bordeaux Montaigne et J.-L. Rouas, CR CNRS. Il fait suite aux nombreuses études déjà menées par T. Shochi sur la caractérisation sonore des affects sociaux chez des sujets japonais et français, de part l'étude des attitudes non spontanées telle que l'admiration, le dégoût, la séduction... Les attitudes étant très liées à la culture, l'objectif à terme est de mettre en avant des profils prototypiques des mêmes attitudes chez des acteurs japonais et des acteurs français afin de fournir les caractéristiques clés permettant de mieux comprendre et se faire comprendre lors de la communication en langue étrangère avec des natifs. L'intérêt est donc de pouvoir aider à l'apprentissage des langues. En effet, la communication repose sur du verbal (le parlé) mais aussi sur des aspects non verbaux (prosodie, gestuelle...). Une méconnaissance de ces aspects non verbaux peut mener à une mauvaise compréhension lors de l'immersion dans une autre culture (Shochi et al., 2016). Dans ce contexte, l'objectif ici était de trouver dans des vidéos des caractéristiques visuelles, sur le visage, communes à plusieurs acteurs jouant une même attitude. Les méthodes existantes sur la reconnaissance des expressions du visage reposent généralement sur l'estimation des "Facial Action Units". Dans nos

travaux nous avons proposé une méthode fusionnant différents descripteurs d'images pour la détection de ces descripteurs (Ming et al., 2015).
Ces recherches nous ont amené à établir une collaboration avec T. Nishida, Université de Kyoto, Japon, avec qui nous avons défini un scénario pour l'acquisition de vidéos de ces attitudes sur plusieurs personnes placées simultanément dans des environnements immersifs en réseau (Nishida et al., 2015).

Scientific Publications

Book Chapters

BC2. H. Boujut, A. Bugeau, J. Benois-Pineau – *Visual search for objects in a complex visual context: what we wish to see* – Chapter in Semantic Multimedia Analysis and Processing - Editors: E. Spyrou, D. Iakovidis, P.J. Mylonas - Publisher: Digital Imaging and Computer Vision, CRC Press, 2014.

BC1. J. Benois-Pineau, A. Bugeau, S. Karaman, R. Mégret – *Spatial and multi-resolution context in visual indexing* – Chapter 4 in Visual Indexing and Retrieval - Editors: J. Benois-Pineau, F. Precioso, M. Cord - Publisher: Springer, 2012.

International Journal

J13. P. Biasutti, J-F Aujol, M. Bredif, A. Bugeau – *Range-Image: Incorporating sensor topology for LIDAR point clouds processing* – Photogrammetric Engineering & Remote Sensing, 2018.

J12. R. Giraud, V.-T. Ta, A. Bugeau, P. Coupé, N. Papadakis – *SuperPatchMatch: an Algorithm for Robust Correspondences of Superpixel Patches* – IEEE Transactions on Image Processing (TIP), 2017.

J11. F. Pierre, J.-F. Aujol, A. Bugeau, V.-T. Ta – *Interactive Video Colorization within a Variational Framework* – SIAM Journal on Imaging Sciences, 2017.

J10. M. Bevilacqua, P. Biasutti, J-F Aujol, M. Brédif, and A. Bugeau, – *Joint Inpainting of Depth and Reflectance with Visibility Estimation* – ISPRS Journal of Photogrammetry and Remote Sensing, Volume 125, pages 16-32, 2017.

J9. F. Pierre, J.-F. Aujol, A. Bugeau, G. Steidl, V.-T. Ta, – *Variational Contrast Enhancement of Gray-Scale and RGB Images* – Journal of Mathematical Imaging and Vision (JMIV), Volume 57, pages 99-116, 2017.

J8. F. Pierre, J.-F. Aujol, A. Bugeau, N. Papadakis, V.-T. Ta, – *Luminance-Chrominance Model for Image Colorization* – SIAM Journal on Imaging Sciences, Volume 8, Issue 1, 2015.

J7. A. Bugeau, V.-T. Ta, N. Papadakis, – *Variational Exemplar-Based Image Colorization* – IEEE Transactions on Image Processing (TIP), Volume 33, Issue 1, 2014.

J6. N. Papadakis, A. Baeza, A. Bugeau, O. D'Hondt, P. Gargallo I Piraces, V. Caselles, X. Armangué, I. Rius, S. Sagàs – *Virtual camera synthesis for soccer game replays.* – Journal of Virtual Reality and Broadcasting, Volume 9, Issue 2012, 2013.

J5. N. Papadakis, A. Bugeau, V. Caselles – *Image editing with spatiogram transfer* – IEEE Transactions on Image Processing (TIP), Volume 21, Issue 5, May 2012.

J4. N. Papadakis, A. Bugeau – *Tracking with occlusions via Graphcuts* – IEEE Transactions Pattern Analysis and Machine Intelligence (TPAMI), Volume 33, Issue 1, Jan. 2011.

J3. A. Bugeau, M. Bertalmío, V. Caselles, G. Sapiro – *A Comprehensive Framework for Image Inpainting* – IEEE Transactions on Image Processing (TIP), Volume 19, Issue 10, Oct. 2010.

J2. A. Bugeau, P. Pérez – *Detection and segmentation of moving objects in complex scenes* – Computer Vision and Image Understanding (CVIU), Volume 113, Issue 4, April 2009.

J1. A. Bugeau, P. Pérez – *Track and Cut: simultaneous tracking and segmentation of multiple objects with graph cuts* – EURASIP Journal on Image and Video Processing - Special Issue on Video Tracking in Complex Scenes for Surveillance Applications, 2008(317278):1-14, 2008.

Preprints

P1. P. Biasutti, J-F Aujol, M. Brédif, A. Bugeau – *Diffusion and inpainting of reflectance and height LiDAR orthoimages* – HAL Preprint 01322822, 2017.

International Conferences

C24. P. Biasutti, J.-F. Aujol, M. Brédif, A. Bugeau. – *Disocclusion of 3D LiDAR point clouds using range images* – ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, 2017.

C23. M. Bevilacqua, J.-F. Aujol, M. Bredif, A. Bugeau. – *Visibility Estimation and Joint Inpainting of Lidar Depth Maps* – Internationnal Conference on Image Processing (ICIP), 2016.

C22. F. Pierre, J.-F. Aujol, A. Bugeau, G. Steidl, V.-T. Ta. – *Hue-Preserving Perceptual Contrast Enhancement* – Internationnal Conference on Image Processing (ICIP), 2016.

C21. T. Shochi, J.-L. Rouas, Z. Ming, M. Guerry, A. Bugeau, E. Donna. – *Cultural differences in pattern matching : multisensory recognition of socio-affective prosody* – International Congress of Psychology, 2016.

C20. F. Pierre, J.-F. Aujol, A. Bugeau, V.-T. Ta. – *A variational approach for color image enhancement* – SIAM Conference on Imaging Science, 2016.

C19. F. Pierre, J.-F. Aujol, A. Bugeau, V.-T. Ta. – *Luminance-Hue Specification in the RGB Space* – Scale space and variational method in computer vision (SSVM), 2015.

C18. G. Letournel, A. Bugeau, V.-T. Ta, J.-P. Domenger, M.C. Manes Gallo. – *Face de-identification with expressions preservation* – IEEE International Conference on Image Processing (ICIP), 2015.

C17. T. Nishida, M. Abe, T. Ookaki, D. Lala, S. Thovuttikul, H. Song, Y. Mohammad, C. Nitschke, Y. Ohmoto, A. Nakazawa, T. Shochi, J.-L. Rouas, A. Bugeau, F. Lotte, Z. Ming, G. Letournel, M. Guerry, D. Fourer– *Synthetic Evidential Study as Augmented Collective Thought Process – Preliminary Report* – International scientific conference for research in the field of intelligent information and database systems (ACIIDS), 2015.

C16. Z. Ming, A. Bugeau, J.-L. Rouas, T. Shochi – *Facial Action Units Intensity Estimation by the Fusion of Features with Multi-kernel Support Vector Machine* – IEEE International Conference on Automatic Face and Gesture Recognition Conference and Workshops (FG), 2015.

C15. F. Pierre, J.-F. Aujol, A. Bugeau, V.-T. Ta. – *Collaborative Image Colorization* – CPCV workshop - European Conference on Computer Vision, 2014.

C14. F. Pierre, J.-F. Aujol, A. Bugeau, N. Papadakis, V.-T. Ta – *Exemplar-based colorization in RGB color space* – IEEE International Conference on Image Processing (ICIP), 2014.

C13. Y. Ren, A. Bugeau, J. Benois-Pineau– *Bag-of-Bags of Words Irregular Graph Pyramids vs Spatial Pyramid Matching for Image Retrieval* – International Conference on Image Processing Theory, Tools and Applications (IPTA), 2014. - Best paper award

C12. Y. Ren, J. Benois-Pineau, A. Bugeau – *A Comparative Study of Irregular Pyramid Matching in Bag-of-Bags of Words Model for Image Retrieval* – International Conference on Image and Signal Processing (ICISP), 2014.

C11. A. Bugeau, V.-T. Ta – *Patch-based image colorization* – International Conference on Pattern Recognition (ICPR), 2012.

C10. S. Karaman, J. Benois-Pineau, R. Mégret, A. Bugeau – *Multi-Layer Local Graph Words for Object Recognition* – International Conference on Multimedia Modeling (ICMM), 2012.

C9. B. Delezoide, F. Precioso, P. Gosselin, M. Redi, B. Merialdo, L. Granjon, D. Pellerin, M. Rombaut, H. Jégou, R. Vieux, A. Bugeau, B. Mansencal, J. Benois-Pineau, H. Boujut, S. Ayache, B. Safadi, F. Thollard, G. Quénot, H. Bredin, M. Cord, A. Benoît , P. Lambert, T. Strat, J. Razik, S. Paris, H. Glotin, *IRIM at TRECVID 2011: High Level Feature Extraction and Instance Search* – TREC Video Retrieval Evaluation workshop, 2011.

C8. A. Hervieu, N. Papadakis, A. Bugeau, P. Gargallo, V. Caselles – *Stereoscopic Image Inpainting using scene geometry*–IEEE International Conference on Multimedia and Expo (ICME), 2011.

C8 G. Facciolo, R. Sadek, A. Bugeau, V. Caselles – *Temporally consistent gradient domain video editing* – 8th International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR), 2011

C7. N. Papadakis, A. Baeza, X. Armangué, I. Rius, A. Bugeau, O. D'Hondt, P. Gargallo I Piraces, V. Caselles, S. Sagàs – *Virtual camera synthesis for soccer game replays* – Conference on Visual Media Production (CVMP), 2010.

C6. A. Bugeau, P. Gargallo, O. D'Hondt, A. Hervieu, N. Papadakis, V. Caselles – *Coherent Background Video Inpainting through Kalman Smoothing along Trajectories*– Modeling, and Visualization Workshop, 2010.

C5. A. Hervieu, N. Papadakis, A. Bugeau, P. Gargallo; V. Caselles – *Stereoscopic image inpainting: distinct depth maps and images inpainting* – International Conference on Pattern Recognition (ICPR), 2010.

C4. A. Bugeau, M. Bertalmío – *Combining texture synthesis and diffusion for image inpainting* – International Conference on Computer Vision Theory and Applications (VISAPP), 2009.

C3. A. Bugeau, P. Pérez – *Track and Cut: simultaneous tracking and segmentation of multiple objects with graph cuts* – International Conference on Computer Vision Theory and Applications (VISAPP), 2008.

C2. A. Bugeau, P. Pérez – *Joint Tracking and Segmentation of Objects using Graph Cuts* – Proc. Conf. Advanced Concepts for Intelligent Vision Systems (ACIVS), 2007.

C1. A. Bugeau, P. Pérez – *Detection and segmentation of moving objects in highly dynamic scenes* – IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2007.

National Conferences

CN8. R. Giraud, V.-T. Ta, A. Bugeau, P. Coupé, N. Papadakis – SuperPatchMatch : Un algorithme de correspondances robustes de patchs de superpixels – Reconnaissance des Formes, Image, Apprentissage et Perception (RFIAP), 2018.

CN7. P. Biasutti, J.-F. Aujol, M. Brédif, A. Bugeau – Estimation de visibilité dans un nuage de points LiDAR – Conférence Française de Photogrammétrie et de Télédétection (CFPT), 2018.

CN6. P. Biasutti, J.-F. Aujol, M. Brédif, A. Bugeau – Desoccultation de nuage de points LiDAR en topologie capteur – Gretsi, 2017.

CN5. P. Biasutti, J.-F. Aujol, M. Brédif, A. Bugeau – Diffusion anisotrope et inpainting d'orthophotographies LiDAR mobile – Reconnaissance des Formes et Intelligence Artificielle (RFIA), 2016.

CN4. F. Pierre, J.-F. Aujol, A. Bugeau, V.-T. Ta – Combinaison linéaire optimale de métriques pour la colorisation d'images – GRETSI, 2015.

CN3. G. Letournel, A. Bugeau A., V.-T. Ta, J.-P. Domenger, M.C. Manes Gallo – *Anonymisation fine de visages avec préservation des expressions faciales*– Reconnaissance des Formes et Intelligence Artificielle (RFIA) , 2014.

CN2. S. Karaman, J. Benois-Pineau, R. Mégret, A. Bugeau – *Mots visuels issus de graphes locaux multi-niveaux pour la reconnaissance d'objets*– Reconnaissance des Formes et Intelligence Artificielle (RFIA) , 2012.

CN1. A. Bugeau, P. Pérez – Sélection de la taille du noyau pour l'estimationà noyau dans des espaces multidimensionnels hétérogènes – GRETSI, 2007.

Research Reports

RR1. A. Bugeau, P. Pérez – *Bandwidth selection for kernel estimation in mixed multi-dimensional spaces* – Technical report, INRIA, RR-6286, 2007.

Patch-based models for image post-production

Introduction

The purpose of image post-production is to generate a modified image from an original one by recovering or adding missing information to the image in order to improve its visual quality or modify its style. It involves processes such as image restoration (*i.e.* denoising or deblurring), contrast or color enhancement, loss data completion, stylization, etc. Digital data post-production is a common problem in any field involving signal processing and applications may thus range from computational photography to medical imaging or satellite imaging.

Image post-production is generally an ill-posed problem. Indeed recovering missing or damaged data cannot be done without any prior assumption. Inspired by the studies made on the visual human system, many methods rely on the predictability or redundancy of images. The notion of redundancy in natural images appeared more than fifty years ago in the context of human vision (Attneave, 1954; Barlow, 1961). According to these studies, the visual system seeks to represent an image with the smallest possible number of information. This is made possible as there exist statistical dependencies across space in any natural image. In order to be efficient, the human visual system reduces these redundancies by removing the statistical dependencies.

Different types of redundancies are present in natural images, each of them leading to different approaches for automatic image processing. For details on natural image statistics we refer the reader to (Simoncelli et al., 2001). In particular, we can differentiate spatial and spectral redundancy. Local spatial redundancy means that there is a strong correlation between neighboring pixels, *i.e.* predictability in local image neighborhood. Spectral redundancies means that there is a strong correlation among the colors or the spectrum within an image. Another important type of redundancy, which will be at the core of this document, corresponds to the self-similarity principle, meaning that an image contains a lot of repetitions of local information (see figure 0.1).

Image redundancy is exploited in many image processing and analysis tasks. For image and video compression, it is only natural to compress the data by removing spatial and temporal redundancy. For denoising, today most powerful methods replace noisy areas

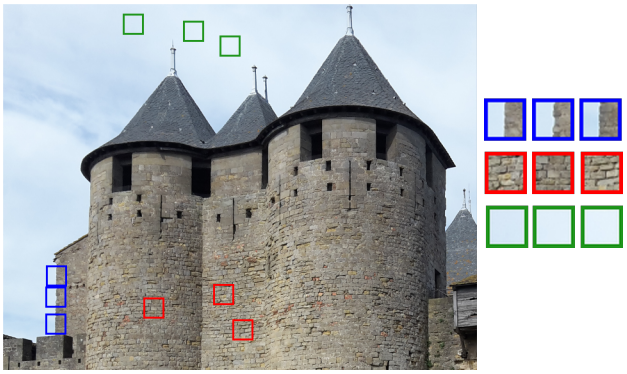


Figure 0.1: Illustration of the self-similarity principle. An image contains a lot of repetitions of local information.

by averaging the values of other similar areas in the same image, thus exploiting the self-similarity principle. In the context of image retrieval, object detection or action recognition, the goal is to retrieve images or areas containing similar representations. Methods usually rely on local spatial redundancy but the self-similarity principle also happens to be useful (Shechtman et al., 2007).

In the field of image post-production, methods exploiting the self-similarity principle are often referred to as patch-based approaches or non-local approaches. The assumption lying behind is that, similar to the human visual system, a computer is also able to predict missing data exploiting the visual redundancy present in a single image or across several similar images. All the works in this trend are inspired by Shannon's work on modeling the English language. In (Shannon, 1948), a Markov model is used to generate an English text by computing the probability distributions of each letter given the previous ones in a large English text. Then, starting from a given letter (the seed), successive letters are found by sampling this Markov chain. This idea has first been extended to images in (Efros et al., 1999) for the purpose of texture synthesis. The texture is modeled as a Markov Random Field assuming that the probability distribution of brightness values for one pixel given the brightness values of its spatial neighborhood is independent from the rest of the image. The neighborhood is a square window called a patch. As we will see all along this document this work is at the origin of many others in various applications. A non exhaustive list of improvements that have been proposed in the literature is: study of similarities between patches, higher-level representation of patches, order in which the pixels (seeds) are sampled, introduction of geometric constraints for better sampling, use of non square window. Despite these various improvements, most patch-based methods remain one-path greedy algorithm: pixels are processed one after the others without being further modified. This property leads to the well known growing garbage problem: errors are propagating from one pixel to the next. The only way to prevent this issue

from happening is to consider multi-pass (or iterative) algorithms. The difficulty becomes to demonstrate the convergence of such algorithm. In order to better understand the behavior and convergence of the algorithms, problems are generally modeled as the minimization of energy functions or solutions of partial derivative equations (PDE). Several energy functions will presented and discussed in this document.

While we will mention other applications of image post-production we will concentrate on image inpainting and image colorization. Inpainting aims at reconstructing a part of an image in a way that is not (easily) detectable by an ordinary observer. Image colorization consists in turning a grayscale image into a color one that must seems natural. Additionally to these two applications that concern the restoration and retrieval of natural images, we will also mention results concerning the completion of laser detection and ranging (LiDAR) clouds of points for orthoimages production and dense depth maps generation.

In chapter 1 we will see in details how the initial texture synthesis approach has been extended for the tackled applications. Chapters 2 and 3 study several models to solve the inpainting and colorization problems with non-greedy algorithms. Finally the last chapter concludes this presentation and opens several new perspectives of research.

Chapter 1

Exploiting the self-similarity principle for image post-production

Every natural image contains redundant information. The human visual system is able to use the redundant information within an image or from previously memorized images in order to predict the content of a missing part of the scene. In this chapter we will see how this behavior translates into algorithm for recovering missing data in the context of image post-production. We will concentrate on patch-based methods that consider the self-similarity principle, assuming that an image has a lot of repetitions of local information.

While there is no clear definition of self-similarity, it is generally understood as follows: if any small template in an image can be approximated by one or several other small templates in that same image, then the image is self-similar. The small templates are generally considered to be small square windows, also called patches, but other less regular shapes may also be considered. In both cases, they are small image parts that capture local color statistics, texture and structure information.

In the context of image restoration and enhancement, the exploitation of patch redundancy started with the seminal work by Efros et al., 1999, for texture synthesis. We will detail this method in section 1.1, and the metrics that can be used to compare patches in section 1.2. This work is at the origin of a huge number of research for various applications. A good survey for paper before 2009 is L.-Y. Wei et al., 2009, and for after is Barnes et al., 2017. Let us briefly mention some applications here, without going into details: denoising (with the famous NonLocal-Means algorithm from Buades et al., 2005), super-resolution (Freeman et al., 2002), stereo-matching (Scharstein et al., 2002; Lu et al., 2013), stylization (Bénard et al., 2013), style transfer (Elad et al., 2017; Frigo et al., 2016), segmentation and labeling (Coupé et al., 2011), optical flow estimation (Fortun et al., 2016), stereoscopic image inpainting (Hervieu et al., 2010; Hervieu et al., 2011) etc.

In this chapter, we will concentrate on applications to image inpainting (section 1.3) and colorization of grayscale images (section 1.4).

1.1 Patch-based texture synthesis

The seminal paper (Efros et al., 1999) presented a simple yet effective patch-based texture synthesis method. The objective is to generate a new texture image $u : \Omega \to \mathbb{R}^3$ from a small texture sample $u_S : \Omega^S \to \mathbb{R}^3$ (figure 1.1).

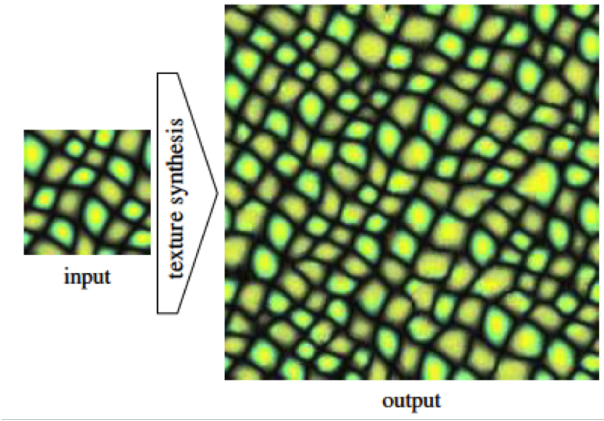


Figure 1.1: Texture synthesis: the objective is to generate a new texture image from a small texture sample. Figure taken from L.-Y. Wei et al., 2009.

The texture is modeled as a Markov Random Field assuming that the probability distribution of brightness values for one pixel given the values of its spatial neighborhood is independent from the rest of the image. The neighborhood is a square window (patch) around the pixel and its size is a global parameter of the algorithm. The input of the algorithm is a set of sample image patches (all patches entirely belonging to the sample texture domain Ω^S). The task is to select an appropriate patch within this set for each of the unknown pixels so as to predict their value. This is done by computing a distance between the known neighborhood of an unknown pixel and each of the sample patches. The unknown pixel is filled-in with the value of the center of the most appropriate patch.

In the following, let $u : \Omega \to \mathbb{R}^3$ be a color image (or $u : \Omega \to \mathbb{R}$ an intensity image) and u_S the sample grayscale or color image. We denote as $\Psi_s(p)$ a patch centered at pixel $p = (x, y)$, of size $N = (2s + 1) \times (2s + 1)$ such that

$$\Psi_s(p) = \{p + k,\ k \in P_s\}$$

with $P_s = \{(i, j), i = -s \cdots s, j = -s \cdots s\}$. For the sake of clarity we will equivalently

use the notation $\Psi_s(p)$ and $\Psi(p)$. We also denote as

$$\Psi_s^u(p) = (u(p_1), \cdots, u(p_N)) \ , \ p_i \in \Psi_s(p) \tag{1.1}$$

the vector of image values within the patch $\Psi_s(p)$.
As described in (Demanet et al., 2003), the texture synthesis problem as just described is akin to finding the *correspondence map* (also called nearest neighbor field, NNF, in Barnes et al., 2009), $\varphi : \Omega \to \Omega^S$, that associates each pixel of the target domain Ω of the image to a pixel from the sample domain Ω^S such that

$$\varphi(p) = \text{argmin}_{q \in \Omega^S} d\left(\Psi_s^u(p), \Psi_s^{u_S}(q)\right) \tag{1.2}$$

where d is a distance between the image value of patches $\Psi_s(p)$ and $\Psi_s(q)$ centered at p and q respectively. Details on possible distances are discussed in section 1.2.

There have been many improvements to the initial patch-based method from Efros et al., 1999. We here briefly describe some common tools which are later used in this document.

Patch-wise approaches The pixel-wise texture synthesis approach as just described is time consuming as synthesizing each pixel requires an exhaustive search among all pixels from the sample image. To speed up the process and better reproduce the input local structures, patch-wise methods synthesize entire patches instead of pixels (Efros et al., 2001; Kwatra et al., 2003; Criminisi et al., 2004; Lefebvre et al., 2005). The common default is the increasing amount of garbage regions and the necessity to add a blending stage to remove discontinuities between adjacent pacth copies.

Multi-resolution scheme Another trend for accelerating and improving texture synthesis algorithm is to incorporate it into a multi-resolution scheme. The unknown image is iteratively approximated using some guidance from coarse to fine levels (L. Wei et al., 2000; Drori et al., 2003).

Approximate Nearest Neighbor (ANN) Over the years, several other strategies have been proposed to accelerate directly pixel-wise approaches, mostly leading to approximate nearest neighbor search (ANN). Note that all the methods in this trend have been extended to a multi-resolution scheme. While it is possible to accelerate exact nearest neighbor algorithms by avoiding any useless calculation (Xiao et al., 2011), the search remains quite inefficient in comparison to ANN. Among ANN methods we can

mention tree-based techniques (L. Wei et al., 2000; Di Blasi et al., 2003; Olonetsky et al., 2012; He et al., 2012) or methods taking advantage of *spatial coherence*. These latter reduce the search space making the assumption that neighbor pixels in the output are likely to have neighbor correspondences in the input (Ashikhmin, 2001; Tong et al., 2002; Busto et al., 2010). The first such work (Ashikhmin, 2001) proposed to perform texture synthesis by looking for the best match for pixel $p \in \Omega$ not in all Ω^S but only among the set $\{\varphi(p+k)-k\}$ of shifted candidates. The idea, as stated by the author, was to increase performance by not "starting the search process from scratch at each new pixel". In practice it also imposes a certain *coherence* in the mapping function φ which clearly improves the visual quality of many synthesis results. This approach was extended to *k-coherence* by Tong et al., 2002.

Exploiting spatial coherence, Barnes et al., 2009, proposed the now famous approximate nearest neighbor algorithm called *patchmatch*. The assumption made is that the correspondence map is most likely to be piece-wise constant. The algorithm has three steps: 1) random initialization of the correspondane map 2) propagation exploiting spatial coherence 3) random search to avoid getting stuck into local minima. Its convergence rate to exact NN has been studied in (Arias et al., 2012).

1.2 Measuring similarity between patches

The underlying questions behind any patch-based methods are: *i)* how to find a discriminant and compact representation of the local statistics lying within a patch? *ii)* how to measure the similarity between these representations? In this section we only focus on the comparison between patches of pixels but other metrics have been designed for more complex templates. For instance, in (Giraud et al., 2017), we have designed metrics for patches of superpixels.

1.2.1 Patch-based descriptors

Many representations of patches have been proposed in the literature. Our goal here is not to make an exhaustive list but to mention the most famous ones and the ones used later in this document. It is important to highlight that the use of these representations highly depends on the targeted application, either we need to find color, texture or structure similarity.

The first low-level feature is the raw pixel intensity or color values themselves. This is the most widely used representation in patch-based methods.

In order to capture local color statistics, the mean and standard deviations of pixel values

within the patch is a very compact representation, though not highly discriminant. Higher dimensional representation are the intensity or color histograms, normalized histograms or cumulative histograms.

Several representations capturing frequencies of important structures can be obtained in the Fourier domain: Discrete Fourier Transform (DFT), Discrete Cosine Transfrom (DCT) for instance. The DFT is combined to other representation for image colorization in (Bugeau et al., 2012). In the same trend, Gabor filters have been used for image colorization in (Chia et al., 2011) and (Gupta et al., 2012).

A simple way to capture intensity or color variation within a template is to rely on image derivatives: intensity gradients and/or magnitude of gradients. Use of these features for image inpainting can be found in (Y. Liu et al., 2013) and (Newson et al., 2014). Higher order derivatives can also be considered. For instance region covariance (Tuzel et al., 2006) computed over the first and second intensity derivatives is used for colorization and color transfer in (Arbelot et al., 2016).

Finally, when measuring similarity across different images, it is often necessary to rely on scale, orientation and illumination invariant representations. Well known descriptors in this trend, built from gradient orientations, are SIFT (Lowe, 1999), SURF (Bay et al., 2006) or DAISY (Tola et al., 2008). These descriptors are well suited for matching highly structured image areas such as corners or keypoints. SIFT has been used for patch-based denoising (Lou et al., 2009) and colorization (Chia et al., 2011), and DAISY for image colorization (Cheng et al., 2015).

None of these representations is well suited for all possible types of textured, structured or uniform areas of natural images. This is mostly why, in the context of image restoration, most methods rely directly on raw pixel values.

In the following, the patch representation computed on $\Psi_s(p)$ is denoted by $\Psi_s^f(p)$ and is stored as a vector of size N:

$$\Psi_s^f(p) = (f(p_1), \cdots f(p_N)) \ , \ p_i \in \Psi_s(p). \tag{1.3}$$

For instance in case the representation are the raw pixels themselves, we have

$$\Psi_s^f(p) = \Psi_s^u(p) = (u(p_1), \cdots u(p_N)) \ , \ p_i \in \Psi_s(p).$$

1.2.2 Similarity metrics

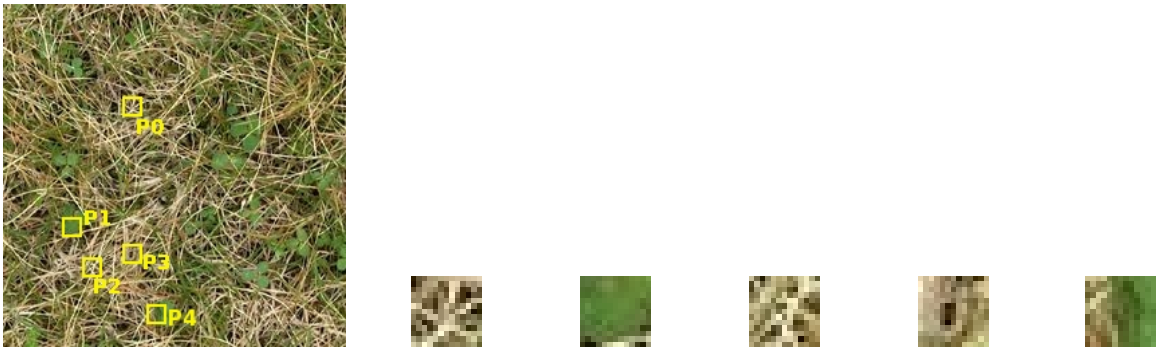
Many metrics exist to compute the similarity between patch representations. The most widely used for image restoration is the squared ℓ_2 norm of the difference between patches,

called sum of squared differences (SSD). For two pixels p and q, it is defined as:

$$d_{SSD}\left(\Psi^u(p), \Psi^u(q)\right) = \sum_{k \in P_s} \|u(p+k) - u(q+k)\|^2.$$

A Gaussian weighted version, d_{SSDG} is also sometimes considered (as in Efros et al., 1999) in order to give more importance to central pixels. Other classical measures include the ℓ_1 norm of the difference between patches, known as the sum of absolute differences (SAD), or zero normalized cross-correlation.

Before going any further, let us analyze the influence of the SSD on an inpainting example. Figure 1.2, taken from (Bugeau et al., 2010a), demonstrates the limitation of the SSD when comparing textures. The textured patch P_0 is, in its overall appearance, very similar to P_2. But, pixel-wise, as the SSD is computed, the difference between pixels at the same location in both patches is greater than the difference between the pixels in P_0 and the average value of P_0. Think of the difference between two sinusoids of average μ which are 180° out of phase: although they look exactly the same, their SSD is greater than the SSD between either sinusoid and the constant function of value μ. This is why patch P_4, which is rather uniform and does not resemble at all P_0, is more similar in terms of SSD to P_0 than P_2 is to P_0. This problem was later discussed in (Newson, 2014) with the following explanation. Assume two patches Q_1 and Q_2 made of i.i.d. pixels with a normal distribution $\mathcal{N}(\mu, \sigma^2)$ and a constant patch Q_3, with the constant equal to μ. The SSD patch distance between Q_1 and Q_2 follows a chi-square distribution $\chi^2(0, 2\sigma^2)$ while the one between Q_1 and Q_3 follows $\chi^2(0, \sigma^2)$. Patch Q_3 is therefore more likely to be chosen as the best candidate for patch Q_1.



a) Original image b) Patch P_0 c) Patch P_1 d) Patch P_2 e) Patch P_3 f) Patch P_4

Figure 1.2: Sum of squared differences between texture and smooth patches: $d_{SSD}(P_0, P_1) = 5290$; $d_{SSD}(P_0, P_2) = 5443$; $d_{SSD}(P_0, P_3) = 5416$; $d_{SSD}(P_0, P_4) = 5070$.

To cope with this problem, in (Bugeau et al., 2010a), we have defined a new similarity

measure:

$$d_{SSDB}\left(\Psi^u(p), \Psi^u(q)\right) := \frac{1}{N}\, d_{SSD}\left(\Psi^u(p), \Psi^u(q)\right) \cdot d_H\left(\Psi^u(p), \Psi^u(q)\right), \tag{1.4}$$

where, for a given pair of patches $\Psi_s(p), \Psi_s(q)$, $d_H\left(\Psi^u(p), \Psi^u(q)\right)$ denotes the Hellinger distance:

$$d_H\left(\Psi^u(p), \Psi^u(q)\right) = \sqrt{1 - \sum_{i=1}^{B} \sqrt{\rho_p(i)\rho_q(i)}}, \tag{1.5}$$

where ρ_p, ρ_q are the histograms of the image u computed on the patches $\Psi^u(p)$, $\Psi^u(q)$, respectively, and B is the number of bins of the histograms. The Hellinger distance is a modified version of the Bhattacharyya probability density distance so that it satisfies the triangle inequality. In practice a histogram is computed on each patch by using 8 bins for each color dimension (*i.e.* 512 bins in case of color images). Let us comment on the rationale behind this choice for d, which combines the SSD and the Hellinger distance. While the Hellinger distance permits to distinguish a smooth patch from a textured one because they have different distributions, it is rotation invariant and we still need the SSD to permit distinction in that case. As later noticed by (Le Meur et al., 2012), when two patches have the same distribution, their distance d_{SSDB} is zero. The authors therefore propose a variant to cope with this limit:

$$d_{SSDB}\left(\Psi^u(p), \Psi^u(q)\right) := \frac{1}{N}\, d_{SSD}\left(\Psi^u(p), \Psi^u(q)\right) \cdot \left(1 + d_H\left(\Psi^u(p), \Psi^u(q)\right)\right). \tag{1.6}$$

This metric is also well adapted for patch-based image denoising (Ebdelli et al., 2013).

As mentioned before, we do not wish to draft an extensive list of all existing similarity metrics. Let us however mention that studies for comparing noisy patches, assuming known the noise model, have been conducted (Deledalle et al., 2012).

As for patch representation, the quality of a similarity metric often depends on the types of data and desired application. This is why most recent papers for image inpainting and image colorization prefer to rely on several types of patch representations thus defining new metrics to compute similarity between patches. We will discuss this matter later in this document.

1.3 From patch-based texture synthesis to patch-based image inpainting

Inpainting is the art of filling an unknwon area of an image or a video in a form that is not (easily) detectable by an ordinary observer. It has become a fundamental area of research in image processing with different applications: restoring missing image blocks (error concealment) in telecommunications, removing scratches or dust in movies, adding or removing elements in images or movies, and image extension. It is an old problem in art restoration as medieval artwork started to be restored as early as the Renaissance (as underlined in Bertalmio et al., 2000). In the art domain, it is also called retouching. The problem was brought to the image processing community by (Masnou et al., 1998) where it was named image disocclusion. The term inpainting was invented for art restorations (Emile-Male et al., 1976) and was first used for digital image restoration in (Bertalmio et al., 2000).

Inpainting is an ill-posed inverse problem which does not have a unique solution. A good solution is one that is undetectable by a human observer. To solve the inpainting problem, some priors must be considered. The commonly assumed prior is that any pixel in an image share the same geometrical and color statistics than its surrounding pixels (local spatial redundancy). Depending on the other priors considered, image completion techniques can be divided into three basic groups: 1) examplar-based or patch-based methods exploiting the self-similarity principle. Recent survey describing this category can be found in (Buyssens et al., 2015); 2) geometrical or diffusion-based methods considering a smoothness prior (see section 2.1.1); 3) hybrid methods combining the two previous types of approaches. Recent surveys describing all these categories can be found in (Bertalmío et al., 2011; Guillemot et al., 2014; Newson, 2014). We only focus in this section on the first category: patch-based methods for image inpainting.

Patch-based image inpainting methods, first introduced in (Bornard et al., 2002), all extend the texture synthesis algorithm of Efros et al., 1999. The extension to inpainting is straightforward: given an image u defined on the set Ω with a hole $\mathcal{H}$, fill-in each pixel inside $\mathcal{H}$ with a value (or combination of values) taken from the known area of the image $\Omega^S = \Omega \setminus \mathcal{H}$ (see figure 1.3).

Filling order Generally, layers of the unknown mask are inpainted successively in an onion peel strategy. As first highlighted in (Drori et al., 2003), the order with which

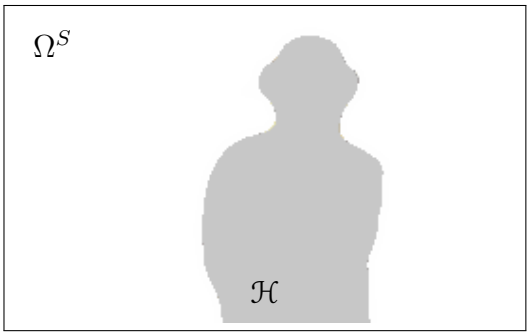


Figure 1.3: Known and unknown areas for inpainting. Left: original area, Right: known area in white and unknown area in gray.

the pixels are filled strongly influence the results. Criminisi et al., 2004, designed a clever ordering procedure with priorities depending on edge strength. The priority order at a pixel is the product of a confidence term, which measures the amount of reliable information surrounding the pixel, and a data term, that encourages linear structures to be synthesized first. The confidence term is the ratio of the number of known pixels divided by the total number of pixels in the patch. The data term is defined as

$$D(p) = \frac{|\nabla u^{\perp}(p).n_p|}{\alpha} \tag{1.7}$$

where $\nabla^{\perp}$ is the orthogonal gradient, α a normalization factor (equal to 255 for grayscale images) and n_p a unit vector orthogonal to the inpainting area. It encourages the linear structures to be synthesized first and depends on the isophotes (contours) that eventually pass by p.

In (Bugeau et al., 2009), we have proposes a new data term based on the structure tensor (Di Zenzo, 1986) computed on the structure image after texture/structure decomposition (Vese et al., 2003). Stucture tensors were also used in the geometry-aware data term proposed in (Buyssens et al., 2015). In (Le Meur et al., 2011), the data term is modified to rely on structure tensor, which better captures the local geometry. In (Xu et al., 2010), a sparsity-based priority has been proposed in order to further improve the data term from (Le Meur et al., 2011).

Post-processing Despite all the great improvements made to patch-based inpainting (see chapter 2 for details on more advanced techniques), a common remaining problem concerns illumination discontinuities at the border of the mask. This problem may have two reasons: there are no candidate patches of the right color or the metric used is not adapted. To deal with this defect, as in (Sun et al., 2005), we have proposed in (Bugeau et al., 2010a) to add a post-processing step of Poisson image editing (Pérez et al., 2003).

The final colors are obtained by solving:

$$\min_{\hat{u}} \int_{\mathcal{H}\cup\partial\mathcal{H}} \|\nabla\hat{u} - \nabla u(\varphi(p))\|_2^2 \text{ with } \hat{u}|_{\partial\mathcal{H}} = u|_{\partial\mathcal{H}}$$

where $\partial\mathcal{H}$ is the outer boundary of $\mathcal{H}$.

Other post-processing approach relies on color histogram equalization, which is a well known tool for contrast enhancement (Kim, 1997; Nikolova et al., 2014; Pierre et al., 2016). Nevertheless, as histograms do not include any information on the spatial repartition of colors, their application to local image editing problems remains limited. To cope with this lack of spatial information, spatiograms have been proposed for tracking purposes (Birchfield et al., 2005). A spatiogram is an image descriptor that combines a histogram with the mean and variance of the position of each color. In (Papadakis et al., 2012), we have addressed the problem of local retouching of images by proposing a variational method for spatiogram transfer. More precisely, a reference spatiogram, computed on Ω^S is used to modify the color values within $\mathcal{H}$ of a given region of interest of the processed image. From figure 1.4, we can see that the spatiogram transfer allows correcting the default of patch-based inpainting results while preserving the textures. This combination produces really plausible reconstructions (see Fig. 1.5(c) for details on the tree example).

1.4 From patch-based texture synthesis to patch-based image colorization

Image colorization consists in recovering a color image from a grayscale one. This application attracts a lot of attention in the image-editing community in order to restore or colorize old grayscale movies or pictures. While turning a color image into a gray-scale one is only a matter of standard, the reverse operation is a strongly ill-posed problem as no information on which color has to be added is known. Therefore priors must be added. In literature, there exist two kinds of priors leading to two different types of colorization methods. In the first category, initiated by Levin et al., 2004, the user manually adds initial colors through scribbles to the grayscale image. The colorization process is then performed by propagating the input color data to the whole image (section 3.1). The second category, called automatic or patch-based colorization, consists in transferring color from one (or many) initial color image considered as example.

For both, the initial grayscale image is considered as the luminance channel which is not

(a) (b) (c)

Figure 1.4: Patch-based inpainting correction with spatiogram transfer. (a) Original image and the inpainting mask in red. (b) Result obtained with (Criminisi et al., 2004) using patches of size 9×9. (c) Post-processing result obtained in (Papadakis et al., 2012).

(a) (b) (c)

Figure 1.5: Patch-based inpainting correction with spatiogram transfer (Papadakis et al., 2012). (a) Zoom on the original image of the second row of figure 1.4 (b) Zoom on the patch-based inpainting result. (c) Zoom on the corresponding spatiogram transfer correction which produces really plausible reconstructions .

modified during the colorization. The objective is then to reconstruct the two chrominance channels, before turning back to the RGB color space. Different luminance-chrominance spaces exist and have been used for image colorization: $l\alpha\beta$ (Ruderman et al., 1998) as in (Welsh et al., 2002), *Lab* as in (Charpiat et al., 2008), YCbCr as in (Yatziv et al., 2006), or YUV as in (Bugeau et al., 2012). In all of our works, we have used the YUV space

since its conversion to RGB is linear.

1.4.1 General pipeline for patch-based image colorization

The first patch-based colorization method was proposed by Welsh et al., 2002. It makes the assumption that pixels with similar intensities or similar neighborhood should have similar colors. It extends the texture synthesis approach by Efros et al., 1999: the final color of one pixel is copied from the most similar pixel in the input color image. The similarity between pixels relies on patch-based metrics. This approach has given rise to many extension in the literature.

In the following, we denote as

- $u_T : \Omega \to \mathbb{R}$ the target grayscale image (which is assumed to be a luminance image)
- $u : \Omega \to \mathbb{R}^3$ the estimated color image, $Y_u : \Omega \to \mathbb{R}$ its luminance (that must be equal to u_T), $u_c = (U_u, V_u) : \Omega \to \mathbb{R}^2$ its chrominances in the YUV color space, and R_u, G_u, B_u its channels in the RGB color space.
- $u_S : \Omega^S \to \mathbb{R}^3$ the source color image.

With these notations, the goal of image colorization is to find u under the constraint that it luminance channel Y_u is equal to u_T. Most patch-based methods for image colorization proposed in the literature can be summarized into a three-steps general pipeline.

Step–1, Pre-Process u_S and u_T

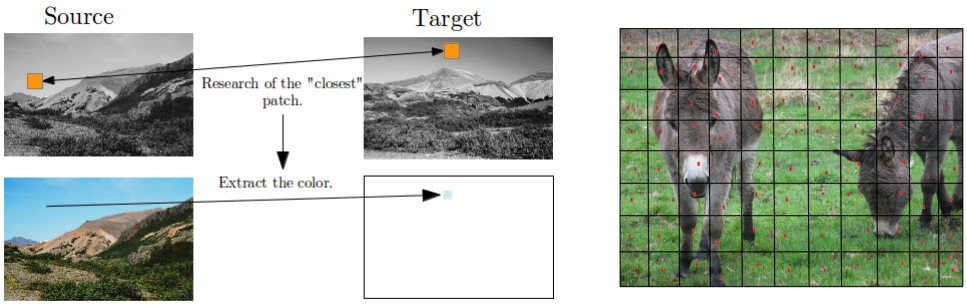
This step consists in converting u_S to a luminance-chrominance space and setting $Y_u = u_T$. Automatic patch-based image colorization methods rely on a copy/paste of the chrominances from the source image to the target image. In order to select the color to transfer, the luminance of u_S (denoted by Y_{u_S}) and u_T should be comparable. Ideally, the transfer process should take into account the global difference in luminance of the two images, *i.e.*, these two images should have similar global statistics such as the mean and the variance. For this purpose, an affine luminance mapping (Hertzmann et al., 2001) is generally performed on Y_{u_S}.

Step–2, Predict Color from u_S and transfer to u

The second step aims at choosing for each pixel $p \in \Omega$ the best pixel $\varphi(p) \in S_n$ which minimizes a distance on the luminance neighborhoods:

$$\varphi(p) = \arg\min_{q \in S_n} d(\Psi^{Y_u}(p), \Psi^{Y_{u_S}}(q)) \ .$$

The process is illustrated on figure 1.6. The set $S_n = \{q_1, \ldots, q_n\}$ of n possible candidate pixels is either the whole pixel grid Ω^S of the source image or a subset of this grid



(a) Search of the candidates.

(b) Subsampling on a regular grid.

Figure 1.6: Illustration of color prediction. For each pixel of the target image, we search for the closest patch in the luminance channel of the source image and copy the chrominances of the central pixel of this patch onto u (see (a)). To speed up the algorithm, in (Welsh et al., 2002) the search is not performed among all pixels, but only on a subsampling (see (b)).

($n \ll \Omega_S$). In (Welsh et al., 2002; Bugeau et al., 2012; Bugeau et al., 2014; Pierre et al., 2014b), approximately $n = 200$ source pixels are selected on a randomly jittered grid (figure 1.6 (b)). Any other acceleration methods for patch search (see section 1.1) can also be used. For instance, in (Pierre et al., 2017a), we rely on patchmatch (Barnes et al., 2009).

Finally, the chrominances of $\varphi(p)$ are transferred to p:

$$u(p) = (Y_u(p), U_{u_S}(\varphi(p)), V_{u_S}(\varphi(p))).$$

In (Welsh et al., 2002), the distance is based on the average of the standard deviation on a 5×5 neighborhood and the luminance of the pixel. In (Charpiat et al., 2008), the color prediction is based on machine learning techniques (either SVM or Parzen windows), discrete optimization, continuous color refinement and image descriptors to capture textures or complex structures.

Step–3, Post-Process u_T

The last step generally consists in a simple color space re-conversion, *i.e.*, from the luminance-chrominance to the RGB color space.

1.4.2 Influence of patch representation and metrics

Natural images are composed of different types of complex structures, redundancies and textures, which are not well captured by the SSD distance only nor by simple statistics as the luminance itself or the standard deviation. In (Bugeau et al., 2012) and our derived work on image colorization (see chapter 3), we have studied the combination of different patch-based features and their associated metrics to better find good candidates. Let us here present and compare the three features used in our works.

While many metrics to compare patches exist (see section 1.2), in (Bugeau et al., 2012) we only concentrated on 3 different features f_l and the corresponding distances d_l with $l \in L = \{1, 2, 3\}$. Each of these three features is computed for different size s of patches. The choice of the metrics and patch descriptors has been made experimentally but they all capture a different characteristic of the grayscale image. In practice, we chose 8 measures, as detailed now. Given a luminance image $Y : \Omega \to \mathbb{R}$, for any pixel $p = (x, y) \in \Omega$, we compute:

RR1. the variance

$$f_1(p, s) = \nu(\Psi^Y(p))$$

which differentiates constant and textured areas. In the experiments, we used two size of patches for this feature: 3, 5.

RR2. the amplitude spectrum of the $2D$ discrete Fourier transform

$$f_2(p, s) = \| \mathrm{DFT}(\Psi^Y(p)) \|_2$$

where $\|.\|_2$ is the amplitude of the complex number. This feature gives information on the high frequencies of the most important structures. In the experiments, we used three size of patches for this feature: 7, 9, 11;

RR3. the cumulative histograms containing H bins with $\{h_1, \ldots, h_H\}$ and $\forall i = 1, \ldots, H$

$$f_3(p, s, h_i) = \sum_{j=1}^{i} \sum_{q \in \Psi_s(p)} \frac{\delta(Y(q), h_j)}{|\Psi_s(p)|}$$

where $\delta(a, h_j) = 1$ if the luminance a belongs to the j-th bin h_j and $\delta(a, h_j) = 0$ otherwise. This cumulative histogram takes into account the luminance distribution inside the patch. In the experiments, we used three size of patches for this feature: 7, 9, 11.

From the features space, we define the associated distances d_l with $l = \{1, 2, 3\}$ such that

for two pixels p, q, we have:

$$
\begin{aligned}
d_1(p,q,s) &= |f_1(p,s) - f_1(q,s)| \ , \\
d_2(p,q,s) &= \sum_{\xi} |f_2(p,s,\xi) - f_2(q,s,\xi)| \text{ and} \\
d_3(p,q,s) &= \sum_{h=1}^{H} |f_3(p,s,h) - f_3(q,s,h)|.
\end{aligned}
$$

As we are comparing 1-dimensional grayscale images, we can notice that the ℓ_1 distance d_3 on the cumulative histograms is equivalent to the ℓ_1 Wasserstein distance between the distributions. This metric is therefore robust to illumination changes.

Figure 1.7 demonstrates experimentally that the same feature is not adapted to colorize all kind of images. A patch-based colorization process similar to (Welsh et al., 2002) is here applied with the different metrics: variance, discrete Fourier transform, Wasserstein and SSD distances. The first feature is computed with 3×3 patches and the three others with 11×11 patches. This experiment demonstrates the interest of combining different features in patch-based methods. We can also observe that the SSD is not well suited for all images as it requires patches to be similar and aligned. As we here only consider $n = 200$ samples, the odds of getting similar source and target patches are very small. That is the reason why we did not considered this metric in our future works.

Finding an optimal metric

In (Pierre et al., 2015a), we have studied the possibility of designing an optimal metric as a linear combination of standard pre-defined metrics. The user provides a set of labeled source images, as well as the initial metrics. Distances between all patches extracted from the source images are first computed. Support vector machines (SVM) are next used to compute the coefficients of the linear combination between the different metrics. Results of this methods are promising. One limitation is that the optimal metric has to be learned again each time a new type of texture is present in an image.

1.4.3 Combining different patch representations to improve patch-based image colorization

As there is no optimal metric for all kinds of images, some authors have proposed to combine several metrics in their image colorization process. Unfortunately, as the metrics are of different nature, they are generally not directly comparable and must be combined in a clever way. Welsh et al., 2002 use the average of the standard deviation and the

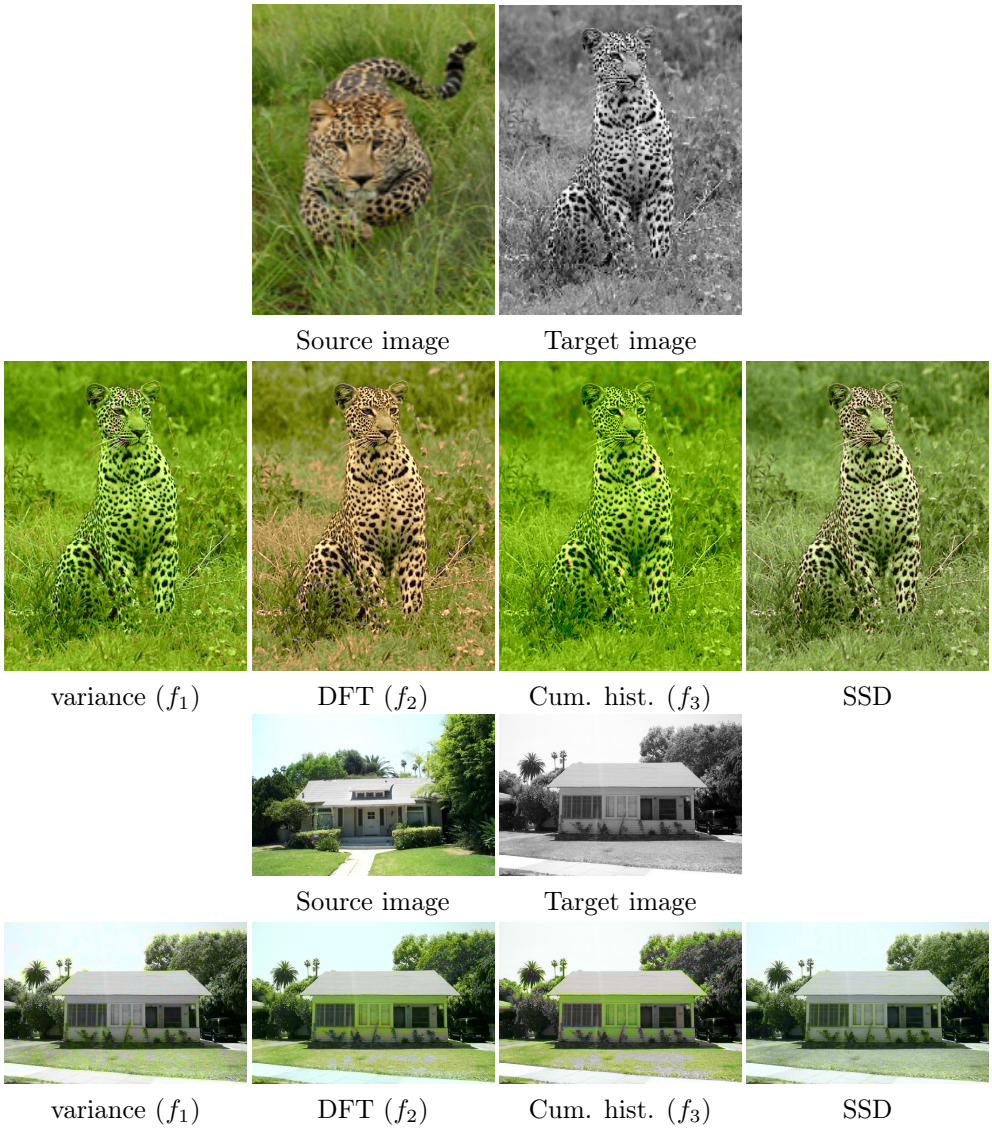


Figure 1.7: Influence of the features on the colorization result.

luminance at each pixel. Gupta et al., 2012 use a combination of distances computed from different features (Gabor, SURF (Bay et al., 2006), mean intensity and standard deviation) extracted from a superpixel segmentation. Arbelot et al., 2016 merge luminance and its first and second order derivatives into a covariance matrix.

Contrary to previous approaches in image colorization, in (Bugeau et al., 2012), we proposed not to predict one unique chrominance per pixel but several. From the features and metrics introduced in previous section, we can define, for each pixel, the set of chrominances from which the final chrominance will be computed. For each point $p \in \Omega$, each

feature l and each size s, we select the candidate $q_{l,s}$ such that:

$$q_{l,s} = \underset{q \in S_n}{\operatorname{argmin}}\, d_l(p, q, s),$$

leading to the chrominance set

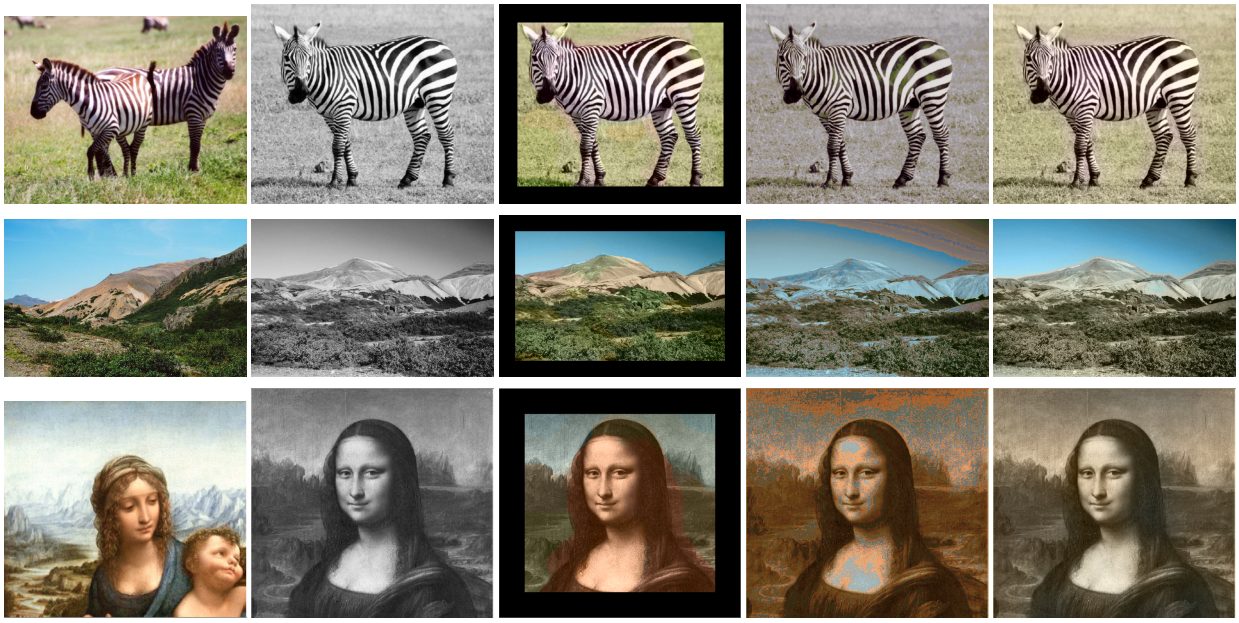
$$C(p) = \{(U_{u_S}(q_{l,s}), V_{u_S}(q_{l,s}))\}\,.$$

For the sake of clarity, we rewrite this set as:

$$C(p) = \{c_i(p) = (U_{u_S}(q_i), V_{u_S}(q_i))\,, \forall i = 1, \ldots, N\}$$

where N is the total number of candidates (8 in our case). We now have a reduced set of possible chrominances for each pixel p of the grayscale image.

In (Bugeau et al., 2012), we select the best chrominance among $C(p)$ as the median value on the main axis of the PCA of $C(p)$. Once the chrominance for each pixel of u has been found, we apply a post-processing step that regularizes the result with total variation.



(a) (b) (c) (d) (e)

Figure 1.8: Some colorization results. (a): source image u_S (b): target image u_T (c): (Charpiat et al., 2008) (results from the website of the authors) (d): (Welsh et al., 2002) (e): (Bugeau et al., 2012)

Figure 1.8 compares our approach with (Welsh et al., 2002) and (Charpiat et al., 2008). Our method globally leads to result of the same quality as the ones with (Charpiat et al.,

2008), with a much simpler approach. As shown in Figure 1.9 and also on the Joconde images, our approach avoids color bleeding along contours thanks to the smoothing effect of regularization post-processing. For the images presented in Figure 1.8(d), the method of (Welsh et al., 2002) clearly produces inconsistent colorization.

Charpiat et al., 2008 Bugeau et al., 2012

Figure 1.9: Zoom on the zebra image from figure 1.8.

1.5 Conclusion and need for non-greedy patch-based methods

In this section we have introduced patch-based methods for texture synthesis, image inpainting and image colorization. Patch-based methods for image inpainting are simple to implement while providing very good results on many different type of images. Automatic patch-based colorization is also simple in essence and able to correctly colorize most grayscale images.

Metrics to compare features of patches have been introduced and we have detailed their use in inpainting and colorization. As we have seen in section 1.4, no optimal metric exists to capture both structure and texture details of all natural images. To cope with this problem, it is often necessary to post-process the results of patch-based algorithms or to propose methods able to combine different metrics as we have done for image colorization.

(a) Original (b) Criminisi et al., 2004 (c) Zoom on (b)

Figure 1.10: Illustration of verbatim copy and contour discontinuities arising with one-pass greedy patch-based algorithms.

All patch-based methods discussed in this section, are one-pass algorithms: once a pixel is filled, its value remains unchanged. These techniques then suffer from common problems with greedy algorithms, and in particular from the filling order which is often critical. This is reflected for example in the propagation of wrongly reconstructed pixels, often observed as growing garbage, verbatim copy or discontinuities in strong structure (see figure 1.10). While some studies propose clever filling orders, the associated algorithms remain greedy and lead to unrealistic images. Many methods, starting with (Demanet et al., 2003) and (Kwatra et al., 2005) propose iterative optimization algorithms of discrete or variational formulation of the problem in order to address these limitations. This mater will be the topic of chapters 2 and 3 for inpainting and colorization respectively.

Chapter 2

Patch-based models for image inpainting

In previous chapter, we have introduced patch-based methods for texture synthesis, image inpainting and image colorization. In this section, we will concentrate on the inpainting application only. Besides patch-based inpainting methods, another trend to tackle this problem is to rely on "geometric" methods. This category represents PDE and variational models (section 2.1.1). They are well suited to reconstruct structures but fail for textured areas. For this reason, several discrete and variational models, combining both types of approaches, have been proposed over the past ten years. This chapter will give an overview of these approaches. Section 2.1 presents geometric models and provides a qualitative comparison with patch-based approaches. Additionally, we detail how these techniques can be used for inpainting LiDAR point clouds. In section 2.2, we concentrate on the modeling of patch-based inpainting problem from the point of view of energy minimization, that provides non greedy patch-based algorithms.

2.1 Patch-based vs. “Geometric” methods

We will start in this section by presenting "geometric" methods for image inpainting and showing qualitative comparisons with patch-based methods (section 2.1.1). Applications to completion of LiDAR point clouds are next explained in section 2.1.2.

2.1.1 Geometric methods

Geometric methods for image inpainting are based on color diffusion using Partial Differential Equations (PDE's) and variational formulations. They have been very successfully

used, in particular for piecewise smooth images or when the hole $\mathcal{H}$ is thinner than the surrounding objects. The first work in this area was by Masnou et al., 1998, whose algorithm performs inpainting by joining with geodesic curves the points of the level lines (iso-value contours) arriving at the boundary of $\mathcal{H}$. They were inspired by the work of Nitzberg et al., 1993, who, in the context of image segmentation, were looking for completion curves in a missing region $\mathcal{H}$ which should be as short as possible and should respect Gestalt's principle of good (smooth) continuation (the human visual system completes missing curves with curves that are as short and as smooth as possible). In order to join with a curve C two points p, q lying on the boundary $\partial\mathcal{H}$ of $\mathcal{H}$, Nitzberg et al., 1993, proposed to choose the curve minimizing an energy functional called Euler's elastica (Mumford, 1994). As its minimization leads to a fourth-order PDE, Ballester et al., 2001 proposed a new functional which is a relaxation of the Elastica. The basic idea is that the level lines of the image are smoothly transferred from a band around $\mathcal{H}$ toward the inside of $\mathcal{H}$, while keeping the image values and their gradient direction compatible. Many other functionals or PDE have been proposed (Chan et al., 2001a; Chan et al., 2001b; Chan et al., 2002)In practice, these methods are extremely computationally consuming since convergence requires a high number of iterations.

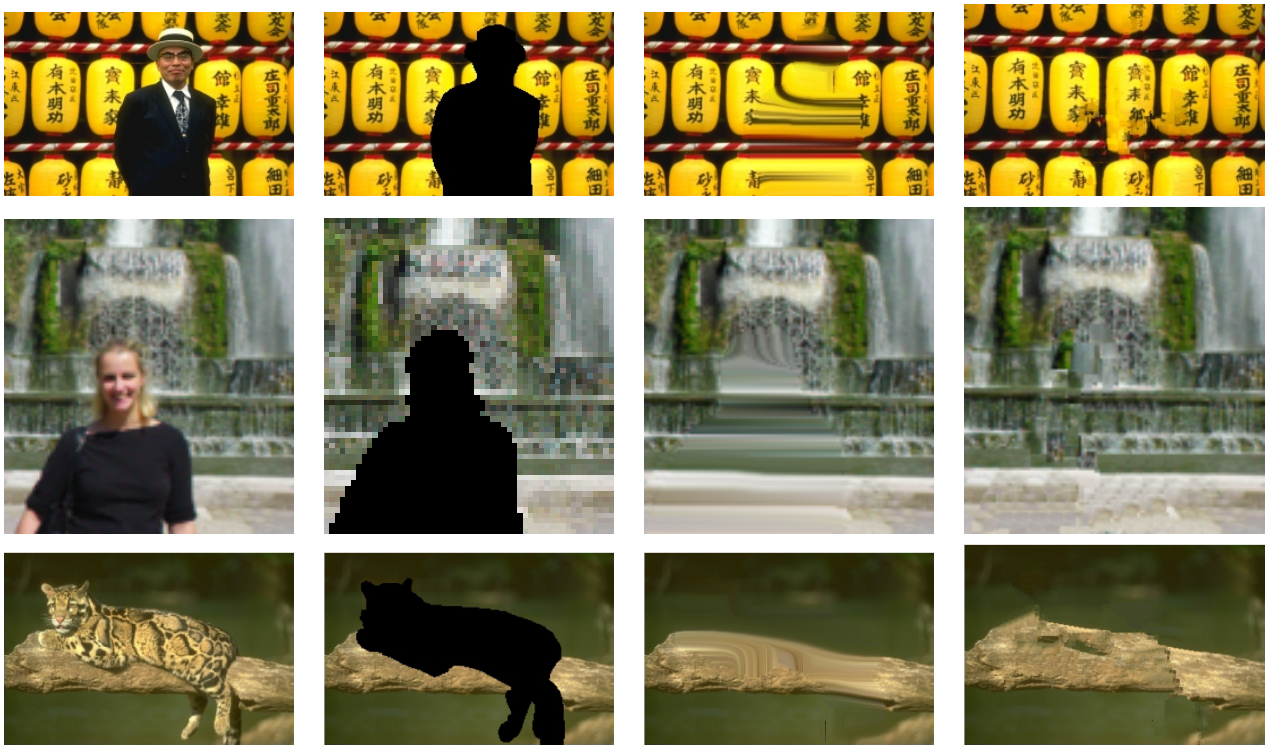
All previous PDEs where based on curvature diffusion. Other PDEs, also relying on the isophotes exist. In (Bertalmio et al., 2000), a third order PDE, solved only inside $\mathcal{H}$ with proper boundary conditions on $\partial\mathcal{H}$, was introduced. Its purpose is to propagate image Laplacians in the isophote directions. The edges recovered with this approach are smooth and continuous at the boundary of the hole. Later, in (Bertalmio, 2006), a third-order PDE complying with the principle of good continuation was proposed. This PDE does not well permit to handle images containing textures and its high order makes it computationally expensive.

Another PDE method dedicated to anisotropic smoothing was developed by Tschumperlé, 2006. Its principle is to retrieve the geometry of the main structures inside the hole using a structure tensor field, and to apply anisotropic diffusion following this geometry.

The global resolution of all the PDEs previously presented is made iteratively, which makes these methods very time consuming. A non iterative method, based on coherence transport, was proposed by Bornemann et al., 2007. It calculates the image values inside the masks by traversing these pixels in just a single pass using the fast marching technique. The mask is filled in a fixed order, from its boundary inwards, using weighted means of already calculated image values. The weights are defined such that the image information is transported in the direction of the isophotes.

The underlying idea behind geometric approaches is to smoothly transport the contours of

the image into the region being inpainted. They are thereby very efficient at reconstructing smooth contours, but for the same reason fail in textured parts of the image. Figure 2.1



Original | Mask | Tschumperlé, 2006 | Criminisi et al., 2004

Figure 2.1: Patch-based vs. "Geometric" methods. These results illustrate the ability of patch-based methods to reconstruct textures and of geometric methods to reconstruct strong structures

presents a comparison between a geometric method (Tschumperlé, 2006) and a patch-based method (Criminisi et al., 2004). We observe the properties that were already mentioned: geometric methods permit to obtain smooth contours and structures but are not able to reconstruct the textures. When looking at the patch-based approach in the fourth column, we observe the opposite, textures are pretty well reconstructed but edges are not continuous.

2.1.2 Application to completion of 3D LiDAR point clouds

In (Biasutti et al., 2017b; Biasutti et al., 2018), we have used a geometric method to reconstruct missing LiDAR points. Let us briefly introduce the context of this work before presenting the proposed reconstruction method.

Over the past decade, street-based Mobile Mapping Systems (MMS) have encountered a large success as the onboard 3D sensors are able to map full urban environments with

a very high accuracy. These systems are now widely used for various applications from urban surveying to city modeling. Several systems have been proposed in order to perform these acquisitions. They mostly consist in optical cameras, 3D LiDAR sensor and GPS combined with Inertial Measurement Unit (IMU), put on a vehicle for mobility purposes (Paparoditis et al., 2012, figure 2.2).

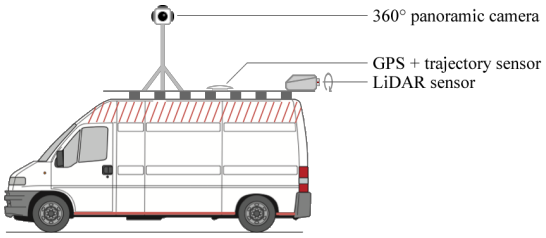


Figure 2.2: Scheme of the Stereopolis-II vehicle (Paparoditis et al., 2012)

LiDAR is a technique used to create a 3D point cloud of an object, a room or even an open area such as urban environment, fields... A sensor is used to illuminate a target by emitting a laser beam and measuring the position of the target point and the reflectance of the beam. The operation is repeated many times while variating the angle of emission and by shifting the sensor in order to create a very accurate (subcentimetric) 3D mapping of the target object. Typically, the sensor is put on a vehicle following the road at a relatively constant speed. Coupling with the MMS trajectory and calibration, the resulting point cloud provides several components such as the *position* (3D coordinates in both the moving sensor frame and a georeferenced earth-fixed frame), the *range* (distance between the measured point and the sensor) and the $reflectance$ (estimated albedo at the wavelength of the laser).

Although MMS lead to very complete 3D mappings of urban scenes by capturing optical and 3D details (pavements, walls, trees, etc.), they often acquire mobile objects that are not persistent to the scene (cars, pedestrians, traffic cones, etc). As LiDAR sensors cannot penetrate through opaque objects, those mobile objects cast shadows behind them where no point has been acquired (Figure 2.3, left). These shadows are also largely visible when the point cloud is not viewed from the original acquisition point of view. They are therefore distracting and confusing for visualization. Thus, the segmentation of mobile objects and the reconstruction of their background remain strategic issues in order to improve the understability of urban 3D scans.

Instead of working directly on the 3D point cloud, making use of the sensor topology, we have proposed in (Biasutti et al., 2017b; Biasutti et al., 2018), a new 2D representation of this type of data, called range image (figure 2.4). LiDAR points are ordered along scanlines, yielding the first dimension of the sensor topology, linking each LiDAR pulse to

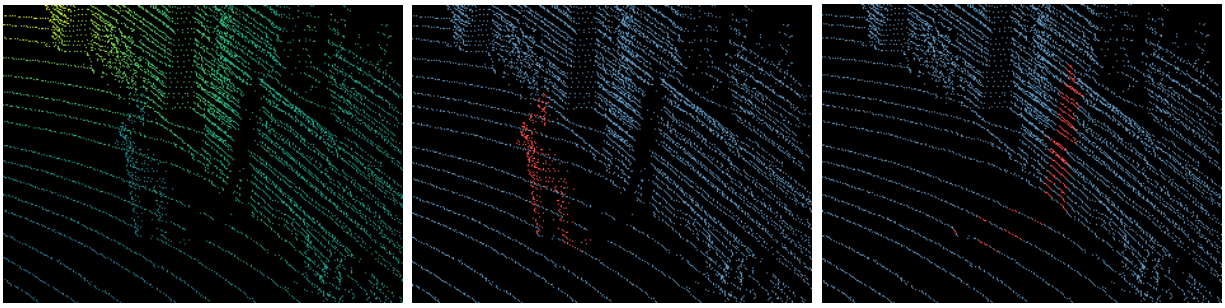


Figure 2.3: Disocclusion of a LiDAR point cloud. (left) original point cloud, (center) segmentation, (right) disocclusion.

the immediately preceding and succeeding pulses within the same scanline. The second dimension is obtained by ordering the consecutive scanlines, thus corresponding to a temporal dimension. The value at each pixel of the image is the range of the corresponding 3D point. This generated range image can now be processed with image processing algorithm.

Figure 2.4: Range image computation using sensor topology. (left) Sensor topology, (center) Acquired point cloud (Geiger et al., 2013), (right) Corresponding range map.

As a first application, we have proposed to use a *a contrario* method (Delon et al., 2007) to automatically segment objects according to their depth. Next, we tackled the problem of removing one object of the scene. As range images do not contain texture, we decided to rely on a geometric approach by diffusing the range information within the missing area. In urban environment the structures that require disocclusion are likely to evolve horizontally, *i.e.* smoothly along the x_W and y_W axis of the real world as defined in figure 2.5. Let us define $\vec{\eta}$, for each point, to be a unitary vector orthogonal to the projection of z_W in the range image u. This vector will define the direction in which the diffusion should be done to respect this prior of horizontal structures. Therefore, in our inpainting method, we propose to extend the level lines of u along $\vec{\eta}$. This can be expressed as $\langle \nabla u, \vec{\eta} \rangle = 0$. Therefore, we define the energy $F(u) = \frac{1}{2}(\langle \nabla u, \vec{\eta} \rangle)^2$. The disocclusion is then computed as a solution of the minimization problem $\min_u F(u)$. The gradient of this energy is given by $\nabla F(u) = -\langle (\nabla^2 u)\vec{\eta}, \vec{\eta} \rangle = -u_{\vec{\eta}\vec{\eta}}$, where $u_{\vec{\eta}\vec{\eta}}$ stands for the second order

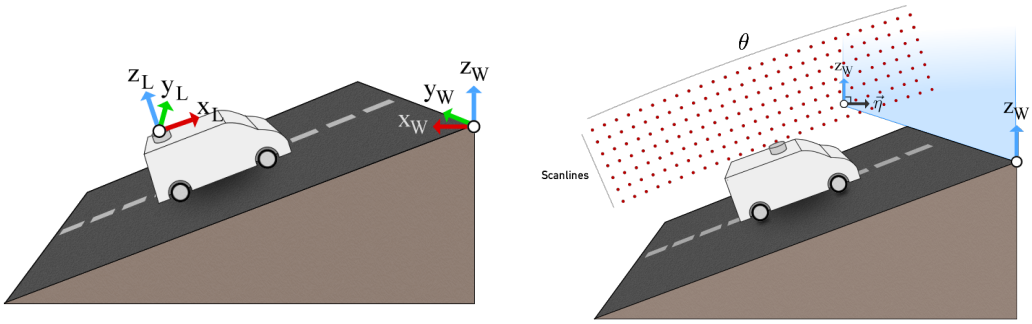


Figure 2.5: Definition of the different coordinates systems (left) and $\vec{\eta}$ (right).

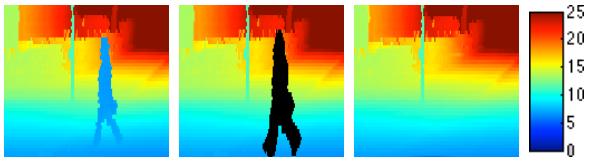
derivative of u with respect to $\vec{\eta}$ and $\nabla^2 u$ for the Hessian matrix. The minimization of F can be done by gradient descent. If we cast it into a continuous framework, we end up with the following equation to solve our disocclusion problem:

$$\begin{cases} \frac{\partial u}{\partial t} - u_{\vec{\eta}\vec{\eta}} = 0 \text{ in } \mathcal{H} \times (0, T) \\ u(0, x) = u_0(x) \text{ in } \mathcal{H} \end{cases} \tag{2.1}$$

where $\mathcal{H}$ is the area of the objects to remove and u_0 is the initial range image with the hole.

Inpainting of a pedestrian is shown in figure 2.6. This result is obtained for a sparse point cloud ($\approx 10^6$ pts) of the KITTI database (Geiger et al., 2013). We can observe that the disocclusion performs very well as the pedestrian has completely disappeared and the result is visually plausible in the inpainted range image.

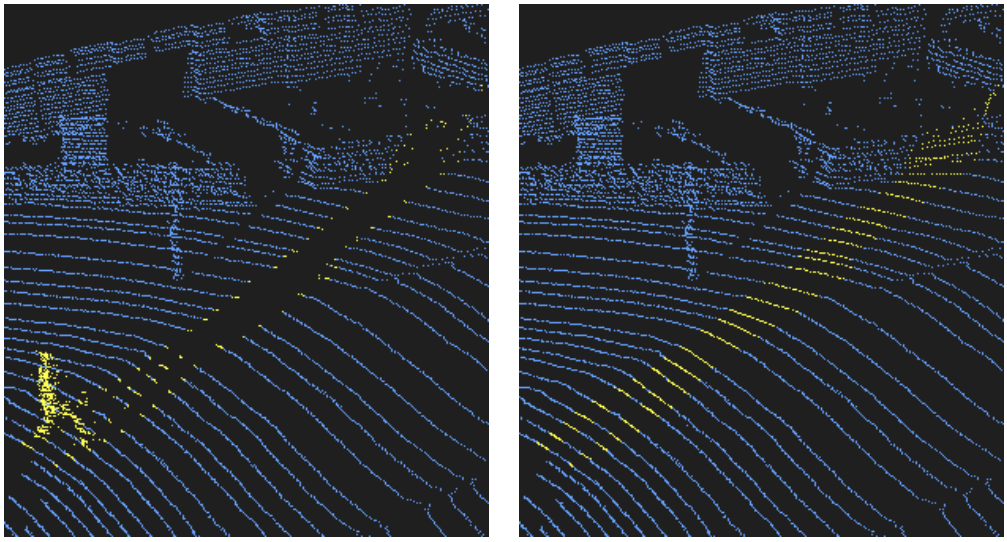
The coherence of the reconstruction can be checked by looking how the acquisition lines are connected in 3D (figure 2.7). Note that a similar analysis can be done on the results presented in Figure 2.3. For more qualitative and quantitative analysis, we refer the reader to (Biasutti et al., 2018).



a. b. c.

Figure 2.6: Result of disocclusion of a pedestrian on the KITTI database (Geiger et al., 2013). (a) is the original range image, (b) the segmented pedestrian (dark), (c) the final disocclusion. Depth scale is given in meters.

We have here illustrated an application, different from natural image inpainting, for which



a. b.

Figure 2.7: Result of the disocclusion of a pedestrian in 3D. (a) is the original mask highlighted in 3D, (b) is the final reconstruction.

geometric methods have a strong interest. In next paragraph, we will show that, even in the context of LiDAR point clouds inpainting, it is sometimes suitable to combine geometric methods with patch-based ones.

2.1.3 Application to the production of orthophotographies from LiDAR point clouds

In this section, we tackle another application related to LiDAR point clouds completion, namely the production of orthophotographies. Orthophotography and Digital Terrain Model (DTM), are defined respectively as the color and ground height orthoimages (*i.e.* raster maps defined on a regular horizontal grid). They are widely used in many application fields such as remote sensing, geographical information and earth observation, mapping, environmental studies... They are often computed from aerial acquisition devices such as satellites, aircrafts and more recently Unmanned Aerial Vehicles (UAVs). Although these techniques provide a very well known and common approach to the problem of orthoimage generation, they are limited in terms of accuracy and resolutions, and suffer from occlusions that are caused by natural and urban environments such as trees, tunnels, tall buildings, etc (figure 2.11 top). Using a Mobile Laser Scan (MLS) coming from a mobile mapping system as presented in previous section can solve these issues. Orthoimage generation from MLS has been scarsely studied in the past. Vallet et al., 2015, first extract the ground points by considering the lowest points projected in each pixel. Next, they perform Gaussian diffusion to connect sparse pixels. In (Biasutti et al., 2017a), we have

proposed a method to produce high resolution orthoimaging for reflectance and height. The proposed framework is summed up in figure 2.8.

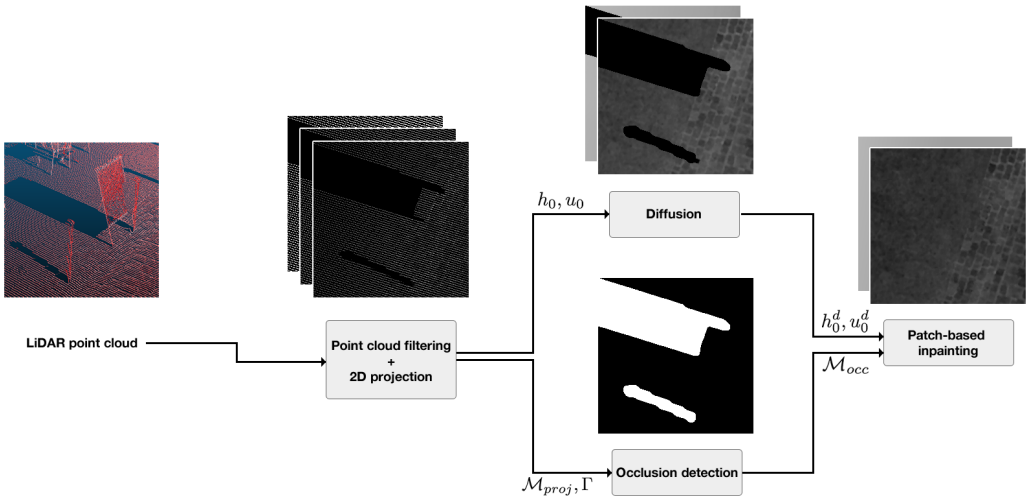


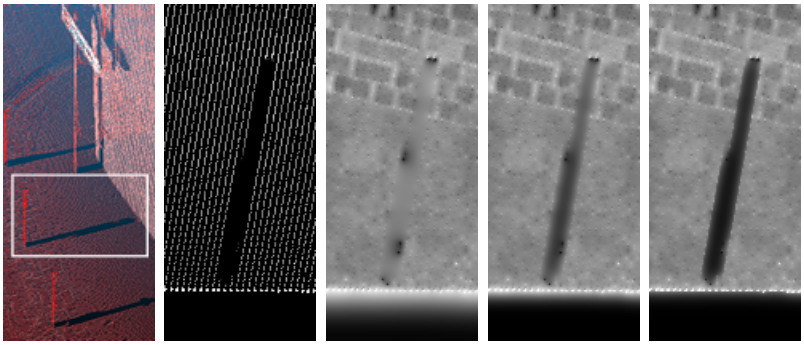
Figure 2.8: Full orthoimage production pipeline from MLS. Framed rectangles are processing steps, arrows are exchanged data.

The first step consists in extracting ground points and their projection onto a 2D plan, leading to two 2D-images: u_0 for reflectance and h_0 for height. This projection is done by removing the z (height) coordinate and rounding the coordinates to the chosen resolution. We also build a mask $\mathcal{M}_{proj}$ of pixels where at least one point was projected. After this first stage, the projections u_0 and h_0 are sparse as they do not cover all the pixels in the images. We can distinguish 3 classes of holes in these (sparse) images.

RR1. Areas that do not correspond to the ground points (inside of buildings, point that are too far from the sensor, ...). These areas do not require any more processing as they should ideally not be reconstructed.

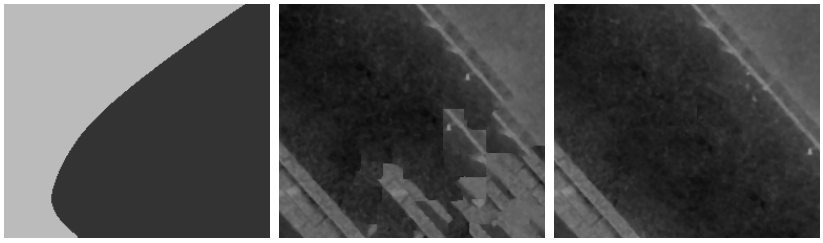
RR2. **Under sampling** of the LiDAR sensing. As the holes are here very small, we have chosen to use a diffusion method in order to reconstruct missing data. We rely on a modified version of the anisotropic diffusion from Perona et al., 1990,by coupling reflectance and height information. A result is presented on figure 2.9.

RR3. **Occlusions** that appear when a relatively vertical object was blocking the beams. Occlusions prevent the sensor from acquiring ground points. The hole being larger, we here rely on a modified version of the patch-based inpainting approach from (Criminisi et al., 2004, section 1.3) by (i) averaging the patch distance over the reflectance and height images; (ii) constraining the selection of candidates to patches



(a) (b) (c) (d) (e)

Figure 2.9: Comparison of different diffusion techniques for filling under sampling holes, without filling the occlusion holes. (a) is the point cloud, (b) its projection (rotated for clarity purpose), (c) is the Gaussian diffusion result, (d) is the Perona-Malik algorithm result and (e) is the result of our proposed modification. We can see that our modification of the Perona-Malik algorithm using both height and reflectance information provides a better conservation of big holes while filling perfectly the stripe holes.



(a) (b) (c)

Figure 2.10: Patch based inpainting of occlusions (a) Occlusion mask: the darkest region is the missing area. (b) Result from (Criminisi et al., 2004). (c) Result with our method. The result is clearly better in (c) as the reconstruction conserves the structures of the image without creating new artifacts such as the ones appearing on the left of (b).

that are at a similar distance to the sensor than that of the current patch. A result is shown on figure 2.10.

A comparison between traditional aerial orthophotography and MLS orthoimage using our framework is given in figure 2.11. The resolution provided by the aerial camera is about 50cm^2 per pixel, whereas our reconstruction is done at 1cm^2 per pixel. Fine textures and very accurate details are noticeable in the reconstruction whereas only main structures can be seen in the aerial orthophotography. Moreover, the aerial orthophotography presents various occlusions (due for instance to trees) that do not appear in our result.

Figure 2.11: Comparison between aerial orthophotography (top) and MLS orthoimage (bottom). The MLS reconstruction contains $\sim 6.10^6$px in which 72% were obtained by diffusion, and 10% by examplar-based inpainting.

2.2 Combining geometric and patch-based methods

Both patch-based and geometric methods have their own advantages and limits for image inpainting. It therefore seems natural to combine these two categories of methods. The solution that has been proposed in (Bertalmio et al., 2003) consists in decomposing the initial image into a structure image and a texture one (Vese et al., 2003). The structure image is completed using a geometric method (Bertalmio et al., 2000) and the texture one with a patch-based approach (Efros et al., 1999). These two images are summed to retrieve the final reconstructed image. An improvement to this method using more recent inpainting methods and enhanced filling order (section 1.3) was later proposed in (Bugeau et al., 2009). While being promising, this first way of combining geometric and patch-based methods remains too dependent on the quality of the texture/structure decomposition.

Other works (Mairal et al., 2008; Elad et al., 2005; Peyre et al., 2007; Fadili et al., 2009) perform inpainting in an energy minimization framework with sparsity regularization:

patches inside $\mathcal{H}$ are synthesized as a sparse linear combination of elements from an image dictionary. As these approaches are limited to the reconstruction of thin regions, they will not be further discussed in this section.

In the following we present energy functions which minimization lead to iterative patch-based methods.

Discrete patch-based modeling

The first discrete modeling for patch-based inpainting was proposed in (Demanet et al., 2003). The authors introduce an energy which is optimized over the correspondence map φ and the image u:

$$E_d(\varphi, u) = \sum_{p \in \mathcal{H}} V_d\left(p, \varphi(p)\right) = \sum_{p \in \mathcal{H}} d_{SSD}\left(\Psi^u(p), \Psi^u(\varphi(p))\right) \tag{2.2}$$

This equation being highly non linear, there is no explicit solution or algorithm to find a global minimum. Demanet et al., 2003 proposed to approximate the solution by repeating until convergence the basic texture synthesis approach with an "onion peel" strategy. The algorithm updates, for each pixel p, $u(p)$ and $\varphi(p)$ conjointly. Note that the authors suggested as a possible improvement to impose restrictions on the correspondence map by penalizing its total variation and favoring that it behaves locally as a translation, but they did not later pursue this line of thought.

A similar energy was also proposed in (Kwatra et al., 2005) for texture synthesis and in (Wexler et al., 2004) for inpainting. The algorithms repeat two steps until convergence: *(1)* a nearest neighbor search (thus updating φ for all pixels); *(2)* a reconstruction step updating u. In both methods, the reconstruction is done through patch fusion as a weighted mean of overlapping patches. In (Wexler et al., 2004), this is formulated as:

$$u(p) = \frac{\sum_{k \in \Psi_s(p)} w(q) u(\varphi(q) + p - q)}{\sum_{q \in \Psi_s(p)} w(q)}$$

where w are weights:

$$w(q) = \exp\left(-\frac{d_{SSD}(\Psi^u(q), \Psi^u(\varphi(q)))}{2\sigma^2}\right)$$

σ being a parameter. Such a reconstruction smooths the final image but maintains spatial regularity along contours.

Regarding the nearest neighbor search, in order to decrease the computational time,

Wexler et al., 2004, perform ANN search (Arya et al., 1993). The "Content-Aware fill" feature of Photoshop CS6 uses the patchmatch ANN search (see section 1.1) together with the reconstruction from (Wexler et al., 2004).

Modeling through graph models

Instead of imposing spatial regularity through patch fusion, some works prefer to add it as a constraint in the energy formulation, and recast inpainting as a labeling problem.

Komodakis et al., 2006, formulate the inpainting problem as an energy minimization over a graph where the nodes are a subset of pixels whose neighborhood intersects $\mathcal{H}$. The energy reads

$$E(\varphi, u) = \sum_{p \in \mathcal{N}} V_d\left(p, \varphi(p)\right) + \lambda \sum_{(p,q) \in \mathcal{E}} V_c\left(p, \varphi(p), q, \varphi(q)\right) \tag{2.3}$$

where

$$V_d\left(p, \varphi(p)\right) = \sum_{r \in \Psi_s(p) \cup \Psi_s(q)} d\left(\Psi^u(\varphi(p) + r - p), \Psi^u(\varphi(q) + r - q)\right). \tag{2.4}$$

λ is a scalar, $\mathcal{N}$ is a subset of pixels regularly spread over the pixel grid Ω and $\mathcal{E}$ is the set of edges with a 4-neighborhood system. The second term (smoothness term) measures how well two patches $\Psi_s(\varphi(p))$ and $\Psi_s(\varphi(q))$ agree at the resulting region of overlap (blue area in figure 2.12). We name this regularity *overlap coherence*. This problem is optimized over a set of possible labels through belief propagation leading to very good results on many images. However, there is no assurance that the iterative process converges to a global minimum. Additionally visual artifacts may appear since the method uses a subset of pixels and copy entire patches.

A graph representation was also used in (Pritch et al., 2009), $\mathcal{N}$ being the complete pixel grid. The first term (data term) constraints candidate patches to belong to the known part of the image. The smoothness term enforces the overlap coherence, both in term of color as in (Komodakis et al., 2006) but also in terms of gradients:

$$V_c\left(p, \varphi(p), q, \varphi(q)\right) = \|u(\varphi(q)) - u(\varphi(p) + q - p)\|^2 + \beta \|\nabla u(\varphi(q)) - \nabla u(\varphi(p) + q - p)\|^2$$

where β is a scalar. The energy in minimized in a multi-resolution scheme through mincut/maxflow (Kolmogorov et al., 2004).

An extension to this approach was later proposed by Y. Liu et al., 2013, with a data term similar to (2.2), and a specific treatment for coarsest resolution in order to deal with lost high frequencies.

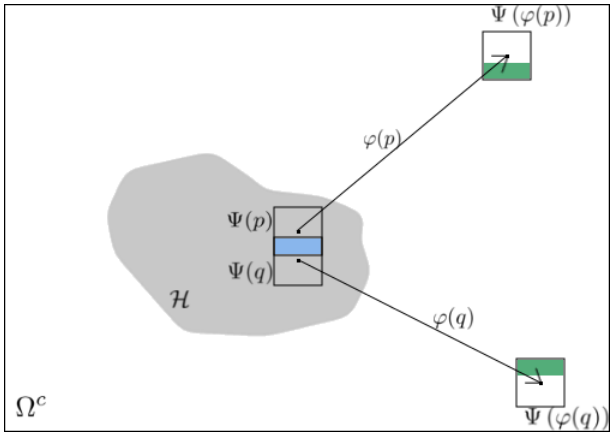


Figure 2.12: Illustration of the overlap coherence: the two green areas must be consistent. The blue region corresponds to the overlapping area $\Psi_s(p) \cup \Psi_s(q)$ and the two green areas are the ones compared in the second term of equation (2.3).

Combining patch-based, geometric and overlap coherence constraints into one model

Aujol et al., 2010, proposed several energy functionals for inpainting with terms related to texture synthesis, spatial coherence (by favoring the mapping function to be locally a roto-translation), and geometry (by minimizing the Total Variation of the "geometry" part of the image, after a cartoon/texture decomposition), but the authors did not actually proposed a minimization algorithm for them.

In (Bugeau et al., 2010a), we proposed a discrete model, made of three terms, also combining texture synthesis, geometry and overlap coherence:

$$E(\varphi, u) = \sum_{p \in \mathcal{H}} V_d\left(p, \varphi(p)\right) + \sum_{p \in \mathcal{H}} V_g\left(p, \varphi(p)\right) + \sum_{p \in \mathcal{H}} V_c\left(p, \varphi(p)\right) \tag{2.5}$$

The first term is the same as the one defined in (2.2), with the metric d_{SSDB} (1.4). The second term encourages continuity in geometric structures:

$$V_g\left(p, \varphi(p)\right) = d_{SSD}(\Psi^{v_d}(p), \Psi^{u}(\varphi(p))), \tag{2.6}$$

where v_d is a *diffusion image* defined as:

$$v_d(p) = \frac{\sum_{q \in \Psi_s(p) \setminus p} w(p,q) u(q)}{\sum_{q \in \Psi_s(p) \setminus p} w(p,q)}$$

where w is a weighting function depending on the type of diffusion used. In our work,

we used the coherence transport diffusion method of (Bornemann et al., 2007) for which $w(p,q)$ is higher if the pixel q is in the direction of the level line at p.

The third terms enforces the overlap coherence. We consider that the family of values $\{\Psi^u(\varphi(q)+p-q) : q \in \Psi_s(p)\}$ is representative of the value of $\Psi^u(p)$ and can be used to estimate it. In order to be less sensitive to the possible aberrant values in the neighborhood of p, we propose to use median instead of mean (as in Komodakis et al., 2006), leading to:

$$V_c\left(p, \varphi(p)\right) = d_{SSD}(\Psi^{v_c}(p), \Psi^u(\varphi(p))), \tag{2.7}$$

where v_c is a *coherence image* defined as:

$$v_c(p) = \text{median}\{\Psi^u(\varphi(q)+p-q) : q \in \Psi_s(p)\}.$$

The multi-resolution algorithm alternates an estimation step and a reconstruction step. The estimation of φ simply consists in estimating the nearest neighbor field with respect to the three patch representations (Ψ^u, Ψ^{v_c} and Ψ^{v_d}) coinjointly. The reconstruction of u simply consists in copying the values of this nearest neighbor for each pixel in $\mathcal{H}$.

Characterizing solutions of patch-based models

For all previous models, there is no proof of convergence to a global minimum. In other words, it is very hard to characterize and improve the solutions of the models. An important contribution was then proposed in (Arias et al., 2011; Arias et al., 2012). The authors propose a functional optimized not over the correspondence map but over a weighting function $w : \mathcal{H} \times \Omega^S \to \mathbb{R}$ indicating the contribution of each pixel in Ω^S to the estimation of $u(p),\ p \in \mathcal{H}$. This formulation is inspired by a variational formulation of non-local means denoising (Gilboa et al., 2007 and Gilboa et al., 2008). It contains two terms: one measuring the similarity between patches in $\mathcal{H}$ and patches in Ω^S, and the other related to the entropy of the weight distributions:

$$E(u,w) = \int_{\tilde{\mathcal{H}}}\int_{\Omega^S} w(p,q)d(\Psi^u(p),\Psi^u(q))dqdp - T\int_{\tilde{\mathcal{H}}}\int_{\Omega^S} w(p,q)\log w(p,q)dqdp \tag{2.8}$$

subject to

$$\int_{\Omega^S} w(p,q)dq = 1\ \forall p \in \tilde{\mathcal{H}}$$

where the sets $\tilde{\mathcal{H}}$ is a dilated version of $\mathcal{H}$. The parameter T controls the selectivity of the weighting function w. The algorithm contains two steps repeated until convergence: estimation of w and reconstruction of u. For all pair of pixels $p \in \mathcal{H}$ and $q \in \Omega^S$, the

estimation step computes:

$$w(p,q) = \exp\left(-\frac{d(\Psi^u(p), \Psi^u(q))}{T}\right),$$

given the current image u. In practice, to avoid prohibitive computation times, a ANN search with patchmatch (Barnes et al., 2009) in first used, and $w(p,q)$ is only computed when q is an approximate nearest neighbor of p. The reconstruction step computes u with an explicit formulation which depends on the distance d: patch non-local means (if d is the Gaussian weighted SSD), patch non-local median (if d is the Gaussian weighted SAD), patch non-local Poisson (if d is the sum of Gaussian weighted SSD over u and ∇u) and a reconstruction from the Affine Invariant Self-Similarity (Fedorov et al., 2016). Convergence to critical points are discussed in Arias et al., 2012.

Extension of this method to video inpainting was proposed in Newson et al., 2014.

with

Results

Figure 2.13 presents a comparison of different inpainting methods discussed previously. All of these methods require the tuning of some parameters. For the method from (Tschumperlé, 2006) as suggested in the original paper, we set the contour preservation parameter p_1 to 0.001, the structure anisotropy $p_2 = 100$, the time step $dt = 150$, and the number of iterations nb equal to 100. For the technique from (Bornemann et al., 2007), we set the averaging radius $\epsilon = 5$, the sharpness parameter $\kappa = 25$, the scale parameter for pre-smoothing $\sigma = 1.4$, and for post-smoothing $\rho = 4$. For all patch-based methods, we used 9×9 patches. We can see that Bugeau et al., 2010a; Arias et al., 2011 and Content-Aware fill (Photoshop CS5) are able to correctly inpaint the covered lanterns, the text on these lanterns and the red and white striped bars, while the other techniques fail in some of these aspects. This result highlight the importance of using a non-greedy approach and the fusion of geometric and patch-based methods.

Finally, Figure 2.14 shows an image where none of the methods discussed in this document seems to work. The problem here is that diffusion methods (Tschumperlé, 2006 and Bornemann et al., 2007) are able to synthesize new information but they are not able to deal with texture, while patch-based methods (Criminisi et al., 2004) treat texture correctly but are not good at creating new information. Even if the models discussed in this section combine these different approaches, they remain based on image self-similarity principle that may not give good results if the information needed is not already present in the image. For most of the mask, no good candidate patches exist.

Figure 2.13: Comparison of several inpainting algorithms. (Bugeau et al., 2010a), (Arias et al., 2011) with nl-poisson (with the implementation from Fedorov et al., 2015) and Content-Aware fill are able to correctly reconstruct the covered lanterns, the text on these lanterns and the red and white striped bars, while the other previous techniques fail in some of these aspects.

2.3 Conclusion

This chapter has first presented geometric methods for image completion. We have explained how both patch-based and geometric methods can be used for LiDAR point cloud completion. We have also illustrated the limits of these two types of approaches for image inpainting when taken separately. Several patch-based models that tend to globally optimize the inpainting problem over the whole missing areas have next been discussed and their advantages highlighted. While today providing state-of-the-art results, they are still not able to solve the inpainting problem in case of complex scenes with large holes (figure 2.14). Improved models for image inpainting could probably benefit from constraints on the correspondence map itself to suggest from where the information should be copied.

Another direction of research to cope with complex scene would be to rely on more information than the image itself. There have been several attempts in the literature to extend the self-similarity principle to a larger set of patches or images. Nevertheless using an external dictionary of patches is currently limited to thin holes restoration and prevent constraints such as overlap coherence into the models. Using a larger set of images was

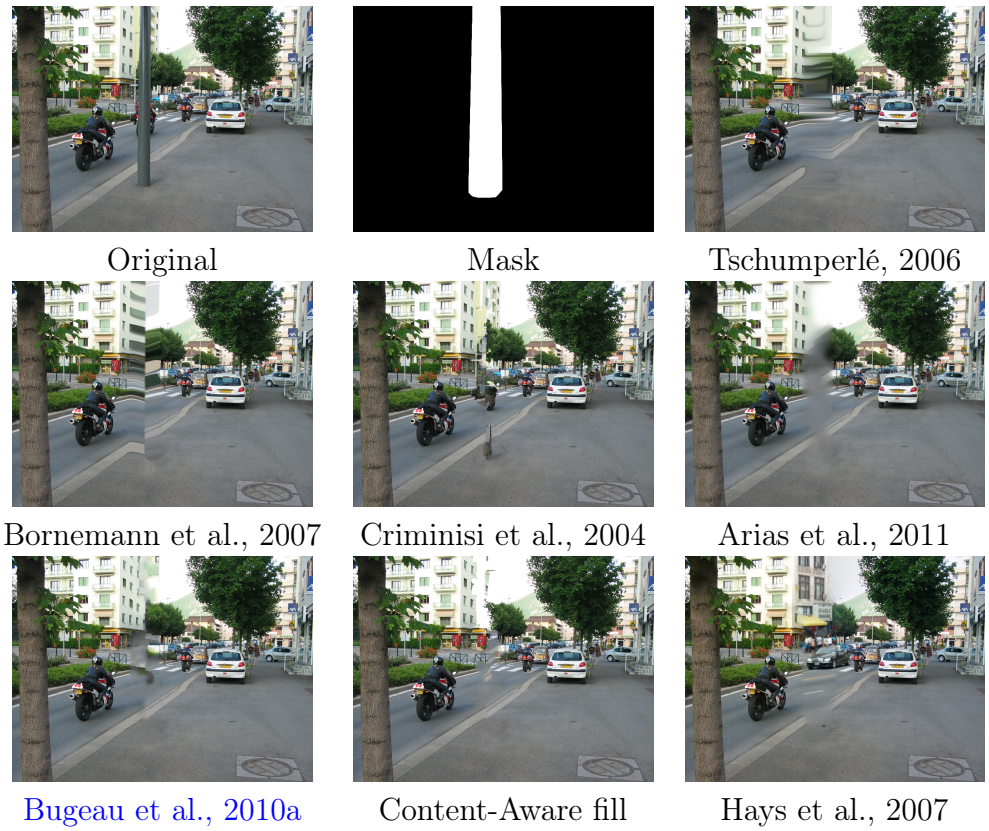


Figure 2.14: Comparison of several inpainting algorithms. None of the aforementioned methods provide a satisfying result on this image.

therefore proposed by Hays et al., 2007. The semantically closer images in the database are first retrieved before being aligned and composited. There are some limits as seen in figure 2.14. In particular, the method highly rely on the semantic retrieval which is likely to fail at recognizing partly occluded objects or non textured areas (for instance pavement in figure 2.14). More recently deep learning has also been used for inpainting purposes (Pathak et al., 2016, and Yang et al., 2017) leading to very promising results, though still a little blurry. As discussed in chapter 4, further research could be conducted in this direction even if it is difficult to characterize the solutions to the problem.

Finally, application to video inpainting is still an open subject. While some methods exist (Granados et al., 2012; Newson et al., 2014), the computational time is still prohibitive to enable their use in practice.

Chapter 3

Patch-based models for image colorization

In this chapter we focus on image colorization. As detailed in chapter 1, colorization is an ill-posed problem as the input image does not have any color prior. To solve this issue, patch-based methods (called automatic methods) consider as input a color image, called source, and copy the color of its pixels to the target grayscale image. Patch-based colorization methods suffer from the lack of spatial regularity and the growing garbage problem.

A second category of approaches for colorization, named manual or diffusion methods, consider as input manually added points of color (called *scribbles*) and propagate these colors to the rest of the image. This is achieved by applying anisotropic diffusion to the chrominance depending on the gradients of the luminance channel. As textures in an image mostly come from the luminance channel, applying diffusion on the chrominance channels produce satisfying results even in texture areas. The main limit of manual methods is the amount of user intervention required.

This chapter aims at proposing some models that enable coupling patch-based approaches with diffusion-based ones by imposing spatial regularity and constraint on the gradients of the chrominance channel w.r.t. the luminance ones. Another limit of automatic method is that, as the color prior only depends on the source image, it is worth choosing this image very carefully which is often a difficult task. Therefore several methods have proposed to use multiple registered similar images collected from internet (see X. Liu et al., 2008; R. Zhang et al., 2016). In another trend, in (Pierre et al., 2014a), we propose to add user interaction with color scribbles, in order to improve patch-based colorization results.

This chapter is organized as follows. First we review diffusion-based manual methods

(section 3.1) before introducing some models that combine patch-based and diffusion-based approaches (section 3.2). We will use the same notations as in chapter 1. We denote as

- $u_T : \Omega \to \mathbb{R}$ the target grayscale image (which is assumed to be a luminance image)
- $u : \Omega \to \mathbb{R}^3$ the estimated color image, $Y_u : \Omega \to \mathbb{R}$ its luminance (that must be equal to u_T), $u_c = (U_u, V_u) : \Omega \to \mathbb{R}^2$ its chrominances in the YUV color space, and R_u, G_u, B_u its channels in the RGB color space.
- $u_S : \Omega^S \to \mathbb{R}^3$ the source color image.

3.1 Diffusion-based manual methods

The first manual colorization method based on diffusion was proposed by Levin et al., 2004. It solves an optimization problem to diffuse the scribbles chrominances with the assumption that chrominances should have small variations where the luminance has small variations:

$$E(u_c) = \sum_{p \in \Omega} \left(u_c(p) - \sum_{q \in \mathcal{N}(p)} w_{pq} u_c(q) \right)^2 \tag{3.1}$$

where $\mathcal{N}(p)$ is the 4-neighborhood of pixel p, w_{pq} are weights defined by

$$w_{pq} \propto \exp\left(-\frac{(Y(p) - Y(q))^2}{2\sigma_p^2} \right)$$

where σ_p is the variance in a window around p. A result of this method is presented on Figure 3.1.

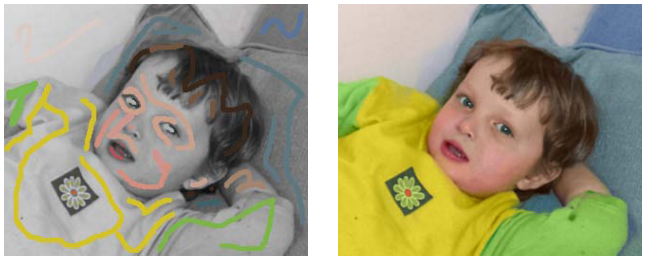


Figure 3.1: A result from the manual method of Levin et al., 2004. The color scribbles are propagated to the rest of the image.

Yatziv et al., 2006 propose a simple yet fast method by using geodesic distances to blend the chrominances given by the scribbles. Heu et al., 2009 use pixel priorities to ensure that important areas end up with the right colors. Other propagation schemes include

probabilistic distance transform (Lagodzinski et al., 2008), discriminative textural features (Kawulok et al., 2012), structure tensors (Drew et al., 2011) or non local graph regularization (Lézoray et al., 2008). As often described in the literature, with these manual approaches, the contours are not well preserved. To cope with this issue, in (Ding et al., 2012), scribbles are automatically generated after segmenting the image and the user only needs to provide one color per scribble. computing quaternion wavelet phases so that the color is propagated along equal phase lines.

However, all manual methods suffers from the following drawback: if the target represents a complex scene the user interaction becomes very important. On the other hand these approaches propose a global optimization over the image thus leading to spatial consistency in the resul

3.2 Combining patch-based and diffusion methods

In this section we will describe some models for image colorization. The purpose for these models is three folds: (i) add spatial consistency in the results obtained with automatic patch-based methods, (ii) add coupling between luminance and chrominance gradients (iii) increase the set of possible chrominances for each pixel with manual intervention.

To overcome the spatial consistency and coupling problems in automatic methods, several works rely on image segmentation. For instance, Irony et al., 2005 propose to determine the best matches between the target pixels and regions in a pre-segmented source image. With these correspondences, micro-scribbles from the source are initialized on the target image and colors are propagated as in (Levin et al., 2004). Gupta et al., 2012 extract different features from the superpixels (X. Ren et al., 2003) of the target image and match them with the source ones. The final colors are computed by imposing spatial consistency as in (Levin et al., 2004). These approaches incorporate diffusion into automatic methods but are highly dependent on the quality of the segmentation. In (Bugeau et al., 2014; Pierre et al., 2015a; Pierre et al., 2014a) and our other papers on image colorization, we have proposed several models to cope with the limitations of automatic patch-based methods. In all these works, we consider that each pixel p in the target image u_T can only take its chrominance (or RGB color) among a reduced set of possible candidates: $C(p) = \left\{c_i(p) \in \Omega^S \ , i = 1, \dots, N\right\}$. These candidates can either be extracted from the source image as in (Bugeau et al., 2012) (see section 1.4.3) or be given by manual intervention (Pierre et al., 2014a).

Colorization is traditionally done in a luminance-chrominance space, estimating the chrominance channels only. This may lead to some issues if the conversion from luminance-chrominance space back to RGB gets outside the range of acceptable values in RGB. The

first solution we propose is to constraint the final chrominances to belong to the set of acceptable chrominance values given an input luminance. The second one is to directly solve the colorization problem in the RGB space. These two directions are now discussed.

3.2.1 Colorization in the YUV color space

In (Bugeau et al., 2012), we have proposed to set the final chrominance at pixel p as the median of the candidate color set $C(p)$ (see section 1.4.3), which approximates the minimization of

$$E(u_c) = \sum_{p\in\Omega} \sum_{i=1\cdots N} |u_c(p) - c_i(p)|.$$

Unfortunately, when working with $d > 1$-dimensional data, this ℓ_1 norm separates the distance between dimensions and the median value does not correspond anymore to one of the candidates. In order to have a smooth derivable data term, in (Bugeau et al., 2014), we considered the ℓ_2^2 norm. Our problem of selecting one single candidate per point, thus avoiding heterogeneous mixing of candidate colors leading to highly unrealistic results, can be re-casted as follows:

$$E(u_c) = \sum_{p} \min_i \|u_c(p) - c_i(p)\|_2^2 \;\text{ s.t. }\; u_c \in \mathcal{R}$$

where $\mathcal{R}$ is the standard range of chrominances. In order to minimize this energy, we consider a relaxation by increasing the state dimension with a new variable

$$w(p) = \{w_i(p), i = 1, \ldots, N\}$$

which represents the probability of choosing the i-th candidate for each point:

$$E(u_c, w) = \sum_{p} \sum_{i} w_i(p)\|u_c(p) - c_i(p)\|_2^2 \;\text{ s.t. }\; u_c \in \mathcal{R} \;\text{ and }\; w \in \mathcal{W}$$

with

$$\mathcal{W} = \left\{ w(p) : \Omega \to [0;1]^N, \text{ s.t. } \sum_{i=1}^{N} w_i(p){=}1 \right\}.$$

If $w_j = 1$ and $w_i = 0$ for $i \neq j$, then the minimum w.r.t. u_c is reached for $u_c = c_j$, *i.e.* when selecting one candidate only. A similar fidelity term was introduced for optical flow estimation from a collection of previously computed motion candidates in (Fortun et al., 2016).

Previous energy thus admits trivial minimizers than can be chosen independently for each pixel p. In our works, we focus on the image colorization problem which involves consistent and spatially coherent image results. In (Bugeau et al., 2014) a Total Variation (TV) based regularization of the unknown variable u_c is introduced in the energy to smooth the solution while keeping sharp discontinuities when required by the data. Unfortunately, this regularization lacks of coupling with the luminance information leading to halo effects in the colorization results (figure 3.2). Hence, in (Pierre et al., 2014a), we introduced the *coupled Total Variation* which is able to couple the chrominance channels with the luminance one [1]:

$$TV_c(u) = \sum_p \sqrt{\gamma\partial_x Y^2 + \gamma\partial_y Y^2 + \partial_x U^2 + \partial_y U^2 + \partial_x V^2 + \partial_y V^2}\ , \tag{3.2}$$

where γ is a parameter which enforces the coupling.

The energy for colorization finally reads:

$$\begin{aligned} E(u_c, w) = \quad & TV_c(u) + \frac{\lambda}{2}\sum_p \sum_i w_i(p)\|u_c(p) - c_i(p)\|_2^2 \\ & \text{s.t. } u_c \in \mathcal{R} \text{ and } w \in \mathcal{W} \end{aligned} \tag{3.3}$$

A primal-dual algorithm, inspired by (Chambolle et al., 2011) is used to minimize this energy. Its convergence to a fixed-point which is a critical pseudo saddle-point of the problem is demonstrated in (Pierre et al., 2015a). In Figure 3.2 we compare the result with ($\gamma = 25$, as in Pierre et al., 2014a) and without the coupled TV ($\gamma = 0$, as in Bugeau et al., 2014). The set of candidates C is defined as in section 1.4.2, *i.e.* using a patch-based search for candidates using 8 different patch features. The coupled TV highly reduces the halos along the contour thus leading to a more realistic result.

Combining automatic and manual colorization

We now present in Figure 3.3 a result when the set of candidates come both from the automatic patch-based method described in section 1.4.3 and from a user manual intervention. At initialization, the weights are uniform for all candidates issued by the patch-based automatic method, while for manual candidates, they depend on the geodesic distance to the scribbles. Using the "automatic candidates" only (Figure 3.3(c)), the sky is not correctly colorized since it appears brown instead of blue as in the ruins main door. Moreover, blue colors appear on the floor. When using only the scribbles (Figure 3.3(d)), we recover a

[1] To simplify the notations, the dependence of each values to the position of the current pixel is removed.

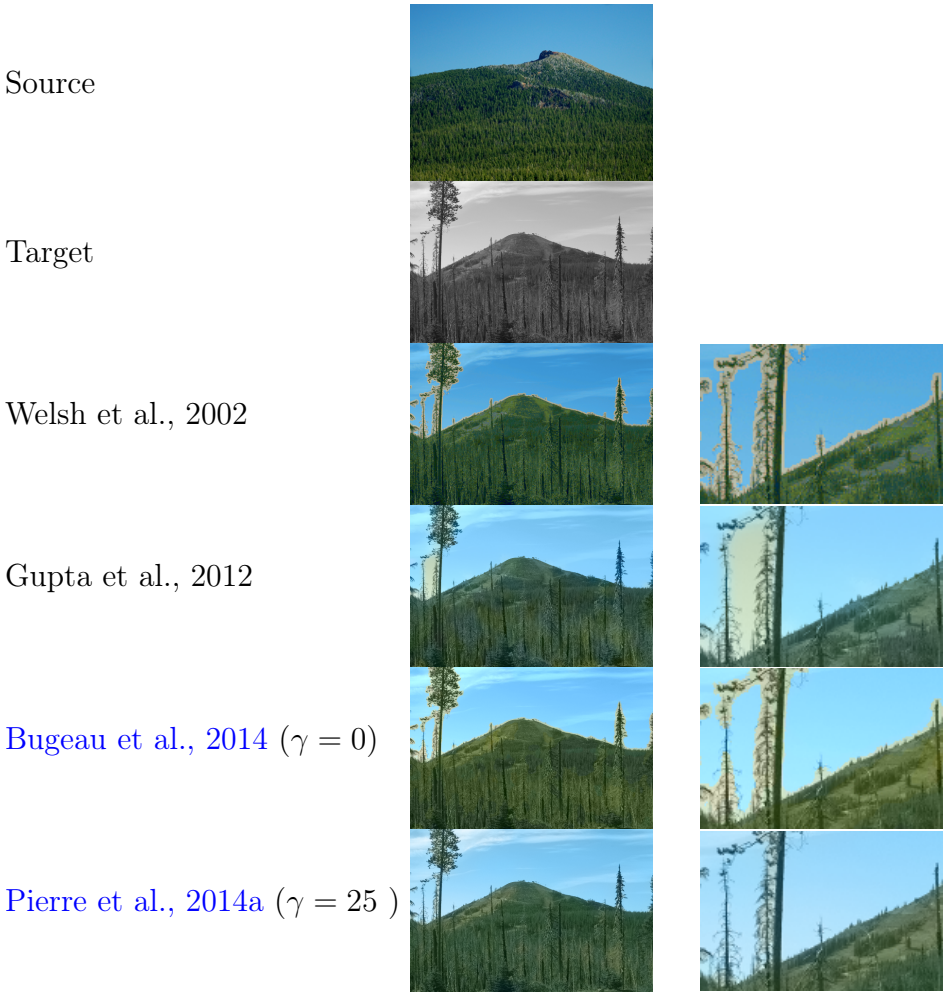


Figure 3.2: Comparison of our methods with state-of-the-art results for images with thin structures. On the left, the source image in color and the target image in gray-scale. The second column is a zoom-in that better illustrates the importance of the coupled TV.

blue sky but not the color of the ruins. When combining both (Figure 3.3(e)), a realistic result is obtained. Note that instead of the geodesic distance, we could have relied on the sroke similarity mask issued from the edge-aware texture descriptor as proposed in (Arbelot et al., 2017).

Extension of this method to video colorization was recently proposed in (Pierre et al., 2017b). The method propagates colors from an initial colorized frame to the whole gray-scale video sequence. The spatio-temporal model also relies on pre-computed possible candidates per pixel. Here, these candidates are coming (1) from the propagation of initial colors with optical flow or block matching (using patchmatch), (2) from manually added scribbles.

(a) Source

(b) Target with scribbles

(c) Automatic

(d) Manual

(e) Automatic+Manual

Figure 3.3: Colorization result obtained when (c) using only candidates from the automatic patch-based method (d) using only candidates from the scribbles (e) using both. Neither the automatic approach nor the manual one are able to properly colorize this image with the given priors. The unified method gives a suitable result.

3.2.2 Colorization in the RGB color space

One common problem with all image colorization methods that aim at reconstructing the chrominances of the target image is that the recovered chrominances combined with the input luminance may not fall into the RGB cube when converting back to the RGB color space. In the model (3.3), this problem is solved with the constraint $u_c \in \mathcal{R}$ and using the YUV space which conversion back to RGB is linear. In this section we propose to work directly in the RGB color space with constraint on the luminance. Considering that $u = (R, G, B)$, we proposed in Pierre et al., 2014b the following model:

$$\begin{aligned} E(u, w) = \quad & TV(u) + \frac{\lambda}{2} \sum_p \sum_i w_i(p) \|u - c_i(p)\|_2^2 \\ & \text{s.t. } u \in [0, 255]^3 \text{ and } Y = u_T \text{ and } w \in \mathcal{W}. \end{aligned} \tag{3.4}$$

We remind that u_T is the target luminance image and Y is the luminance corresponding to u:

$$\forall p \in \Omega \quad Y(p) = A.u(p)$$

with $A = (0.2990, 0.5870, 0.1140)$. The candidates c_i are now defined in the RGB space.

The important contribution in this work concerns the respect of both constraints

$$u(p) \in [0, 255]^3 \text{ and } Y(p) = u_T(p) \ \forall p. \tag{3.5}$$

This is equivalent to computing a projection of$u(p)$ onto the intersection of the cube $[0, 255]^3$ and the affine plan defined by all 3D points having a luminance equal to $u_T(p)$ (dark yellow plan on figure 3.5(a)). We add one constraint when doing this projection in order to preserve the hue of u, which represents the human perception of the pure color. Indeed, an orthogonal projection onto these constraints modifies the hue (3.4). Therefore we detail now an oblique projection that preserves constraints (3.5) and the hue.

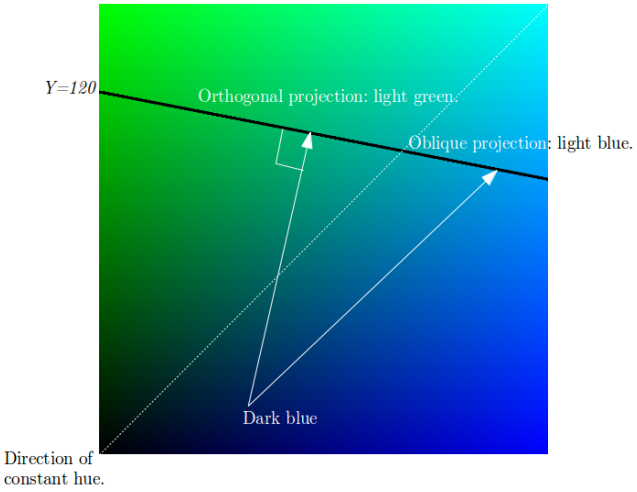


Figure 3.4: The orthogonal projection of a color (here dark blue) on the constraint $\{u(p) \in [0, 255]^3 \text{ and } Y(p) = u_T(p)\}$ modifies the hue. The color is here projected on light green instead of blue. An oblique projection is able to avoid this issue.

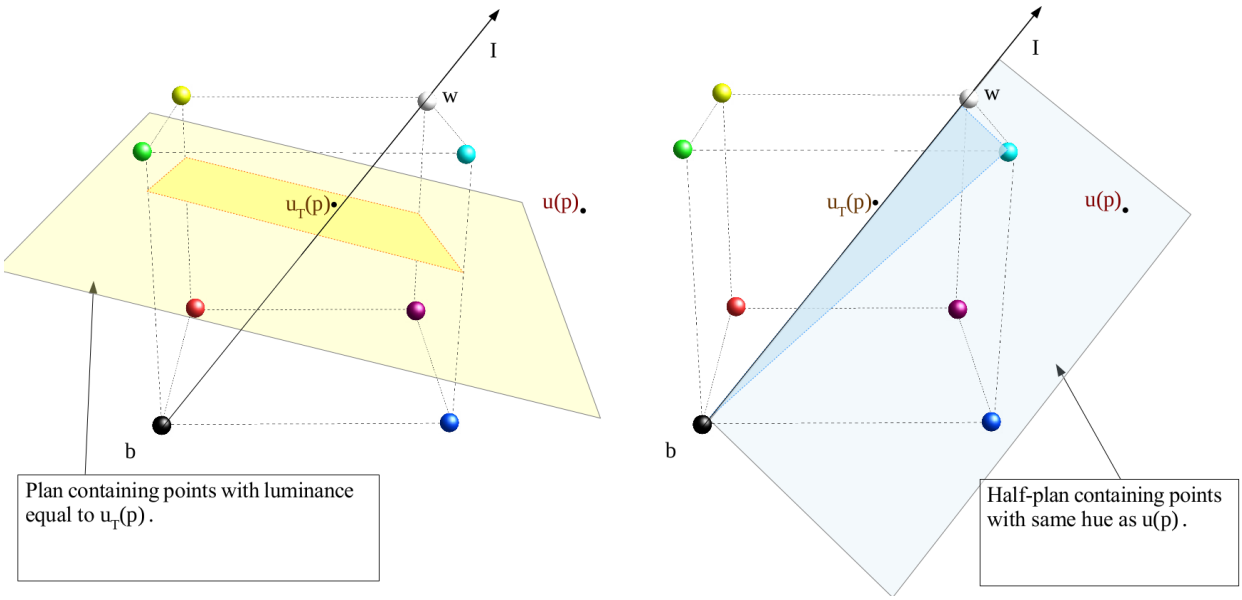
The hue H in the HSI (Gonzalez et al., 2007) color space is defined as

$$H = \begin{cases} \theta & \text{if } B \leq G \\ 360 - \theta & \text{otherwise,} \end{cases} \tag{3.6}$$

where θ is an angle in degrees, such as:

$$\theta = \arccos\left\{\frac{\frac{1}{2}\big((R-G)+(R-B)\big)}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right\}. \tag{3.7}$$

All the points in the RGB color space which have the same hue as a given color $u(p)$ live on a half-plan containing $u(p)$ and bounded by the line I going through coordinates (0,0,0) and (255,255,255) (light blue plan on figure 3.5(b), which contains $u(p)$). The projection onto the



(a) (b)

Figure 3.5: Geometric properties of color spaces. (a) affine plan defined by all colors having a luminance equal to $u_T(p)$ (b) Half plan defined by all colors having the same hue as $u(p)$.

three constraints was achieved in (Pierre et al., 2015c). First, $u(p)$ is projected onto the yellow plan of luminance equal to $u_T(p)$ in the direction of I, leading to point u' (figure 3.6). Next if u' is out of the RGB cube, the Smit's algorithm for raytracing (Williams et al., 2005) is used to compute the contact point of the RGB cube $[0, 255]^3$ and the line D containing the green segment intersecting both plans.

Figure 3.7 presents two colorization results obtained by minimizing the model (3.4) with hue constraint and compare them with other methods. The results of (Irony et al., 2005) and (Charpiat et al., 2008) are not realistic. Due to the lack of regularization, images of (Welsh et al., 2002) show artifacts since areas that were originally homogeneous now present irregularities. The RGB colorization (3.4) with the hue constraint better preserves the homogeneous parts such as the sky. The quality of the results is comparable to the one of (Gupta et al., 2012), without requiring any segmentation step.

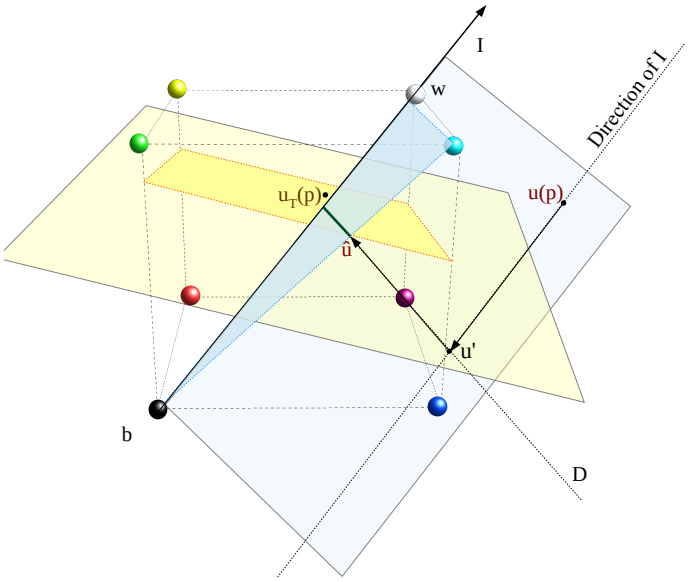


Figure 3.6: Illustration of the projection of $u(p)$ onto the constraints.

3.3 Conclusion

This chapter has introduced several models for image colorization combining patch-based automatic and manual diffusion-based techniques. We have highlighted the importance of avoiding color bleeding along luminance contours and the necessity of managing correctly conversion between color spaces. The quality of the results are very promising and could probably be further enhanced by proposing even more candidates coming from additional patch representations (such as Arbelot et al., 2016) and more manual intervention.

Automatic image colorization methods rely on the careful selection of the source image. If this source is not close enough to the target in terms of context, luminance or texture statistics, no good results can be expected by automatic patch-based methods. As proposed, this problem can be solved by adding user intervention. Nevertheless, this task can become very time consuming in case of complex images. One solution is to rely not on one source image but on several (*e.g.* X. Liu et al., 2008), or, as it will be discussed in chapter 4, on deep learning architecture (*e.g.* R. Zhang et al., 2016), with the difficulty of characterizing the solutions of the problem and to impose constraints reducing visual artifacts.

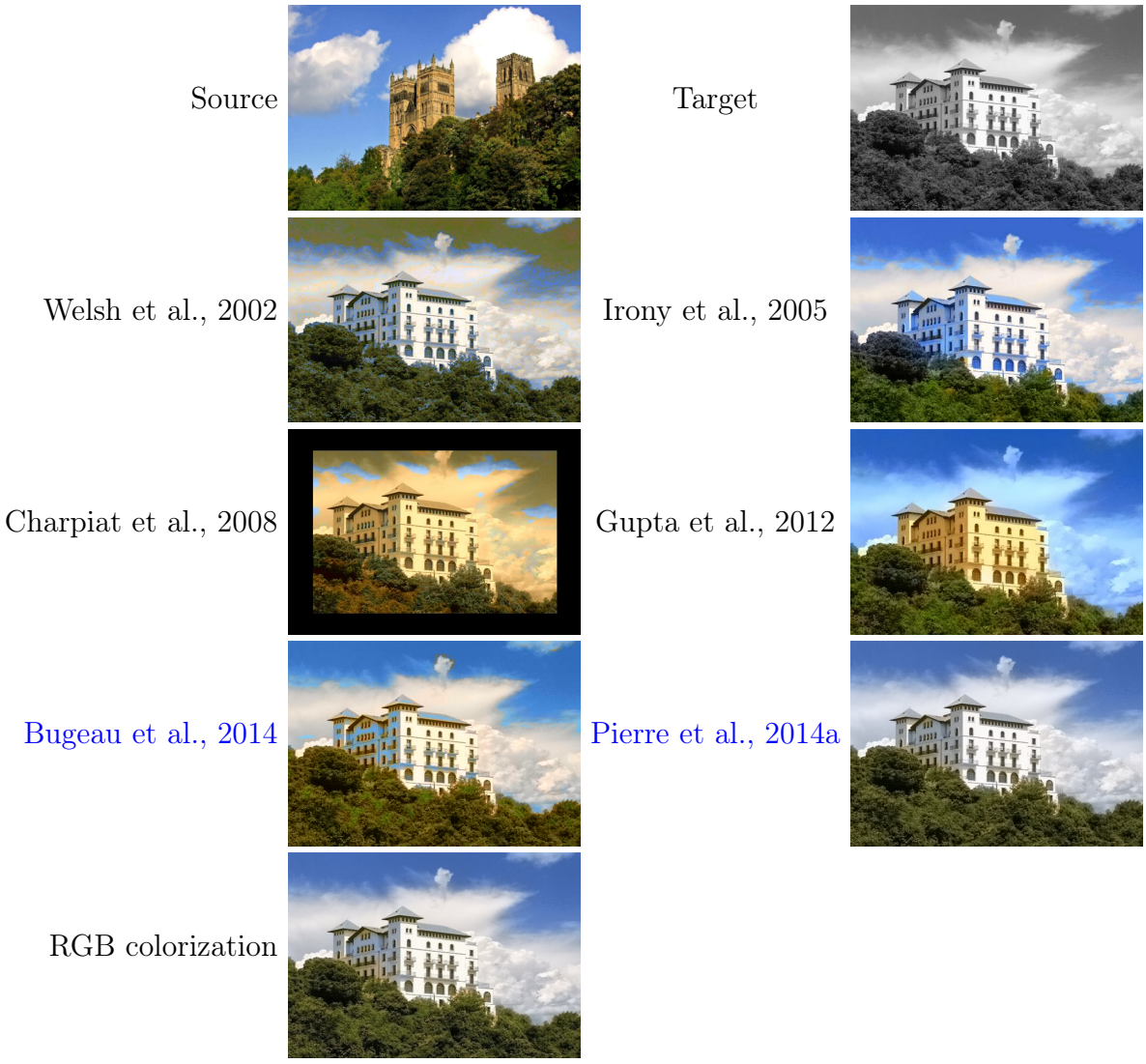


Figure 3.7: Comparison of the RGB colorization (3.4) including hue constraint (Pierre et al., 2015c) with state-of-the-art methods.

Chapter 4

Conclusion and Perspectives

In this manuscript, several models for image inpainting and colorization have been introduced. We have first highlighted the importance of finding a good patch representation and its associated metric in order to capture different properties of the data: high and low frequencies, luminance, color, etc. Secondly we have demonstrated how global optimization can serve the purpose of patch-based approaches. Most discrete and variational models presented incorporate local spatial regularizations of the colors to avoid unrealistic color discontinuities in the final result. Coupled with data terms constraining the set of admissible values for each pixel, these models lead to plausible results with textures and continuous structures. Let us now draw some perspectives.

4.1 Fusion of heterogeneous data

We have discussed in section 2 the application of simple image inpainting methods to the completion of LiDAR point clouds. One difficulty with LiDAR depth information is to precisely locate the contours betweens elements of the scene. For instance, when the tire of a car is in contact with the road, the depth of nearby points are very similar. Coupling depth with reflectance enables to better classify the point as part of the road or the tire but this information is not always accurate enough. Nowadays, most vehicles (planes, unmanned air vehicles, trucks) are also equipped of other sensors such as optical cameras, hyperspectral cameras and GPS combined with Inertial Measurement Unit (IMU). Coupling the data together could lead to very precise and complete 3D reconstruction of urban scenes. The first challenge is to align the different data together. The sensors are placed at different locations on the vehicle and acquire asynchronous data at various resolutions. When a very precise GPS combined with Inertial Measurement Unit is present, and if the vehicle has a relatively slow motion, existing registration techniques already provide satisfying results. Otherwise, it becomes necessary to detect common information such as structures within all the data, which requires further investigation depending on the type

of data considered. In (Castorena et al., 2016) for instance, edge alignment between LiDAR and optical data are considered. Nevertheless this methods implies that edges coincide in both modalities which is not always the case in practice.

Once the registration has been done, specific data fusion, depending on the application, must be considered. In previous works (Bevilacqua et al., 2016; Bevilacqua et al., 2017), we have designed a method to add depth information to every pixel of one image. As several LiDAR points may correspond to the same pixel, it is necessary to choose which depth is appropriate for that pixel. We solved this problem through minimization of a variational functional encouraging spatial regularity of the depth information. This work must be extended to consider the multiple images available for the same part of the scene.

At the opposite, one may want to add color to the point cloud in order to facilitate its visualization. In this case, one point corresponds to one different pixel in each image. Existing softwares generally choose the color from the image having its optical center closer to the point. This simple approach creates color discontinuities that could be avoided by designing a model including visibility and spatial coherency.

Data fusion must also be investigated in the context of structure from motion (SfM). SfM techniques aim at estimating 3D structures from a sequence of 2D images. This is done by first matching images keypoints before computing transformation matrices then used to reconstruct 3D positions of points. The color associated to each point is the one coming from the camera closest to the point, leading to color discontinuities. Here again a global modeling of the problem could enhance the visual quality of the results.

4.2 Regularity of the correspondence map

As any image contains local spatial redundancies, it is natural to consider that a correspondence map should be piecewise constant. A uniform area of the image should indeed correspond to an area of the same shape in the image itself or in another image. In particular, Aujol et al., 2010 (and later Arias et al., 2012) claim that the correspondence map should be a roto-translation and propose a corresponding variational model that is highly non linear, without any associated algorithm. Without directly regularizing the map, the overlap constraint defined for inpainting in section 2.2 tends to produce piecewise translations. Even without imposing any constraint within the algorithm or the model, we can observe experimentally that patch-based methods naturally tend to produce correspondence maps that are piecewise constant. Figure 4.1 for instance was obtained with the patchmatch algorithm and clearly demonstrates that entire part of Ω are translated onto $\mathcal{H}$ to produce the final result. While it could be argued that this is a consequence of the propagation step of the patchmatch algorithm, let us mention that a similar property is also observed with (Efros et al., 1999). What is noticeable when looking deeply into the correspondance maps generated by any patch-based method is that in between regular

pieces lie much smaller areas, that can be as little as isolated pixels. These areas are the ones causing visual artifacts appearing as noise in many restoration results. Controlling regularity of the correspondence map would undoubtedly improve the quality of patch-based restoration methods. A work in that direction was recently proposed for texture synthesis in (Gutierrez Ortega et al., 2017) where the problem is recasted as an optimal assignment of patches with controls on the permutations. This idea should be further explored to more general problems.

Figure 4.1: Regularity of the correspondence map. Inpainting of the image with the patchmatch algorithm leads to a correspondence map which is a piecewise translation. Figure taken from (Arias et al., 2012).

4.3 Deep Learning for image editing

Since 2012, deep learning approaches, and in particular Convolutional Neural Networks (CNNs), have become very popular in the community of computer vision and computer graphics. They have been used for various post-production tasks Mao et al., 2016, such as texture synthesis G. Liu et al., 2016, inpainting Pathak et al., 2016, colorization R. Zhang et al., 2016, style transfer Gatys et al., 2016, super-resolution Dong et al., 2014 or denoising K. Zhang et al., 2017. After a long offline training step, they can infer the output images much faster than previous approaches. Instead of relying on an image model, these approaches require a 'training set', made of images and the corresponding expected output images. During the training phase, the parameters of a Deep Network are optimized by minimizing a loss function on the training set, *i.e.* by comparing the generated output with the groundtruth. Before presenting our perspectives in this context, let us provide a very brief state-of-the-art of deep learning methods for image post-production.

4.3.1 State-of-the-art of deep learning approaches for image and video post-production

The objective of image post-production is to generate an output image given an input image and some application related constraints. In deep networks, the way to infer the output given an input is learnt on training images with groundtruth, given low to high level representations (features) of the input. End-to-end networks directly learn how to compute these features at

the same time they learn how to reconstruct the final image. Otherwise, the features can be previously extracted with traditional methods as the ones introduced in section 1.2 or with a pre-trained networks such as VGG (Simonyan et al., 2014) for instance. Let us now detail some of the methods developed for texture synthesis, inpainting and colorization.

Texture synthesis using deep learning

The problem of texture synthesis can be formulated as follows: given an input sample of texture u_S, generate an output image u which apparently contains the same texture as the input one. In (Gatys et al., 2015) and later in (G. Liu et al., 2016), it has been shown that texture synthesis can be defined as a constrained optimization problem using statistical features (represented with a Gram matrix) generated by the VGG network pre-trained for object recognition. Given the j^{th} set of features $f_{ij}^u(p)$ at layer i of the VGG network applied on image u, the statistical features are defined as:

$$G_{ij}^u = \frac{1}{|\Omega|} \sum_{p \in \Omega} \sum_{q \in \Omega} f_{ij}^u(p) f_{ij}^u(q)^T.$$

The final texture image is obtained by iteratively updating image u, starting from a white noise image, in order to minimize:

$$E(u) = \sum_{i,j} w_{ij} \left(G_{ij}^u - G_{ij}^{u_S} \right)^2,$$

w_{ij} denoting weights of loss at layer i and feature j. Constraints on the Fourier spectrum were later added in (G. Liu et al., 2016) to better reconstruct low frequency structures.

Inpainting using deep learning

Deep learning-based methods have also been proposed for inpainting. Pathak et al., 2016 propose an encoder-decoder pipeline to encode the context information and predict the unavailable content with respect to a learned loss function. This pipeline is called *Context Encoders*. The encoder is a Convolutional Neural Network (CNNs) trained to learn feature maps from Pascal VOC 2007 images (Everingham et al., 2007) in which random regions are automatically removed. From these feature maps, the decoder generates the value of the missing pixels using aother CNN. As the traditional ℓ_2 loss tends to blur the result by averaging the multiple modes in predictions (Figure 4.2(c)), the authors propose to combine it with an adversarial loss. The adversarial loss is based on a Generative Adversarial Network (Goodfellow et al., 2014) that jointly learns a generative model of the data distribution together with a discriminative model which provides loss gradients to the generative models. This loss has the effect of picking a particular mode from the data distribution, resulting in less blurry results (Figure 4.2(d)).

The results from this method often lack of texture details leading to blurry areas at the border of the missing area. Yang et al., 2017 proposed another pipeline made of two networks. The first

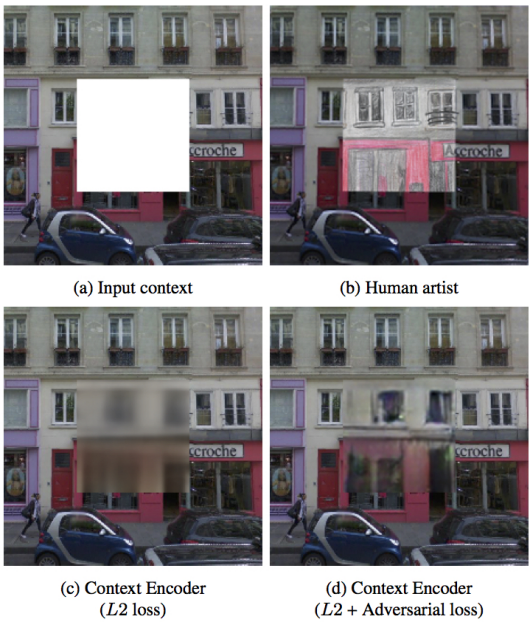


Figure 4.2: Illustration of context encoders result. Figure taken from Pathak et al., 2016.

network, similar to the context encoder, reconstructs an image $\hat{u}$ from the input image u_0. The second is the pre-trained VGG network which is used to compute, from an image u, features f at a pre-determined layer. A discrete model

$$E(u, f, \varphi) = \sum_{p \in \mathcal{H}} \|u(p) - \hat{u}(p)\|_2^2 + \lambda_1 \|\Psi^f(p) - \Psi^f(\varphi(p))\|_2^2 + \lambda_2 \|\nabla u\|_2^2 \tag{4.1}$$

is then iteratively solved. We can observe that this model in fact combines an *a priori* on the solution resulting from the context encoder, a patch-based term involving correspondence maps, and spatial regularization term. While the results are still blurry, it is a first step for combining patch-based and geometric models together with convolutional neural networks.

Colorization using deep learning

Recently, CNNs have been used in the context of image colorization. The idea is to automatically learn on a database with groundtruth how to colorize a given grayscale image.

Cheng et al., 2015, start by extracting and concatenating low, medium and high level semantic features from the grayscale version of the images in the database. They feed these features into a three-layer fully connected neural network trained with a ℓ_2 loss. In order to remove some ambiguities in the training, the database is clustered into scene category and one network is learned per category. During inference, the grayscale image is assigned to its closest category and colorized using the corresponding network. Chrominances are finally refined with joint bilateral filtering (Petschnigg et al., 2004) to remove possible artifacts. The method is highly dependent on the categories present during learning. Deshpande et al., 2015, also propose an

architecture that is limited to scene categories and trained on a small database of 240 images. The success of these two approaches is limited to simple images without foreground objects.

Iizuka et al., 2016, train, in an end-to-end fashion, different features which are fed to the colorization network made of a set of up-convolutional layers. A ℓ_2 loss function is used. In order to remove some ambiguities in the learning, the authors add a classification network within their architecture and use a cross-entropy loss for this task. Therefore, once again, even if a very large database is used, the colorization is more likely to fail once an image from a new category has to be colorized.

Instead of predicting pixel chrominances directly, Larsson et al., 2016, and R. Zhang et al., 2016, predict color histograms. The idea lying behind is that many elements in a image can be colorized with many suitable colors. Therefore, they prefer predicting several possible colors per pixel before estimating the final color (as we did since (Bugeau et al., 2012)). The colorization process can be formulated:

$$u_c = \mathcal{G} \circ \mathcal{F}(u_T) \tag{4.2}$$

where $\mathcal{F} : \Omega \to [0,1]^N$ is a mapping function that associates to each pixel $p \in \Omega$ a probability distribution over N possible chrominance values[1]. The mapping function $\mathcal{G} : \Omega \times N \to \mathbb{R}^2$ estimates the final chrominance for each pixel given its own probability distribution. Let us explain how $\mathcal{F}$, $\mathcal{G}$ and N are defined in both papers.

In (Larsson et al., 2016), N is the number of bins resulting from the discretization of the luminance/chrominance (in Lab color system) axis by evenly spaced Gaussian quantiles. In (R. Zhang et al., 2016), they quantize the ab space into bins with grid size 10 and keep the N values which are in gamut (*i.e.* such that the corresponding Lab value belong to the space of existing values in this color space).

In both paper, the mapping functions $\mathcal{F}$ are learned with deep networks whose architectures are not discussed here. During training, the input are the grayscale images from a huge database and the output are the quantized histograms of the corresponding color images. In (Larsson et al., 2016), the loss function is the cross entropy while in (R. Zhang et al., 2016), it is a weighted version of this cross entropy. The weights are based on what they call the pixel color rarity and permits rare colors to be predicted.

Once the network has been learned, we can predict, for each pixel of an input grayscale image u_T, a distribution of chrominances: $h_c = \mathcal{F}(u_T)$. The final step of the algorithm is to estimate the final color $u_c(p) = \mathcal{G}\left(h_c(p)\right)$. In (Larsson et al., 2016), $\mathcal{G}$ is either the median or expectation of $h_c(p)$ leading to desaturated prediction. In (R. Zhang et al., 2016), annealed-mean are introduced:

$$\mathcal{G}\left(h_c(p)\right) = \frac{\sum_{i=1} h_c(p,i)^{1/T}.i}{\sum_{i=1} h_c(p,i)^{1/T}}$$

[1]Note the in (Larsson et al., 2016) the authors proposed to retrieve either the chrominances values in the Lab color space or the Hue/Chroma in the HSL (Hue-Saturation-Lightness) space but we do not detail further this point in this document

where $h_c(p, i)$ is the i-th bin of $h_c(p)$, and $T = 1/6$ is called the temperature.
The results are very impressive on a wide variety of natural images (figure 4.3). Some artefacts still remain: color bleeding over strong gradients, drab or unrealistic colors, etc.

Target | Pierre et al., 2015a | R. Zhang et al., 2016

Figure 4.3: Comparison of the colorization results from our models in RGB (Pierre et al., 2015a) and in YUV Pierre et al., 2014b and from the fully automatic CNN approach of R. Zhang et al., 2016 (files obtained using `http://demos.algorithmia.com/colorize-photos/`). CNNs provide very impressive results for this task but some artifacts still remain.

Constrained Deep Learning based models for image and video post-production.

While proposing impressive results with a very low computational time, current methods based on deep learning still have some limitations. First, we often observe loss in details and color bleeding or blurring along edges. A perspective is to design a framework combining Deep Learning

(either CNNs or autocoders), patch-based and variational approaches together. The framework should be robust to the type of degradation (*e.g.* type of noise in denoising, size of the hole in inpainting) and to be as generic as possible to be less dependent on the application. The corresponding models and loss functions should include *i)* spatial regularity constraints to avoid blur and color bleeding; *ii)* features constraints to avoid loss in details, *e.g.* by comparing the patch representations of inferred images with those of the groundtruth (in the same trend as the post-processing step in (Yang et al., 2017)); *iii)* visual quality constraints. Several research directions can be followed for this purpose. The first one consists in designing loss functions which include these constraints. Secondly, it is possible to define post-production as the problem of minimizing a variational model for which the data term (or the regularization term) is learnt with deep learning while the regularization term (resp. the data term)is given explicitly. A first work in that direction is Meinhardt et al., 2017, where the regularization term is modeled implicitely with a neural networks. Authors of Ulyanov et al., 2018 also claim that the term regularization term could be drop if the data term is minimized in a space of images generated by a simple CNN.

Secondly, results inferred by networks are highly dependant on the training images. In the context of image restoration, the data are obtained by degrading high resolution images to serve as groundtruth. The degradations applied may not represent correctly originally degraded data. For instance, in case of image colorization, grayscale images obtained by converting a RGB image into luminance are not completely representative of original grayscale images and this is even more true for old photographs (Figure 4.3, third line). One solution is to incorporate into the database images that have been manually restored. The other solution is to design methods that are able to corrupt appropriately the data, as done for instance in the work of (Newson et al., 2017) that approximates the physical reality of film grain.

Finally, there are currently very few works using deep learning for video post-production. Ruder et al., 2016 rewards consecutive frames that are consistent with each other by defining a temporal loss function based on optical flow computation (Revaud et al., 2015). We will follow this idea to extend deep learning based models from image to video post-production. In parallel, other networks such as Recurrent Neural Networks (RNN), which have shown to be very powerful for video classification, could be studied to be applied for video post-production.

4.4 Visual quality assessment

One difficulty when performing image restoration concerns the validation of the results. As soon as the problem is to recover missing data, it is very difficult to perform quantitative evaluation. It is possible to manually remove some contents from an image, restore and compare it to the original image with measures such as PSNR or SSIM (Wang et al., 2004). Nevertheless, as shown on figure 4.4, a metric such as PSNR does not characterize the plausibility of the result.

Up to now, the best way to validate the algorithms is to rely on naive human observers (*e.g.*

(a) Initial image (b) Image manually degraded: PSNR = 30.55. (c) Image re-colorized: PSNR = 30.51.

Figure 4.4: Non validity of PSNR for colorization (b): the red channel of the original image has been manually modified. (c): result of the colorization of the original image. Figure taken from (Pierre, 2016).

Turing test). The question generally asked are how realistic are the images. While this is the best evaluation possible, running such a test is highly time consuming and must be adapted each time a new method or experiment is performed. It would be interesting to design methods allowing for automatic validation. Some works already exist for color transfer application (Hristova et al., 2017) and could maybe be adapted for colorization. In another trend, Jin et al., 2016 propose to measure image aestethics using a CNN. Applications of the measures to image post-production can be considered. Furthermore such metric could be added as constraints in all image post-production models.

Bibliography

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P. and Süsstrunk, S. (2012). 'SLIC superpixels compared to state-of-the-art superpixel methods'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.11, pages 2274–2282 (see page xv).

Arbelot, B., Vergne, R., Hurtut, T. and Thollot, J. (2016). 'Automatic Texture Guided Color Transfer and Colorization'. In: *Expressive* (see pages 9, 20, 52).

Arbelot, B., Vergne, R., Hurtut, T. and Thollot, J. (2017). 'Local texture-based color transfer and colorization'. In: *Computers & Graphics* 62, pages 15–27 (see page 48).

Arias, P., Caselles, V. and Facciolo, G. (2012). 'Analysis of a Variational Framework for Exemplar-Based Image Inpainting'. In: *Multiscale Modeling & Simulation* 10.2, pages 473–514 (see pages 8, 38–39, 56–57).

Arias, P., Facciolo, G., Caselles, V. and Sapiro, G. (2011). 'A variational framework for exemplar-based image inpainting'. In: *International Journal of Computer Vision* 93.3, pages 319–347 (see pages 38–41).

Arya, S. and Mount, D. M. (1993). 'Approximate Nearest Neighbor Queries in Fixed Dimensions'. In: *ACM-SIAM Symposium on Discrete Algorithms* (see page 36).

Ashikhmin, M. (2001). 'Synthesizing Natural Textures'. In: *ACM Interactive 3D Graphics*, pages 217–226 (see page 8).

Attneave, F. (1954). 'Some informational aspects of visual perception.' In: *Psychological Review* 61.3, pages 183–193 (see page 1).

Aujol, J.-F., Ladjal, S. and Masnou, S. (2010). 'Exemplar-based inpainting from a variational point of view'. In: *SIAM Journal on Mathematical Analysis* 42.3, pages 1246–1285 (see pages 37, 56).

Ballester, C., Bertalmio, M., Caselles, V., Sapiro, G. and Verdera, J. (2001). 'Filling-in by joint interpolation of vector fields and grey levels'. In: *IEEE Transactions on Image Processing* 10, pages 1200–1211 (see page 26).

Barlow, H. B. (1961). 'Possible Principles Underlying the Transformations of Sensory Messages'. In: *Sensory Communication*. The MIT Press, pages 216–234 (see page 1).

Barnes, C., Shechtman, E., Finkelstein, A. and Goldman, D. (2009). 'PatchMatch: a randomized correspondence algorithm for structural image editing'. In: *ACM Transactions on Graphics* 28.3, page 24 (see pages 7–8, 17, 39).

Barnes, C. and Zhang, F.-L. (2017). 'A survey of the state-of-the-art in patch-based synthesis'. In: *Computational Visual Media* 3.1, pages 3–20 (see page 5).

Bay, H., Tuytelaars, T. and Van Gool, L. (2006). 'Surf: Speeded up robust features'. In: *European Conference on Computer Vision*. Springer, pages 404–417 (see pages xiv, 9, 20).

Bénard, P., Cole, F., Kass, M., Mordatch, I., Hegarty, J., Senn, M. S., Fleischer, K., Pesare, D. and Breeden, K. (2013). 'Stylizing Animation by Example'. In: *ACM Transactions on Graphics* 32.4, 119:1–119:12 (see page 5).

Benois-Pineau, J., Bugeau, A., Karaman, S. and Mégret, R. (2012). 'Visual Indexing and Retrieval'. In: edited by J. Benois-Pineau, F. Precisoso and M. Cord. Springer. Chapter Spatial and multi-resolution context in visual indexing (see page xiv).

Bertalmio, M. (2006). 'Strong-continuation, contrast-invariant inpainting with a third-order optimal PDE'. In: *IEEE Transactions on Image Processing* 15.7, pages 1934–1938 (see page 26).

Bertalmío, M., Caselles, V., Masnou, S. and Sapiro, G. (2011). 'Inpainting'. In: *Encyclopedia of Computer Vision* (see page 12).

Bertalmio, M., Sapiro, G., Caselles, V. and Ballester, C. (2000). 'Image Inpainting'. In: *ACM Transactions on Graphics* (see pages 12, 26, 34).

Bertalmio, M., Vese, L., Sapiro, G. and Osher, S. (2003). 'Simultaneous structure and texture image inpainting'. In: *IEEE Transactions on Image Processing* 12.8, pages 882–889 (see pages viii, 34).

Bevilacqua, M., Aujol, J.-F., Biasutti, P., Brédif, M. and Bugeau, A. (2017). 'Joint inpainting of depth and reflectance with visibility estimation'. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 125, pages 16–32 (see pages xi–xii, 56).

Bevilacqua, M., Aujol, J.-F., Bredif, M. and Bugeau, A. (2016). 'Visibility estimation and joint inpainting of lidar depth maps'. In: *IEEE International Conference on Image Processing*. Institute of Electrical and Electronics Engineers (IEEE) (see pages xi, 56).

Biasutti, P., Aujol, J.-F., Brédif, M. and Bugeau, A. (2016). 'Diffusion anisotrope et inpainting d'orthophotographies LiDAR mobile'. In: *Congrès national sur la Reconnaissance des Formes et l'Intelligence Artificielle* (see page xii).

Biasutti, P., Aujol, J.-F., Brédif, M. and Bugeau, A. (2017a). *Diffusion and inpainting of reflectance and height LiDAR orthoimages*. Technical report hal-01322822. preprint (see pages xii, 31).

Biasutti, P., Aujol, J.-F., Brédif, M. and Bugeau, A. (2017b). 'Disocclusion of 3D LiDAR point clouds using range images'. In: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* (see pages xii, 27–28).

Biasutti, P., Aujol, J.-F., Brédif, M. and Bugeau, A. (2018). 'Range-Image: Incorporating sensor topology for LIDAR point clouds processing'. In: *Photogrammetric Engineering & Remote Sensing* (see pages xii, 27–28, 30).

Birchfield, S. and Rangarajan, S. (2005). 'Spatiograms versus Histograms for Region-Based Tracking'. In: *IEEE Conference on Computer Vision and Pattern Recognition* (see page 14).

Bornard, R., Lecan, E., Laborelli, L. and Chenot, J.-H. (2002). 'Missing data correction in still images and image sequences'. In: *ACM international conference on Multimedia* (see page 12).

Bornemann, F. and März, T. (2007). 'Fast Image Inpainting Based on Coherence Transport'. In: *Journal of Mathematical Imaging and Vision* 28.3, pages 259–278 (see pages 26, 38–41).

Buades, A., Coll, B. and Morel, J.-M. (2005). 'A non-local algorithm for image denoising'. In: *IEEE Conference on Computer Vision and Pattern Recognition*. Volume 2, pages 60–65 (see page 5).

Bugeau, A. and Bertalmio, M. (2009). 'Combining texture synthesis and diffusion for image inpainting'. In: *International Conference on Computer Vision Theory and Applications* (see pages viii, 13, 34).

Bugeau, A. and Pérez, P. (2007). 'Joint Tracking and Segmentation of Objects using Graph Cuts'. In: *Conference on Advanced Concepts for Intelligent Vision Systems* (see page xiii).

Bugeau, A. and Pérez, P. (2008a). 'Track and Cut : simultaneous tracking and segmentation of multiple objects with graph cuts'. In: *International Conference on Computer Vision Theory and Applications* (see page xiii).

Bugeau, A., Bertalmío, M., Caselles, V. and Sapiro, G. (2010a). 'A Comprehensive Framework for Image Inpainting'. In: *IEEE Transactions on Image Processing* 19.10, pages 2634–2645 (see pages viii, 10, 13, 37, 39–41).

Bugeau, A., Gargallo, P., D'Hondt, O., Hervieu, A., Papadakis, N. and Caselles, V. (2010b). 'Coherent Background Video Inpainting through Kalman Smoothing along Trajectories'. In: *Vision, Modeling, and Visualization* (see page viii).

Bugeau, A. and Pérez, P. (2008b). 'Track and Cut: Simultaneous Tracking and Segmentation of Multiple Objects with Graph Cuts'. In: *EURASIP Journal on Image and Video Processing* (see page xiii).

Bugeau, A. and Ta, V.-T. (2012). 'Patch-based image colorization'. In: *International Conference on Pattern Recognition*, pages 3058–3061 (see pages ix, 9, 15, 17–18, 20–22, 45–46, 60).

Bugeau, A., Ta, V.-T. and Papadakis, N. (2014). 'Variational Exemplar-Based Image Colorization'. In: *IEEE Transactions on Image Processing* 23.1, pages 298–307 (see pages ix–x, 17, 45–48, 53).

Busto, P. P., Eisenacher, C., Lefebvre, S. and Stamminger, M. (2010). 'Instant Texture Synthesis by Numbers'. In: *Vision, Modeling, and Visualization (2010)*. Edited by R. Koch, A. Kolb and C. Rezk-Salama. The Eurographics Association (see page 8).

Buyssens, P., Daisy, M., Tschumperlé, D. and Lézoray, O. (2015). 'Exemplar-Based Inpainting: Technical Review and New Heuristics for Better Geometric Reconstructions'. In: *IEEE Transactions on Image Processing* 24.6, pages 1809–1824 (see pages 12–13).

Castorena, J., Kamilov, U. and Boufounos, P. (2016). 'Autocalibration of LIDAR and Optical Cameras via Edge Alignment'. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing* (see page 56).

Chambolle, A. and Pock, T. (2011). 'A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging'. In: *Journal of Mathematical Imaging and Vision* 40.1, pages 120–145 (see page 47).

Chan, T., Kang, S. and Shen, J. (2002). 'Euler's elastica and curvature based inpaintings'. In: *Journal of Applied Mathematics* 63, pages 564–592 (see page 26).

Chan, T., Osher, S. and Shen, J. (2001a). 'The digital TV filter and nonlinear denoising'. In: *IEEE Transactions on Image Processing* 10.2, pages 231–241 (see page 26).

Chan, T. and Shen, J. (2001b). 'Nontexture Inpainting by Curvature-Driven Diffusions'. In: *Journal of Visual Communication and Image Representation* 12.4, pages 436–449 (see page 26).

Charpiat, G., Hofmann, M. and Schölkopf, B. (2008). 'Automatic Image Colorization via Multimodal Predictions'. In: *European Conference on Computer Vision*, pages 126–139 (see pages 15, 17, 21–22, 51, 53).

Cheng, Z., Yang, Q. and Sheng, B. (2015). 'Deep colorization'. In: *IEEE International Conference on Computer Vision*, pages 415–423 (see pages 9, 59).

Chia, A. Y.-S., Zhuo, S., Gupta, R. K., Tai, Y.-W., Cho, S.-Y., Tan, P. and Lin, S. (2011). 'Semantic Colorization with Internet Images'. In: *ACM SIGGRAPH ASIA* (see page 9).

Coupé, P., Manjón, J. V., Fonov, V., Pruessner, J., Robles, M. and Collins, D. L. (2011). 'Patch-based segmentation using expert priors: Application to hippocampus and ventricle segmentation'. In: *NeuroImage* 54.2, pages 940–954 (see page 5).

Criminisi, A., Pérez, P. and Toyama, K. (2004). 'Region filling and object removal by exemplar-based inpainting'. In: *IEEE Transactions on Image Processing* 13.9, pages 1200–1212 (see pages 7, 13, 15, 22, 27, 32–33, 39–41).

Deledalle, C.-A., Denis, L. and Tupin, F. (2012). 'How to Compare Noisy Patches? Patch Similarity Beyond Gaussian Noise'. In: *International Journal of Computer Vision* 99.1, pages 86–102 (see page 11).

Delon, J., Desolneux, A., Lisani, J.-L. and Petro, A. B. (2007). 'A nonparametric approach for histogram segmentation'. In: *IEEE Transactions on Image Processing* 16 (see page 29).

Demanet, L., Song, B. and Chan, T. (2003). *Image Inpainting by Correspondence Maps: a Deterministic Approach*. Technical report 03-04. UCLA CAM R (see pages 7, 23, 35).

Deshpande, A., Rock, J. and Forsyth, D. (2015). 'Learning Large-Scale Automatic Image Colorization'. In: *IEEE International Conference on Computer Vision* (see page 59).

Di Blasi, G. and Reforgiato, D. (2003). 'Fast colorization of gray images'. In: *Eurographics Italian* (see page 8).

Di Zenzo, S. (1986). 'A Note on the Gradient of a Multi-image'. In: *Computer Vision, Graphics, and Image Processing* 33.1, pages 116–125 (see page 13).

Ding, X., Xu, Y., Deng, L. and Yang, X. (2012). 'Colorization Using Quaternion Algebra with Automatic Scribble Generation.' In: *Advances in Multimedia Modeling* (see page 45).

Dong, C., Loy, C. C., He, K. and Tang, X. (2014). 'Learning a deep convolutional network for image super-resolution'. In: *European Conference on Computer Vision*, pages 184–199 (see page 57).

Drew, M. S. and Finlayson, G. D. (2011). 'Improvement of Colorization Realism via the Structure Tensor.' In: *International Journal on Image Graphics* 11.4, pages 589–609 (see page 45).

Drori, I., Cohen-Or, D. and Yeshurun, H. (2003). 'Fragment-based image completion'. In: *ACM Transactions on Graphics* 22.3, page 303 (see pages 7, 12).

Ebdelli, M., Meur, O. L. and Guillemot, C. (2013). 'Analysis of patch-based similarity metrics: Application to denoising'. In: *IEEE International Conference on Acoustics, Speech and Signal Processing* (see page 11).

Efros, A. and Freeman, W. (2001). 'Image Quilting for Texture Synthesis and Transfer'. In: *ACM Transactions on Graphics* (see page 7).

Efros, A. and Leung, T. (1999). 'Texture Synthesis by Non-parametric Sampling'. In: *IEEE International Conference on Computer Vision* (see pages ix, 2, 5–7, 10, 12, 16, 34, 56).

Elad, M. and Milanfar, P. (2017). 'Style Transfer Via Texture Synthesis'. In: *IEEE Transactions on Image Processing* 26.5, pages 2338–2351 (see page 5).

Elad, M., Starck, J., Querre, P. and Donoho, D. (2005). 'Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA)'. In: *Applied and Computational Harmonic Analysis* 19.3, pages 340–358 (see page 34).

Emile-Male, G. and Underwood, J. A. (1976). *The restorer's handbook of easel painting.* New York ; London [etc.] : Van Nostrand Reinhold (see page 12).

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J. and Zisserman, A. (2007). 'The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results'. In: *International Journal of Computer Vision* (see page 58).

Fadili, M., Starck, J.-L. and Murtagh, F. (2009). 'Inpainting and Zooming Using Sparse Representations'. In: *Computer Journal* 52.1, pages 64–79 (see page 34).

Fedorov, V., Arias, P., Facciolo, G. and Ballester, C. (2016). 'Affine Invariant Self-Similarity for Exemplar-Based Inpainting'. In: *International Conference on Computer Vision Theory and Applications* (see page 39).

Fedorov, V., Facciolo, G. and Arias, P. (2015). 'Variational Framework for Non-Local Inpainting'. In: *Image Processing On Line* 5, pages 362–386 (see page 40).

Fortun, D., Bouthemy, P. and Kervrann, C. (2016). 'Aggregation of local parametric candidates with exemplar-based occlusion handling for optical flow'. In: *Computer vision and image understanding* 145, pages 81–94 (see pages 5, 46).

Freeman, W. T., Jones, T. R. and Pasztor, E. C. (2002). 'Example-based super-resolution'. In: *IEEE Computer Graphics and Applications* 22.2, pages 56–65 (see page 5).

Frigo, O., Sabater, N., Delon, J. and Hellier, P. (2016). 'Split and Match: Example-Based Adaptive Patch Sampling for Unsupervised Style Transfer'. In: *IEEE Conference on Computer Vision and Pattern Recognition* (see page 5).

Gatys, L. A., Ecker, A. S. and Bethge, M. (2015). 'Texture Synthesis Using Convolutional Neural Networks'. In: *Conference on Neural Information Processing Systems* (see page 58).

Gatys, L. A., Ecker, A. S. and Bethge, M. (2016). 'Image style transfer using convolutional neural networks'. In: *IEEE Conference on Computer Vision and Pattern Recognition* (see page 57).

Geiger, A., Lenz, P., Stiller, C. and Urtasun, R. (2013). 'Vision meets Robotics: The KITTI Dataset'. In: *International Journal of Robotics Research* (see pages 29–30).

Gilboa, G. and Osher, S. (2007). 'Nonlocal linear image regularization and supervised segmentation'. In: *SIAM Multiscale Modeling and Simulation* 6, pages 595–630 (see page 38).

Gilboa, G. and Osher, S. (2008). 'Nonlocal operators with applications to image processing'. In: *Multiscale Modeling & Simulation* 7.3, pages 1005–1028 (see page 38).

Giraud, R., Ta, V.-T., Bugeau, A., Coupé, P. and Papadakis, N. (2017). 'SuperPatchMatch: an Algorithm for Robust Correspondences of Superpixel Patches'. In: *IEEE Transactions on Image Processing* (see pages xv–xvi, 8).

Gonzalez, R. C. and Woods, R. E. (2007). *Digital image processing 3rd edition* (see page 50).

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2014). 'Generative Adversarial Nets'. In: *Advances in Neural Information Processing Systems* (see page 58).

Granados, M., Tompkin, J., Kim, K., Grau, O., Kautz, J. and Theobalt, C. (2012). 'How Not to Be Seen - Object Removal from Videos of Crowded Scenes'. In: *Computer Graphics Forum* 31.2, pages 219–228 (see page 41).

Guillemot, C. and Le Meur, O. (2014). 'Image Inpainting : Overview and Recent Advances'. In: *IEEE Signal Processing Magazine* 31.1, pages 127–144 (see page 12).

Gupta, R. K., Chia, A. Y.-S., Rajan, D., Ng, E. S. and Zhiyong, H. (2012). 'Image colorization using similar images'. In: *ACM International Conference on Multimedia*, pages 369–378 (see pages 9, 20, 45, 48, 51, 53).

Gutierrez Ortega, J. A., Rabin, J., Galerne, B. and Hurtut, T. (2017). 'Optimal Patch Assignment for Statistically Constrained Texture Synthesis'. In: *Scale Space and Variational Methods in Computer Vision* (see page 57).

Hays, J. and Efros, A. (2007). 'Scene Completion Using Millions of Photographs'. In: *ACM Transactions on Graphics* 26.3 (see page 41).

He, K. and Sun, J. (2012). 'Computing nearest-neighbor fields via Propagation-Assisted KD-Trees'. In: *IEEE Conference on Computer Vision and Pattern Recognition* (see page 8).

Hertzmann, A., Jacobs, C. E., Oliver, N., Curless, B. and Salesin, D. H. (2001). 'Image analogies'. In: *ACM Computer graphics and interactive techniques*, pages 327–340 (see page 16).

Hervieu, A., Papadakis, N., Bugeau, A., Gargallo, P. and Caselles, V. (2010). 'Stereoscopic Image Inpainting: Distinct Depth Maps and Images Inpainting'. In: *International Conference on Pattern Recognition* (see pages viii, 5).

Hervieu, A., Papadakis, N., Bugeau, A., Gargallo, P. and Caselles, V. (2011). 'Stereoscopic image inpainting using scene geometry'. In: *IEEE International Conference on Multimedia and Expo* (see pages viii, 5).

Heu, J., Hyun, D.-Y., Kim, C.-S. and Lee, S.-U. (2009). 'Image and video colorization based on prioritized source propagation.' In: *IEEE International Conference on Image Processing* (see page 44).

Hristova, H., Meur, O. L., Cozot, R. and Bouatouch, K. (2017). 'Perceptual Metric for color transfer methods'. In: *IEEE International Conference on Image Processing* (see page 63).

Iizuka, S., Simo-Serra, E. and Ishikawa, H. (2016). 'Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification'. In: *ACM Transactions on Graphics* 35.4 (see page 60).

Irony, R., Cohen-Or, D. and Lischinski, D. (2005). 'Colorization by example'. In: *Eurographics conference on Rendering Techniques*. Eurographics Association, pages 201–210 (see pages 45, 51, 53).

Jin, B., Segovia, M. V. O. and Süsstrunk, S. (2016). 'Image aesthetic predictors based on weighted CNNs'. In: *IEEE International Conference on Image Processing* (see page 63).

Karaman, S., Benois-Pineau, J., Mégret, R. and Bugeau, A. (2012). 'Multi-layer Local Graph Words for Object Recognition'. In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, pages 29–39 (see page xiv).

Kawulok, M., Kawulok, J. and Smolka, B. (2012). 'Discriminative Textural Features for Image and Video Colorization'. In: *IEICE Transaction on Information and Systems* 95-D.7, pages 1722–1730 (see page 45).

Kim, Y.-T. (1997). 'Contrast enhancement using brightness preserving bi-histogram equalization'. In: *IEEE Transactions on Consumer Electronics* 43.1, pages 1–8 (see page 14).

Kolmogorov, V. and Zabin, R. (2004). 'What energy functions can be minimized via graph cuts?' In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26.2, pages 147–159 (see page 36).

Komodakis, N. and Tziritas, G. (2006). 'Image Completion Using Global Optimization'. In: *IEEE Conference on Computer Vision and Pattern Recognition* (see pages 36, 38, 40).

Kwatra, V., Essa, I., Bobick, A. and Kwatra, N. (2005). 'Texture optimization for example-based synthesis'. In: *ACM Transactions on Graphics* (see pages 23, 35).

Kwatra, V., Schödl, A., Essa, I., Turk, G. and Bobick, A. (2003). 'Graphcut Textures: Image and Video Synthesis Using Graph Cuts'. In: *ACM Transactions on Graphics* (see page 7).

Lagodzinski, P. and Smolka, B. (2008). 'Digital image colorization based on probabilistic distance transformation'. In: *50th International Symposium ELMAR*. Volume 2, pages 495–498 (see page 45).

Larsson, G., Maire, M. and Shakhnarovich, G. (2016). 'Learning Representations for Automatic Colorization'. In: *European Conference on Computer Vision*, pages 1–16 (see page 60).

Lazebnik, S., Schmid, C. and Ponce, J. (2006). 'Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories'. In: *IEEE Conference on Computer Vision and Pattern Recognition* (see page xiv).

Le Meur, O., Gautier, J. and Guillemot, C. (2011). 'Examplar-based inpainting based on local geometry'. In: *IEEE International Conference on Image Processing* (see page 13).

Le Meur, O. and Guillemot, C. (2012). 'Super-Resolution-Based Inpainting'. In: *European Conference on Computer Vision*, pages 554–567 (see page 11).

Lefebvre, S. and Hoppe, H. (2005). 'Parallel controllable texture synthesis'. In: *ACM Transactions on Graphics* (see page 7).

Letournel, G., Bugeau, A., Ta, V.-T. and Domenger, J.-P. (2015). 'Face de-identification with expressions preservation'. In: *IEEE International Conference on Image Processing* (see page xvii).

Levin, A., Lischinski, D. and Weiss, Y. (2004). 'Colorization using optimization'. In: *ACM Transactions on Graphics* 23.3, pages 689–694 (see pages ix, 14, 44–45).

Lézoray, O., Ta, V.-T. and Elmoataz, A. (2008). 'Nonlocal graph regularization for image colorization'. In: *International Conference on Pattern Recognition*, pages 1–4 (see page 45).

Liu, G., Gousseau, Y. and Xia, G. S. (2016). 'Texture synthesis through convolutional neural networks and spectrum constraints'. In: *International Conference on Pattern Recognition* (see pages 57–58).

Liu, X., Wan, L., Qu, Y., Wong, T.-T., Lin, S., Leung, C.-S. and Heng, P.-A. (2008). 'Intrinsic colorization'. In: *ACM Transactions on Graphics* 27.5, page 1 (see pages 43, 52).

Liu, Y. and Caselles, V. (2013). 'Exemplar-Based Image Inpainting Using Multiscale Graph Cuts'. In: *IEEE Transactions on Image Processing* 22.5, pages 1699–1711 (see pages 9, 36).

Lou, Y., Favaro, P., Soatto, S. and Bertozzi, A. (2009). 'Nonlocal Similarity Image Filtering'. In: *International Conference on Image Analysis and Processing*, pages 62–71 (see page 9).

Lowe, D. G. (1999). 'Object recognition from local scale-invariant features'. In: *IEEE International Conference on Computer Vision*. Volume 2, pages 1150–1157 (see page 9).

Lu, J., Yang, H., Min, D. and Do, M. N. (2013). 'Patch Match Filter: Efficient Edge-Aware Filtering Meets Randomized Search for Fast Correspondence Field Estimation'. In: *IEEE Conference on Computer Vision and Pattern Recognition* (see page 5).

Mairal, J., Elad, M. and Sapiro, G. (2008). 'Sparse representation for color image restoration'. In: *IEEE Transactions on Image Processing* 17, pages 53–69 (see page 34).

Mao, X., Shen, C. and Yang, Y.-B. (2016). 'Image Restoration Using Very Deep Convolutional Encoder-Decoder Networks with Symmetric Skip Connections'. In: *Advances in Neural Information Processing Systems* (see page 57).

Masnou, S. and Morel, J. M. (1998). 'Level Lines based Disocclusion'. In: *IEEE International Conference on Image Processing*, pages 259–263 (see pages 12, 26).

Meinhardt, T., Moller, M., Hazirbas, C. and Cremers, D. (2017). 'Learning Proximal Operators: Using Denoising Networks for Regularizing Inverse Imaging Problems'. In: *arXiv preprint: 1704.03488* (see page 62).

Ming, Z., Bugeau, A., Rouas, J.-L. and Shochi, T. (2015). 'Facial Action Units Intensity Estimation by the Fusion of Features with Multi-kernel Support Vector Machine'. In: *IEEE International Conference on Automatic Face and Gesture Recognition Conference and Workshops* (see page xviii).

Mumford, D. (1994). 'Algebraic Geometry and its Applications'. In: Chandrajit Bajaj, New York, Springer-Verlag. Chapter Elastica and Computer Vision, pages 491–506 (see page 26).

Newson, A., Galerne, B. and Delon, J. (2017). 'A stochastic film-grain model for resolution-independent rendering'. In: *Computer Graphics Forum* (see page 62).

Newson, A. (2014). 'On Video Completion: Line Scratch Detection in Films and Video Inpainting of Complex Scenes'. Theses. Telecom ParisTech (see pages 10, 12).

Newson, A., Almansa, A., Fradet, M., Gousseau, Y. and Pérez, P. (2014). 'Video inpainting of complex scenes'. In: *SIAM Journal on Imaging Sciences* 7.4, pages 1993–2019 (see pages 9, 39, 41).

Nikolova, M. and Steidl, G. (2014). 'Fast ordering algorithm for exact histogram specification'. In: *IEEE Transactions on Image Processing* 23.12, pages 5274–5283 (see page 14).

Nishida, T., Abe, M., Ookaki, T., Lala, D., Thovuttikul, S., Song, H., Mohammad, Y., Nitschke, C., Ohmoto, Y., Nakazawa, A., Shochi, T., Rouas, J.-L., Bugeau, A., Lotte, F., Ming, Z., Letournel, G., Guerry, M. and Fourer, D. (2015). 'Synthetic Evidential Study as Augmented Collective Thought Process'. In: *International scientific conference for research in the field of intelligent information and database systems* (see page xviii).

Nitzberg, M., Mumford, D. and Shiota, T. (1993). *Filtering, Segmentation, and Depth*. Springer-Verlag, Berlin (see page 26).

Olonetsky, I. and Avidan, S. (2012). 'TreeCANN - k-d Tree Coherence Approximate Nearest Neighbor Algorithm'. In: *European Conference on Computer Vision*, pages 602–615 (see page 8).

Papadakis, N., Bugeau, A. and Caselles, V. (2012). 'Image Editing With Spatiograms Transfer'. In: *IEEE Transactions on Image Processing* 21.5, pages 2513–2522 (see pages 14–15).

Papadakis, N. and Bugeau, A. (2011). 'Tracking with Occlusions via Graph Cuts'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.1, pages 144–157 (see pages xiii–xiv).

Paparoditis, N., Papelard, J.-P., Cannelle, B., Devaux, A., Soheilian, B., David, N. and Houzay, E. (2012). 'Stereopolis II: A multi-purpose and multi-sensor 3D mobile mapping system for street visualisation and 3D metrology'. In: *Revue française de photogrammétrie et de télédétection* 200 (see page 28).

Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T. and Efros, A. (2016). 'Context Encoders: Feature Learning by Inpainting'. In: *IEEE Conference on Computer Vision and Pattern Recognition* (see pages 41, 57–59).

Pérez, P., Gangnet, M. and Blake, A. (2003). 'Poisson image editing'. In: *ACM Transactions on Graphics* (see page 13).

Perona, P. and Malik, J. (1990). 'Scale-space and edge detection using anisotropic diffusion'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12.7, pages 629–639 (see page 32).

Petschnigg, G., Szeliski, R., Agrawala, M., Cohen, M., Hoppe, H. and Toyama, K. (2004). 'Digital Photography with Flash and No-flash Image Pairs'. In: *ACM Transactions on Graphics* 23.3, pages 664–672 (see page 59).

Peyre, G., Fadili, J. and Starck, J. (2007). 'Learning adapted dictionaries for geometry and texture separation'. In: *SPIE The International Society for Optical Engineering*. Volume 6701. 2 (see page 34).

Pierre, F., Aujol, J.-F., Bugeau, A., Papadakis, N. and Ta, V.-T. (2015a). 'Luminance-Chrominance Model for Image Colorization'. In: *SIAM Journal on Imaging Sciences* 8.1, pages 536–563 (see pages x, 19, 45, 47, 61).

Pierre, F. (2016). 'Methodes variationnelles pour la colorisation d'images, de videos, et la Correction des couleurs'. PhD thesis. Universite de Bordeaux (see page 63).

Pierre, F., Aujol, J.-F., Bugeau, A., Steidl, G. and Ta, V.-T. (2016). 'Hue-preserving perceptual contrast enhancement'. In: *IEEE International Conference on Image Processing* (see pages x, 14).

Pierre, F., Aujol, J.-F., Bugeau, A., Steidl, G. and Ta, V.-T. (2017a). 'Variational Contrast Enhancement of Gray-Scale and RGB Images'. In: *Journal of Mathematical Imaging and Vision* 57.1, pages 99–116 (see pages x–xi, 17).

Pierre, F., Aujol, J.-F., Bugeau, A. and Ta, V.-T. (2014a). 'A Unified Model for Image Colorization'. In: *European Conference on Computer Vision, CPCV Workshop*, pages 1–12 (see pages x, 43, 45, 47–48, 53).

Pierre, F., Aujol, J.-F., Bugeau, A. and Ta, V.-T. (2015b). 'Combinaison lineaire optimale de metriques pour la colorisation d'images.' In: *GRETSI*, pages 1–4 (see page ix).

Pierre, F., Aujol, J.-F., Bugeau, A. and Ta, V.-T. (2015c). 'Luminance-Hue Specification in the RGB Space'. In: *Scale Space and Variational Methods in Computer Vision*, pages 413–424 (see pages x, 51, 53).

Pierre, F., Aujol, J.-F., Bugeau, A. and Ta, V.-T. (2017b). 'Interactive Video Colorization within a Variational Framework'. In: *Journal of Mathematical Imaging and Vision* (see pages x, 48).

Pierre, F., Aujol, J.-F., Bugeau, A., Ta, V.-T. and Papadakis, N. (2014b). 'Exemplar-based colorization in RGB color space'. In: *IEEE International Conference on Image Processing*, pages 1–5 (see pages x, 17, 49, 61).

Pritch, Y., Kav-Venaki, E. and Peleg, S. (2009). 'Shift-map image editing'. In: *IEEE International Conference on Computer Vision* (see page 36).

Ren, X. and Malik, J. (2003). 'Learning a classification model for segmentation'. In: *IEEE International Conference on Computer Vision* (see page 45).

Ren, Y., Benois-Pineau, J. and Bugeau, A. (2014a). 'A Comparative Study of Irregular Pyramid Matching in Bag-of-Bags of Words Model for Image Retrieval'. In: *Lecture Notes in Computer Science*. Springer International Publishing, pages 539–548 (see pages xiv–xv).

Ren, Y., Bugeau, A. and Benois-Pineau, J. (2014b). 'Bag-of-bags of words irregular graph pyramids vs spatial pyramid matching for image retrieval'. In: *International Conference on Image Processing Theory, Tools and Applications* (see pages xiv–xv).

Revaud, J., Weinzaepfel, P., Harchaoui, Z. and Schmid, C. (2015). 'EpicFlow: Edge-Preserving Interpolation of Correspondences for Optical Flow'. In: *IEEE Conference on Computer Vision and Pattern Recognition* (see page 62).

Ruder, M., Dosovitskiy, A. and Brox, T. (2016). 'Artistic style transfer for videos'. In: *German Conference on Pattern Recognition* (see page 62).

Ruderman, D. L., Cronin, T. W. and Chiao, C.-C. (1998). 'Statistics of cone responses to natural images: Implications for visual coding'. In: *Journal of the Optical Society of America A* 15.8, pages 2036–2045 (see page 15).

Scharstein, D. and Szeliski, R. (2002). 'A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms'. In: *International Journal of Computer Vision* 47.1, pages 7–42 (see page 5).

Shannon, C. E. (1948). 'A Mathematical Theory of Communication'. In: *Bell System Technical Journal* 27.4, pages 623–656 (see page 2).

Shechtman, E. and Irani, M. (2007). 'Matching Local Self-Similarities across Images and Videos'. In: *IEEE Conference on Computer Vision and Pattern Recognition* (see page 2).

Shochi, T., Rouas, J.-L., Ming, Z., Guerry, M., Bugeau, A. and Donna, E. (2016). 'Cultural differences in pattern matching : multisensory recognition of socio-affective prosody'. In: *International Congress of Psychology* (see page xvii).

Simoncelli, E. P. and Olshausen, B. A. (2001). 'Natural image statistics and neural representation'. In: *Annual Review of Neuroscience* 24.1, pages 1193–1216 (see page 1).

Simonyan, K. and Zisserman, A. (2014). 'Very Deep Convolutional Networks for Large-Scale Image Recognition'. In: *arXiv preprint: 1409.1556* (see page 58).

Sun, J., Yuan, L., Jia, J. and Shum, H.-Y. (2005). 'Image completion with structure propagation'. In: *ACM Transactions on Graphics* 24.3, page 861 (see page 13).

Tola, E., Lepetit, V. and Fua, P. (2008). 'A fast local descriptor for dense matching'. In: *IEEE Conference on Computer Vision and Pattern Recognition* (see page 9).

Tong, X., Zhang, J., Liu, L., Wang, X., Guo, B. and Shum, H.-Y. (2002). 'Synthesis of bidirectional texture functions on arbitrary surfaces'. In: *ACM Transactions on Graphics* (see page 8).

Tschumperlé, D. (2006). 'Fast Anisotropic Smoothing of Multi-Valued Images using Curvature-Preserving PDE's'. In: *International Journal of Computer Vision* 68.1, pages 65–82 (see pages 26–27, 39–41).

Tuzel, O., Porikli, F. and Meer, P. (2006). 'Region Covariance: A Fast Descriptor for Detection and Classification'. In: *European Conference on Computer Vision* (see page 9).

Ulyanov, D., Vedaldi, A. and Lempitsky, V. (2018). 'Deep Image Prior'. In: *IEEE Conference on Computer Vision and Pattern Recognition* (see page 62).

Vallet, B. and Papelard, J.-P. (2015). 'Road orthophoto/DTM generation from mobile laser scanning'. In: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* 3 (see page 31).

Vese, L. and Osher, S. (2003). 'Modeling Textures with Total Variation Minimization and Oscillating Patterns in Image Processing'. In: *Journal scientific Computing* 19.1-3, pages 553–572 (see pages 13, 34).

Wang, Z., Bovik, A. C., Sheikh, H. R. and Simoncelli, E. P. (2004). 'Image Quality Assessment: From Error Visibility to Structural Similarity'. In: *IEEE Transactions on Image Processing* 13.4, pages 600–612 (see page 62).

Wei, L. and Levoy, M. (2000). 'Fast texture synthesis using tree-structured vector quantization'. In: *ACM Transactions on Graphics* (see pages 7–8).

Wei, L.-Y., Lefebvre, S., Kwatra, V. and Turk, G. (2009). 'State of the Art in Example-based Texture Synthesis'. In: *Eurographics 2009, State of the Art Report, EG-STAR* (see pages 5–6).

Welsh, T., Ashikhmin, M. and Mueller, K. (2002). 'Transferring color to greyscale images'. In: *ACM Transactions on Graphics* 21.3, pages 277–280 (see pages ix, 15–17, 19, 21–22, 48, 51, 53).

Wexler, Y., Shechtman, E. and Irani, M. (2004). 'Space-time video completion'. In: *IEEE Conference on Computer Vision and Pattern Recognition* (see pages 35–36).

Williams, A., Barrus, S., Morley, R. K. and Shirley, P. (2005). 'An efficient and robust ray-box intersection algorithm'. In: *ACM SIGGRAPH 2005 Courses*, page 9 (see page 51).

Xiao, C., Liu, M., Yongwei, N. and Dong, Z. (2011). 'Fast Exact Nearest Patch Matching for Patch-Based Image Editing and Processing'. In: *IEEE Transactions on Visualization and Computer Graphics* 17.8, pages 1122–1134 (see page 7).

Xu, Z. and Sun, J. (2010). 'Image Inpainting by Patch Propagation Using Patch Sparsity'. In: *IEEE Transactions on Image Processing* 19.5, pages 1153–1165 (see page 13).

Yang, C., Lu, X., Lin, Z., Shechtman, E., Wang, O. and Li, H. (2017). 'High-Resolution Image Inpainting using Multi-Scale Neural Patch Synthesis'. In: *IEEE Conference on Computer Vision and Pattern Recognition* (see pages 41, 58, 62).

Yatziv, L. and Sapiro, G. (2006). 'Fast image and video colorization using chrominance blending'. In: *IEEE Transactions on Image Processing* 15.5, pages 1120–1129 (see pages 15, 44).

Zhang, K., Zuo, W., Chen, Y., Meng, D. and Zhang, L. (2017). 'Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising'. In: *IEEE Transactions on Image Processing* 26.7, pages 3142–3155 (see page 57).

Zhang, R., Isola, P. and Efros, A. A. (2016). 'Colorful Image Colorization'. In: *European Conference on Computer Vision*, pages 1–16 (see pages 43, 52, 57, 60–61).