1. Опис задачі:

   Завдання полягало в тому, щоб написати алгоритм обходження зображення по контуру, а саме «Backward contour tracing» і порівняти його роботу з алгоритмом Жука.

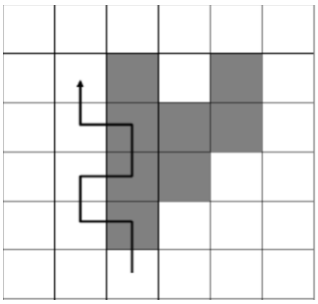
   Опис експериментів:

   Застосувати основний алгоритм, алгоритм Жука та OpenCV на різних типах зображення і порівняти отримані результати.

2. Опис алгоритму Жука:

   Алгоритм працює на основі двох простих правил:

   - якщо активний піксель належить однорідній області, то проводиться поворот ліворуч
   - якщо активний піксель не належить однорідній області, то проводиться поворот праворуч. Алгоритм зупиняє свою роботу, якщо він повернувся в стартовий піксель



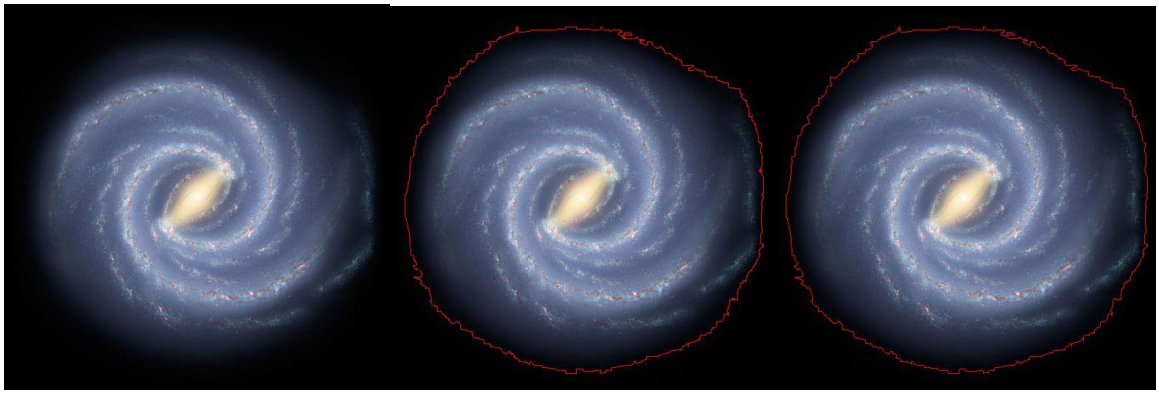
3. Опис основного алгоритму:
   - Проводиться пошук стартового пікселя Pa
   - Проводиться пошук сусіднього контурного пікселя Pn за годинниковою стрілкою і P'n - проти
   - Якщо Pn != P'n , то стартовий піксель визнається також і кінцевим Ps = Pe . Сусідній контурний піксель, отриманий на кроці вище, заноситься в масив контурних пікселів і стає активним Pa = Pcn
   - Проводиться пошук наступного сусіднього контурного пікселя Pcn. Позиція стартової перевірки d визначається як (d'+2)mod8, де d' – позиція, з якої було знайдено активний піксель Pa . Якщо сусідній піксель є контурним та не співпадає з кінцевим пікселем, то він заноситься в масив контурних пікселів і стає активним Pa = Pcn
   - Якщо знайдений сусідній піксель вже є в масиві контурних, то він видаляється і рахується фоновим. Статус активного пікселя присвоюється попередньому контурному пікселю
   - Якщо Pcn == Pe, то алгоритм завершує свою роботу

4. Результати експериментів:

Щоб результат був найбільш правильним, для кожного зображення потрібно виставляти різницю в кольорі, тобто, на скільки компоненти r, g, b одного пікселя будуть відрізнятися від компонентів іншого.

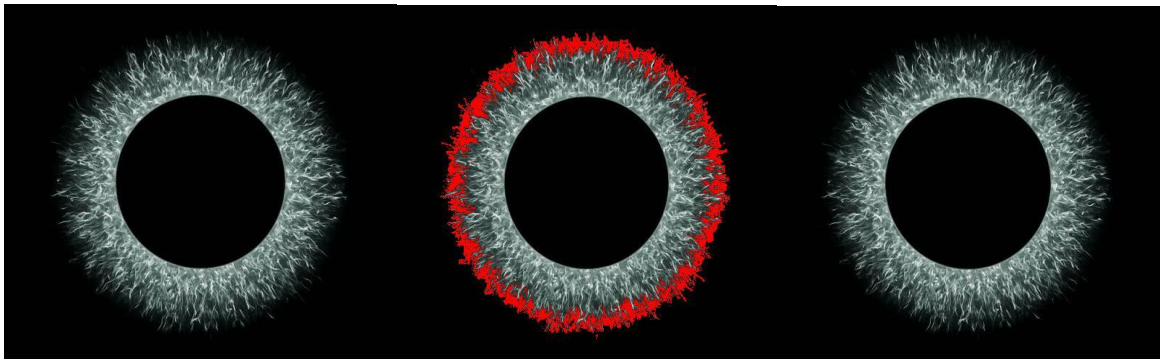
Для кожного експерименту виставлялася різниця та засікався час роботи.

(Час в секундах)

Оригінал	Алгоритм Жука	Backward contour tracing
	Час: 0.9807350635528564 Різниця: 110	Час: 0.87687087059021 Різниця: 110
		
	Час: 0.7295770645141602 Різниця: 6	Час: 0.6766960620880127 Різниця: 6
		

Час: 10.915980815887451 не обходить це зображення з

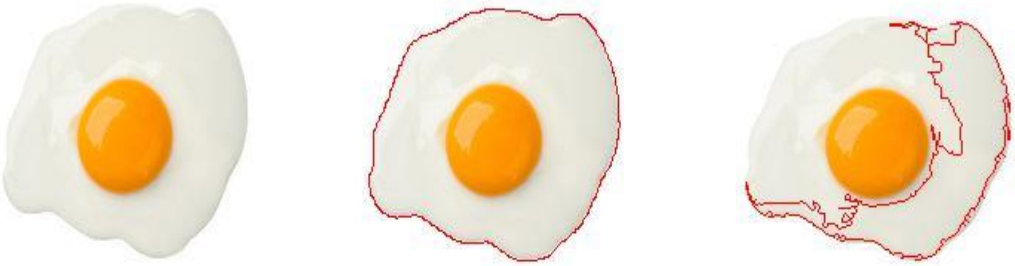
Різниця: 17 з будь-якою різницею



Час: 0.14807915687561035 Час: 0.13663363456726074

Різниця: 10 Різниця: 11

(При різниці 10 результат був взагалі некоректний)

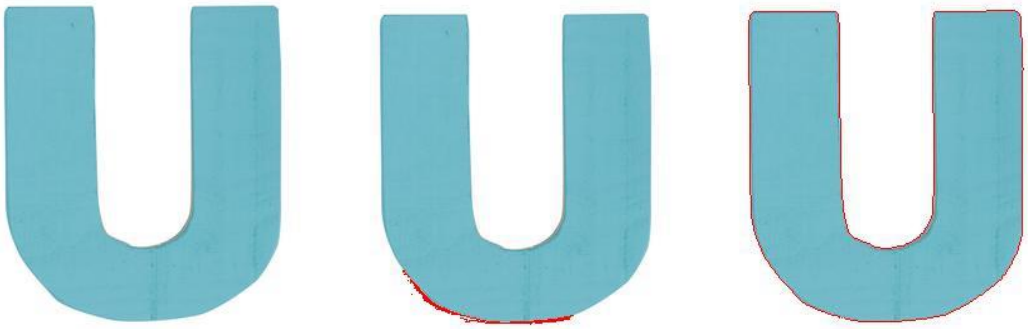


Час: 0.2715342044830322 Час: 0.23027491569519043

Різниця: 45 Різниця: 45

(з різницею 45+ результат був кращий але

Дуже далекий від правильного)



Час: 0.46532106399536133 Час: 0.4495968818664551

Різниця: 45 Різниця: 45

Час: 0.7529420852661133 Час: 0.6121268272399902

Різниця: 50 Різниця: 60

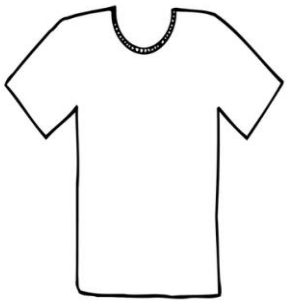
(з 50 не обводить, 60 - найбільш оптимальний варіант)

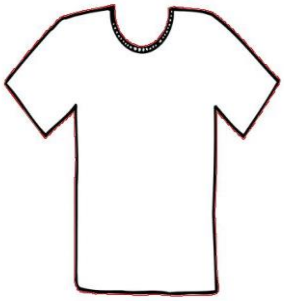
Час: 0.7400109767913818

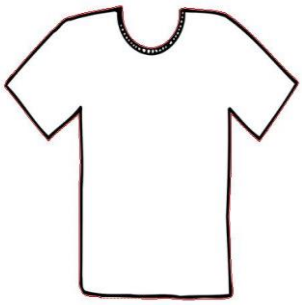
Різниця: 50

Час: 0.6134791374206543

Різниця: 50







Час: 0.16852688789367676

Різниця: 40

Час:

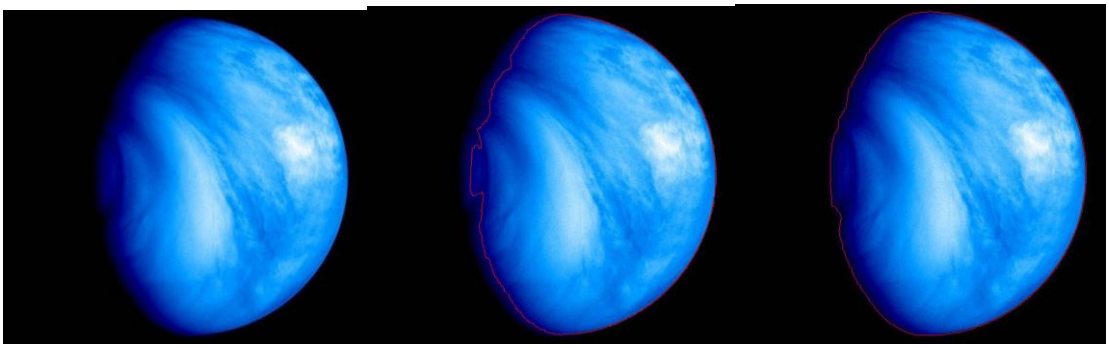
Різниця: 45

Час: 1.3134980201721191

Різниця: 45

Час:0.9451990127563477

Різниця: 45



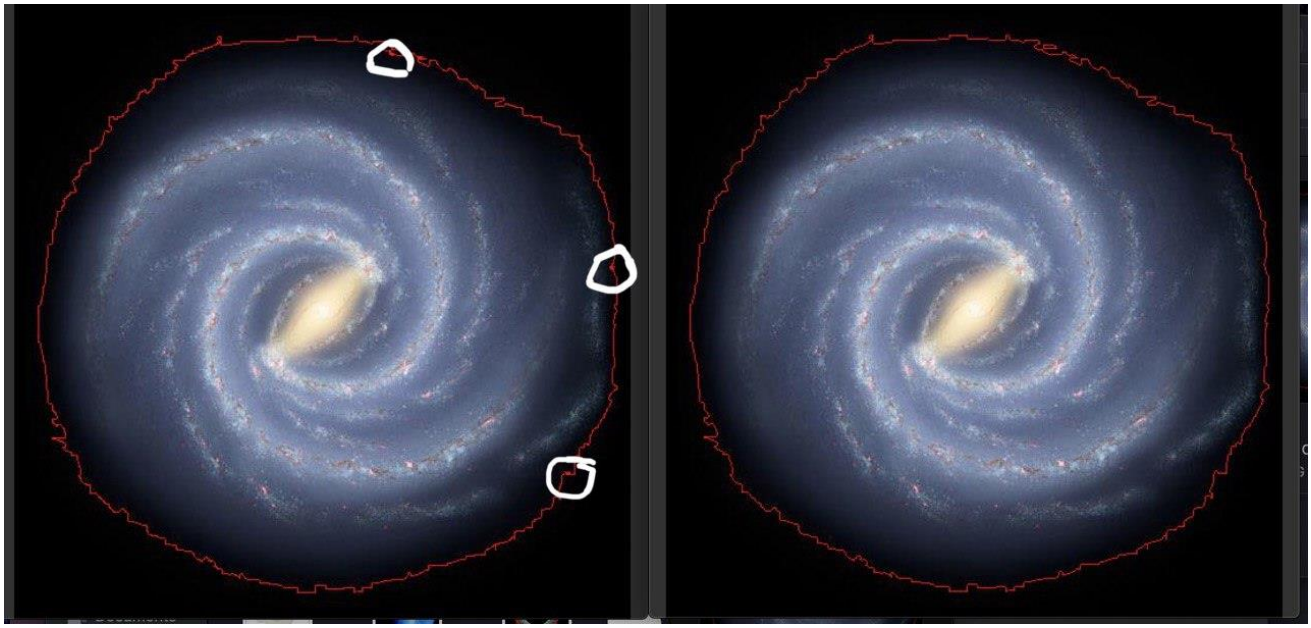
5. Висновки:

У всіх випадках алгоритм Backward contour tracing працював швидше за Жука, інколи різниця в часі була досить велика. Також основний алгоритм описує 8-зв’язний контур, Жука - 4-зв’язний. На простих зображеннях різницю обході двох алгоритмів побачити важко, а на складних різниця суттєва.

Під час роботи алгоритму Жука на контурі зображення можна побачити великі скупчення кольору його обходу(в даному випадку червоний), щодо основного алгоритму, то він дозволяє уникати цього.

Приклад:

Алгоритм Жука Backward contour tracing



Великим недоліком алгоритму Backward contour tracing є те що, він не добре обходить зображення на яких об’єкт схожого кольору до кольору фону, а також

Щодо алгоритму OpenCV, то він працює як алгоритм Жука, але розібратися в його реалізації не можна, бо в ньому використовується бібліотека, без доступу до коду.

Результат обходу залежить від різниці компонентів r, g, b одного пікселя й іншого. З огляду на зображення(фон, об’єкт, плавні/різкі переходи) можна підібрати оптимальну різницю.

Посилання на репозиторій: https://github.com/Oleh93/backward_tracing_ad_fontes