**<1장표지>**

**안녕하세요 5조 발표를 맡게 된 문석현입니다**

**<2장>**

**발표는 주제, 중간발표 요약, 코드에 대해 직접 구현한 부분과 라이브러리를 활용한 부분에 대한 구분 및 기능 설명, 프로그램 시연 영상, 프로그램 개선점 순서로 진행하겠습니다.**

**<3장>**

**저희 조는**

**1) Image Stitching을 주제로 두 영상을 프레임 별로 결합하여 비디오로 저장하는 프로젝트를 진행하였습다. 이미지 스티칭이란 고해상도 이미지 및 넓은 시야의 이미지를 생성하기 위해 다중의 영상을 정합하는 방법을 의미합니다.**

**(2) 비디오 스티칭은 여러 분야에서 응용 가능합니다. 대표적으로 CCTV 영상의 실시간 비디오 스티칭을 통한 범죄 현장 예방, VR 기술에서의 활용 등 여러 사례가 있습니다.**

**<4장>**

**다음은 코드의 전체 플로우에 대해 말씀드리겠습니다. 먼저 두 비디오를 input한 후 비디오의 각 프레임에 대해 SURF 알고리즘을 사용하여 특징점 추출하고 디스크립터를 지정시킨 후, 대응되는 특징점들을 매칭합니다. 매칭쌍을 통해 계산한 호모그래피를 사용하여 한쪽 프레임들을 warping하여 스티칭을 진행합니다.**

**<5장>**

**다음은 중간발표때까지 진행했던 내용입니다. 저희 조는 2개의 동일한 기종의 스마트폰을 사용하였고 영상의 흔들림 최소화하기 위해 고정된 위치에서의 촬영한 영상을 사용했습니다. 코드 구현은 전체 틀을 잡는 방향으로 대부분 Opencv의 라이브러리의 함수를 활용하여 진행하였습니다.**

**<6장>**

**다음은 중간발표 시점에서 발견되는 문제점입니다. 사진은 영상 내의 랜덤한 3개의 프레임입니다. 영상의 frame별로 homography를 계산을 진행하여, 각자 다른 homography를 사용하여 warping을 하였기 때문에 Stitching을 진행하면 결과 영상이 다르게 나오게 되었습니다. 이를 개선하기 위해 단일한 homography을 계산하여 전체 warping에 적용하였습니다.**

**<7장>**

**또한 사진에서 보시는 바와 같이 부자연스러운 스티칭 경계가 발생하여 영상의 결과가 매끄럽지 않은 것을 확인할 수 있었습니다. 그래서 중간발표 이후, 이를 해결하기 위해서 단순 나열식의 Stitching이 아닌 블렌딩을 통해 프레임 쌍을 결합하였습니다.**

**그리고 스티칭 이후, 결과 영상에 오른쪽 프레임의 Warping으로 인해 검은색 잔여 영역이 발생했는데, 깔끔한 결과 영상을 위하여 Cropping 작업을 추가하였습니다.**

**<8장>**

**다음은 최종 플로우 차트입니다. 중간발표 때는 라이브러리만을 활용하여 Stitching을 진행했으나 위에서 언급한 것처럼 문제점들을 해결하기 위해 이미지 블렌딩 과정과 Cropping 과정을 직접 구현하여 추가하였습니다.**

**<9장>**

**라이브러리 활용에서는 SURF algorithm이 구현되어 있는 SURF클래스를 활용하였는데 라이선스 문제가 있어 opencv에서 교육용으로 제공하고 있는 SURF를 사용했습니다. 또한 bruteforce로 매칭을 해주는 BF Matcher, 그리고 Warpperspective 함수를 사용하였습니다.**

**직접 구현에서 호모그래피는 DLT 알고리즘을 사용하여 homography를 추정을 했습니다. 이미지 블렌드의 경우 멀티 밴드 블렌딩을 구현하였습니다.**

**<10장>**

**프로젝트의 기능을 코드의 동작 순서에 맞추어 설명하겠습니다.**

**처음으로, 두 영상의 frame으로부터 stitching을 시작하기에 앞서 SURF알고리즘을 사용하여features 추출 및 descriptor를 계산을 합니다. 화면에서 보시는 두 영상에서 feature들이 추출됨을 확인할 수 있습니다.**

**<11장>**

**다음으로 BFmatcher를 사용해 두 frame에서 동일한 특징점인 것들을 matching 하였습니다. homography 계산을 위해 별도로 좋은 matching에 대한 특징점들을 따로 저장을 하였습니다. 화면에서 좋은 매칭으로 연결된 특징점들을 확인할 수 있습니다.**

**<12장>**

**다음으로 프레임 별로 다르게 나오는 영상 부분 문제가 있었고 이를 해결하기 위해 하나의 프레임에 대한 고정된 호모그래피를 적용하여 문제를 해결했습니다. 원래 두 동영상을 결합하기 위해 프레임별로 다르게 나오는 영상들을 local homography를 적용 모든 프레임 값을 계산하려 했으나 충분한 자료가 부족하고 저희 조가 촬영한 비디오 특성 상 고정된 위치에서 촬영을 했기 때문에 global homography를 적용해도 무방하다는 피드백을 받아 호모그래피를 추정하는 알고리즘인 DLT를 사용해 호모그래피를 추정하였습니다.**

**<13장>**

**DLT 알고리즘은 충분한 특징점들의 대응들이 주어졌을 때 H를 계산하기에 간단하기 때문에 구현에 사용했습니다. Homogeneous coordinate를 적용해 특징점들의 좌표를 다음과 같이 나타낼 수 있고 변환행렬인 homography H에 대해 행렬 식을 풀어서 우측과 같은 두개의 방정식을 나타낼 수 있습니다. 즉, 한 쌍의 대응점에 대해서 2개의 식이 생기고 이 방정식을 행렬의 곱으로 나타낼 수 있는데 총 n개의 feature들에 대해 행렬식을 풀어 나타내면 다음과 같이 크기가 2n\*9인 행렬 A와 호모그래피 원소들을 열벡터로 표현한 행렬의 곱으로 나타낼 수 있습니다.**

**<14장>**

**다음으로 SVD를 사용해서 V의 맨 마지막에 열 벡터를 호모그래피로 estimation 하는 방식을 사용 A를 udv의 transpose Decomposition을 통해 호모그래피 추정이 가능해집니다. 이 때 homography estimation을 하기 위해서 앞에서 구한 행렬 A를 Singular value Decomposition을 적용해 V의 transposition으로부터 V를 구할 수 있고, V의 맨 마지막 열벡터로부터 homography를 추정한 호모그래피 H를 얻을 수 있게 됩니다. 이때 SVD는 opencv에서 제공하는 함수를 사용하였습니다.**

**<15장>**

**언급했던 바와 같이 두개의 영상에서 맨처 음의 frame에 대해 특징점들을 구하고 매칭하여 DLT알고리즘으로 호모그래피 추정을 한 다음, 이 호모그래피를 영상전체에 적용을 해서 문제점을 해결하였습니다.**

**<16장>**

**DLT알고리즘을 조금 더 정확하게 하기 위해서 normalized DLT을 사용하여 호모그래피를 추정을 또한 해보았습니다.**

**먼저 각각의 영상에서 구한 특징점들 xi와 x’i에 대해 각각 similarity transform T에 의해 정규화된 특징점들을 구할 수 있는데 이때 similarity transform T는 특징점들의 centroid를 원점으로 보내고 그 원점에서 특징점들까지의 거리가 루트 2가 되게 하는 변환행렬 T로 이 정규화하는 과정을 통해서 similarity transformation T와 T’을 각각 구할 수 있습니다.**

**그 다음 정규화된 두개의 특징점들의 집합들 간에 호모그래피 추정을 아까 말씀드린 DLT 알고리즘을 적용해서 구해서 다음과 같이 호모그래피를 추정할 수 있습니다.**

**<17장 표지>**

**다음으로 이미지 와핑 작업인데 이미지 와핑 작업을 하는 이유는 이미지의 형태를 변환시켜 두 이미지간 블렌딩 작업을 용이하게 하기 위해서 입니다. 이를 위해 warpperspective함수를 사용 두 이미지의 행과 열을 변환시켜 두 이미지를 스티칭 하는데 사용하였습니다.**

**<18장 표지>**

**다음으로는 이미지 블렌딩에 대해 설명드리도록 하겠습니다. 앞서 중간 발표시 경계면에서 부자연스러운 현상이 발생하였습니다. 이를 해결하기 위해 이미지 블렌딩 기법을 사용하였데는 순서는 보시는 것과 같이 각 프레임에 대하여 가우시안 피라미드를 통해 라플라시안 피라미드 생성하고 블렌드 한 후 층별 이미지를 합하는 과정을 멀티 밴드 블렌딩 작업이라고 합니다.**

**<19장 표지>**

**먼저 LapacianBlending 함수에 좌,우 이미지, 블렌딩 배경 마스크, 밴드 레벨을 입력합니다. 블렌딩 배경 마스크는 두 이미지의 경계를 결정해주는 경계입니다. 경계 설정에 따라 영상의 부드러움의 정도가 달라질 수 있습니다. 밴드 레벨은 앞서 설명한 피라미드의 층 깊이를 뜻합니다. 밴드가 깊을 수록 다양한 주파수의 이미지를 blend할 수 있습니다.**

**<20장 표지>**

**먼저 각 프레임에 대하여 Laplcain Pyramid를 빌드합니다. Opencv 제공의 pryDown 함수와 pryUp 함수를 통해 가우시안 피라미드를 구축하고 그 차를 통해 Laplacian Pyramid를 얻을 수 있습니다. buldLaplacianPyramid의 함수를 보시면 세번째에 smallestLevel에 대한 마스크를 따로 추출합니다. 이는 후에 가장 low frequency인 밴드 레벨 마스크에서 high frequency인 밴드 레벨의 마스크로 점차적으로 블렌딩 하는 멀티밴드블렌딩의 알고리즘에서 필요한 마스크입니다.**

**<21장 표지>**

**다음은 두 라플라시안 피라미드를 블렌드하여 하나의 블렌디드 라플라시안 피라미드로 만드는 과정입니다. 아래의 식을 살펴보면 X 성분이 좌,우 라플라시안 이미지 그리고 M 성분이 블렌딩시 필요한 가우시안 마스크입니다. 각 밴드 레벨에서 왼쪽 라플라시안 이미지와 블렌딩 마스크의 곱과 오른쪽 라플라시안 이미지와 (1-블렌딩 마스크)의 곱을 합하면 블렌디드 라플라시안 피라미드를 얻게 됩니다.**

**<22장 표지>**

**다음으로 가장 낮은 주파수를 갖는 라플라시안 이미지에서 다음 밴드레벨 이미지의 사이즈에 맞추어 합하여 블렌딩을 수행하느 부분입니다. 순차적으로 각 모든 밴드 레벨의 라플라시안 이미지가 더해지면 최종 이미지를 얻을 수 있습니다.**

**<23장 표지>**

**마지막으로 Rect 함수를 사용하여 사용자가 미리 측정한 좌표에 맞추어 사각형을 만든 후 원하는 이미지 영역을 Cropping하는 작업을 수행하였습니다.**

**<24장>**

**이제 결과물에 대해 말씀드리겠습니다**

**우선 중간 발표 결과물에 경우 두 이미지의 영상이 매 프레임별로 호모그래피가 달라 와핑이 다르게 되어 불안정 모습이 보이고 부자연스러운 스티칭 경계 영상 및 검정색 화면이 남아 개선해야 될 점들이 있었습니다.**

**<25장>**

**이 문제점들을 해결한 최종 결과 영상입니다.**

**homography dlt 함수를 적용하여 두 이미지가 영상이 불안정한 부분이 해소되었음을 알 수 있습니다. 또한 영상의 스티칭된 경계부분을 보시면 이미지 블렌딩을 적용을 했기 때문에 전과 다르게 자연스러워 진 것을 확인할 수 있습니다. 또한, 여백으로 보이던 검정 부분이 Crop되었음을 알 수 있었습니다.**

**<26장>**

**또한 normalized DLT로 homography를 추정하여 구한 결과 영상입니다.**

**기대와 달리 앞서 보여드린 영상과 다르게 스티칭된 경계부분이 뚜렷하고 영상이 매끄럽지 않음을 알 수 있습니다.**

**<27장>**

**마지막으로 코드의 한계점 및 개선 필요 사항에 대해 말씀드리겠습니다. 우선 프로젝트 테스트 영상 이외에 다양한 영상에 적용해보지 못해 영상의 신뢰성(reliable) 검증이 필요해 보입니다.**

**다음으로 고정된 두개의 카메라로 촬영된 영상에만 적용이 가능한 점입니다. 사람이 수동적으로 겹치는 부분을 일정부분 이상으로 고려하여 촬영해야 하는 문제점이 있습니다. 또한 마지막으로 보신 영상에서 적용한 normalized DLT가 기본 DLT 알고리즘보다 더 좋은 성능을 가질 것으로 예상했으나 더 낮은 성능을 보였습니다. 이 부분에 대해서는 영상의 문제점인지 알고리즘의 문제점인지 더 연구를 해봐야할 것 같습니다.**

**<28장>**

**다음으로 영상 촬영조건에 따라 stitching의 결과의 품질이 달라지게 됩니다. 영상 내에서 특징점이 잘 추출되는 부분에서는 충분한 점들을 추출하여 호모그래피를 계산을 할 수 있으나 그렇지 않은 경우, 저희 영상에서는 차선 부분에서 특징점들이 충분히 뽑히지 못해 약간 어긋나 스티칭품질이 낮은 문제가 발생함을 알 수 있었습니다.**

**마지막으로 영상에서 블렌딩의 경계와 cropping 영역을 직접 사용자가 설정해줘야 한다는 추가과정이 있었습니다. 이 부분을 인지하고 자동으로 처리해주는 방법에 대해서는 더 많은 연구가 필요해보입니다.**

**<29장>**

**이상으로 발표를 마치겠습니다 감사합니다**