

MTH6412B: Projet Voyageur de Commerce

Phase 5: Reconstruction d'images

• Auteur: El Hadji Abdou Aziz Ndiaye (1879468)

• Code source [Répertoire Github](#)

Importation du code

```
reconstruct_picture
Read a tour file and a shuffle image, and output the image reconstructed using the tour.

begin
    using Plots
    using Printf
    using Random
    using FileIO
    using Images
    using ImageView
    using ImageMagick
    include("project/phases5/node.jl")
    include("project/phases5/edge.jl")
    include("project/phases5/read_tour.jl")
    include("project/phases5/test_tour.jl")
    include("project/phases5/mst_kruskal.jl")
    include("project/phases5/mst_prim.jl")
    include("project/phases5/tsp_rsl.jl")
    include("project/phases5/tsp_hk.jl")
    include("project/phases5/tsp.jl")
    include("project/phases5/im_reconstruct.jl")
    include("project/phases5/test_node.jl")
    include("project/phases5/tests/test_edge.jl")
    include("project/phases5/tests/test_tour.jl")
    include("project/phases5/tests/test_mst_kruskal.jl")
    include("project/phases5/tests/test_mst_prim.jl")
    include("project/phases5/tests/test_tsp_rsl.jl")
    include("project/phases5/tests/test_tsp_hk.jl")
    include("shredder-julia/bin/tools.jl")
end
```

Révision du code de la phase 4

Le code de la phase 4 a été entièrement repris dans la phase 5.

L'algorithme de Held et Karp a été légèrement modifié pour sortir une tournée à la place d'un 1-tree.

Tests des algorithmes de la phase 4

```
Test de la structure de données 'Node' : -v
Test de la structure de données 'Edge' : -v
Test de la structure de données 'Graph' : -v
Test de la structure de données 'ConnectedComponents' : -v
Test de l'algorithme de Kruskal : -v
Test de l'algorithme de Prim : -v
Test de l'algorithme de RSL : -v
Test de l'algorithme HK : -v
```

```
with terminal do
    run.test_node()
    run.test_edge()
    run.test_graph()
    run.test_connected_components()
    run.test_kruskal()
    run.test_prim()
    run.test_tsp_rsl()
    run.test_tsp_hk()
end
```

Reconstruction des images

La reconstruction des images se fait en utilisant directement les algorithmes de RSL et de HK.

Pour l'algorithme de RSL, les poids des arêtes reliées au noeud zéro ont été ajustés pour éviter d'avoir un arbre de recouvrement trivial.

Dans certains cas, le tour a été inversé (grâce à la fonction `inverse_tour` pour obtenir des résultats fidèles avec les images originaux.

L'ensemble des résultats ci-dessous peuvent être obtenu en lancant le programme principal.

Abstract light painting

Original

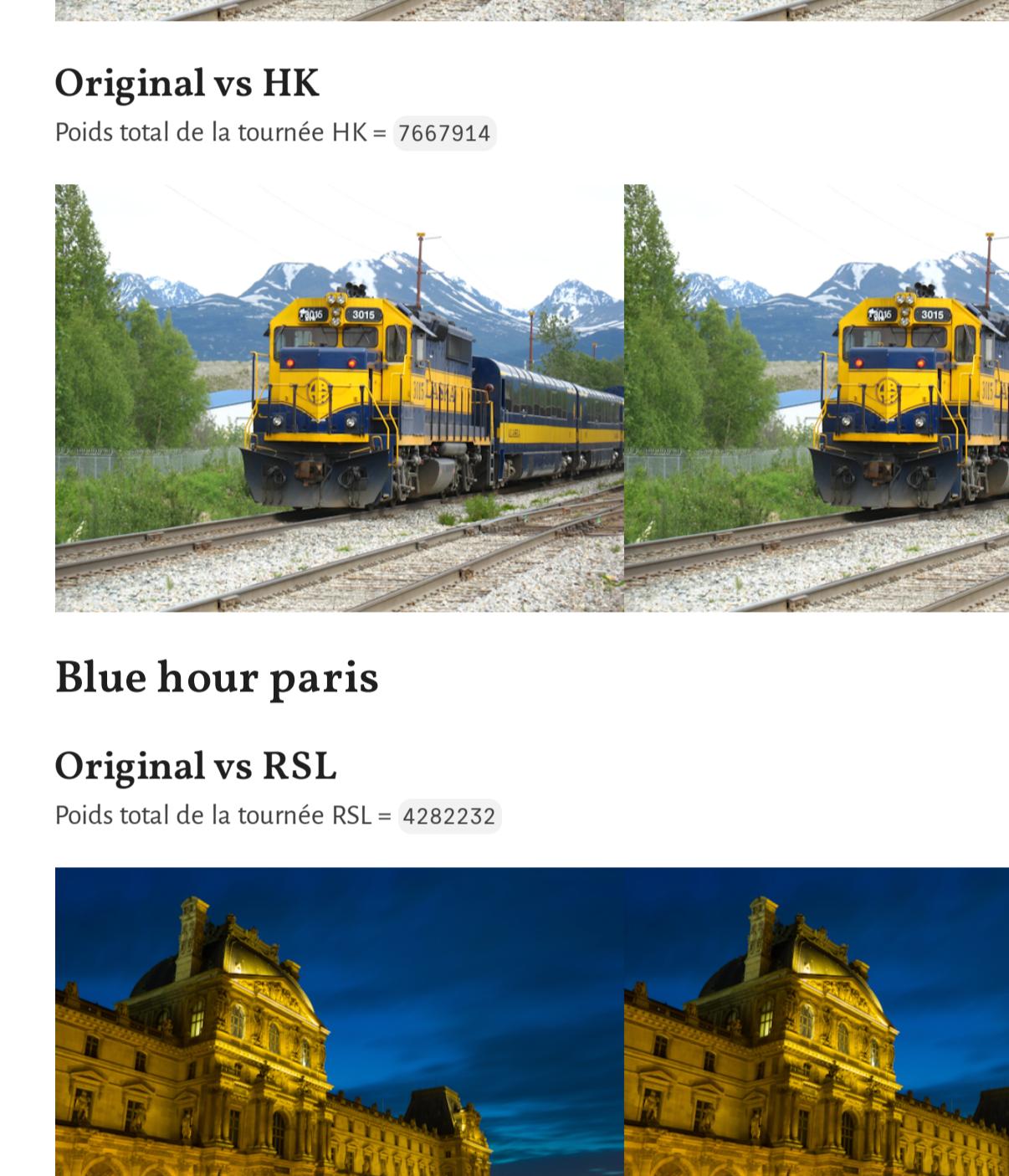


Image reconstruite avec l'algorithme de RSL

Poids total de la tournée = 12839731

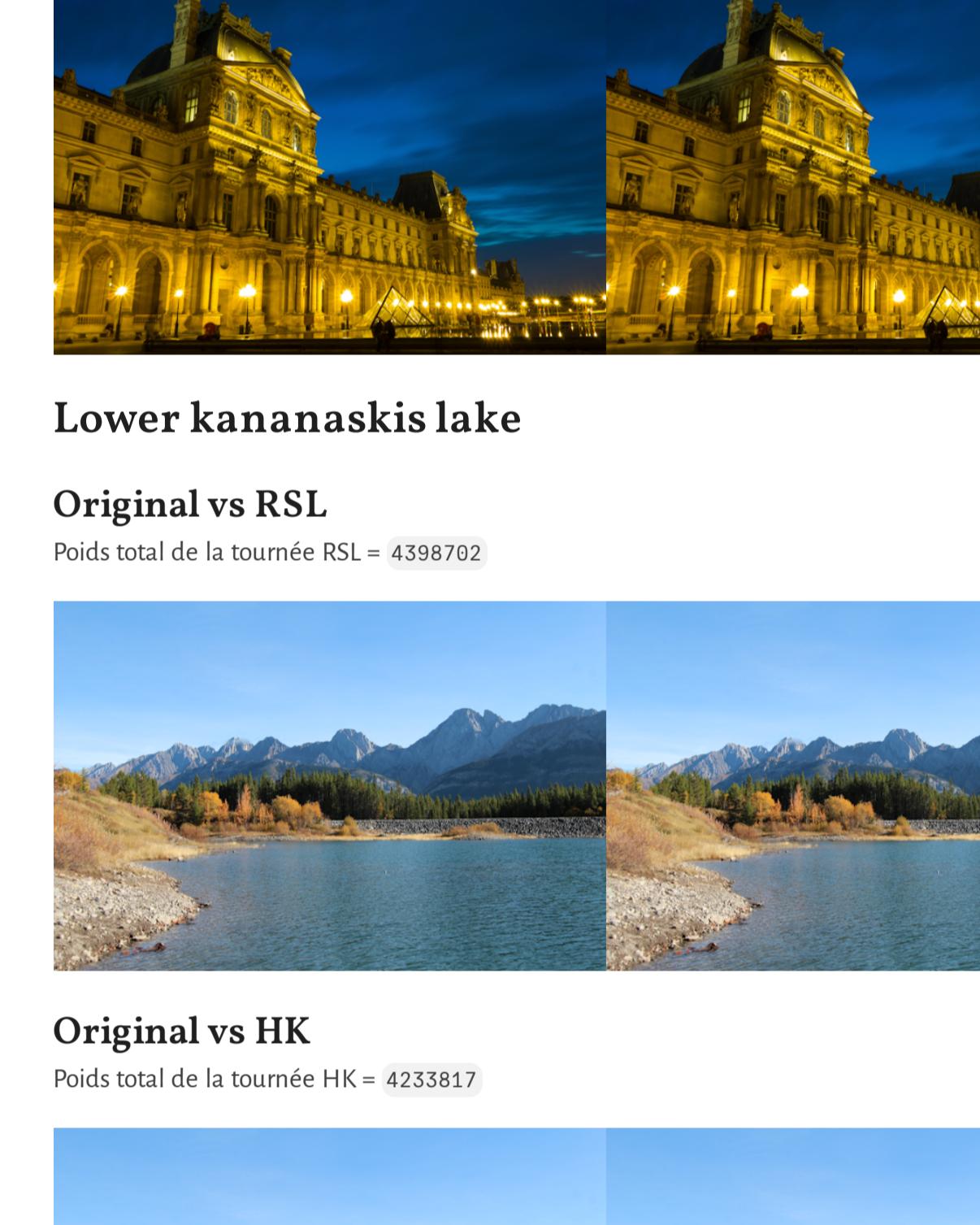
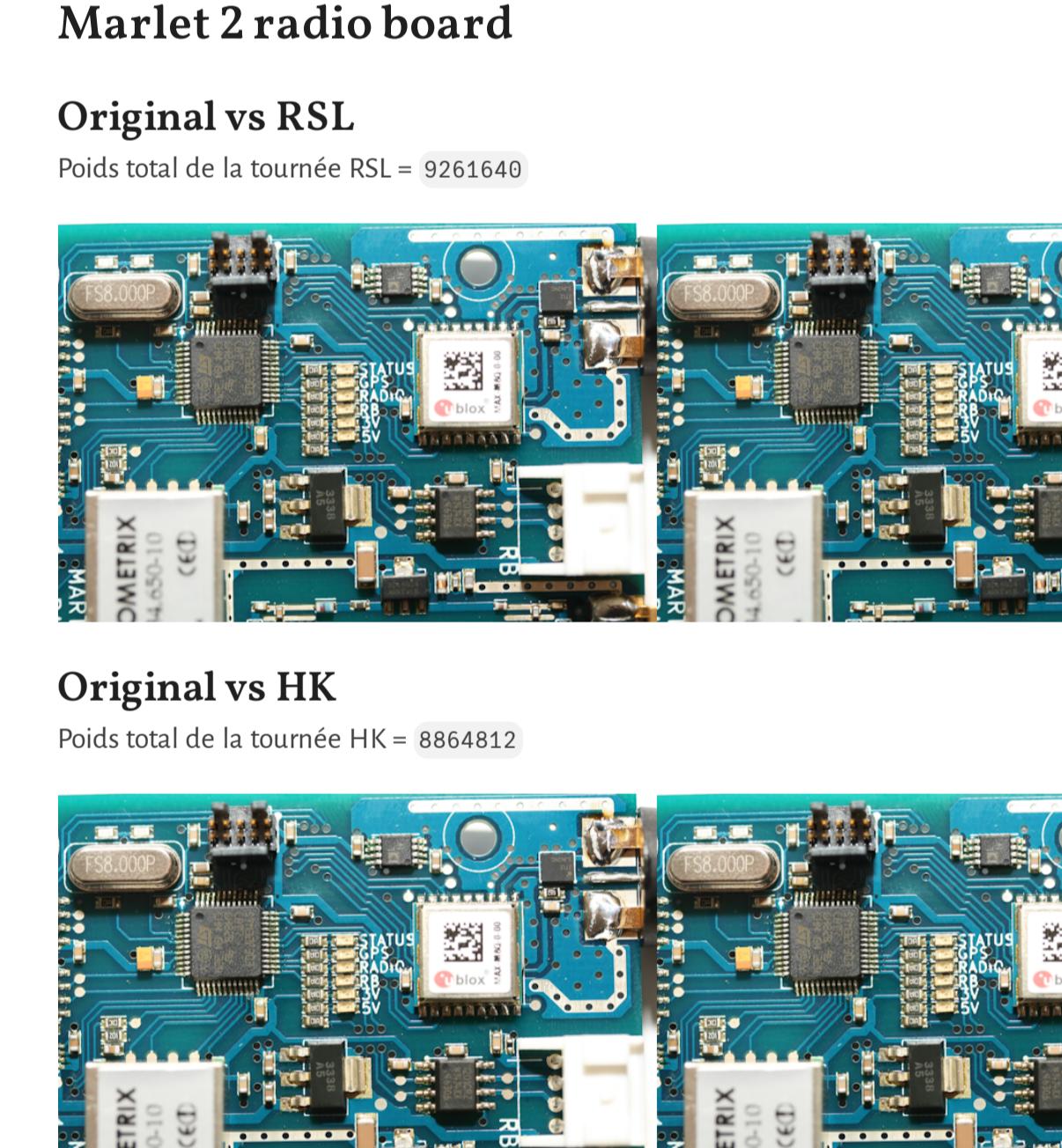


Image reconstruite avec l'algorithme de HK

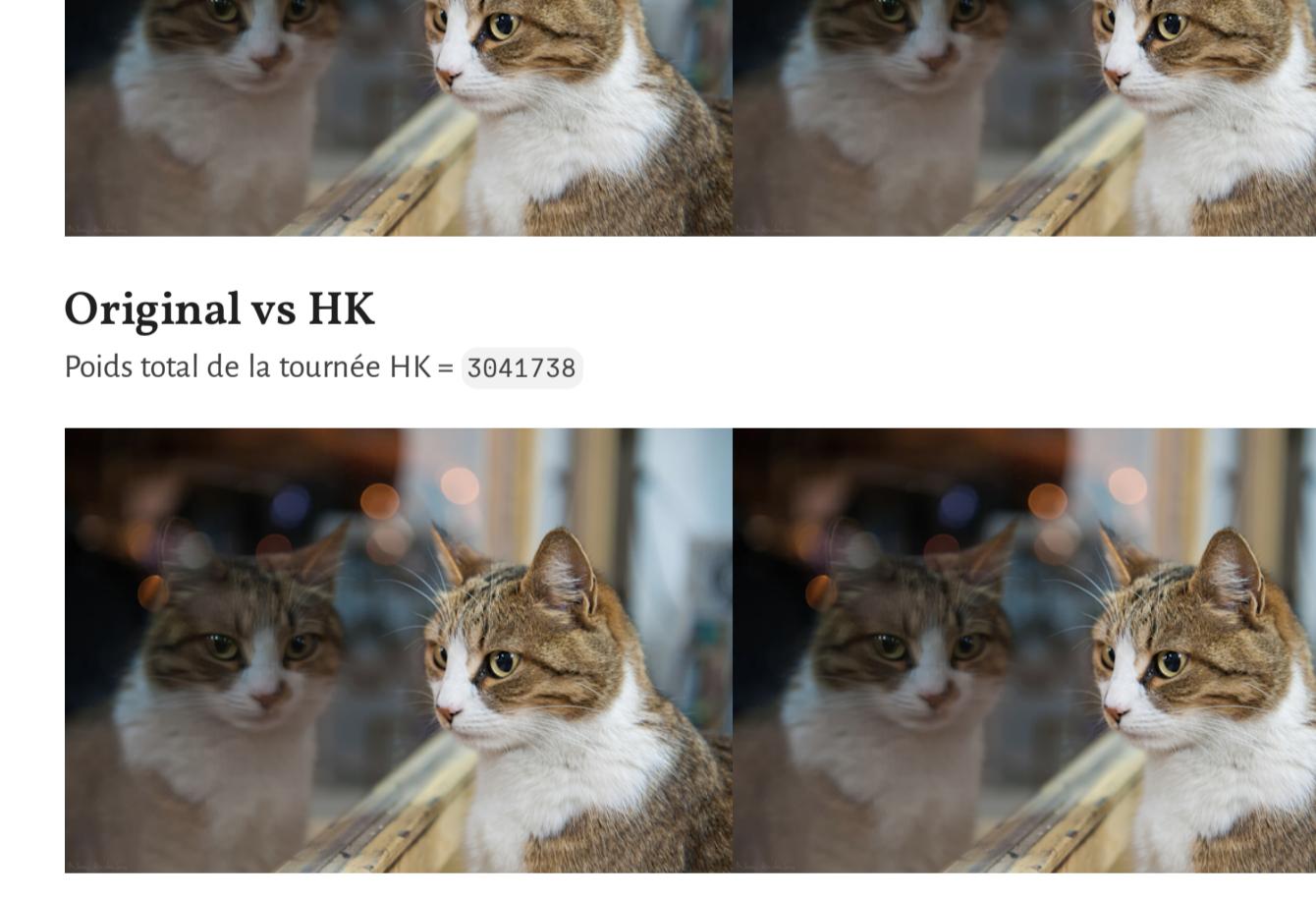
Poids total de la tournée = 12314767



Alaska railroad

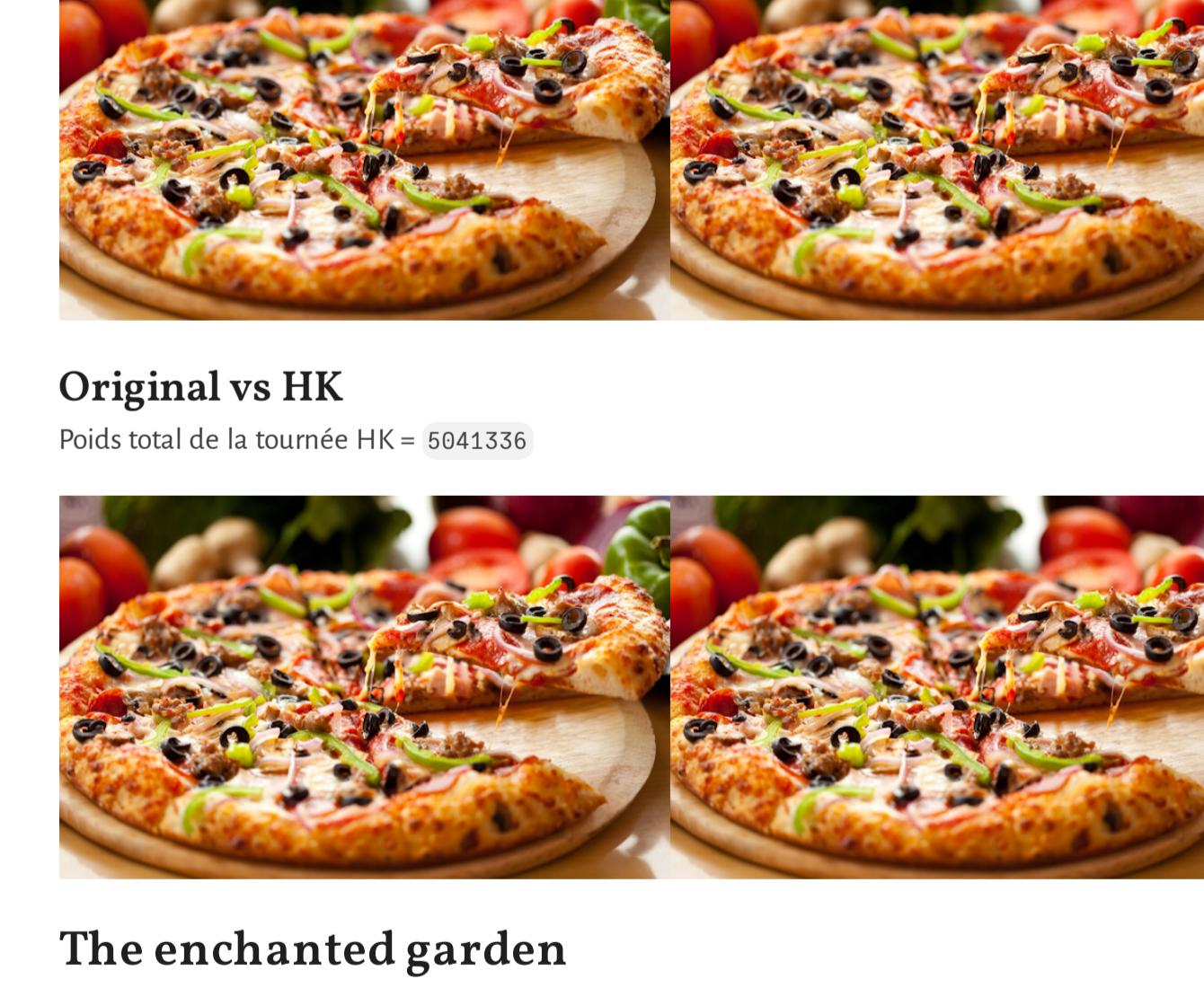
Original vs RSL

Poids total de la tournée RSL = 7978214



Original vs HK

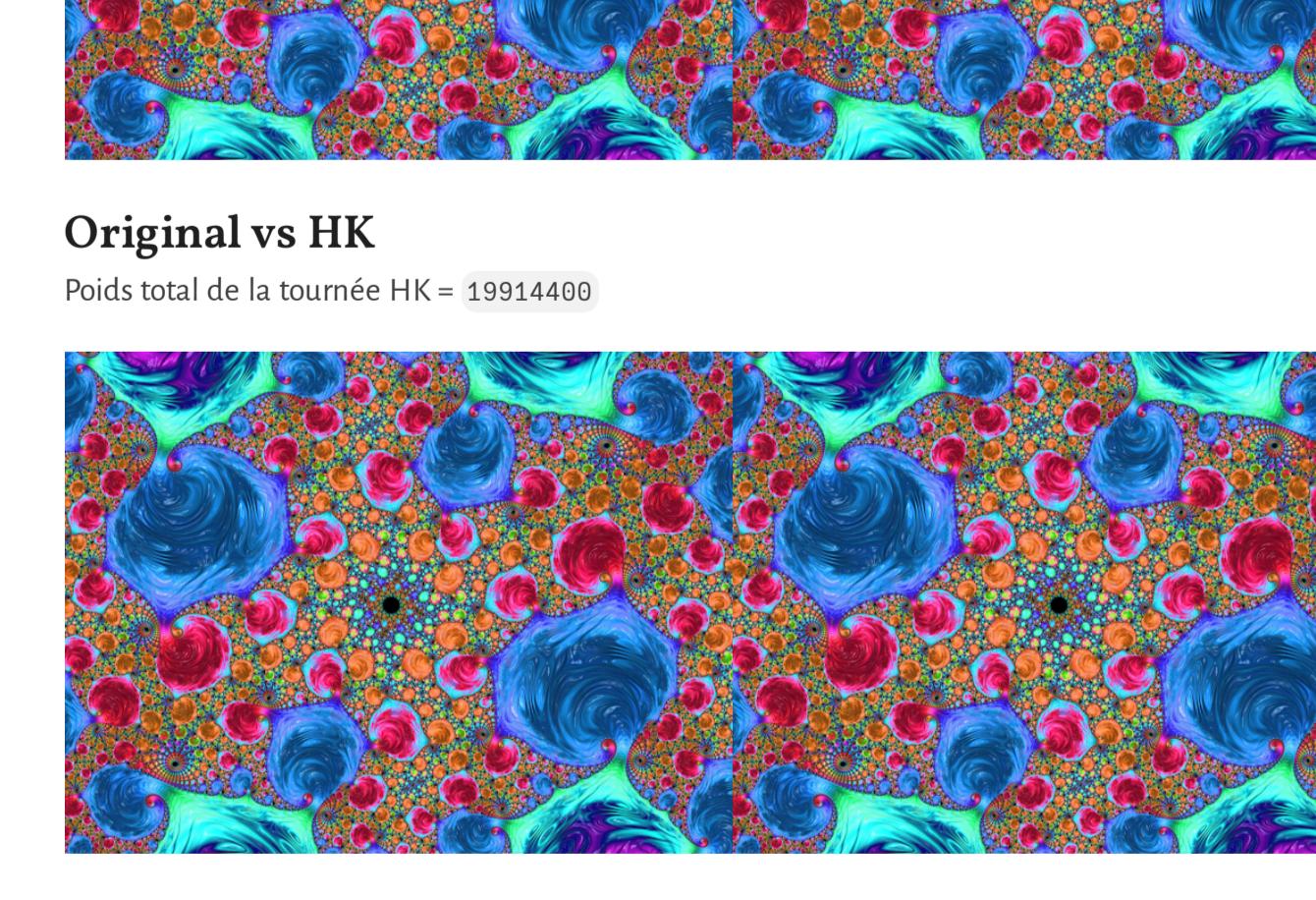
Poids total de la tournée HK = 7667914



Blue hour paris

Original vs RSL

Poids total de la tournée RSL = 4282232



Original vs HK

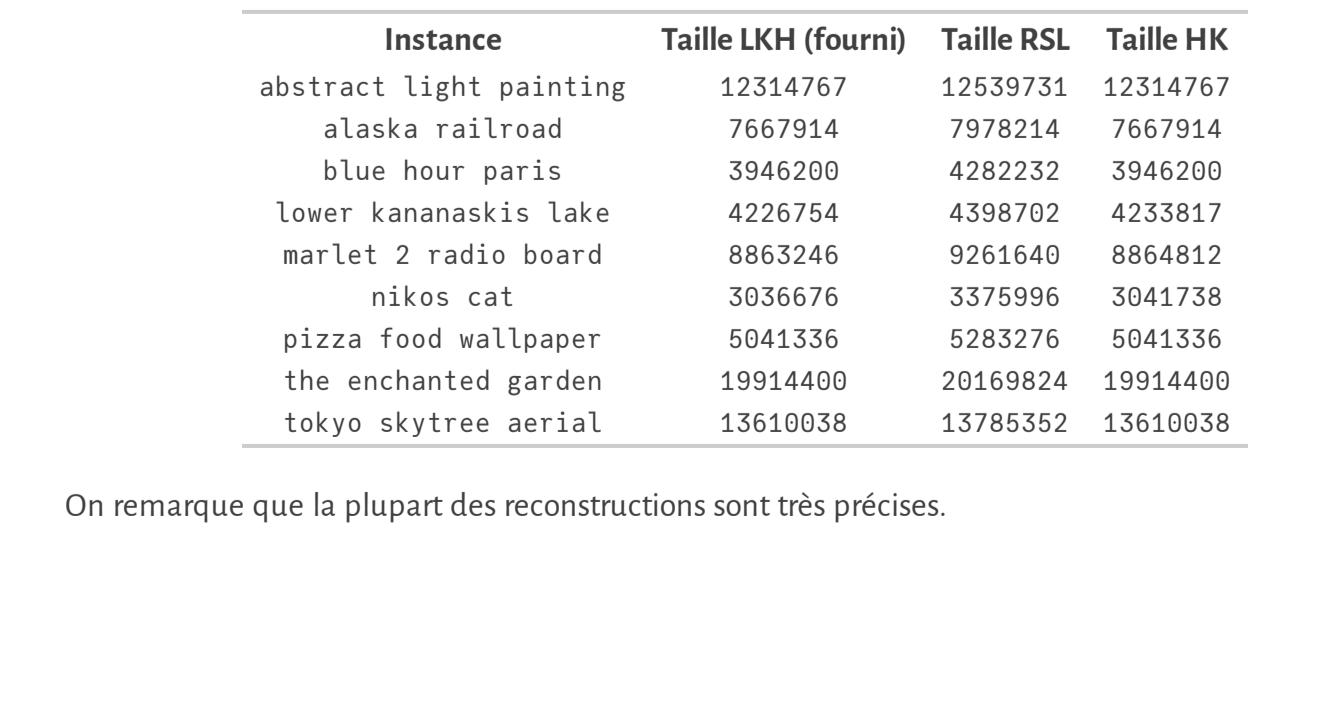
Poids total de la tournée HK = 3946208



Lower kananaskis lake

Original vs RSL

Poids total de la tournée RSL = 4398702



Original vs HK

Poids total de la tournée HK = 3946200



Résumé des résultats

Instance	Taille LKH (fourni)	Taille RSL	Taille HK
abstract light painting	12314767	12539731	12314767
alaska railroad	7978214	4282232	3946208
blue hour paris	3946200	4282232	3946200
lower kananaskis lake	4226754	4398702	4235817
marlet 2 radio board	8863246	9261640	8864812
nikes cat	3036676	3375996	3041738
pizza food wallpaper	5041336	5283276	5041336
the enchanted garden	19914400	20169824	19914400
tokyo skytree aerial	13610038	13785352	13610038

On remarque que la plupart des reconstructions sont très précises.