

Hướng Dẫn Sử Dụng Chương Trình Bài Tập Giữa Kỳ

Nhóm Thực Hiện

Trần Huỳnh Trung Hiếu
MSSV: N21DCCN122

Nguyễn Thị Thanh Huyền
MSSV: N21DCCN130

Nguyễn Thị Huyền My
MSSV: N21DCCN47

Tô Phan Kiều Thương
MSSV: N21DCCN184

*Khoa Công Nghệ Thông Tin,
Học Viện Công Nghệ Bưu Chính Viễn Thông Cơ Sở TP.HCM*

Tháng 5, 2025

Mục lục

1	Giới thiệu	3
2	Cấu trúc chương trình	3
3	Yêu cầu cài đặt	4
4	Hướng dẫn chạy chương trình	4
4.1	Chạy giao diện chính (<code>main.py</code>)	4
5	Ví dụ minh họa sử dụng chương trình	10
5.1	Tạo ảnh mới	10
5.2	Thêm đối tượng	11
5.3	Áp dụng biến đổi	12
5.4	Thêm Hàm Chi Phí	13
5.5	Tính Chi Phí	14
5.6	Tạo chuỗi biến đổi	14
6	Tổng kết	15
7	Tài liệu tham khảo	15

1 Giới thiệu

Chương trình **Object Image Editor** là một ứng dụng giao diện đồ họa (GUI) được phát triển bằng Python, sử dụng thư viện `tkinter` và `PIL` để quản lý và chỉnh sửa các ảnh chứa các object hình chữ nhật. Chương trình cho phép người dùng thêm ảnh, chỉnh sửa object, áp dụng các phép biến đổi, chuyển đổi object giữa các ảnh, và tạo chuỗi biến đổi với tính toán chi phí. Ứng dụng được thiết kế với giao diện trực quan, sử dụng màu nền `#f0f4f8`, font `Helvetica`, và các canvas có thanh cuộn để hiển thị ảnh. Giao diện chính bao gồm nhiều tab chức năng: Home, Transformation Library Manager (TLM), Cost Function Server (CFS), Object Converter (OC), Sequence Editor, và About.

Mục tiêu của chương trình là cung cấp một công cụ linh hoạt để thao tác trên các ảnh và object, đồng thời hỗ trợ tính toán chi phí cho các phép biến đổi, phù hợp với các yêu cầu xử lý ảnh cơ bản trong học thuật và thực tiễn.

2 Cấu trúc chương trình

Chương trình được tổ chức thành nhiều module, mỗi module đảm nhận một chức năng cụ thể:

- `main.py`: Tập chính, khởi tạo giao diện với `ttk.Notebook` chứa các tab: Home, TLM, CFS, OC, Sequence Editor, About.
- `tab/Home_tab.py`: Quản lý hiển thị ảnh, thêm ảnh mới, và chỉnh sửa object (tọa độ, màu RGB).
- `tab/TLM_tab.py`: Quản lý thư viện phép biến đổi, hỗ trợ áp dụng toán tử từ JSON hoặc nhập thủ công.
- `tab/CFS_tab.py`: Quản lý hàm chi phí, đọc và ghi vào `cost_functions.json`.
- `tab/OC_tab.py`: Chuyển đổi object giữa hai ảnh, tính toán chuỗi biến đổi và chi phí.
- `tab/Sequence_tab.py`: Tab Sequence Editor, cho phép tạo và áp dụng chuỗi biến đổi trên các object.
- `object_manager.py`: Quản lý cơ sở dữ liệu ảnh (`ImageDatabase`) và các object (`ImageObjectRegion`).
- `transformation_manager.py`: Quản lý thư viện phép biến đổi (`TransformationLibraryManager`).

- `cost_function_server.py`: Xử lý tính toán chi phí dựa trên `cost_functions.json`.
- `object_converter.py`: Hỗ trợ chuyển đổi object giữa hai ảnh.
- `data/transformations.json`: Lưu danh sách toán tử và tham số.
- `data/cost_functions.json`: Lưu danh sách hàm chi phí.

3 Yêu cầu cài đặt

Clone dự án từ Github: <https://github.com/huyenmy239/transform-similarity-retrieval> Để chạy chương trình, bạn cần cài đặt Python và các thư viện sau:

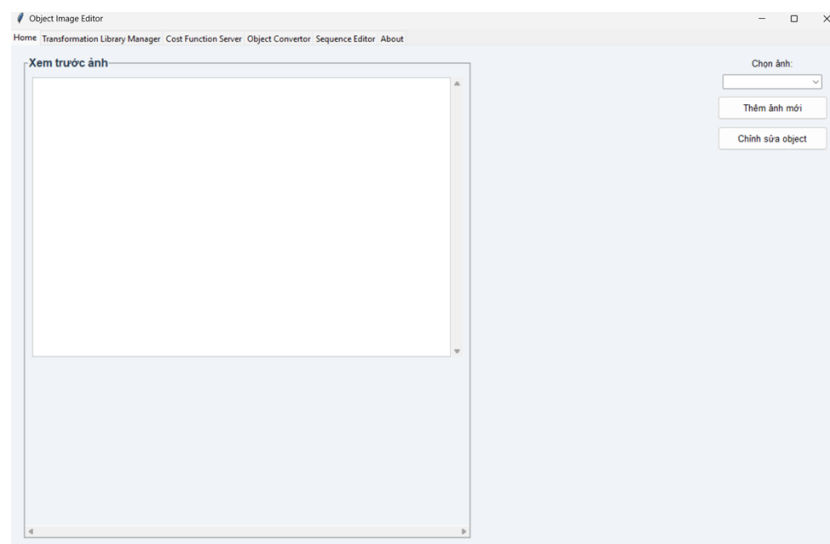
```
1 pip install -r requirements.txt
```

4 Hướng dẫn chạy chương trình

4.1 Chạy giao diện chính (`main.py`)

1. Mở terminal (dòng lệnh) trong thư mục chứa mã nguồn.
2. Chạy lệnh:

```
1 python main.py
```



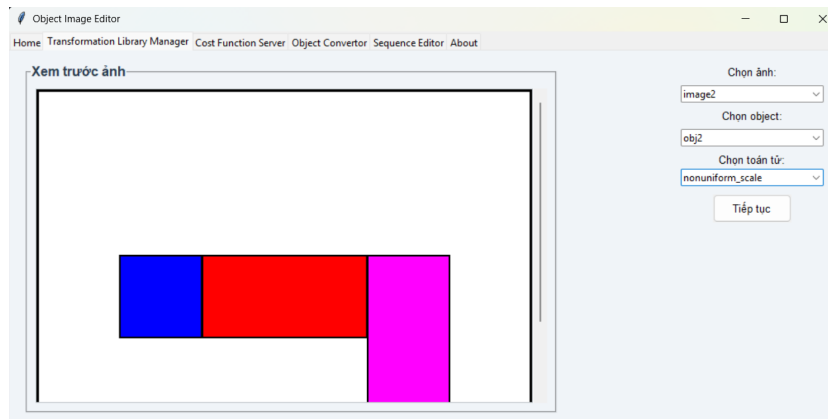
Hình 1: Giao diện chính

3. Giao diện chính sẽ hiện ra với 6 tab: Home, Transformation Library Manager, Cost Function Server, Object Converter, Sequence Editor, About.
4. Thực hiện theo hướng dẫn trong từng tab để sử dụng các chức năng:
 - **Home:** Quản lý danh sách ảnh, thêm ảnh mới, và chỉnh sửa object.



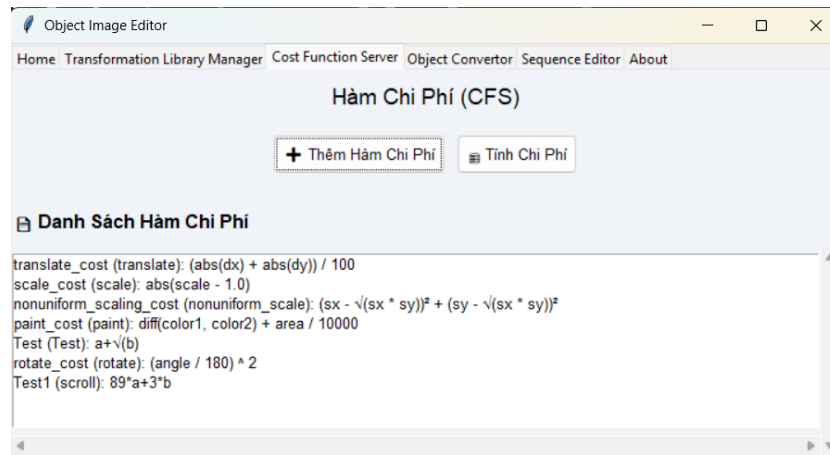
Hình 2: Giao diện home

- (a) Chọn ảnh từ Combobox “Chọn ảnh” để xem trước trên canvas.
 - (b) Nhấn “Thêm ảnh mới”, nhập tên, tọa độ ảnh (ví dụ: anh1, 0, 0, 100, 100), nhấn “Tạo ảnh”.
 - (c) Nhấn “Chỉnh sửa object”, thêm/xóa object, chỉnh sửa tọa độ (x1, y1, x2, y2) và màu RGB (ví dụ: (255,0,0)), nhấn “Lưu”.
- **Transformation Library Manager (TLM):** Áp dụng phép biến đổi lên object.



Hình 3: Giao diện tab Transformation Library Manager

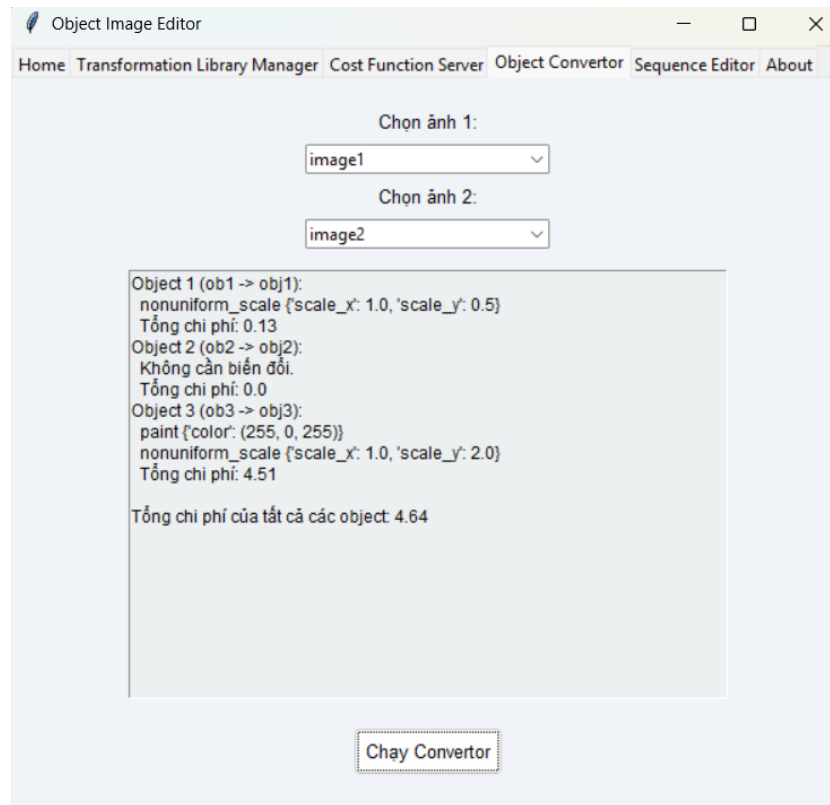
- (a) Chọn ảnh từ Combobox “Chọn ảnh” để hiển thị trên canvas.
 - (b) Chọn object từ Combobox “Chọn object”.
 - (c) Chọn toán tử (ví dụ: translate, scale) từ Combobox “Chọn toán tử”, nhấn “Tiếp tục”.
 - (d) Trong cửa sổ tùy chọn, chọn:
 - “Chọn từ file JSON”: Chọn tham số từ `transformations.json` (ví dụ: `{“dx”: 10, “dy”: 20}`), nhấn “Áp dụng”.
 - “Nhập tham số thủ công”: Nhập tham số (ví dụ: `dx=10, dy=20`), nhấn “Áp dụng”.
 - (e) Kiểm tra canvas để xem đối tượng sau khi biến đổi.
- **Cost Function Server (CFS):** Quản lý và tính toán hàm chi phí.



Hình 4: Giao diện tab Cost Function Server

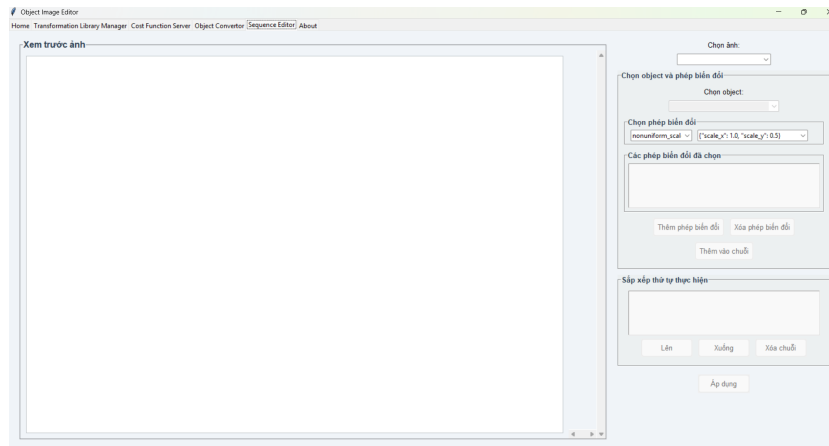
- (a) Xem danh sách hàm chi phí trong Listbox.
- (b) Nhấn “Thêm Hàm Chi Phí”, nhập tên (ví dụ: `translate_cost`), loại phép biến đổi (`translate`), công thức (ví dụ: $dx + dy$), sử dụng máy tính công thức, nhấn “Xác Nhận”.
- (c) Nhấn “Tinh Chi Phí”, chọn hàm chi phí, nhập tham số (ví dụ: $dx=10$, $dy=20$), nhấn “Tinh” để xem kết quả trong message-box.

- **Object Convertor (OC):** Chuyển đổi object giữa hai ảnh.



Hình 5: Giao diện tab Object Converter

- (a) Chọn hai ảnh từ Combobox “Chọn ảnh 1” và “Chọn ảnh 2” (ảnh 2 không trùng ảnh 1).
 - (b) Nhấn “Chạy Convertor” để xem chuỗi biến đổi và chi phí trong Text widget.
 - (c) Kiểm tra kết quả: chuỗi biến đổi cho từng object (ví dụ: translate {“dx”: 10, “dy”: 20}) và tổng chi phí.
- **Sequence Editor:** Tạo và áp dụng chuỗi biến đổi.

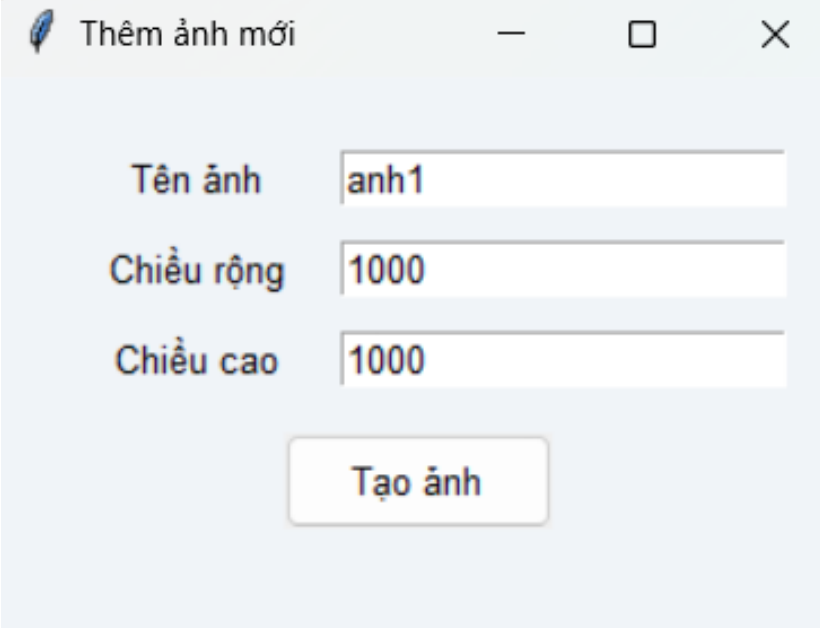


Hình 6: Giao diện tab Sequence Editor

- (a) Chọn ảnh từ Combobox “Chọn ảnh” để hiển thị trên canvas.
 - (b) Chọn object từ Combobox “Chọn object”.
 - (c) Chọn toán tử và tham số (ví dụ: translate, {“dx”: 10, “dy”: 20}) từ Combobox, nhấn “Thêm phép biến đổi” để thêm vào Listbox tạm thời.
 - (d) Nhấn “Thêm vào chuỗi” để thêm vào Listbox chuỗi chính.
 - (e) Sắp xếp chuỗi bằng nút “Lên”/“Xuống” hoặc xóa bằng “Xóa chuỗi”.
 - (f) Nhấn “Áp dụng” để chạy chuỗi, xem kết quả trong Text widget (giai đoạn, chi phí, lỗi nếu có).
- **About:** Xem thông tin về chương trình và tác giả.
 - (a) Chuyển sang tab để đọc thông tin tính (tên chương trình, nhóm thực hiện).

5 Ví dụ minh họa sử dụng chương trình

5.1 Tạo ảnh mới



Thêm ảnh mới

Tên ảnh

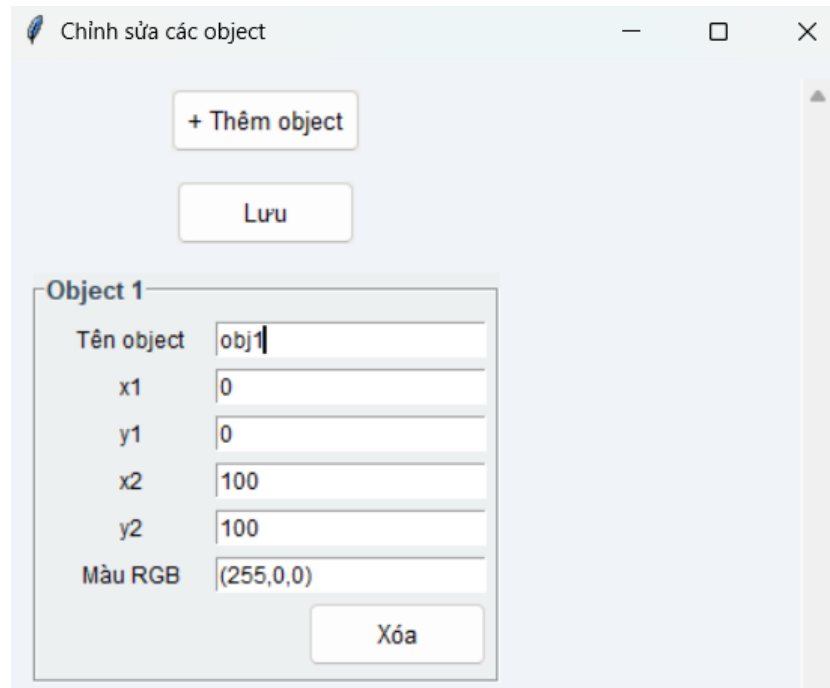
Chiều rộng

Chiều cao

Hình 7: Thêm ảnh mới

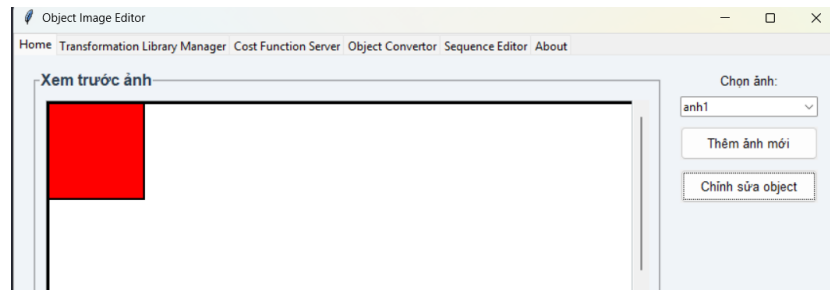
1. Nhấn nút “Thêm ảnh mới”.
2. Nhập tên ảnh: **anh1**, chiều rộng : 1000, chiều cao : 1000.
3. Nhấn “Tạo ảnh”.

5.2 Thêm đối tượng



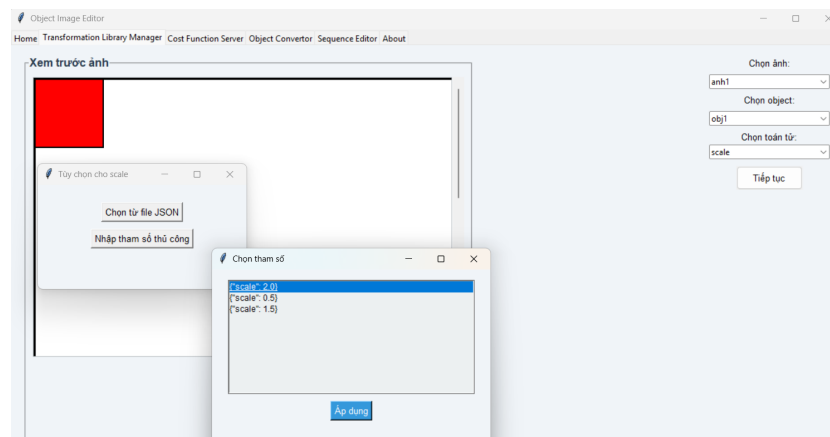
Hình 8: Thêm đối tượng mới cho ảnh

1. Chọn ảnh vừa tạo.
2. Nhấn “Chỉnh sửa ảnh”.
3. Nhấn “Thêm đối tượng”.
4. Nhập tên đối tượng
5. Nhập tọa độ đối tượng: $(x1=0, y1=0, x2=100, y2=100)$, màu sắc: $(255, 0, 0)$.
6. Nhấn “Lưu”.



Hình 9: Đối tượng mới được tạo thành công

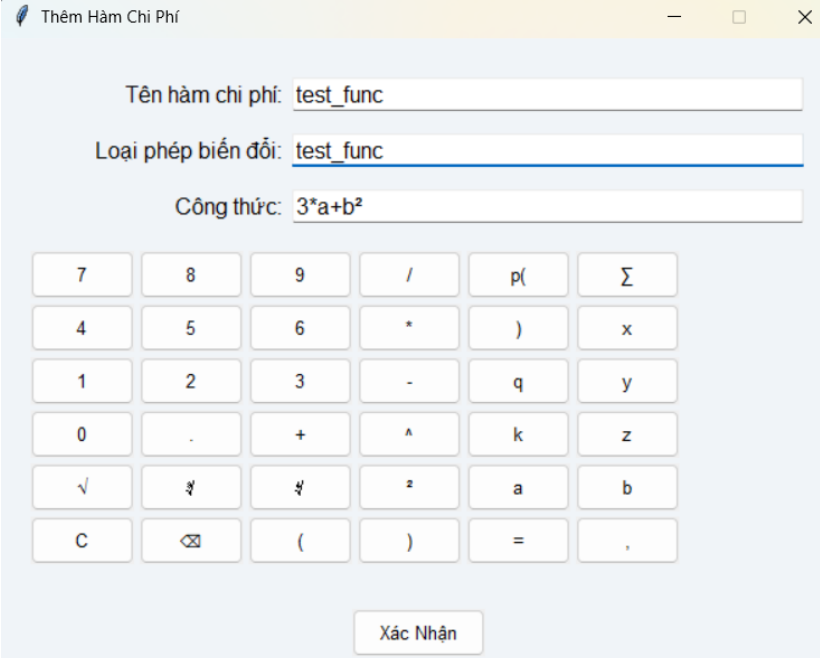
5.3 Áp dụng biến đổi



Hình 10: Chuyển đổi ảnh

1. Chọn tab "Transformation Library Manager".
2. Chọn ảnh có đối tượng.
3. Nhấn “Biến đổi đối tượng”.
4. Chọn phép biến đổi **scale**, nhấn "Tiếp tục"
5. Nhập từ json, chọn tham số có sẵn: **"scale":2.0**.
6. Nhấn “Áp dụng”.

5.4 Thêm Hàm Chi Phí



Thêm Hàm Chi Phí

Tên hàm chi phí:

Loại phép biến đổi:

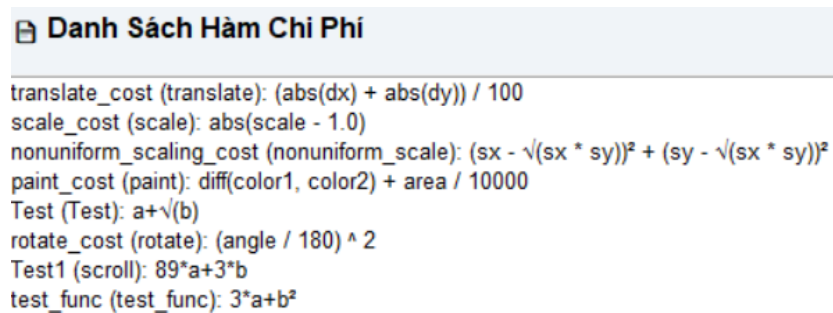
Công thức:

7	8	9	/	p(Σ
4	5	6	*)	x
1	2	3	-	q	y
0	.	+	^	k	z
$\sqrt{}$	$\sqrt[3]{}$	$\sqrt[n]{}$	2	a	b
C	\times	()	=	,

Xác Nhận

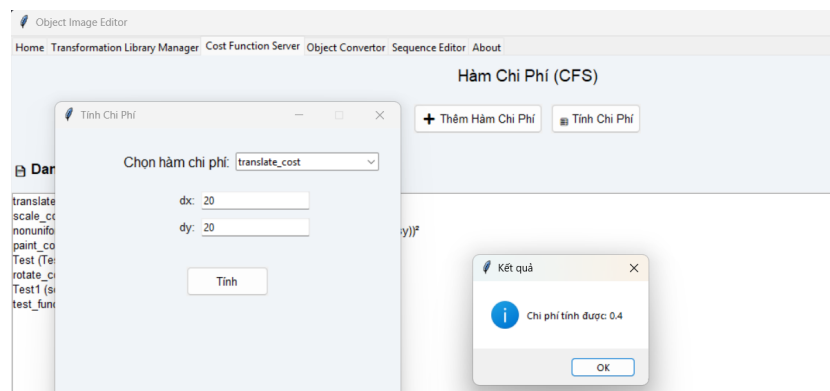
Hình 11: Thêm mới hàm chi phí

1. Chọn tab “Cost Function Server”.
2. Nhấn “Thêm Hàm Chi Phí”.
- 3.
4. Nhập tên hàm chi phí: `test_func`, loại phép biến đổi: `test_func`, công thức: `dx + dy`.
5. Nhấn “Xác Nhận”.



Hình 12: Hàm chi phí được thêm thành công

5.5 Tính Chi Phí



Hình 13: Tính hàm chi phí

1. Nhấn “Tính Chi Phí”.
2. Chọn hàm chi phí: `translate_cost`.
3. Nhập tham số: `dx=10`, `dy=20`.
4. Nhấn “Tính”.
5. Kết quả sẽ hiển thị trong messagebox.

5.6 Tạo chuỗi biến đổi

1. Mở tab “Sequence Transformation”.
2. Chọn ảnh và đối tượng.

3. Thêm các phép biến đổi: `scale`, `rotate`, `translate`.
4. Nhấn “Thêm vào chuỗi”.
5. Sắp xếp chuỗi biến đổi theo thứ tự mong muốn rồi nhấn “Áp dụng”.
6. Quan sát kết quả và chi phí trong hộp văn bản.

6 Tổng kết

Chương trình `Object Image Editor` là công cụ hữu ích giúp người dùng thử nghiệm các phép biến đổi đối tượng ảnh một cách trực quan và dễ dàng. Tài liệu này cung cấp hướng dẫn chi tiết từng bước để chạy và sử dụng các chức năng chính của chương trình.

7 Tài liệu tham khảo

- Thư viện Python: `tkinter`, `PIL`, `json`.
- Tài liệu LaTeX: <https://www.overleaf.com/learn>.