Cartoon Maker

The Kyu’s

대그룹 1

개인 보고서

072124

김승규

**목차**

1. 인터페이스
   1. 동영상 파일 열기
   2. Cam 띄우기
   3. 영상 뷰
   4. 영상 편집 도구
   5. 동영상 파일 저장
2. 흑백 만화
   1. Halftoning
   2. Edge Detection
3. 칼라 만화
   1. 만화화
   2. 기타 효과
4. 기타
   1. 시연 영상
   2. 발표자료

인터페이스

1. 동영상 파일 열기

MFC의 Document/View, MDI 구조에 맞춰서 열기 버튼을 누르면 열리는 파일 입력 대화상자를 통해 영상 파일의 경로를 입력 받아서 OpenCV의 cvCaptureFromFile함수를 사용하여 영상을 읽었습니다. 프레임은 OpenCV에서 지원하는 cvRetrieveFrame함수를 사용하여 캡쳐 후 OpenCV에서 지원하는 IplImage 형식으로 배열에서 저장했습니다. 영상을 읽어오는 과정에 약간의 시간이 소요되므로 다른 작업이 가능하도록 작업자 스레드를 사용하여 구현하였습니다.

1. Cam 띄우기

최종 프로젝트에서 사용되지 않은 부분이지만, 영상을 파일이 아닌 Cam을 통해 입력 받기 위해 만들었습니다. OpenCV를 사용해서 구현하였습니다. 원래는 정상 작동하였으나 통합하는 과정에서 문제가 생겨 사용하지 않게 되었습니다.

1. 영상 뷰

영상을 뷰에 띄워주는 작업입니다. 사용자가 원하는 프레임의 영상을 뷰에 띄워줍니다. 처음 영상을 입력 받으면 첫번째 프레임을 띄워줍니다. IplImage형식으로 저장된 이미지를 픽셀마다 읽어서 뷰에 찍어줍니다.

1. 영상 편집 도구

영상을 편집하는 도구 부분도 구현했습니다. Pen, FloodFill, Get Color부분을 직접 구현하였습니다. Pen과 Get Color는 기본적으로 있는 함수들을 이용하여 구현하였고, FloodFill은 BFS방식으로 Queue를 사용하여 구현하였습니다.

1. 동영상 파일 저장

만화 애니메이션을 만들기 위해 동영상 파일로 저장을 해줍니다. OpenCV에서 지원하는 함수를 사용하여 구현하였습니다.

흑백만화

실제 프로젝트에서는 사용하지 않았지만, 원래 계획은 흑백 만화를 만들고 색깔을 입히는 방식으로 생각했기 때문에 흑백 만화에 대한 연구를 많이 하였습니다.

1. Halftoning

Dithering, Thresholding, Hatching 등 다양한 기법을 통해 만화로 만들어보려고 노력했습니다. 최종 프로젝트에도 Hatching이 들어가긴 했지만 결과물이 좋지 않아서 중점적으로 다루지 않은 부분입니다.

1. Edge Detection

중간발표 때 까지도 만화화에 가장 핵심이 되는 내용이었습니다. Sobel, Susan Edge, Canny등 다양한 Edge Detection을 실험해보았습니다. Sobel과 Canny는 OpenCV에서 지원되는 함수를 사용하였고, Susan Edge는 연구실에서 사용하던 소스를 사용하였습니다.

칼라만화

1. 만화화

FloodFill방식과 유사하게 한 점에서부터 BFS로 RGB값이 차이가 적은 연속된 영역으로 탐색합니다. Edge Detection을 통해 얻은 테두리를 벗어나지 않게 해서 탐색한 영역을 평균값으로 칠합니다.

1. 기타 효과

가우시안 필터링을 통해 이미지를 약한 흐릿하게 만들었습니다. 또 대비를 높여서 밝은 부분은 더 밝게 어두운 부분은 더 어둡게 표현하였습니다.

기타

1. 시연 영상

여러 가시 실험과 시연에 사용된 SBS 드라마 ‘타짜’ 7회 영상에서 일부를 추출하였습니다. 만화를 원작으로 만든 드라마이기 때문에 다시 만화로 바꾸어 보는 것이 흥미로운 점이라고 생각합니다.

1. 발표자료

프로젝트를 진행하면서 만들었던 모든 발표자료를 만들었습니다.