머신러닝 기법을 적용한 CS-RANSAC 알고리즘

고승현*, 윤의녕*, 주마벡*, 조근식*
*인하대학교 컴퓨터정보공학과
e-mail : kosehy@gmail.com, entymos@hotmail.com, jumabek4044@gmail.com, gsjo@inha.ac.kr

CS-RANSAC Algorithm using Machine Learning Technique

Seunghyun Ko*, Ui-Nyoung Yoon*, Jumabek Alikhanov*, Geun-Sik Jo*
*Dept. of Computer Science & Information Engineering, Inha University

요 약

증강현실에서 영상과 증강된 콘텐츠 간의 이질감을 줄이기 위해서 정확한 호모그래피 행렬을 추정해야 하며, 정확한 호모그래피 행렬을 추정할때 RANSAC 알고리즘이 널리 사용된다. 그러나 RANSAC 알고리즘은 랜덤 샘플링 과정을 반복적으로 거치기 때문에 불필요한 연산 과정이 발생하고 이로 인해 알고리즘의 효율이 저하된다. 이러한 단점을 극복하기 위해 DCS-RANSAC 알고리즘이 제안되었다. 제안된 DCS-RANSAC 알고리즘은 이미지를 특징점 분포 패턴에 따라 그룹으로 분류하고 각 그룹에 제약조건 문제를 적용하여 불필요한 연산 과정을 줄이고 정확도를 향상시킨 알고리즘이다. 그러나 DCS-RANSAC 알고리즘에서 사용된 이미지 그룹 데이터는 수동적인 방법을 통해 직관적으로 분류되어 있지만 특징점 분포 패턴이 다양하지 않아 분류시 정확도가 저하되는 경우가 있다. 위의 문제점을 해결하기 위해 본 논문에서는 머신러닝 기법을 통해 이미지들을 자동으로 분류하고 각 그룹마다 각기 다른 제약조건을 적용하는 MCS-RANSAC 알고리즘을 제안한다. 제안하는 알고리즘은 머신러닝 기법을 사용하여 전처리 단계에서 이미지를 분류하고 분류된 이미지에 제약조건을 적용시켜 알고리즘의 처리시간을 줄이고 정확도를 향상시켰다. 실험 결과 본 논문에서 제안하는 MCS-RANSAC 은 DCS-RANSAC 알고리즘에 비해 수행시간이 약 6% 단축되었고 호모그래피 오차율은 약 15% 줄어들었으며 참정보 비율은 2.8% 증가한 것으로 확인되었다.

1. 서론

최근 실시간으로 실제 환경에 존재하는 객체 위에 컴퓨터로 생성한 가상의 객체 혹은 영상을 합성시켜 실제의 사물로 보이게 하는 증강현실(AR: Augmented Reality)에 대한 연구가 활발히 진행되고 있다[1,2]. 이러한 가상의 객체를 2D 이미지 기반으로 저작한 후에 저작된 이미지와 실세계의 영상을 결합시킬 때 결과물이 이질감 없이 보여지기 위해서는 가상의 객체를 증강시키고자 하는 영상의 위치와 방향을 정확하게 추정해야 한다. 이를 위해서는 목표 객체를 매 프레임마다 정확하게 인식하고 추적하는 기술이 요구된다. 추적기술은 크게 마커, 비마커, 하이브리드 기반 기술로 나눌 수 있다. 여기서 비마커 기반의 추적기술은 마커나 하이브리드 방식보다는 복잡한 연산과 수행시간을 필요로 하지만 점, 선, 에지 등과 같은 실제 환경의 자연적인 특징들을 이용 할 수 있어서 최근 들어 활발하게 사용되고 있다[1].

카메라의 자세와 방향을 파악하여 가상의 객체를 실제 영상위에 보다 정확하고 이질감 없게 증강시키기 위해 비마커 기반의 추적 기술에서는 주로 호모그래피(Homography) 행렬이 사용된다[3].

호모그래피 행렬을 추정하는데 있어 RANSAC 알고리즘이 널리 사용되는데 이는 두개의 연속된 프레임에 위치한 동일한 부분을 반복적으로 탐색하여 두 프레임 사이의 호모그래피 행렬을 추정하는데 사용된다. 하지만 RANSAC 알고리즘은 SURF 에 의해 추출된 특징점들 중에서 랜덤샘플링을 할 때 선형이나 군집을 이루는 특징점들을 선택 할 경우에는 잘못된 호모그래피 행렬을 추정할 수 있다. 이 문제점을 해결하기 위해 RANSAC 의 랜덤샘플링 과정에서 제약만족문제 (Constraint Satisfaction Problem)를 동적으로 적용한 DCS-RANSAC 알고리즘이 연구 되었다[4-6].

DCS-RANSAC 은 이미지들을 특징점 분포 패턴에 따라 다섯가지 그룹으로 분류하고 K-Means 클러스터링 알고리즘을 이용하여 분류된 그룹 각각의 최적의 단위격자 크기를 동적으로 구하여 이를 CS-RANSC 알고리즘에 적용하여 알고리즘의 성능을 높였다. 하지만 특징점 분포 패턴에 따라 분류된 다섯가지 이미지 그룹은 수동적인 방법을 통해 직관적으로 분류되어 있고 실험에 쓰인 UKBench dataset[9]의 다양한 특징점 분포 패턴을 모두 포함하지 못하는 단점이 있다[7-8].

본 논문에서는 전처리 과정에서 머신러닝 기법인 K-Means 클러스터링 알고리즘을 이용하여 특징점 분포

패턴에 기준으로 이미지들을 자동으로 분류하고 이 결과를 SVM 기법[10]에 적용하여 입력된 이미지를 자동으로 식별하는 방법을 제안한다[11].

본 논문의 구성은 다음과 같다. 1 장의 서론에 이어 2 장에서는 관련연구를 기술하고, 3 장에서는 제안하는 알고리즘을 기술한다. 4 장에서는 제안된 방법에 대한 실험결과를 다루고, 마지막으로 5 장에서는 결론을 도출한다.

2. 관련연구

본 장에서는 RANSAC 알고리즘과 RANSAC 알고리즘의 샘플링 과정에서 발생하는 단점들을 줄일 수 있는 알고리즘을 소개한다.

2.1 RANSAC

RANSAC 알고리즘은 랜덤으로 샘플 데이터를 선택하여 구하고자하는 수학적 모델을 계산하고 가까이에 있는 데이터들의 개수를 세어 개수가 큰 모델을 기억하고 이 과정을 N 번 반복한 후 지지하는 데이터의 개수가 많은 모델을 구하는 방법이다[4].

RANSAC 알고리즘에는 특징점을 임의로 선택하는 랜덤 샘플링방식이 사용된다. 이때 선택된 특징점들의 분포가 선형 혹은 군집을 이루는 경우가 발생하는데 이러한 경우에는 정확한 호모그래피를 추정하기가 어렵다. 이는 알고리즘의 수행시간을 증가시키고 효율을 저하시키는 문제점이 발생한다.

2.2 삼각형의 넓이를 이용한 RANSAC(TR-RANSAC)

이 알고리즘은 RANSAC 알고리즘의 샘플링 단계에서 3 개의 특징점을 이은 삼각형의 넓이를 사용한 필터링을 사용한다. 샘플링 단계에서 선택된 특징점들의 선형 혹은 군집이 되는 것을 방지하기 위해, 3 개의 특징점들을 선택하여 삼각형의 넓이를 구하고 이들의 넓이의 합이 일정 임계값을 넘어설 경우에는 샘플 특징점으로 선택하고 아닐 경우에는 필터링하는 알고리즘이다[12].

2.3 CS- RANSAC

CS-RANSAC 알고리즘은 RANSAC 알고리즘에서 발생할 수 있는 랜덤 샘플링에 의해 선택된 특징점의 분포가 선형 혹은 군집을 이룰 경우 부정확한 호모그래피 추정을 방지하기 위해 제약 만족 문제를 적용한 알고리즘이다[7]. 이미지를 N×N 으로 나누고 선택된 특징점의 가로, 세로, 대각선에 위치한 특징점들을 중 하나의 단위격자의 특징점을 선택하는 선형제약조건과, 주변 2 칸, 대각선 1 칸에 포함된 단위격자의 특징점을 선택하지 않는 거리제약조건을 가진다. 하지만 랜덤으로 인한 샘플링으로 인해 낮은 정확도의 호모그래피 행렬을 추정할 수 있는 특징점들을 다시 샘플링 단계에서 선택되는 경우가 있고 이는 알고리즘의 효율을 저하시키는 결과를 가져온다.

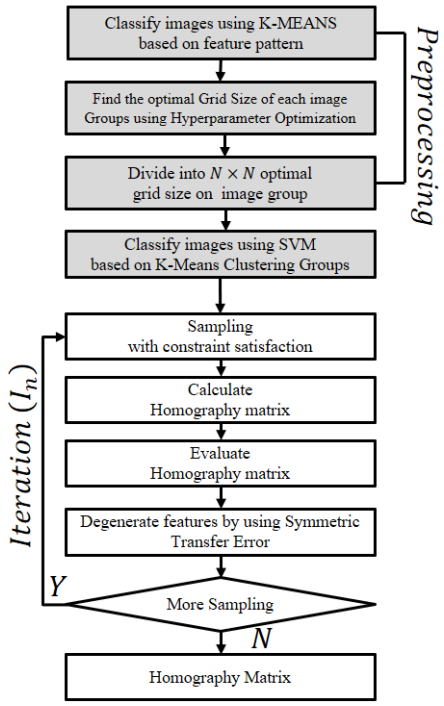
2.4 DCS-RANSAC

CS-RANSAC 의 문제점을 해결하기 위해 이미지를 특징점 분포 패턴에 따라 분류된 이미지에 K-Means 클러스터링 기법을 사용하여 각 그룹의 최적의 단위격자 크기를 찾아 불필요한 연산을 줄이는 DCS-RANSAC 알고리즘이 연구되었다[6].

하지만 K-Means 클러스터링 결과를 적용시키는 과정에서 사용되는 이미지 그룹은 수동적인 방법을 통해 직관적으로 분류된 그룹으로 실제 실험에 쓰이는 테스트 이미지의 다양한 특징점 분포 패턴을 모두 포함하지 못하여 이미지 분류시에 정확도가 저하되는 문제가 있다.

3. 머신러닝 기법을 적용한 CS-RANSAC 알고리즘

본 논문에서는 머신러닝 기법인 K-Means, SVM, 그리고 Hyperparameter Optimazation[13]을 사용하여 입력된 이미지들을 특징점 분포 패턴에 따라 자동으로 분류하는 방법(MCS-RANSAC)을 제안한다. 그림 1 은 기존의 DCS-RANSAC 에 제안하는 방법이 추가된 알고리즘의 순서도이다. 제안하는 알고리즘에서는 전처리 단계에서 K-Means 알고리즘을 이용하여 모든 이미지를 특징점 분포 패턴에 따라 클러스터링 한다. 실험을 통해 구한 최적의 클러스터링 수를 사용하여 이미지들을 8 가지 그룹으로 나누고 각각의 이미지 그룹들의 최적의 단위 격자 크기를 실험을 통해 구한 처리시간, 호모그래피 오차율, 참정보 비율 값에 Hyperparameter Optimazation 기법을 적용하여 구한다. 위의 전처리 단계는 데이터셋이 변하지 않는 한 한번만 작동한다. SVM Classifier 를 사용하여 입력된 이미지를 8 그룹에 자동으로 분류를 하고 전처리 단계에서 구한 최적의 격자 수를 적용하여 샘플링 단계에서 탐색 범위를 줄임으로써 알고리즘의 처리시간을 줄이고 정확도를 향상시킨다.



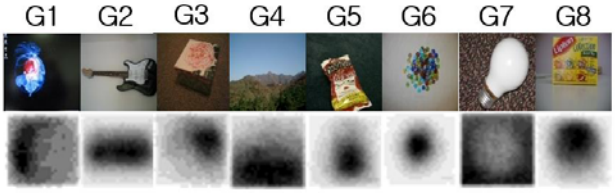
(그림 1) MCS-RANSAC 의 순서도

4. 실험 및 결과

본 논문에서는 MCS-RANSAC 의 검증과 성능평가를 위해 총 4 가지의 실험을 수행하였다. 첫번째 실험은 K-Means 클러스터링 방법을 사용하여 이미지들의 특징점 분포 패턴을 기반으로 최적의 그룹수를 정한다. 두 번째 실험은 각 이미지 그룹별로 처리시간, 호모그래피 오차율, 참정보 비율을 가지고 최적의 단위 격자 크기를 측정한다. 세번째 실험은 제안하는 방법과 DCS-RANSAC 의 수행시간, 호모그래피 오차율, 참정보 비율을 측정 및 비교한다. 마지막 실험은 8 개의 이미지 그룹에 대하여 RANSAC, CS-RANSAC, TR-RANSAC, MCS-RANSAC 의 성능을 수행시간, 호모그래피 오차율, 참정보 비율을 사용하여 비교하였고 MCS-RANSAC 을 제외한 나머지 알고리즘은 논문[7]를 참고하여 단위 격자 크기를 3 개의 알고리즘 모두 최적의 결과 값을 갖는 17X17 로 하였다.

4.1 머신러닝을 이용한 이미지 분류

K-Means 클러스터링 알고리즘과 SVM 알고리즘을 사용하여 UKBench dataset 을 이미지의 특징점 분포 패턴에 따라 최적의 그룹 수를 찾는 실험을 진행하였다. K-Means 클러스터링 알고리즘에 사용할 클러스터링 숫자를 5-60 까지 변경하여 알고리즘을 수행하였다. 수행결과 겹치는 그룹이 많이 발생하고 겹치는 그룹 대부분의 데이터 수가 적어서 유사한 그룹들을 서로 병합하여 최종적으로 8 개 그룹을 생성하였다. 이를 SVM 알고리즘에 적용시킨 결과 약 80%의 식별확률을 기록했다.



(그림 3) 특징점 분포 패턴에 따라 분류된 8 개의 그룹

특징점 분포 패턴에 따라 분류된 그룹은 그림 3 과 같이 8 개의 그룹이며 각 그룹에 대한 설명은 아래와 같다.

G1: 이미지 왼쪽 부분에 분포
G2: 이미지 중앙에 길게 분포
G3: 이미지 오른쪽 상단에 분포
G4: 이미지 아래부분에 넓게 분포
G5: 이미지 중앙 약간 오른쪽에 분포
G6: 이미지 중앙에 분포
G7: 이미지 중앙을 제외한 부분에 분포
G8: 이미지 중앙 위쪽에 분포

4.2 각 이미지 그룹별 최적화된 격자 크기

위의 실험에서 구한 8 개의 이미지 그룹들에 맞는 최적의 단위 격자 크기를 구하기 위해 MCS-RANSAC 을 수행한 후 Hyperparameter Optimization 기법을 사용하여 호모그래피 오차율과 수행시간은 최소값, 참정보 비율은 최대값을 수렴하는 단위 격자 크기를 구하였다. 실험에 사용된 이미지는 UKBench dataset 를 사용하였고 알고리즘은 100 번씩 반복 수행하였다.

실험 결과 각 이미지 그룹마다 최적의 단위 격자 크기가 다르게 나타나는 것을 볼 수 있다. 대체적으로 특징점이 몰리는 그룹의 경우 세밀한 크기의 단위 격자를 가지고 특징점 분포가 넓게 분포된 그룹의 경우에는 넓은 크기의 단위격자를 갖는 것을 볼 수 있다.

<표 1> 각 이미지 그룹별 최적의 단위 격자 크기

	Optimal Grid Size
G1	18
G2	15
G3	26
G4	10
G5	20
G6	15
G7	28
G8	10

4.3 DCS-RANSAC vs MCS-RANSAC

DCS-RANSAC 과 MCS-RANSAC 의 성능 측정 및 비교를 하는 실험을 진행하였다. 데이터는 DCS-RANSAC 에서 사용된 이미지를 사용하였고 DCS-RANSAC 과 MCS-RANSAC 을 둘다 수행하여 평균값을 비교하였다.

실험결과 제안한 방법(MCS-RANSAC)은 DCS-RANSAC 알고리즘보다 수행속도가 평균 6.4%, 호모그래피 오차율은 13.2% 감소하고, 참정보 비율은 2.8% 증가한 것으로 확인되었다. 이는 MCS-RANSAC 의 머신러닝을 사용한 자동 이미지 분류방법이 DCS-RANSAC 의 수동적인 이미지 분류 방법에 비해 더욱 더 세밀한 정보를 얻은 결과로 볼 수 있다

<표 2> DCS-RANSAC 과 MCS-RANSAC 의 성능 비교

결과값 \ 알고리즘	DCS-RANSAC	MCS-RANSAC
처리시간(ms)	64.548	60.3905
에러율(pixel)	31.124	27.025
참정보 비율(%)	66.22	68.09125

4.4 MCS-RANSAC 의 성능 평가

성능비교는 8 개의 이미지 그룹에 대하여 RANSAC, CS-RANSAC, TR-RANSAC, MCS-RANSAC 의 4 가지 알고리즘을 동일한 환경에서 각 그룹마다 4 쌍의 이미지를 적용하여 각각의 알고리즘을 100 번 반복수행 하여 얻은 값들의 평균값을 표 3,4,5 와 같이 비교하였다.

실험결과 MCS-RANSAC 이 다른 알고리즘에 비해 호모그래피 오차율은 표 4 와 같이 약 9.6%에서 47.7%까지 평균 23.8%의 향상을 보였고 참정보 비율은 표 5 와 같이 약 0.1%에서 6.4%까지 평균 2.6%의 성능 향상을 보였다. 각 이미지 그룹마다 최적의 단위 격자 크기를 적용했기 때문에 호모그래피 오차율을 감소시키고 참정보 비율을 향상 시킬 수 있었다. 수행시간은 표 3 과 같이 MCS-RANSAC 의 성능이 TR-RANSAC 보

다 저조한 것으로 확인되었으나 이는 특징점 샘플링 과정에서 MCS-RANSAC 은 수행시간, 호모그래피 오차율 및 참정보 비율에 초점을 둔 반면 TR-RANSAC 은 선형과 군집 문제만을 해결하는데 초점을 두었기 때문이다. 또한 MCS-RANSAC 의 샘플링단계에서 특징점 분포가 희박한 부분의 특징점이 선택되어 불필요한 연산이 증가하는 단점이 있다.

<표 3> MCS-RANSAC 과 다른 알고리즘 처리시간(ms) 비교

그룹 / 알고리즘	G1	G2	G3	G4	G5	G6	G7	G8
RANSAC	135.094	146.943	149.472	145.794	226.433	32.3153	397.146	71.5493
CSP	137.997	115.559	126.022	138.694	225.401	33.4417	412.73	61.7471
TR_RAN	42.3752	29.5712	29.5092	144.808	56.1969	14.0531	211.881	11.4088
MCS_RAN	129.432	113.116	87.4341	103.419	150.172	30.458	377.752	14.1373

<표 4> MCS-RANSAC 과 다른 알고리즘 호모그래피 오차율(pixel) 비교

그룹 / 알고리즘	G1	G2	G3	G4	G5	G6	G7	G8
RANSAC	4.34796	76.7108	55.4865	32.9834	223.145	87.126	4.70586	16.468
CSP	4.43244	43.1652	47.7426	35.5777	165.837	65.8425	5.1116	13.8197
TR_RAN	4.3814	62.3884	47.8437	43.0346	185.632	88.5261	4.88687	13.0972
MCS_RAN	5.1013	40.1236	36.1636	29.8028	147.622	66.2243	3.48751	11.1617

<표 5> MCS-RANSAC 과 다른 알고리즘 참정보 비율(%) 비교

그룹 / 알고리즘	G1	G2	G3	G4	G5	G6	G7	G8
RANSAC	67.3149	57.2766	58.2586	42.992	53.3966	92.9518	41.6231	95.1633
CSP	67.4581	61.617	58.8793	43.9385	56.514	93.0181	41.7204	95.3265
TR_RAN	67.1906	58.2194	58.6814	43.0679	54.2184	92.9801	41.5808	95.277
MCS_RAN	67.4928	62.1915	59.9483	44.0642	57.3128	93.0723	41.8495	95.3503

5. 결론 및 향후 연구

본 논문에서는 머신러닝 기법을 이용하여 입력된 이미지를 특징점 분포 패턴에 따라 각 이미지 그룹별로 구별하고 그룹마다 최적화된 단위격자 크기를 적용하여 CS-RANSAC 알고리즘의 효율성을 향상시키는 알고리즘을 제안하였다. 실험결과 MCS-RANSAC 알고리즘은 DCS-RANSAC 보다 수행속도가 평균 6.4%, 호모그래피 오차율은 13.2% 감소하고, 참 정보 비율은 2.8% 증가하였다. RANSAC, CS-RANSAC, TR-RANSAC, DCS-RANSAC 알고리즘에 비해 호모그래피 오차율은 평균 23.8%의 향상을 보였고 참정보 비율에서는 평균 2.6%의 향상을 보였다. 수행시간은 TR-RANSAC 에 비해 저조하게 나왔지만 이는 샘프링 단계에서의 불필요한 연산으로 인해 발생하는 문제라고 볼 수 있다.

향후 연구에서는 전처리 단계에서 각 이미지 그룹에 따라 특징점 분포가 희박한 부분을 필터링하여 샘플링 과정에서의 불필요한 연산을 줄일 수 있는 연구가 필요하다.

참고문헌

[1] K. Y. Eom, G. J. Kim, M. H. Kim, “A Study of Markerless Object Recognition and Tracking for Augmented Reality,” Communications of the KIISE, vol28, no.8, pp. 54-66, 2010. (in Korean)

[2] F.Zhou, H.B.Duh, and M.Billinghurst, “Trends Augmented Reality Tracking, Interaction and Display: A Review of Ten Years of ISMAR,” IEEE International Symposium on Mixed and Augmented Reality, pp. 193-202, 2008.

[3] S. J. D. Prince, K. Xu, and A. D. Cheok, “Augmented reality camera tracking with homographies,” Computer Graphics, vol. 22, no. 6, pp. 39-45, 2002.s

[4] M.A. Fischler and R.C. Bolles, “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography,” Communications of the ACM, pp. 381-395, 1981.

[5] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, “SURF: Speeded Up Robust Features,” Computer Vision and Image Understanding, vol. 110, no. 3, pp. 346-359, 2008.

[6] D. Chandra, “Constraint-based Sampling in RANSAC Algorithm with a Flexible Domain Size,” Inha Univercity, 2014.

[7] G. S. Jo, K. S. Lee, C. Devy, C. H. Jang, and M. H. Ga, “RANSAC versus CS-RANSAC,” American Association for Artificial Intelligence(AAAI), pp. 1350-1356, 2015.

[8] C. H. Jang, K. S. Lee, and G. S. Jo “CSP Driven RANSAC Algorithm for Improving Accuracy of Planar Homography” Journal of Korean Institute of Information Science and Engineers, Vol. 39, No. 11, pp. 876-888, 2012.

[9] UKbench dataset(Center for Visulization&Virtual Environments), http://vis.uky.edu/~stewe/ukbench/, 2015.

[10] SVM, https://en.wikipedia.org/wiki/Support_vector_machine, 2016.

[11] C.-C. Chang and C.-J. Lin, ”LIBSVM : a library for support vector machines,” ACM Transactions on Intelligent Systems and Technology, pp. 2:27:1--27:27, 2011.

[12] E.Vincent and R.Laganiere, “Detecting Planar Homographies in an Image Pair,” IEEE International Symposium on Image and Signal Processing and Analysis, pp. 182-187, 2001.

[13] Bishop and Christopher “Pattern Recognition and Machine Learning” Springer, 2007.