

Introduction *,**

Objectif

- ▶ **Supprimer des objets large** définis par l'utilisateur, à partir d'images numériques.
- ▶ Corrections d'artefacts ou Restauration d'image
- ▶ Rendu réaliste ou vraisemblable pour l'œil humain

Principe

- ▶ Remplir des régions masquées, par **propagation de texture** le long des structures linéaires
- ▶ Utilisation du Principe de **connectivité** pour définir l'ordre de remplissage : propager les textures tout en conservant les structures linéaires de l'image
- ▶ Recherche de **patch similaire** par analyse couleur

Historique / État de l'art



Exemple. De gauche à droite. Originale, Suppression de la région, Remplissage de textures.

Méthode

Respect du principe de connectivité et les 3 points de l'algo

Algorithme en 3 étapes

- ▶ Calcul des priorités en bordure du masque
 - ▷ Terme de données * Confiance
 - ▷ Terme de données : Quantité de variation des textures autour du pixel courant
 - ▷ Confiance : importance accordée au pixel courantfigures exemples avec les ovales
 - ▶ Propager texture et information structurelle
 - ▷ La priorité est donnée aux zones de fortes variations (terme de données)
 - ▷ Propager la texture en commençant par ses zones permet de conserver les structures linéaires de l'image
 - ▷ les données de textures manquantes autour du pixel de plus forte priorité sont issues de la zone de l'image la plus ressemblante (SSD dans l'espace couleur Lab)figures exemples avec le test priority carre si on a un truc pas trop moche en résultat
 - ▶ Mise à jour de la confiance : les nouveaux pixels copiés obtiennent la confiance du pixel courant de sorte que sa valeur diminue vers l'intérieur du masque indiquant l'incertitude sur la valeur du pixel
- figures d'actualisation de la confiance

Résultats

Ce qui marche et comment

résultat sur les ovales , sur le test priority, sur des vrais images

Les limitations de notre algo : la montgolfiere , les patchs opposés ?