공간 표현과 입력 영상 관점에서 살펴본 Visual Localization 연구 동향 분석

최 진 원*, 허 동 욱*, Nguyen Cong Quy*, 서 찬 호*, 장 인 성**, 최 성 록*
(*서울과학기술대학교 컴퓨터공학과, **한국전자통신연구원 모빌리티인프라연구실)

1. 서론

Visual localization (영상기반 측위기술)은 카메라에 촬영된 영상을 분석하여 영상이 촬영된 카메라의 위치를 알아내는 기술이다. Visual localization은 카메라가 갖는 많은 장점으로 인해 다양한 분야와 제품에 활용된다. 우선 visual localization은 자동차나 로봇, 드론의 자율주행이나 모바일매핑시스템(mobile mapping system; MMS)을 위한 위치 정보를 제공할 수 있다. 특히 터널, 실내, 지하주차장, 높은 빌딩숲 등 GPS가 동작하지 않거나 정확도가 크게 떨어지는 환경에서 visual localization은 GPS 측위의 효과적인 대안이다. 스마트폰이나 태블릿과 같은 단말기는 카메라를 기본적으로 탑재하고 있어 실내외 위치기반서비스(location-based service; LBS)를 제공하기 위해 visual localization을 활용할 수 있다. 가상현실(VR)이나 증강현실(AR)에서도 가상의 환경이나 물체를 원하는 위치에 렌더링하기 위해 단말기의 위치와 자세를 정확히 알아야 하고 이를 위해 visual localization 기술을 활용한다.

측위를 위해 다양한 센서를 사용할 수 있지만, 카메라는 측위에 가장 많이 선택되는 센서들 중 하나이다. 우선 카메라는 가장 쉽게 접할 수 있는 센서 중 하나이다. 대부분의 스마트 단말기는 카메라를 탑재하고 있고, 자동차의 블랙박스 카메라나 도심 환경의 CCTV 카메라도 매우 흔히 볼 수 있다. 또 카메라는 상대적으로 매우 경제적인 센서이다. 측위에 많이 사용되는 LiDAR와 비교하여 카메라는 가격이 매우 싸고, 외부에 빛(에너지)을 방출하는 능동형(active) 센서가 아니라 빛을 받아들이는 수동형(passive) 센서로 전력 소모가 상대적으로 적다. 또 카메라는 회전 구동부가 포함된 LiDAR나 안테나가 필요한 GPS에 비해 매우 작고 가벼운 센서이다. 카메라가 획득한 영상은 환경의 많은 요소들로 영향을 받고 이로 인해 다양하고 많은 정보를 담고 있다. 그러나 이러한 점은 오히려 visual localization의 난제가 되기도 한다. 야외에서 햇빛의 양은 시간의 정보를 포함하고 있지만, 같은 장소에서 촬영한 영상 사이에 큰 차이를 만들기 때문에 올바른 장소 인식에 어려움이 된다. 극단적으로 햇빛의 양이 없는 야간은 카메라 영상을 전혀 사용하지 못하는 어려운 환경이 되기도 한다. 또한 비나 눈, 안개와 같은 날씨나 계절의 변화도 해당 시점의 정보가 될 수 있지만, 마찬가지로 동일 장소의 영상에 큰 차이를 만들기 때문에 visual localization의 난제이다. 이 외에 카메라 영상은 3차원 공간의 2차원 투영으로 카메라 영상만을 이용해 실제 공간의 크기나 카메라 움직임의 크기를 구분하지 못하는 근본적인 한계인 scale ambiguity 문제를 갖는다. 따라서 visual localization에서 카메라가 갖는 이러한 난제와 한계를 극복하고자 하는 다양한 방향의 연구들이 이뤄지고 있다.

본 기고에서는 visual localization의 최신 연구 동향을 살펴하고자 한다. 본 연구 동향 분석에서는 visual localization은 주어진 영상 데이터의 형태와 활용 방법, 그리고 공간(또는 지도)을 표현하는 방법에 따라 나누어 보고자 한다.

2장에서는 이러한 영상 입력과 공간 표현의 관점에서 visual localization 기술들을 소개하고, 다양한 하위 접근법들과 대표 연구들을 살펴본다. 3장에서는 visual localization에서 특히 많이 사용되고 있는 이미지 매칭 기법을 이용한 기술들을 소개한다. 이미지 매칭은 visual localization 뿐만 아니라 컴퓨터비전과 영상처리 등에 광범위하게 사용되는 기반 기술이다. Image Matching Challenge는 주어진 이미지들을 정합하여 각 이미지들의 상대적인 위치와 방향각을 측정하는 대회로, 3장에서는 Image Matching Challenge에 주로 많이 사용되는 계층적 visual localization 기법을 살펴본다. 특히 이미지 매칭을 위해 최근 제안된 전역 특징(global descriptor)와 지역 특징(local feature)들을 추가로 살펴본다. 4장에서는 3장에서 소개하지 않는 최근 visual localization 기법들을 공간 표현의 관점에서 나누어 소개한다. 5장에서는 visual localization 연구에 많이 활용되는 공개 데이터셋을 소개하고, visual localization의 성능평가를 위한 평가 지표를 소개한다. 끝으로 6장에서는 전체 연구 동향 분석 내용을 정리하고, 향후 필요한 연구 방향에 대해 논의한다.

2. Visual Localization의 분류

카메라는 그 구성에 따라 다양한 타입의 영상 데이터를 제공한다. 일반적인 구성은 한 개의 카메라로 구성된 단안(monocular) 카메라로 하나의 이미지를 제공한다. 두 개의 카메라로 구성된 스테레오(stereo) 카메라는 다른 시점의 두 장의 이미지를 제공하고, 두 장의 이미지를 매칭하여 깊이(depth) 영상을 추가로 얻을 수 있다. ToF(Time-of-Flight) 카메라는 LiDAR와 마찬가지로 빛을 방출하고 반사된 빛이 입사되는 시간을 측정하여 깊이 영상을 획득한다. 이렇게 RGB 영상 외에 깊이(depth) 영상을 추가로 얻을 수 있는 스테레오 카메라나 ToF 카메라를 RGB-D 카메라라고 하기도 한다.

본 연구 동향 분석에서는 (깊이영상을 사용하지 않고) 일반 단안(monocular) 카메라의 영상을 이용한 visual localization 기법들로 한정한다. 또한 GPS, IMU, LiDAR 등 다양한 센서를 융합한 visual localization 기법들도 소개하지 않는다. RGB-D 카메라나 GPS, IMU, LiDAR에서 얻은 부가적인 센서 데이터를 통해 보다 정확하고 신뢰성 높은 visual localization 기술을 만들 수 있다. 그러나 본 동향 분석에 이러한 기술들을 포함하면 범위가 너무 광범위해지고, 영상 데이터 본연의 활용보다 다른 모달리티의 정보 융합에 대한 부분이 더욱 강조될 수 있다.

2.1. Monocular Visual Localization의 분류

Monocular visual localization은 하나의 카메라에서 얻은 영상을 이용하여 영상이 촬영된 카메라의 위치를 추정하는 기술이다. Monocular visual localization에서는 입력(query) 데이터로 하나의 카메라에서 얻은 1 장의 영상만을 사용할 수도 있고, 현재의 영상을 포함한 과거 N장의

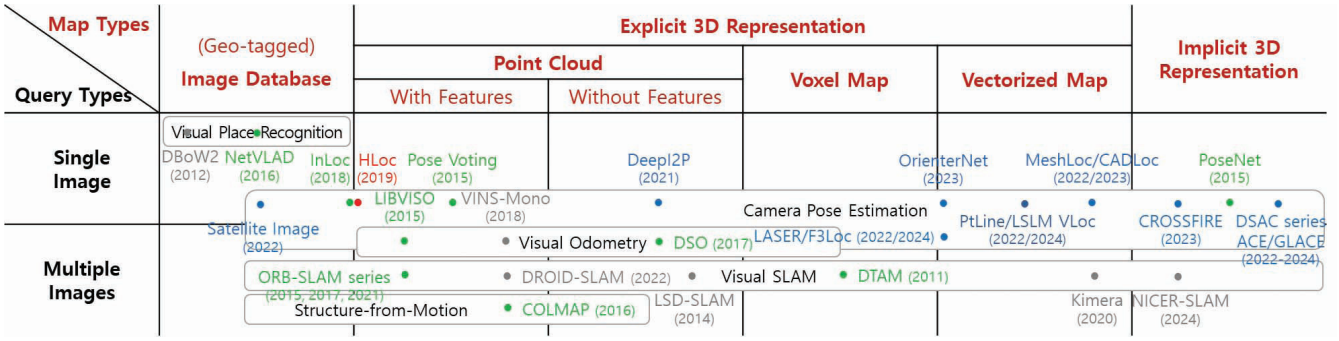

그림 1. **공간 표현(가로축)과 입력 영상(세로축)** 관점에서의 visual localization 기술 분류: 녹색으로 표시된 대표 기술은 2.2절에서 소개, 빨간색으로 표시된 기술은 3.1절, 파란색으로 표시된 기술은 4장에서 소개.

영상을 사용할 수도 있다. 어떤 연구들은 visual localization을 1장의 영상만 사용하는 좁은 의미로 사용하고, 이를 분명히 하기 위해 image-based localization 또는 visual re-localization과 같은 용어를 사용하기도 한다.

카메라의 위치를 표현하는 기준이 되는 공간(또는 지도)은 다양한 형태로 표현될 수 있다. 우선 주어진 공간의 여러 지점들에서 획득된 영상 세트, 즉 영상 데이터베이스 자체가 공간의 표현이 될 수 있다. 이때 카메라의 위치는 데이터베이스에서 가장 가까운 영상의 인덱스 또는 유사한 k 개 영상의 지리적 좌표(geotag)를 조합한 좌표로 표현될 수 있다. 주어진 공간은 다양한 형태의 3차원 모델로 표현할 수 있고, 이러한 3차원 모델의 좌표계를 기준으로 카메라의 위치와 카메라의 방향각이 표현된다. 가장 널리 사용되는 3차원 모델은 점군(point cloud)이다. 많은 이미지 매칭 및 visual localization 기법들이 점(point)을 단위로 동작하고, 점군은 이러한 기법들에 가장 자연스러운 공간 표현 방법이다. 점군의 각 점은 3차원 위치와 RGB 값 외에 visual localization이나 이미지 매칭에 사용되는 추가적인 특징(feature)을 포함할 수도 있다. 또 2차원 이미지를 비트맵과 벡터 형태로 표현하듯이, 3차원 공간도 복셀(voxel) 또는 벡터(vector) 형태로 표현할 수 있다. 이러한 표현 방법은 점군 표현법보다 좀 더 적은 용량(compact)으로 높은 밀도(dense)의 표현이 가능하다. 복셀 표현은 단순하고 규칙적인(regular) 표현이 가능하지만, 복셀의 크기에 따른 해상도의 제약이 있다. 폴리곤 메시(polygon mesh)나 선분(line segment)은 보다 간결하고 명확한 표현이 가능하지만, 3차원 모델의 수정이 상대적으로 어려운 단점이 있다. 점군, 복셀, 벡터를 이용한 표현 방법은 3차원 공간의 형태를 직접적이고 바로 알 수 있는 방법으로 나타내는 명시적(explicit)인 표현 방법이다. 그러나 TSDF (Truncated Signed Distancetrans Formation)나 NeRF (neural radiance field), 3D Gaussian splatting과 같이 비명시적(implicit) 표현 방법도 있다. 비명시적인 표현 방법은 공간의 형태를 바로 알 수 없다는 단점이 있지만, 보다 적은 용량으로 보다 밀도 높고 보다 정밀한 표현을 가능한 장점이 있다.

Visual localization을 그림 1과 같이 입력 영상의 개수(세로축)와 공간 표현의 방법(가로축)에 따라 분류할 수 있다. 그림 1의 가로축의 공간 표현은 왼쪽으로 갈수록 영상에 보다 가까운 방법이 배치되도록 정렬하였다. 최근 소개된 연구 동향 조사[1]도 공간 표현 관점에서 기술을 분류하였는데, 해당 연구 동향 조사는 그림 1의 1장의 영상을 활용 연구들을 monocular re-localization의 관점만을 다루었지만, 최신 연구들을 매우 상세하게 분석하였다.

2.2. 입력 영상 관점에서의 Visual Localization의 접근법

Monocular visual localization의 다양한 접근법은 현재 위치를 알아내는 데 사용된 입력(query) 영상의 개수에 따라 그림 1과 같이 두 가지 방법론으로 나눠볼 수 있다.

우선 1 장의 영상만을 입력 영상으로 사용하는 기법으로 visual place recognition과 camera pose estimation이 있다. Visual place recognition은 구축된 영상 데이터베이스에서 현재 주어진 입력 영상과 가장 가까운 영상을 찾는 기술로 카메라의 위치로 영상 데이터베이스에 찾은 영상의 인덱스 (또는 해당 장소의 ID)를 반환한다. Visual place recognition은 image retrieval의 한 형태로 볼 수 있으며 핵심 개념이나 기법을 공유하기도 한다. **NetVLAD** [2]는 CNN 네트워크를 통해 얻은 기본 특징벡터들을 장소 구분에 효과적이 되도록 그룹핑(aggregation)된 특징벡터를 이용한 visual place recognition 기법이다. Visual place recognition은 visual SLAM이나 structure-from-motion에서 주어진 입력 영상과 연결된 과거의 다른 영상을 찾는 visual loop closure의 역할을 수행할 수 있다. Camera pose estimation은 주어진 영상을 촬영한 카메라의 3차원 위치와 방향각, 즉 자세를 알아내는 기술이다. 3차원 카메라 자세는 주어진 3차원 모델 (또는 지도)의 절대 좌표계를 기준으로 표현되기도 하며, 영상 데이터베이스의 영상에 대한 상대 좌표계를 기준으로 표현되기 한다. 전자는 absolute pose estimation (또는 visual positioning system; VPS), 후자는 relative pose estimation으로 불린다. 대규모(large-scale)의 3차원

모델에서 주어진 영상에 담고 있는 정보는 보통 1% 이하의 극히 일부분이다. 따라서 전체 3차원 모델에 대해 주어진 영상의 특징을 매칭하여 카메라의 자세를 찾는 RANSAC 기반의 일반적인 접근은 99% 이상의 관련 없는 데이터(outlier)로 인해 불가능한 문제이다. 따라서 전체 3차원 모델에서 주어진 영상과 관련된 부분(candidate)을 찾아 탐색 공간을 줄이는 과정이 필요하다. **Pose Voting** [3]은 기하학적인 조건들을 이용해 특징점의 시점 정보 등을 가지고 있는 3차원 점군에서 후보지를 필터링하고 RANSAC이 아닌 투표(voting) 기법을 통해 이러한 문제를 해결하였다. **InLoc** [4]은 NetVLAD [2]를 이용해 주어진 실내 공간 내에 주어진 영상과 관련된 후보 영역을 찾아내고, NetVLAD의 VGG-16 backbone의 결과를 dense matching하여 카메라의 3차원 자세를 알아내었다. 실내 환경은 텍스쳐가 단순하거나 없는 부분이 많은데, InLoc에서는 dense matching을 통해 이를 극복하였고, dense matching의 결과에 PnP 알고리즘을 이용한 RANSAC을 더해 카메라의 3차원 자세를 추정하였다. **PoseNet** [5]은 영상이나 3차원 모델이 아닌 CNN을 통해 공간을 표현하는 방법을 사용하였다. PoseNet은 23 레이어의 GoogLeNet을 backbone으로 사용하고, 마지막 레이어에서 7차원(위치: 3차원, 방향각: 4차원)의 벡터를 예측(regression)하는 기법을 사용하였다. 처음으로 영상 데이터베이스나 3차원 모델이 아닌 CNN이라는 암시적 공간 표현에 바탕을 둔 (absolute) pose regression 기법이다.

현재의 영상뿐만 아니라 과거의 영상 또는 과거의 영상에서 얻은 결과를 추가로 사용하는 기법으로 visual odometry (이하 VO), visual SLAM, structure-from-motion (이하 SfM)가 있다. 이 세 가지 기법은 포함 관계를 갖는다 (VO $\subset$ visual SLAM $\subset$ SfM). VO는 첫 번째 영상의 좌표계를 기준으로 이후 연속된 영상의 카메라의 자세를 연속적으로 추정한다. Visual SLAM은 VO에 visual loop closure를 추가하여 과거에 지나갔던 위치를 인지하고 이를 3차원 지도에 보정한다. VO 동일한 위치를 반복 방문하였을 때 카메라 자세에 누적된 에러(drift error)를 갖는데 반해, visual SLAM 이러한 문제가 덜 하다. 그러나 visual SLAM은 전체 공간의 지도를 기억하고 일관성을 유지하여야 하지만, VO는 현재 영상 근처의 좁은 영역의 3차원 모델만 기억하면 된다. SfM은 입력된 영상들이 연속된다는 가정을 하지 않고, 각 영상의 자세와 3차원 모델을 추정한다. VO와 visual SLAM과 달리 SfM의 입력된 영상 사이에 순서가 없기 때문에 주어진 모든 영상 쌍(pair)을 매칭하며, 영상 사이의 연결 관계를 찾는 과정이 추가적으로 필요하다. 이러한 정합 과정은 visual place recognition이나 visual SLAM의 loop closure와 유사하다. **LIBVISO** [6]는 오프소스로 공개된 초기 VO로서 KITTI 데이터셋 [51]과 함께 이후 많은 기술들의 baseline으로 참고되었다. **ORB-SLAM** [7]은 대표적인 특징점 기반의 visual SLAM 기술로 특징점과 키프레임 기반의 그래프 최적화를 이용한 기법이다. ORB-SLAM은 단안 카메라 외에 v2에서 스테레오 카메라와 RGB-D 카메라를 지원하였고, v3에서는 어안(fisheye) 카메라와 IMU 융합을 지원한다. **COLMAP** [8]은 ORB-SLAM과 마찬가지로 특징점과 그래프 최적화를 이용한 SfM 기술로 높은 완성도와 뛰어난 성능으로 지금까지 여러 응용에 널리 사용되고 있다. COLMAP은 SfM 결과를 이용한 multi-view stereo 기법을 통한 고밀도 점군 생성과 Poission 표면 복원 기법을 통한 폴리곤 메쉬 생성도 가능하다. 대부분의 기술들은 특징점을 기반으로 동작하는데, RGB-D 카메라의 등장으로 특징점 추출을 하지 않고 영상의 보다 많은(dense) 영역을 그대로(direct) 이용하는 방법이 단안 카메라에도 많이 적용되었다. **DTAM** [9]은 이러한 접근의 시작이었고, 이후 많은 direct method가 연구되었다. **DSO** [10]는 이러한 접근에서 가장 널리 알려진 연구로 카메라의 카메라의 자세 뿐만 아니라 시변하는 광학적(photometric) 파라미터를 동시에 최적화하는 기법을 사용하였다.

연속된 다수의 영상을 활용하는 VO나 visual SLAM은 카메라가 장착된 자율차, 로봇, 드론 등의 위치추정에 많이 사용되고, SfM은 (특히 여러 카메라 또는 임의 위치에서 촬영한 데이터를 활용한) 3차원 물체 및 공간 복원에

많이 활용된다. 1장의 영상을 활용하는 visual place recognition과 camera pose estimation은 visual SLAM이나 SfM의 loop closure나 이미지 매칭에 활용되기도 하지만, 카메라에서 영상을 상시 계속 얻기 힘든 환경인 스마트폰의 위치기반 서비스에 많이 사용된다.

3. Image Matching Challenge의 Visual Localization 기술

Image Matching Challenge [11] (이하 IMC)는 이미지 매칭과 visual localization 문제의 다양한 시도와 기술들의 성능을 경쟁하는 대회이다. IMC는 본래 이미지 매칭에 초점을 둔 대회이지만, 이미지 매칭의 성능을 이미지가 촬영된 카메라의 위치와 방향각의 정확도로 평가하기 때문에 visual localization 또한 매우 중요한 대회이다. IMC는 2019년 워크샵을 시작으로 2020년부터 매년 기술의 성능을 평가하는 대회를 함께 개최하고 있고, 2022년부터 대회가 Kaggle을 통해 운영되고 있다. 특히 2024년에는 낮과 밤의 보다 극단적인 조명 변화나 계절에 따른 큰 시각적 변화, 투명하거나 반사도 높은 물체 등 도전적인 데이터셋이 많이 추가되었고, 많은 참여 기술들이 좋은 성능을 달성하지 못해 이 분야가 여전히 많은 발전이 필요함을 시사하였다.

3.1. Hierarchical Visual Localization 접근법

IMC에서 camera pose estimation이 필요하고, 전통적인 2D-2D 지역(local) 특징점 매칭 결과를 전체 3D 모델에서의 2D-3D 지역 특징점 매칭으로 확장하는 기법은 대규모(large-scale) 3차원 모델 환경에서 엄청난 계산량이 필요하고 정확성도 크게 떨어진다. 따라서 앞에 언급한 PoseVoting [3]이나 InLoc [4]과 같이 지역 특징점 매칭에 앞서 매칭 대상이 될 후보 영역을 선택하는 과정을 일반적으로 많이 사용한다.

HLoc [12]은 IMC에 참여하는 많은 팀들이 사용하는 HF-Net을 이용한 계층적(hierarchical) camera pose estimation 기법이다. HLoc은 NetVLAD [2] 형태의 전역(global) 특징벡터와 k-NN 알고리즘을 이용해 후보 영역을 선택하고, 공통으로 관찰된 시점(co-visibility)을 고려한 후보 영역 클러스터링을 통해 최종 후보 영역을 결정한다. 선택된 후보 영역에 대해 SuperPoint [17] 지역(local) 특징점 매칭을 수행하여 얻은 2D-3D 매칭 결과에 대해 RANSAC과 PnP 알고리즘을 이용해 최종 카메라 자세를 계산한다. HLoc에서는 HF-Net을 통해 공동된 MobileNet backbone을 바탕으로 하나의 CNN 네트워크에서

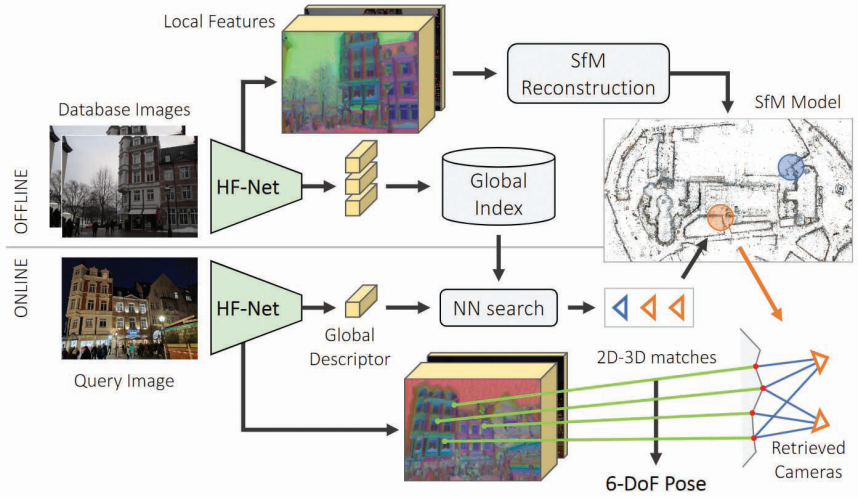

그림 2. HLoc: HF-Net을 이용한 계층적 camera pose estimation 과정[12].

NetVLAD 스타일의 전역 특징벡터와 SuperPoint 스타일의 지역 특징점과 특징벡터를 동시에 획득한다. 이미지 전체에 대한 특징을 이용해 전체 3차원 모델에서 탐색을 하고, 이후 이미지의 지역적 특징 매칭하여 카메라의 자세를 추정하는 이러한 과정을 coarse-to-fine localization으로 부르기도 한다. 이러한 계층적 접근은 InLoc [4]도 마찬가지이고, visual localization 뿐만 아니라 **RoMa** [13]와 같이 이미지 매칭에도 활용되기도 한다.

RoMa [13]는 스케일, 조명, 시점, 질감의 변화와 같은 극한 매칭 조건에서도 두 영상을 촘촘히 매칭하는 계층적 dense feature matching 기법이다. RoMa는 coarse global matching 단계와 fine refinement 단계로 구성된다. Coarse global matching 단계에서는 시점과 조명 변화에 강하고 일반화 성능이 높은 **DINOv2** [15]를 fine-tuning없이 coarse feature encoder로 사용한다. 두 영상의 DINOv2 결과는 Transformer를 사용해 전역적(global) 유사성을 바탕으로 매칭한다. Fine refinement 단계에서는 VGG-19를 fine feature encoder로 사용하고, warp refiner를 이용하여 전단계의 coarse matching 결과와 통합한다. Warp refiners는 coarse 및 fine feature map과, coarse matching 단계에서 생성된 warp 및 전역적 유사성을 확률적으로 표현한 신뢰도(certainty)를 입력으로 사용한다. RoMa는 coarse matching 단계와 fine refinement 단계에서 각각 다른 손실함수(loss function)를 사용한다. Coarse matching 단계에서는 매칭 분포가 멀티모달(multi-modal)일 가능성이 높기 때문에 Kullback-Leibler divergence를 최소화하는 regression by classification 방식을 사용한다. Fine refinement 단계에서는 어느 정도 매칭된 결과를 세밀하게 조정하는 단계이므로 유니모달(uni-modal) 분포로 모델링하고, Charbonnier loss를 이용한 robust regression 방식을 사용한다.

IMC에 참여한 많은 팀들은 앞서 언급한 2단계의 계층적 camera pose estimation 기법을 기본적으로 사용하고, 특히 특징벡터의 추출과 매칭 기법의 개선과 조합에 많은 고려를 하였다. 따라서 다음 절에서는 최근 제안된 전역 특징(global image descriptor 또는 image encoding)와 지역 특징(local feature)과 이들의 매칭 기술에 대해 살펴보고자 한다.

3.2. Global Image Descriptors 기술들

DINO [14]는 자기지도학습(self-supervised learning)을 이용한 시각기반모델(vision foundation model)로, ViT (Vision Transformer) 모델을 라벨링없이 데이터를 이용해 스스로 학습하여 이미지의 시각적 특징 표현을 학습한다. DINO의 자기지도학습 방법으로 학생/교사 네트워크를 이용한 지식증류(knowledge distillation) 기법을 이용한다. 기존 지식증류 기법은 교사 네트워크의 지식을 증류하여 필요한 부분만 (네트워크의 크기가 작은) 학생 네트워크에 학습 시켜 네트워크 크기를 줄이는 설계하는 방법이

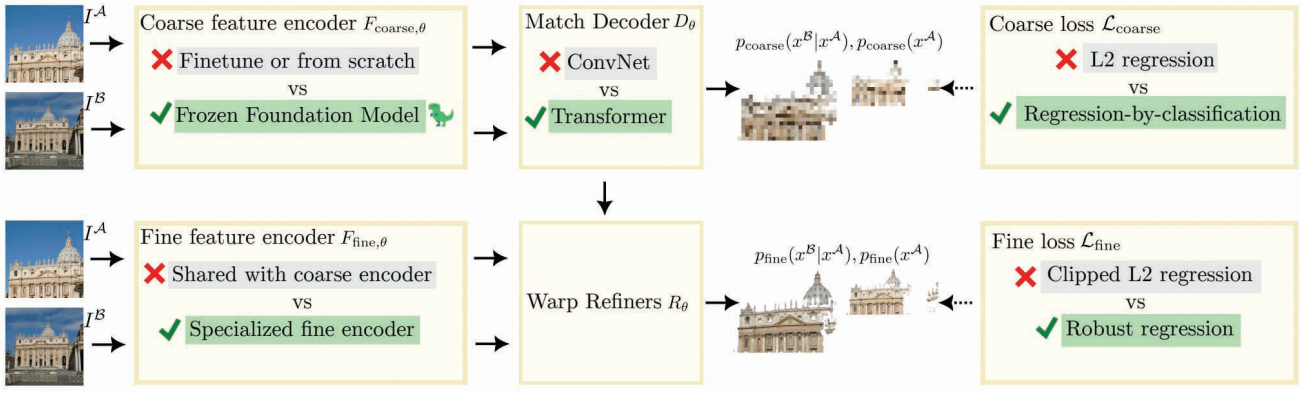

그림 3. RoMa의 두 단계 접근: Coarse global matching 단계(상)과 fine refinement 단계(하) [13].

다. DINO는 이와 달리 동일한 아키텍처를 가진 학생 네트워크와 교사 네트워크를 이용한다. 주어진 이미지에 데이터 증강(data augmentation)을 적용해 서로 다른 이미지를 생성하고, 이를 각각 학생 네트워크와 교사 네트워크에 입력하여 학습한다. 이때, 교사 네트워크는 이미지 전체의 전역적(global) 특징을 보게 하고, 학생 네트워크는 이미지의 일부만 보게 한다. 그리고 cross entropy 관점에서 네트워크의 출력이 일치하는 방향으로 학습한다. 전체적인 학습 과정에서 교사 네트워크는 학습을 하지 않고,

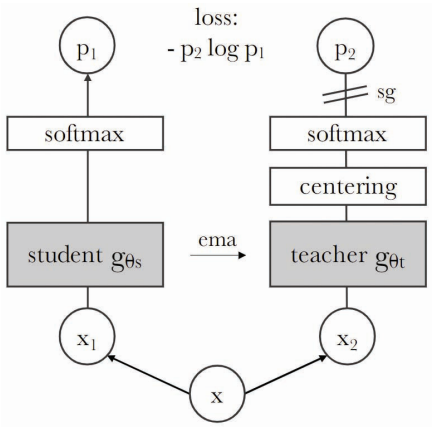

그림 4. DINO의 knowledge distillation 기법: Teacher 네트워크는 학습(sg)하지 않고, student 네트워크의 값을 EMA를 통해 업데이트 받음.

학습된 학생 네트워크의 파라미터를 지수이동평균(exponential moving average; EMA)으로 교사 네트워크에 업데이트한다. DINO는 이러한 학습 과정에 의해 여러 다운스트림 작업에서 뛰어난 성능을 보인다. 특히, 이미지 분류, 객체 검출, 영역 분할(segmentation) 작업에서 지도 학습 방식으로 훈련된 모델과 비교할 만한 결과를 얻을 수 있으며, 명시적인 클러스터링 목표 없이도 의미있는 시각적 클러스터를 형성할 수 있음을 보여준다.

DINOv2 [15]는 DINO의 개선된 버전으로, 142M의 대규모 이미지 데이터셋을 활용하여 범용적인 시각적 특징 표현을 학습할 수 있게 하였다. 이러한 대규모 이미지 데이터셋의 학습으로 일반화 능력이 크게 향상되었으며, 다양한 훈련 전략과 여러가지 최적화 기법을 적용하여 학습 속도 및 추론 속도를 향상시켰다. 특히, 데이터 레이블 없이도 이미지의 전역적 특징을 학습하여 이미지 검색에서 많이 활용되었다.

3.3. Local Feature Points, Descriptors, 그리고 Matching 기술들

지역 특징점에 대한 연구는 컴퓨터비전에서 가장 활발히 연구되었던 분야이다. 근래와 같이 기반모델(foundation model)에 대한 개념이 없던 시절에도 지역 특징점은

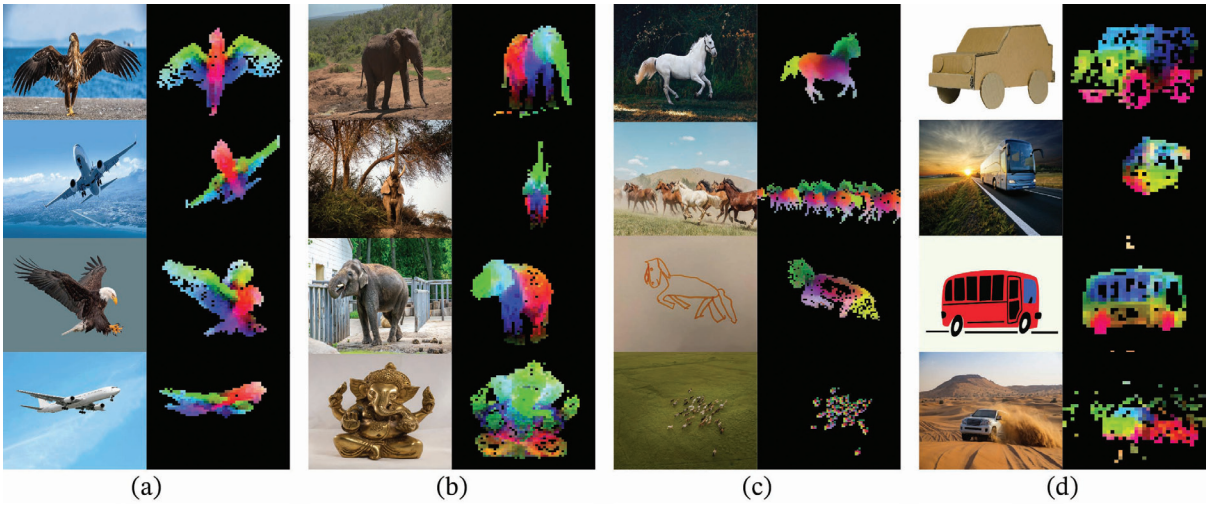

그림 5. 각 컬럼의 이미지의 DINOv2 descriptor의 PCA 결과의 첫 3개 벡터의 표시: DINOv2 descriptor가 영상에 보이는 주요 물체에 보다 집중하고, 유사 물체의 유사 부분에서 유사한 벡터를 가짐을 확인 가능.

지금의 기반모델과 같이 여러 컴퓨터비전 문제에 공통적으로 활용되는 핵심 기술로 여겨졌다. Harris corner나 SIFT와 같은 그레디언트(gradient) 기반의 지역 특징점과 FAST, ORB와 같이 바이너리(binary) 기반의 지역 특징점도 연구되었다. 최근에는 딥러닝을 이용한 지역 특징점과 descriptor, 그리고 매칭 기법이 많이 연구되고 있다. 최근의 지역 특징은 Image Matching WebUI [16]를 통해 간단히 실험해볼 수 있다.

SuperPoint [17]는 CNN 기반의 네트워크로 특징점의 위치와 descriptor를 획득하는 기법으로 자기지도학습(Self-supervised learning)을 이용하는 특징이 있다. SuperPoint는 참값을 알고 있는 합성 데이터를 학습하여 사전에 획득한 MagicPoint를 기초특징검출기(base detector)로 사용한다. SuperPoint의 자기지도학습은 크게 두 가지 과정으로 구성되어 있다. 우선 특징점의 위치 학습을 위한 참값은 주어진 학습 이미지에 여러가지 변형을 주고 생성한 여러 이미지에 MagicPoint를 적용한 결과를 합한 heapmap으로 표현된다. 또한 descriptor의 학습을 위한 올바른 매칭은 이미 알고 있는 이미지의 변형을 통해 얻을 수 있다. SuperPoint는 HPatches 데이터셋에서 SIFT, ORB, LIFT와 비교하여 평균 5% 더 높은 반복성과 10-15% 더 높은 매칭 정확도를 갖는다. 또 호모그래피 추정 실험에서는 약 50% 더 적은 평균 거리 오류를 나타내었다. SuperPoint의 학습 이미지는 주어진 이미지에서의 호모그래피 변환으로 표현되기 때문에 먼 거리(wide baseline) 영상 매칭에 잘 동작하지 않고, 이후 SuperGlue [22]와 같은 추가 기술이 개발되었다.

ALIKED [18]는 여러가지 이미지 변형을 사용하여 이

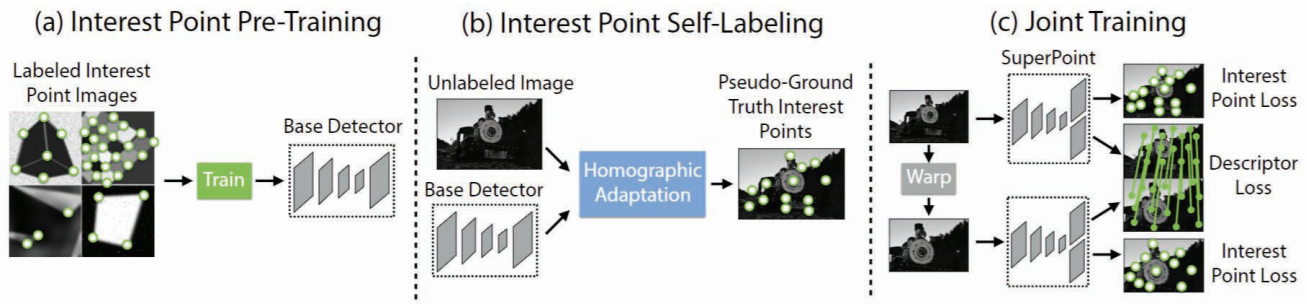

그림 6. SuperPoint의 자기지도학습(self-supervised learning)의 핵심 3가지 과정[17].

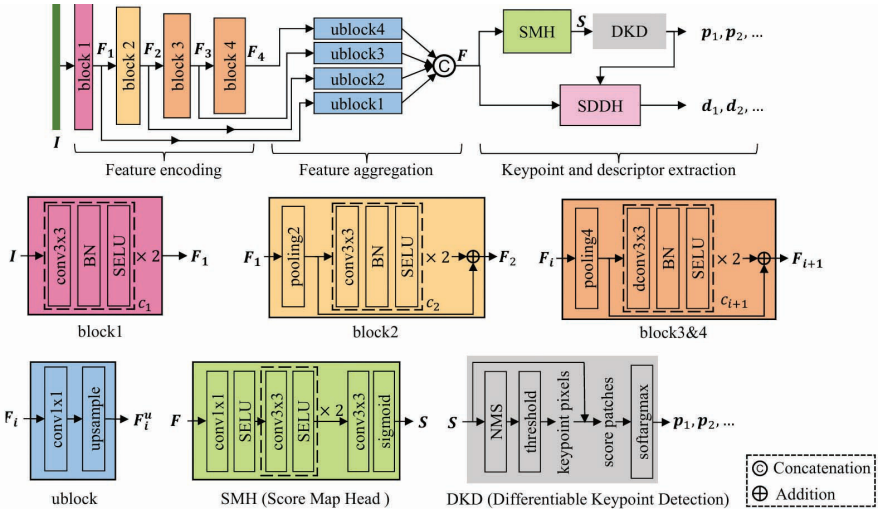

그림 7. ALIKED의 신경망 구조[18].

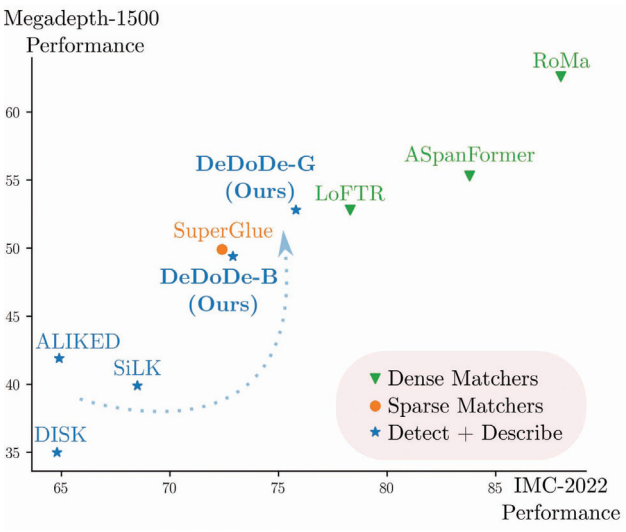

그림 8. 기존 detect-and-describe 접근 방식과 sparse/dense 이미지 매칭의 성능 예[19].

미지 특징점(keypoint)과 descriptor를 효율적이고 견고하게 추출하기 위해 설계된 모델이다. 이전 방법들은 고정 크기의 일반적인 합성곱을 사용하여 이미지를 인코딩할 때 기하학적 불변성(invariance)이 부족하고, 변형 가능한 descriptor를 추출할 때 처리 속도가 느리다는 한계가 있었다. 이를 해결하기 위해 ALIKED는 중복 계산을 크게 줄이고 다양한 기하학적 변환을 모델링할 수 있도록 하는 Sparse Deformable Descriptor Head (SDDH)를 도입했다. 이를 바탕으로, ALIKED는 시각적 측정 작업을 위해 SDDH를 활용한다. 이 네트워크는 Neural Reprojection Error (NRE) 손실을 밀집(dense)에서 희소(sparse) 컨텍스트로 조정하게 하기 때문에 희소 descriptor 훈련을 용이하게 하고 네트워크 훈련 중 중복 계산을 더욱 최소화한다. ALIKED는 특징 인코딩, 특징 그룹핑(aggregation), 그리고 특징점 및 descriptor 추출의 세 가지 주요 구성 요소로 이루어져 있다. 특징 인코더는 입력 이미지를 다중 스케일 특징으로 변환한다. 특징 집계 구성 요소는 다중 스케일 특징을 집계하여 localization과 representation 능력을 향상시킨다. 마지막으로, 특징점 및 descriptor 추출 구성 요소는 그룹화된(aggregated) 특징을 사용하여 특징점과 descriptor를 추정한다. ALIKED는 여러 장점을 가지고 있지만, 몇 가지 한계도 존재한다. 스케일과 관점에서 큰 차이가 있는 이미지 매칭 작업에서는 정확한 매칭을 얻기 어려울 수 있다. 또한, 계산 리소스를 절약하기 위해 ALIKED의 SDDH는 변형 가능한 위치 추정을 위해 단일 레이어만 사용하므로 복잡한 이미지 변형을 모델링하는 데 한계가 있다. 결과적으로, 이미지 스케일의 큰 차이가 있는 경우, ALIKED가 잘 작동하지 않을 수 있다.

DeDoDe [19]는 SfM으로부터 얻은 3D 트랙을 검출기(detector) 훈련 시 선행 정보로 사용하는 descriptor에 구애받지 않는 모듈형 기하학적 특징점(keypoint) 검출 방법이다. 전통적인 특징점 검출 방법은 일반적으로 두 단계 과정을 거친다. 먼저 검출기를 사용하여 유망한 특징점을 식별한 후, 별도의 descriptor로 이를 설명한다. 이 과정은 종종 검출기와 descriptor 간의 시너지가 부족하다. 이에 반해, 최근의 학습 기반 방법들은 두 구성 요소의 목표를 일치시키는 공동 학습 패러다임을 사용한다. 그러나 이러한 방법에서 검출기와 descriptor 간의 상호 의존성은 때때로 성능을 저하시킬 수 있다. DeDoDe는 descriptor와 검출기의 학습 과정을 과정을 분리하면서도 목표를 일치시키는 새로운 접근 방식을 채택하고 있다. 검출기는 대규모 SfM 데이터셋의 트랙을 사용하여 3D 일관성으로부터 직접 특징점을 식별하도록 훈련된다. 이후 descriptor는 검출된 특징점을 대상으로 상호 최근접 이웃 목표를 최대화하는 방식으로 별도로 훈련된다. 그러나 DeDoDe는 몇 가지 한계를 가지고 있다. 기본 검출기에 의존하기 때문에 안정적인 특징점을 놓칠 수 있으며, 증강된 데이터로 훈련되지 않아 성능과 일반화 능력이 저하될 수 있다. 또한, DeDoDe는 많은 잠재적 특징점을 식별하지만, 이들의 방향과 스케일을 추정하지 못하는 단점이 있다. 이는 별도의 네트워크를 통합하여 지역 프레임을 추정함으로써 개선될 수 있다. 이러한 문제를 해결하기 위해, **DeDoDe v2** [20]가 제안되었다. DeDoDe v2는 훈련 파이프라인에 non-max suppression와 향상된 데이터 증강을 도입하여 보다 많은 특징점이 추출되도록 하였고, 이전보다 향상된 성능과 짧은 훈련 시간을 달성하였다.

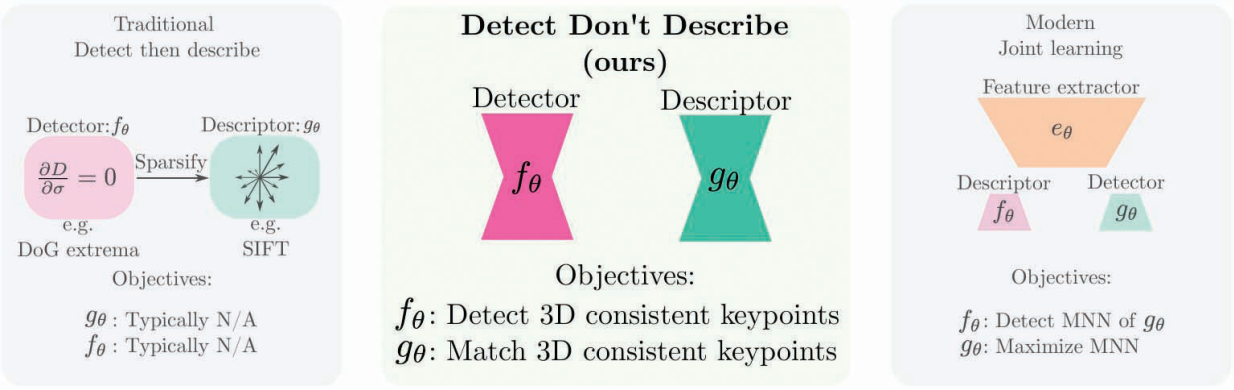

그림 9. 기존 detect-and-describe 접근 방식(좌)과 DeDoDe(중), 그리고 공통 학습 기반 접근 방식(우)[19].

XFeat [21]은 고해상도 이미지에서도 빠르고 효율적인 지역 특징 추출이 가능한 모델이다. 기존 신경망 기반 특징 추출 방법들[17, 18]은 높은 계산량이 요구된다. XFeat은 모델의 아키텍처를 경량화하면서도 성능을 유지하는 전략을 이용하여 하드웨어에 특화된 최적화 없이도 GPU가 없는 환경에서도 실시간 특징 추출을 가능하도록 만들어졌다. XFeat은 합성곱 층에서 채널 수를 최소화하고, 공간 해상도가 감소함에 따라 채널 수를 급격히 증가시키는 전략을 사용하여 초기 층의 계산 비용을 크게 줄이면서도 네트워크의 표현 능력을 유지한다. XFeat의 backbone은 6개의 주요 블록으로 구성되며, 각 블록은 2D 합성곱, ReLU 활성화 함수, 배치 정규화로 이루어진 기본 층들의 조합이다. 이 구조는 {4, 8, 24, 64, 64, 128}의 채널 구성을 따르며, 마지막에는 다중 해상도 특징을 통합하는 융합 블록이 추가된다. 이러한 구조는 H/32 x W/32의 최종 공간 해상도를 달성하며, 고해상도 입력에 대해 효율적인 처리를 가능하게 한다. XFeat은 기존의 깊이별 분리 가능한 합성공이나 균일한 채널 감소 방식과는 달리, 초기 층의 계산 부하를 집중적으로 줄이면서도 네트워크의 전반적인 표현 능력을 유지하는 새로운 접근법을 제시한다. 이 모델은 특히 카메라 위치 추정, 시각적 위치 추정, SfM 등 픽셀 수준의 정확한 매칭이 필요한 작업에서 뛰어난 성능을 보인다.

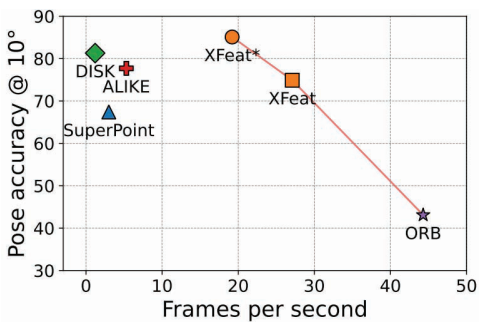

그림 10. XFeat의 효율적인 성능과 속도[21].

SuperGlue [22]는 두 개의 지역(local) 특징 세트를 매칭하면서 동시에 일치하지 않는 점들을 필터링할 수 있는 매칭 기법이다. 이 네트워크는 그래프 신경망과 어텐션에 기반한 유연한 컨텍스트 집계 메커니즘을 도입하여 3D 장면의 기저와 특징 할당을 동시에 추론한다. 이로 인해 부분적인 점 가시성(visibility)과 가림(occlusion)를 어느 정도 극복하며 특징 매칭이 가능하다. SuperGlue는 이미지 쌍에 대해 컨텍스트 그룹핑(aggregation), 매칭, 필터링을 동시에 수행하는 학습 가능한 미들엔드 역할을 한다. 전통적인, 수작업으로 설계된 휴리스틱 방법과 달리, SuperGlue는 기하학적 변환 및 3D 세계의 규칙성을 학습하여 더 정교한 솔루션을 제공한다. SuperGlue의 구조는 어텐션 그래프 신경망과 최적 매칭 레이어와 같이 두 가지 요소로 이루어져 있다. 어텐션 그래프 신경망은 키포인트 인코더를 사용하여 특징점 위치와 시각적 descriptor를 단일 벡터로 매핑한다. 이후 교차(cross) 어텐션 레이어를 번갈아 사용하여 더 강력한 표현을 생성한다. 최적 매칭 레이어는 점수 매트릭스를 생성하고, Sinkhorn 알고리

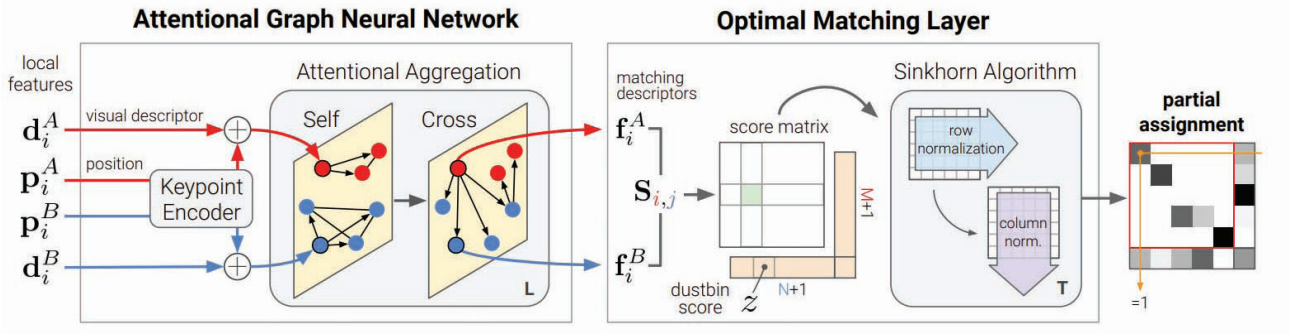

그림 11. SuperGlue의 신경망 구조[22].

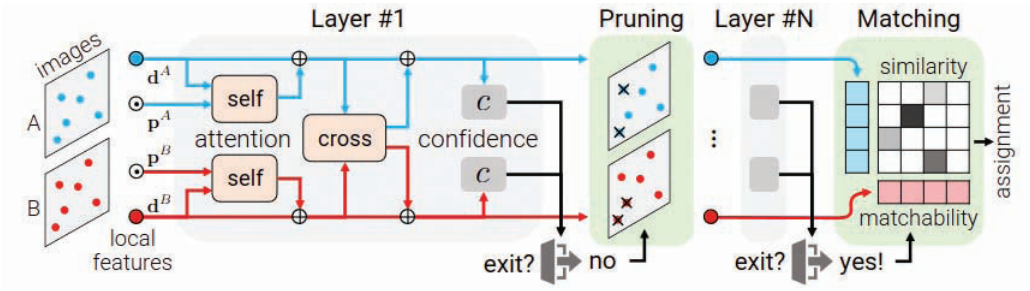

그림 12. LightGlue의 신경망 구조[23].

즘을 사용하여 최적의 부분 할당을 찾는다. 이러한 조합을 통해 SuperGlue는 이미지 쌍 간의 특징을 효율적이고 정확하게 매칭할 수 있어 다양한 다른 응용에 널리 사용되고 있다.

LightGlue [23]는 SuperGlue의 설계를 보완하여 효율성과 정확성을 동시에 향상시킨 매칭 기법이다. LightGlue는 적응적 매칭(adaptive matching)을 사용하여 매칭 난이도에 따라 사용하는 레이어 수를 조절한다. 예를 들어 시각적 중복이 많거나 형태 변화가 적은 경우 매칭 난이도가 낮다고 판단해 추론 속도를 높인다. LightGlue는 self-attention 메커니즘을 사용해 이미지 내 및 이미지 간의 관계를 학습하여 매칭 정확성을 높였다. 또한 초기 단계에서 매칭 가능성이 낮은 특징 점들을 필터링하여, 중요한 연산에 집중 할 수 있도록 해준다. LightGlue는 IMC 2021에서 주요 알고리즘으로 주목 받았는데, 다양한 PhotoTourism 데이터셋에서 SuperPoint [17], DISK, LoFTR [24] 등과 비교해 높은 성능을 보였다.

4. 공간 표현 관점에서의 Visual Localization 연구 동향 조사

Visual localization의 다양한 기법들은 3차원 공간, 즉 지도를 표현하는 방법에 따라 그림 1과 같이 나눠볼 수 있다. 다양한 공간 표현 방법은 각각 장단점이 있고, 각 공간 표현 방법에 따른 visual localization 기법들은 공간 표현의 장점을 바탕으로 단점을 극복하며 높은 정확도를 갖도록 하는 것을 목표로 한다.

4.1. Image Database 공간 표현을 이용한 기술들

Image database를 공간 표현으로 사용하는 visual localization 기법은 일반적으로 image matching 기술로 치환된다. 그러나 주어진 입력(query) 영상과 image database 영상의 시점이나 특성이 크게 다른 경우가 있다. 예를 들어 1차원 시점의 입력 영상이 주어지고, image database의 영상으로 3차원 시점의 위성사진이 사용될 수 있다. 이러한 위

성사진은 항공기나 인공위성을 통해 사전에 구축되어 있는 경우가 많기 때문에 SLAM이나 SfM 등을 통해 공간 모델을 만들 필요가 없는 장점이 있다.

Shi et al.의 연구 [26]는 위성사진을 projective 변환과 polar 변환을 통해 지상에서의 1차원 시점의 영상과 유사하게 변환시켜 visual localization에 활용하는 기법이다. 이 방법은 HLoc [12]과 마찬가지로 2단계로 구성되어 있다. 첫 번째 단계인 coarse geo-localization은 1차원 시점의 영상과 변환된 위성사진을 CNN을 이용해 특징벡터를 추출하고 이를 매칭하여 가장 매칭이 잘 된 위치를 획득하고, 이후 영상을 이동(shift)하며 매칭하여 카메라의 방향각을 추정한다. 두 번째 단계인 fine-grained geo-localization에서는 몇 가지 위치 후보에 대한 이미지 매칭을 통해 지상에서의 위치를 보다 정확하게 보정하는 역할을 한다.

Shi et al.의 후속 연구 [27]는 위성사진을 변환하는 선행 연구에서 fine-grained geo-localization 부분을 LM 기반의 최적화 과정으로 바꾸었다. 또한 기존의 크기가 작아지는 CNN에서 출력된 특징벡터의 크기가 입력 이미지와 유사한 U-Net을 사용하였는데, 기존 projective 기하학적 변환이 위성사진에 대해 이뤄지지 않고 특징벡터에 대해

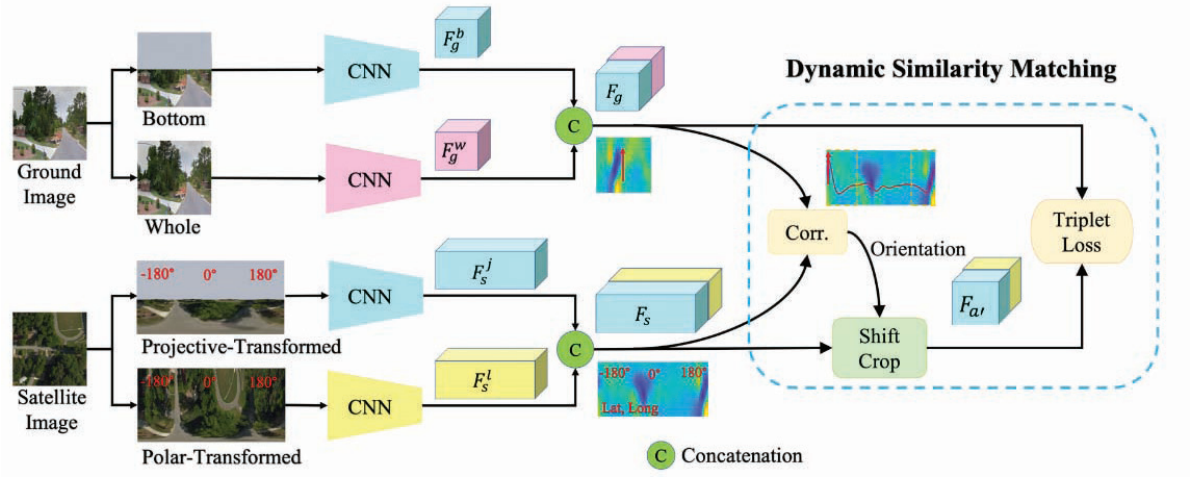

그림 13. Shi et al. 연구 [26]의 coarse geo-localization 과정.

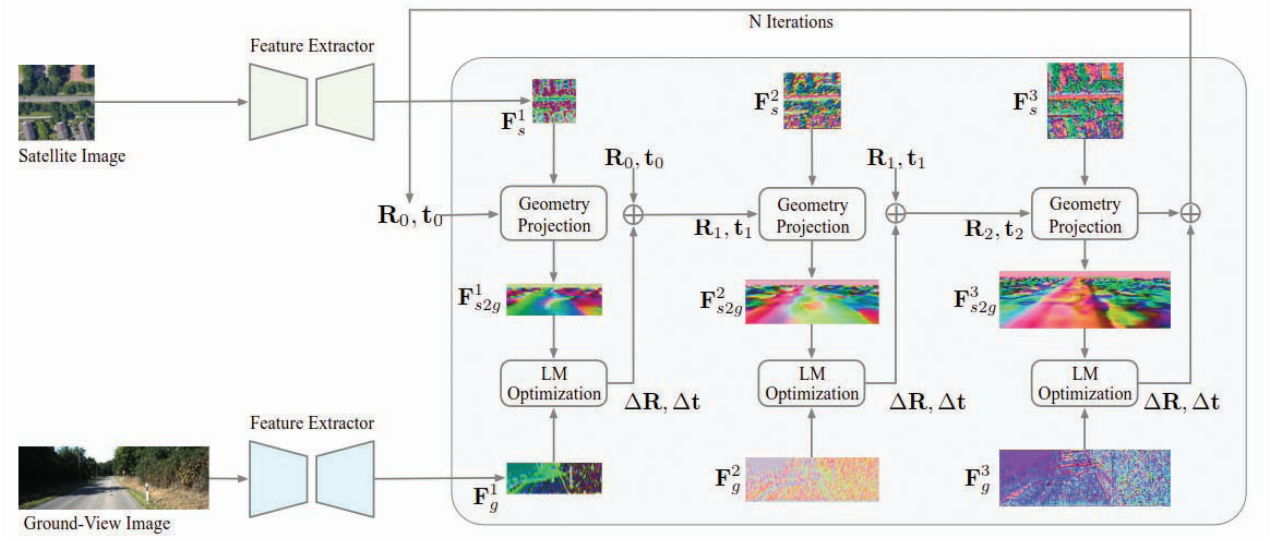

그림 14. Shi et al. 후속 연구 [27]의 전체 흐름도.

이뤄진다. 또한 기하학적 변환이 몇 개의 후보로 고정되지 않고, LM 기반의 반복적인 최적화 과정을 통해 개선되면 이뤄진다. KITTI 데이터셋과 위성사진을 이용하여 약 5m 이내의 위치 오차를 가졌다.

4.2. Point Cloud 공간 표현을 이용한 기술들

점군(point cloud)을 이용한 공간 표현에 각 점이 이미지에서 추출한 특징벡터를 포함하고 있는 경우, 특징벡터를 이용한 HLoc [12] 형태의 계층적 visual localization이 가능하다. 그러나 점군 데이터에 특징벡터를 포함하고 있지 않은 경우 주어진 입력 이미지와 공간이 표현된 점군 사이의 갭(gap)을 고려한 visual localization이 필요하다.

DeepI2P [28]는 이러한 문제를 해결하기 위해 이미지와 점군 사이의 서로 다른 특성(modality)를 고려한 visual localization 기법이다. DeepI2P는 크게 두 가지 단계로 구성되어 있다. 첫 번째 단계인 frustrum classification 단계는 점군에서 (주어진 영상을 고려해) 주어진 영상의 시야각 내에 존재하는 점들을 구분하는 과정으로 PointNet과 attention fusion을 이용한 신경망으로 구성되어 있다. 두 번째 단계인 pose optimization은 카메라의 시야각으로 존재하는 점들을 카메라에 투영하는 형태로 카메라의 자세를 획득하는 단계이다. DeepI2P의 실험 결과에서 인상적인 부분은 MonoDepth2를 통해 깊이(depth) 데이터를 획득하고 이를 이용해 ICP 기반의 점군-점군 매칭을 진행할 때, 일반적인(naive) 접근도 비교적 잘 동작하는 점이다.

4.3. Vectorized Map 공간 표현을 이용한 기술들

벡터 형태의 공간 표현은 공간을 밀도있게 표현하면서 공간 표현에 필요한 메모리를 획기적으로 줄일 수 있다. 그래픽이나 게임에서 많이 사용하는 폴리곤 메시(polygon mesh), CAD 도면나 평면도(floor map), 혹은 도로지도(road map) 등 벡터 형태로 공간을 표현하는 방법은 다양하며, 각각 고유한 여러가지 특징을 가지고 있다. 따라서 벡터 형태의 공간 표현이 갖는 장점을 극대화 할 수 있는 여러 visual localization 기법들이 연구되어 왔다. 또한 일부 기법들은 벡터로 주어진 공간 표현을 샘플링(rasterization)을 통해 비트맵 형태로 변환하여 활용하기도 한다.

MeshLoc [29]은 폴리곤 메시로 표현된 3차원 모델을 이용한 영상 측위 기법이다. MeshLoc에서 사용하는 폴리곤 메시는 SfM을 통해 생성한 기존 3차원 점군과 마찬가지로 매우 세밀하고 많은 정보를 포함하고 있다. 따라서 MeshLoc의 전체 과정은 HLoc [12]과 전체적으로 매우 유사하다. 1단계는 전역 특징벡터와 k-NN을 이용한 후보 영역 탐색이고, 2단계는 LoFTR [24]과 SuperGlue [22]를 이용한 지역 특징 매칭이다. 이후 RANSAC과 PnP 알고리즘을 이용한 카메라 자세를 계산한다. MeshLoc을 통해 간접적으로 폴리곤 매쉬 표현의 효율성을 살펴볼 수 있다. 그림 16과 같은 예에서 기존 SfM을 통해 얻은 점군과 descriptor는 약 7.36 GB인데 반해, 폴리곤 메시를 이용해 모델의 디테일의 큰 손실 없이 약 600 MB 정도까지 모델 저장에 필요한 메모리를 줄일 수 있다. **CADLoc** [30]은

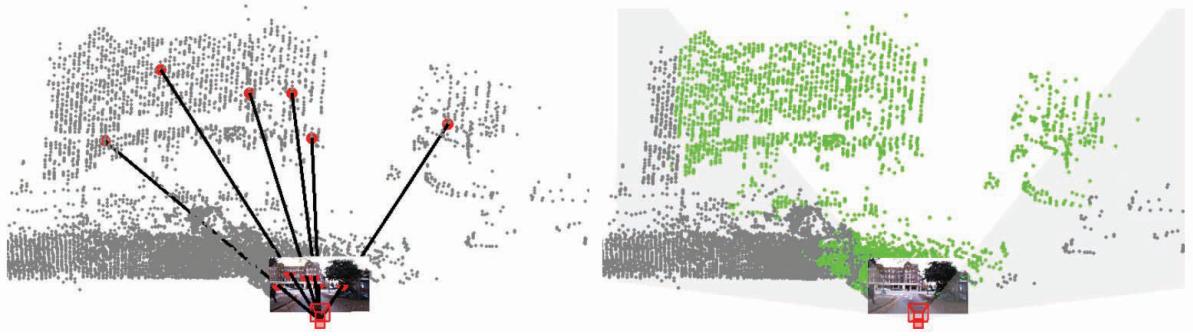

그림 15. 특징벡터 매칭을 이용한 2D3D-MatchNet(좌)와 특징벡터없는 점군을 이용한 DeepI2P(우)의 비교[28].

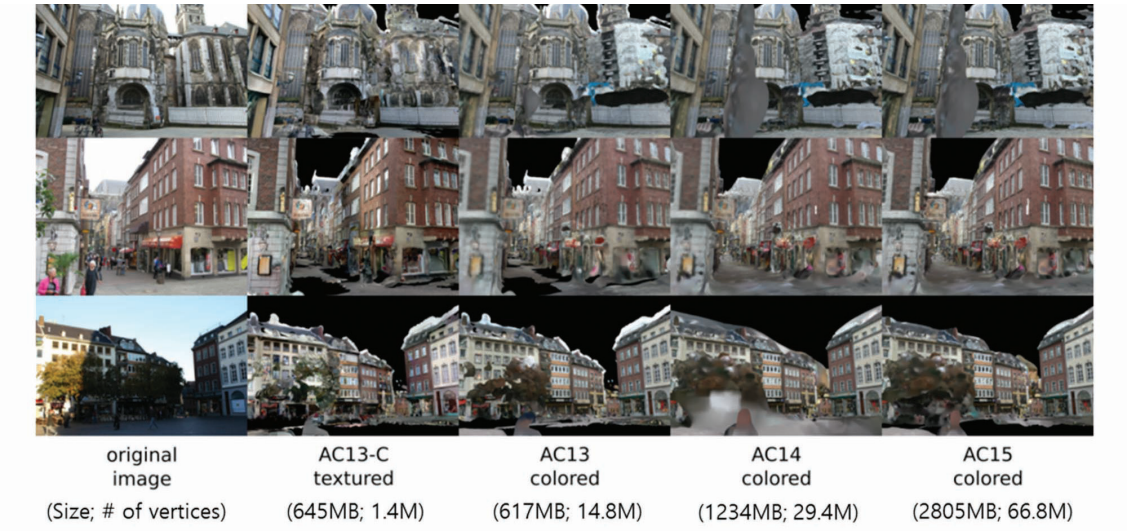

그림 16. MeshLoc에서 사용되는 다양한 디테일의 폴리곤 메시 모델 [29].

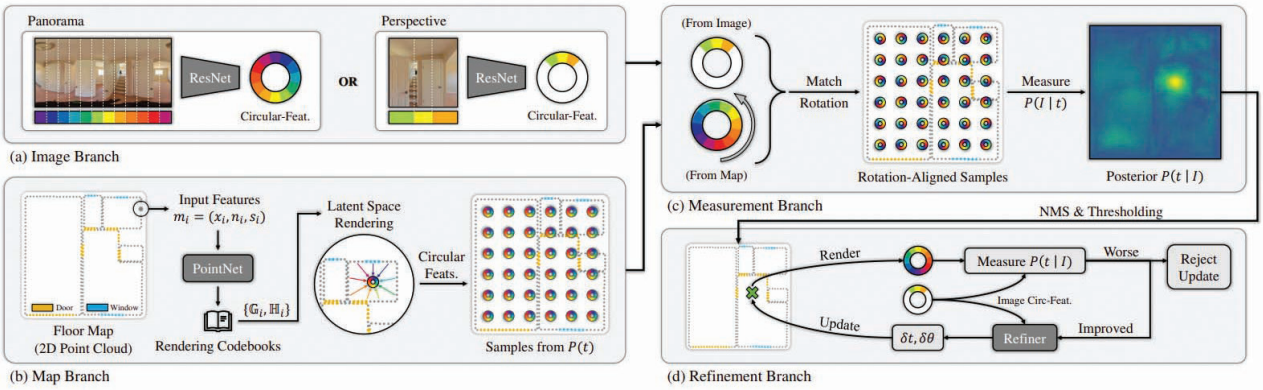

그림 17. LASER의 전체 흐름도 [31].

MeshLoc의 후속 연구로 인터넷 지도 서비스에서 얻을 수 있는 부정확하고 대략적인 3차원 모델을 포함한 다양한 디테일의 3차원 폴리곤 메시에 대한 성능평가가 가능한 벤치마크이다.

LASER [31]는 2차원 평면도(floor map)와 Monte Carlo localization (MCL) 프레임워크를 이용한 영상기반 2차원 측위 기법이다. LASER의 전체적인 동작은 그림 17와 같다. (a) 주어진 입력 영상은 ResNet-50 신경망을 통해 인코딩한다. 이때 생성된 2차원 특징벡터는 행 방향으로 average pooling을 수행하여 1차원 특징벡터로 만든다. (b) 주어진 2차원 평면도는 2차원 점군 형태로 변환하여 PointNet 신경망을 통해 공간 내 각 위치를 인코딩한다. 이때 평면도에 주어진 창문이나 문과 같은 시멘틱한 특성을 인코딩에 반영하였다. MCL 프레임워크는 deep metric learning 형태로 구현하였고, 특히 입자필터(particle filter) 방식의 MCL의 초기 입자 생성을 위해 주어진 공간의 각 위치를 샘플링할 때 방향각을 고정하여 적은 수의 샘플만 사용하였다. (c) 실제 이미지 인코딩 결과와 각 위치 샘플(입자)을 정합할 때 원형(circular) 매칭 형태로 사용하였여 고정된 방향각의 문제를 해결하였다. (d) 마지막 보정 단계(refinement branch)에서는 보다 정확한 카메라의 2차원 자세를 추정을 위해 실시간으로 임의 위치에 대한 인코딩이 필요한데, 이는 코드북 방식을 이용하여 약 10 kHz 정도로 빠르게 동작하도록 구현하였다.

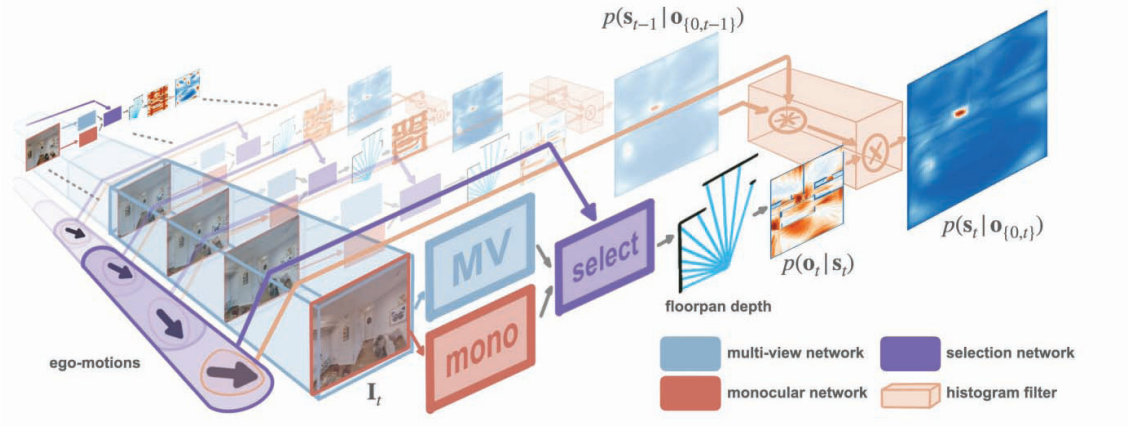

그림 18. F3Loc의 전체 흐름도 [32].

F3Loc [32]도 2차원 평면도(floorplan)와 RGB 이미지 시퀀스를 이용한 영상기반 2차원 측위 기법이다. LASER [31]는 벽과 코너 및 창문이나 문과 같은 의미적(semantic) 정보를 인코딩하여 비교하는 기법인데 반해, F3Loc은 영상에서 얻어지는 깊이(depth) 정보를 2차원 LiDAR 데이터와 유사하게 획득하고 이를 적극적 활용하였다. 깊이 영상의 획득은 단일 영상을 이용하는 방법과 다중 영상을 이용한 방법을 동시에 사용하며, 둘 사이의 가중치는 신경망을 통해 학습된 값을 사용하였다. F3Loc도 LASER와 마찬가지로 2차원 측위 기법으로 평면 위의 위치와 방향각, 3자유도 자세를 제공한다. 따라서 영상이 앞으로 기울어지거나(pitch) 전방 축을 기준으로 돌아있는(roll) 경우, 이를 보정해주는 부분이 적용되어 있다. 깊이 정보를 활용하는 F3Loc은 기존 2차원 LiDAR와 MCL을 활용하는 실내 측위 기법과 유사한데, 입자필터(particle filter) 대신 히스토그램필터(histogram filter)를 사용하는 것이 특징이다. 히스토그램필터를 사용하는 경우 카메라 자세는 3차원 배열로 표현되고, 그룹 컨볼루션과 같은 연산을 통해 효율적으로 처리할 수 있다.

OrienterNet [33]은 top-view의 도로지도를 이용하여 1차원 도로 시점 영상의 자세를 추정한다. OrienterNet의 접근은 전술한 위성사진을 이용한 기법과 매우 유사한데, 도로지도는 공간 표현에 필요한 메모리를 극단적으로 더 줄일 수 있다고 한다. (참고: 1 km^2 크기의 공간에 대해

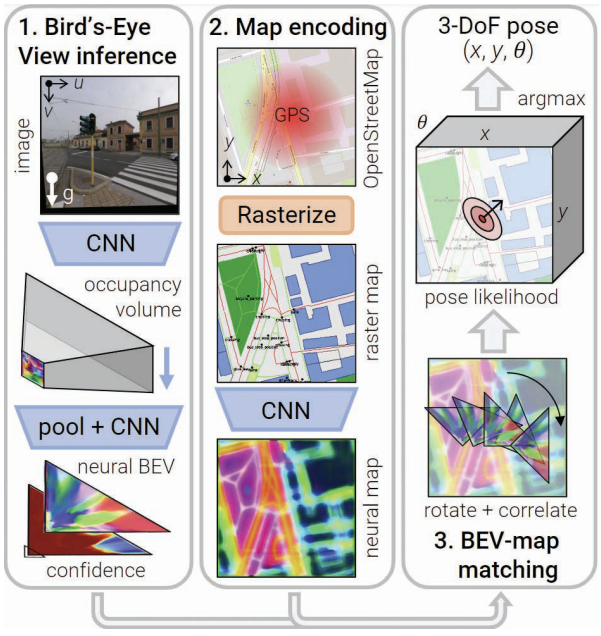

그림 19. OrienterNet의 전체 흐름도 [33].

SfM의 결과는 약 42 GB, 위성사진은 약 75 MB (550배 감소), 도로지도는 4.8 MB (8800배 감소)) OrienterNet은 주어진 1차원 시점의 영상을 단안 깊이 추정(monocular depth estimation)과 시맨틱 추론(semantic inference) 결과를 이용해 top-view의 neural image로 인코딩한다. Top-view의 도로영상 또한 CNN 신경망을 이용해 유사한 형태의 neural image로 인코딩한다. 마지막으로 top-view로 표현된 두 영상을 템플릿 매칭(template mathcing)을 통해 정합하여 카메라의 위치와 방향을 추정한다. 또한 1장의 영상

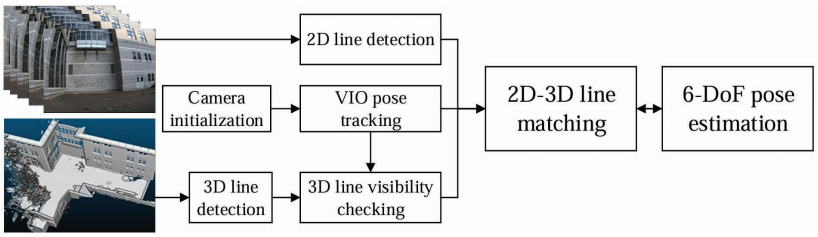

그림 20. 2D–3D line matching을 이용한 카메라 자세 추정 과정 [34].

을 이용한 visual localization 외에 Markov localization 프레임워크 관점에서 N장의 연속된 영상의 결과를 순차적으로 이용하여 훨씬 높은 성능을 얻었다.

2D–3D Line Matching [34]은 LiDAR 등으로 획득된 3차원 점군에서 3D 선분을 추출하고 이를 visual localization에 활용한 연구이다. 기본적으로 주어진 공간 표현은 점군의 형태이지만, 이를 3D 선분과 같이 보다 간소한(compact) 벡터 형식으로 변환하여 매칭에 사용하여 보다 가볍고 빠르게 동작 가능하다. 기본적으로 VO/VIO 기법인 VINS–Mono [35]를 이용해 카메라의 상대적인 자세 변화를 추정하고, 조금씩 발생하는 누적(drift) 에러와 급격한 자세 변화에 따른 추정 불능(lost) 문제를 2차원–3차원 선분 매칭을 통해 극복한다. 2차원–3차원 선분 매칭은 카메라의 시야각(field–of–view; FoV)를 고려한 가시성(visibility) 체크 후, 선택된 선분들에 대해 LM 기반의 자세 최적화를 통해 이뤄진다. 특히 선분이 적게 보이는 경우와 추정값이 튀는 문제는 sliding window 형태 최적화를 통해 극복하였다.

PtLine [36]은 3차원 선분 지도와 입력 영상에서 얻은 2차원 선분과 점 특징(feature)을 이용한 visual localization 기법이다. 입력 영상에서 선분 추출을 위해 VLSE (Visual Line Segment Extractor)라는 CNN 기반 모델을 사용한다. 이 모델은 선분의 양 끝점 위치를 직접 예측하는 방식을 사용하는데, 선분 표현에 있어 기존의 길이와 각도 대신 수평 및 수직 거리를 사용하여 긴 선분 표현이 갖는 오차를 줄인다. 선분 매칭 및 필터링은 기하학적 매칭 방법을 사용한다. 입력 영상에서의 2차원 선분과 image database 상의 2차원 선분 사이의 에피폴라 제약조건을 이용해 2D–2D 후보 선분 매치를 획득하고, 이를 3D 공간에 매핑한 후 재투영 오차를 최소화하는 방식으로 2D–3D 선분 매치를 선택한다. 이 방법은 시점 변화, 스케일 변화, 텍스처 부족, 조명 변화 등의 상황에서도 강건하게 동작한다. 이후, 자세 조정 단계에서 점과 선분 특징을 결합한(joint) 최적화 방법을 사용한다. 선분 특징의 경우 중심 거리 오차와 각도 오차를 고려하며, 점 특징의 재투영 오차도 같이 최적화한다. 이를 통해 다양한 장면에서 더 강건한 자세 추정을 가능하게 한다.

LSLM VLoc [37]은 선분으로 표현된 지도를 이용한 위치 추정 방법이다. 이 방법은 텍스처가 부족하거나 조명 변화가 심한 환경에서도 성능이 저하되는 포인트 기반 방법의 문제를 해결하고자 한다. 라인 특징은 이러한 환경에서도 더 안정적으로 검출되고 매칭될 수 있어, 위치 추정의 견고성을 향상시킬 수 있다. 또한, 3D 포인트 클라우드는 대규모 환경을 표현할 때 수십만에서 수백만 개의 포인트로 구성되어 메모리 요구사항이 높아지는 반면, 라인 기반 표현은 더 효율적으로 환경을 표현할 수 있다. 이를 위해 LSLM VLoc은 고품질의 라인 맵 구축을 위해 딥러닝 기반 라인 추출 및 매칭 방법을 활용한다. 이 접근 방식은 기존 방법에서 발생하는 부분적 가려짐으로 인한 불연속적인 라인 세그먼트 문제를 해결하여 라인 맵의 구조적 결함과 노이즈를 최소화한다. 이를 통해 더 정확하고 신뢰성 있는 라인 맵을 구축한다. 이후, 라인 추출 및 매칭, 초기 자세 추정, 자세 조정 단계를 거쳐 자세를 추정한다. 효율적인 라인 매칭을 위해 계층적 매칭 전략과 라인 세그먼트간의 colinearlity를 고려한 전략이 사용되었다. 이 전략은 기존 방법의 문제인 가려짐과 길이 불일치

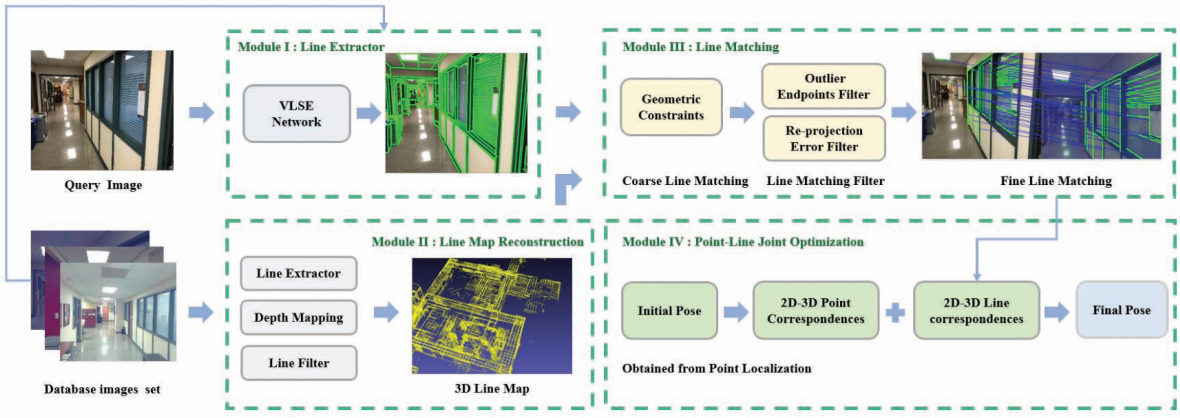

그림 21. PtLine의 선분 추출과 매칭을 이용한 전체 흐름도 [36].

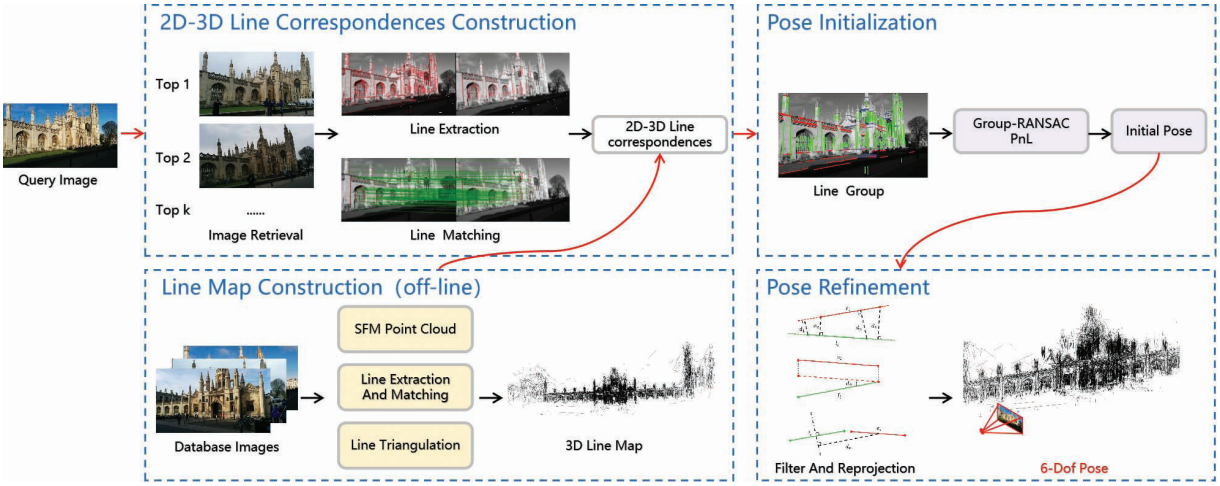

그림 22. LSLM VLoc의 선분 추출과 매칭을 이용한 전체 흐름도 [37].

문제를 해결하고, 매칭 속도를 향상시켰다. 또한, 초기 자세 추정 단계에서 Group-RANSAC PnL 알고리즘을 활용하였다. 이 방법은 라인 세그먼트의 공간적 다양성을 확보하여 자세 추정의 안정성을 향상시키는 데 중점을 둔다. Manhattan world 가정을 기반으로 라인 세그먼트를 세 개의 주요 직교 방향으로 그룹화하고, 각 반복마다 서로 다른 그룹에서 라인 매칭을 샘플링함으로써 선택된 라인 매칭이 공간적 방향 면에서 뚜렷하게 구별되도록 한다. 이를 통해 자세 추정의 정확성과 효율성을 향상시키며, 특히 텍스처가 부족하거나 조명 변화가 심한 환경에서도 성능을 발휘할 수 있다. 이후 초기 자세 추정을 조정하기 위해 라인 재투영 오차를 최소화하는 반복적 최적화 알고리즘을 사용하였다. 이는 쿼리 이미지의 2D 라인과 그에 대응하는 3D 라인의 재투영이 얼마나 잘 정렬되는지 평가하는 손실 함수를 정의는 것이다. 라인은 가려짐이나 불일치로 기존 재투영 방식으로는 한계가 있다. 이를 해결하기 위해 수직 방향 거리 오차, 수직 이등분 거리 오차 그리고 각도 오차를 고려한 새로운 재투영 오차 손실 함수를 적용하였다.

4.4. Implicit 공간 표현을 이용한 기술들

암시적(implicit) 3차원 표현 방법은 데이터를 통해 3차원 형태를 직관적으로 바로 알 수는 없지만, 적은 메모리로도 공간을 연속적으로 또는 보다 세밀하고 밀도 높게

표현할 수 있다. 최근 NeRF와 3D Gaussian splatting와 같이 새로운 형태의 효과적인 암시적 표현 방법이 많이 제안되었고, 많은 visual localization 기법들이 이를 활용하고 있다.

Loc-NeRF [38]는 입자필터(particle) 기반 MCL 기반 위치 추정 방법과 NeRF를 결합한 새로운 위치 추정 방법이다. 학습된 NeRF를 인코딩된 지도로 하여 쿼리 이미지와 로봇의 움직임 추정치를 입력으로 받아 로봇의 6-DoF 포즈를 추정한다. 핵심 아이디어는 입자필터 업데이트에 있어서 NeRF의 랜더링된 이미지를 이용하여 위치 추정을 한다. Loc-NeRF의 입자필터는 예측, 업데이트, 리샘플링의 세 단계로 구성된다. 예측 단계에서는 이전 시간의 입자 집합과 로봇의 움직임 측정값을 바탕으로 현재 시간의 입자 집합을 예측한다. 이 과정에서 로봇의 움직임 모델을 사용하며, 주행거리 오차를 고려하기 위해 가우시안 노이즈를 추가한다. 이러한 접근은 로봇의 움직임을 정확하게 모델링하면서도 불확실성을 고려할 수 있게 한다. 업데이트 단계는 현재 수집된 카메라 이미지를 사용하여 각 입자의 가중치를 조정한다. 이 단계에서는 측정 가능도를 근사하는 함수를 사용하여 실제 이미지와 NeRF가 예측한 이미지를 비교한다. 이 비교를 통해 각 입자가 실제 로봇의 위치를 얼마나 잘 표현하는지 평가하고, 그에 따라 가중치를 할당한다. 효율적인 계산을 위해 이미지의 일부 픽셀만을 사용하여 가중치를 업데이트한다. 리샘플링 단계는 업데이트된 가중치를 바탕으로 입자 집합을 재구성한다. 이 과정에서 높은 가중치를 가진 입자는 더 많이 선택되고, 낮은 가중치를 가진 입자는 제거된다. 이를 통해 로봇의 실제 위치를 더 잘 표현하는 입자들과 집합을 구성한다. 또한, 계산 효율성을 높이기 위해 입자 어닐링(annealing) 기법을 도입한다. 이 기법은 입자의 분포 상태에 따라 예측 노이즈와 입자 수를 동적으로 조정한다. 입자들이 특정 영역에 집중되면 노이즈와 입자 수를 줄여 더 정밀한 위치 추정을 가능하게 하고 동시에 계산 부하를 줄인다.

CROSSFIRE [39]는 NeRF 기반 공간 표현과 특징 추출을 결합한 위치 추정 방법이다. 핵심 아이디어는 NeRF의 렌더러(renderer)와 CNN 기반의 특징 추출기를 동시에 학습시켜, 일관된 특징 표현을 생성하는 것이다. 이를 통해 NeRF 렌더러가 특정 시점에 대해 descriptor, RGB 이미지, 깊이 이미지를 렌더링하고 쿼리 이미지의 디스크립터와 매칭을 통해 위치를 추정한다. 제안된 방법의 학습 과정은 자기 지도 학습 방식으로 이루어진다. RGB 이미지, 평균 제곱 오차, 구조적 비유사성, 그리고 깊이 총 변동을 최소화하는 손실 함수를 사용하여 NeRF 를 학습한다. 동시에 CNN 특징 추출기와 신경망 렌더러가 생성한 디스크립터 맵 사이의 유사성을 최대화하고, 3D 공간에서 멀리 떨어진 점들 사이의 매칭을 억제하는 손실함수를 통해 일관된 특징 표현을 학습한다. 위치 추정 과정은 반복적인 특징 매칭을 통해 이루어진다. 초기 위치 추정치를 시작점으로 하여, CNN으로 추출한 쿼리 이미지의 특징과 신경망 렌더러가 생성한 특징을 매칭한다. 이를 통해 2D-3D 대응점을 얻고, RANSAC과 결합된 PnP 알고리즘을 사용하여 카메라 포즈를 추정한다. 신경망 렌더러의 유연성을 활용하여 이 과정을 반복함으로써 위치 추정의 정확도를 향상시킨다. 이 접근 방식은 기본적인 위치 추정 전략을 사용하면서도 높은 정확도의 위치 결정을 달성할 수 있음을 보여준다.공간 모델에 대한 명시적인 고려가 전혀 없이 주어진 입력 이미지의 카메라 자세를 바로 추정(regression)하는 방법도 있다. PoseNet [5]은 이러한

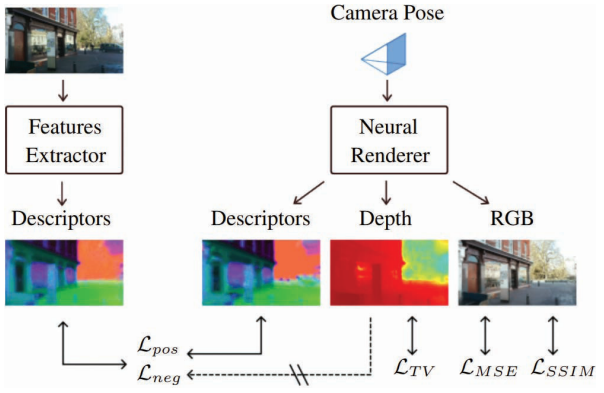

그림 23. 입력 이미지의 descriptor와 NeRF 랜더러와의 손실함수 관계 [39].

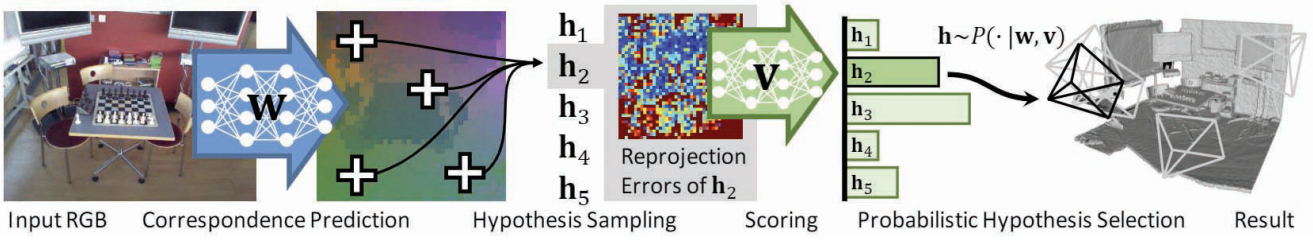

그림 24. DSAC의 scene coodinate regression (w 파트)과 미분가능한 RANSAC (v 파트)을 이용한 전체 흐름도 [41].

접근의 시작점으로 생각할 수 있는데 CNN 신경망의 입력으로 영상을 받아 end-to-end 접근에서 바로 (절대 좌표계 기준의) 카메라의 자세를 추정한다. 이러한 접근을 absolute pose regression이라고 하는데, 이후 다양한 접근들을 사용한 연구들이 있었다. 그 중 대표적인 방법은 **scene coodinate regression** (이하 SCR) [40]이다. SCR은 ML 모델의 출력으로 입력 영상의 각 픽셀의 3차원 위치를 출력하고, 이를 기존 RANSAC과 PnP 알고리즘을 이용하여 카메라 자세를 획득한다. PoseNet에서 사용한 absolute pose regression 보다 end-to-end 접근에서 한 발 물러선 접근으로 볼 수 있다.

DSAC [41]은 SCR 관점에서 CNN을 통해 픽셀의 3차원 위치(scene coordinate)을 알아내는 모듈(그림 24의 w 파트)과 RANSAC 알고리즘을 미분가능하도록 만들어 신경망을 통해 구성한 모듈(그림 24의 v 파트)을 통해 카메라의 자세를 추정한다. 특히 기존 RANSAC 알고리즘을 신경망으로 대체하였는데, (강화학습의 policy gradient 과정과 유사하게) 가장 최고의 가설을 선택(hypothesis selection) 과정을 확률적 샘플링으로 대체하여 미분 가능하도록 만들어 학습이 가능하게 만들었다. 따라서 DSAC은 두 개의 순차적인 신경망 모듈을 이용해 다시 end-to-end 형태의 visual localization이 가능하도록 하였다. 이후 개선된 **DSAC++** [42]은 기존의 42x42 이미지 패치를 입력으로 사용하던 SCR 파트를 VGG 스타일의 FCN(Fully Convolutional Network)으로 대체하고, DSAC의 가설 선택 과정의 손실함수 부분을 보다 부드럽고(smooth)하고 값의 범위가 한정되도록 만들었다. 또 기존 RGB-D 이미지를 이용하던 부분을 RGB 이미지를 이용할 수 있도록 개선하였다. 이후 보다 효율적인 네트워크 디자인과 카메라 자세 최적화 부분을 개선한 **DSAC*** [43]도 제안되었다.

DSM [44]은 cost volume을 활용한 dense scene matching(DSM) 모듈을 이용한 위치 추정 방법이다. Cost volume은 기존 스테레오 매칭, optical-flow, multi-view-stereo 등에서 유사도를 계산하기 위해 사용되었다. DSM은 2D-3D 포인트의 매칭 유사도을 통해 정확한 위치 추정을 위해 cost volume을 사용하였다. Cost volume을 구성하기위해 쿼리 이미지의 픽셀과 3D scene 포인트 사이의 상관관계를 계산한다. 이때 유사도는 코사인 유사도로 계산된다. 매 scene 마다, 2D-3D 상관관계의 수가 다르기 때문에 cost volume의 크기가 달라지며 이는 CNN 처리가 어려워진다. DSM은 상위 k개의 후보를 정렬하여 CNN의 입력으로 사용한다. 이후 CNN을 통해 신뢰도를 추정하고 쿼리 이미지의 3차원 좌표를 추정한다. 그리고 RANSAN과 PnP 알고리즘을 통해 자세를 추정한다.

ACE [45]는 end-to-end 학습을 위해 사용되었던 네트워크 대신, re-projection error기반의 최적화와 MLP 헤드를 이용해 수 시간에서 수일이 걸리던 학습 시간을 5분 이내로 줄이면서도 높은 성능을 유지하는 가속화된 좌표 인코딩(acclerated coordniate encoding; 이하 ACE) 방법이다. ACE는 학습 가속화를 위해 먼저 그레디언트 분리(gradient decorrelation)를 이용한 학습 방법을 사용한다. 일반적은 학습 과정에서 CNN을 통한 패치는 공통된 하나의 이미지에서 예측된다. 이 패치를 최적화할 때 패치간 손실과 그레디언트는 큰 상관관계를 가지고 있어 느린 수렴 속도를 보인다. ACE는 패치를 무작위로 섞음으로써 패치간 그레디언트를 분리(decorrelation)하여 학습에 더 빠른

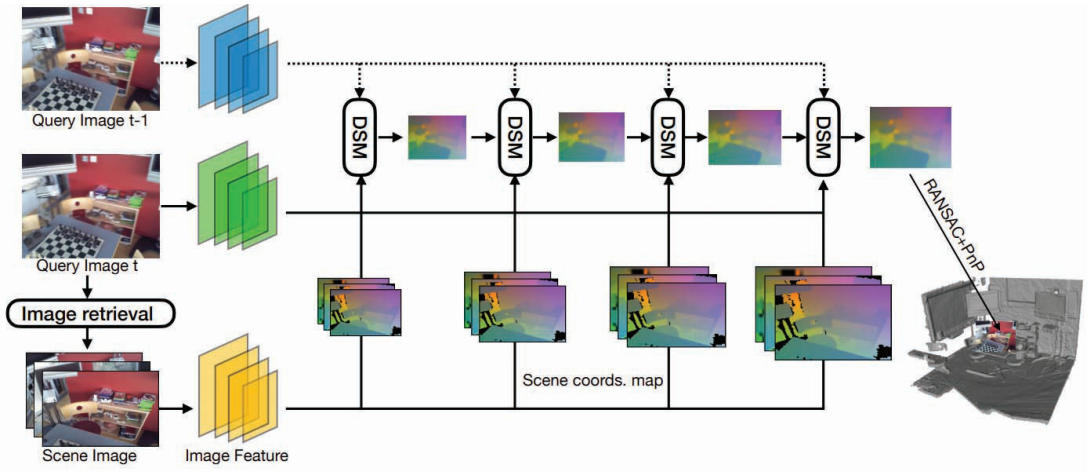

그림 25. 위치 추정 파이프라인 [44].

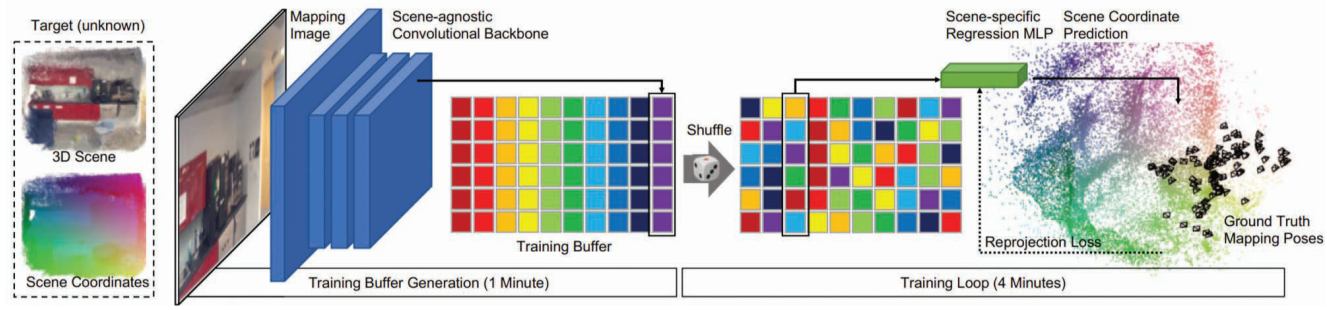

그림 26. ACE의 학습 과정 [45].

수렴 속도를 보인다. 이 과정에서 회귀(regreesion) 네트워크에 CNN backbone과 MLP 헤드를 분할한다. MLP 헤드는 CNN backbone과 달리 인접 픽셀에 접근할 필요 없어 패치를 무작위로 섞는 것이 가능하여 더 효율적인 학습을 가능하도록 한다. ACE는 또한 커리큘럼 학습을 사용한다. 이는 기존 end-to-end 방식이 오히려 많은 비용이 요구되어 학습 시간을 증가시키는 문제를 해결한다. 이는 moving inlier threshold를 사용하여 학습이 진행됨에 따라 threshold를 변화시켜 좋은 예측에 집중하고, RANSAC에 의해 필터링된 덜 정확한 예측을 무시하여 가속화한다. 이외에도 반정밀도(half-precision) 부동소수점 가중치로 학습하여 저가형 GPU에서도 가속화된 학습을 가능하도록 한다.

5. Visual Localization의 성능 평가

5.1. 공개 데이터셋

Visual localization 기술의 성능 평가를 위한 다양한 데이터셋들이 공개되어 있다. KITTI 데이터셋 [46]과 Long-term Visual Localization 데이터셋 [47]은 가장 널리 알려진 데이터셋으로 알고리즘의 성능 평가를 위한 지표, 그리고 벤치마크 결과을 홈페이지에 공개하고 있다.

- **(실외) KITTI Vision Benchmark Suite Homepage**
  - ○ Visual odometry/SLAM에 가장 널리 사용되는 데이터셋 및 벤치마크 제공(+ stereo camera, LiDAR 센서 데이터 제공)
  - ○ 고성능 RTK-GPS를 통해 획득된 참값(ground-

truth; 이하 GT) 제공(위성 사진을 이용한 연구[52][26]에 활용되기 함)

- **(실내/외) Long-term Visual Localization Homepage**
  - ㅇ조도/날씨 변화가 큰 상황의 다양한 영상 데이터셋 및 벤치마크 제공
  - ㅇ데이터셋 모음: Aachen Day-Night, (Extended) CMU-Seasons, RobotCar Seasons, InLoc, SILDa Weather and Time of Day, Symphony Seasons, Gangnam Station and Hyundai Department Store, ETH-Microsoft, Cross-Seasons Correspondence

그 외에 아래와 같이 정말 다양한 공개 데이터셋들이 있고, 이들을 다양한 분류 체계를 통해 정리한 페이지들[48] [49]도 존재한다.

- **(실외) Oxford RobotCar Dataset Homepage**
- **(실외) Ford Multi-AV Seasonal Dataset Homepage**
- **(실외) NCLT (North Campus Long-Term) Vision and LIDAR Dataset Homepage**
- **(실외) Complex Urban Dataset Homepage**
- **(실외) CrowdDriven (Mapillary) Dataset Homepage**
- **(실외/위성) CVUSA (Cross-view USA) Dataset Homepage**
- **(실외/위성) CVACT Dataset Github**
- **(실외) Google Landmarks Dataset v2 Github**
- **(실외) Cambridge Landmarks Dataset Github**
- **(실내) InLoc Dataset Github**
- **(실내) Zillow Indoor Dataset (ZInD) Github**
- **(실내/가상) Structured3D Dataset Github**

5.2. 평가 지표

Visual place recognition 기술이 찾은 영상 인덱스 (또는 장소 ID)는 이미지 분류(image classification) 문제에서 사용하는 재현율(recall)과 정밀도(precision)를 통해 평가한다. Visual place recognition 문제는 클래스(class)가 각각의 장소로 정의된 매우 많은 수의 클래스에 대한 이미지 분류(image classification)와 같고, 따라서 동일한 평가 지표와 결과 해석을 적용할 수 있다. 평가 지표를 적용할 때, 정해진 거리 임계값 이내의 장소를 찾은 경우 올바르게 찾음(hit)으로 판단한다. 또 top-K recall과 같이 1개의 최종 결과가 아니라 K개의 결과에 대해 재현율이나 정밀도를 표현할 수 있다.

그 외의 visual localization 기술이 추정한 3차원 카메라 자세는 참값(GT)이 제공되는 경우 쉽게 오차를 계산할 수 있다. 3차원 카메라 자세는 카메라의 3차원 위치와 3차원 방향각으로 나뉘어지고, 각각 위치오차(position error)와 회전오차(orientation error)로 나뉘어 표현한다. 추정값과 참값, 즉 한 쌍의 3차원 위치 사이의 이동 오차는 일반적으로 두 3차원 점 사이의 직선 거리, 즉 유클리디안(Euclidean) 거리로 계산된다. 마찬가지로 한 쌍의 3차원 방향각 사이의 회전오차는 두 방향각 사이의 차, 즉 첫 번째 방향각에서 두 번째 방향각으로 회전할 때 필요한 회전 각도로 계산된다. 3차원 방향각은 회전행렬이나 사원수(quaternion) 또는 axis-angle 표현법으로 나타낼 수 있는데, 회전행렬의 경우

$$\theta = \arccos\left(\frac{\mathrm{tr}(R_\Delta)-1}{2}\right) \text{ where } R_\Delta = R_2^\top R_1$$

와 같이 두 방향각 사이의 상대적인 3차원 회전 R_Δ을 계산할 수 있고, 대각합(trace) tr과 역코사인함수를 이용해 회전 각도 θ를 계산할 수 있다. 만약 R_Δ를 axis-angle 표현법으로 나타낸 경우, 표현된 각도(angle) 값이 바로 두 방향각 사이의 각도이다.

Visual localization 기술이 찾은 3차원 카메라 자세가 절대 좌표계가 아닌 (처음 또는 특정 영상을 기준으로 한) 지역 좌표계로 표현되는 경우가 많다. 따라서 추정된 카메라 자세와 절대 좌표계의 참값을 그대로 비교하여 오차를 계산할 수 없고, 추정값과 참값을 특정 지점 (또는 영상)을 기준으로 한 상대적인 카메라 자세로 변환하고 비교한다. 이렇게 계산된 위치오차는 상대위치오차(relative position error; RPE)와 같이 표기하여 절대위치오차(absolute position error; APE)와 구분하여 표현하기도 한다. 특

히 monocular visual localization 기법은 scale ambiguity 한계로 인해 카메라 궤적의 크기를 알 수 없다. 따라서 visual odometry/SLAM, SfM에서는 카메라 궤적의 참값과 추정값의 닮음변환(similarity transformation)을 찾고 이를 보정한 후, 위치 오차와 회전 오차를 계산한다[7].

Visual odometry/SLAM의 경우 커스텀 데이터셋의 참값 확보의 어려움 때문에 카메라 궤적의 마지막 위치만 비교하여 절대궤적오차(absolute trajectory error; ATE)로 표현한다. 그러나 ATE는 카메라 궤적이 겹침이 있는 경우, 정확하게 정확도를 표현하기 어려운 문제가 있다. 따라서 KITTI Odometry Benchmark [46]와 같은 성능 평가에서는 주어진 궤적을 각각 100m, 200m, …, 800m 단위로 쪼개고 해당 궤적 조각(path segment)에서 위치오차와 회전오차를 계산하고, 궤적 조각의 길이로 나누어 정규화한 후, 평균을 도출하여 사용한다.

실제 성능 평가는 다양한 카메라 시점에서 촬영된 여러 영상을 이용하므로 여러 영상에서 얻은 각 평가 지표의 대표값을 이용할 수 있다. 대표값으로 평균(mean)이나 평균제곱근(root mean sqaure; RMS)을 이용할 수 있지만, 평균은 이상치(outlier) 결과에 값이 크게 망가지기 때문에 중앙값(median)을 사용하는 경우가 더 많다. 그러나 중간값 또한 전체 결과의 한 단면이기 때문에 극단적으로 [그림 27]와 같은 형태의 누적확률분포(cumulative probability districution)로 전체 영상에 대한 결과를 모두 표시할 수 있다. 중간값은 누적확률이 0.5인 지점의 단면만을 보여준다. 이러한 그래프를 이용한 표현은 결과를 압축적으로 표현하기 힘들기 때문에 특정 체크 지점을 정하고 해당 위치의 값이나 비율을 이용한다. 각 연구들은 환경에 따라 표 1과 같이 각각 다양한 체크 지점(check point)을 사용한다.

6. 결론 및 요약

본 기고에서 visual localization의 최신 연구 동향을 살펴보았다. 많은 visual localization 기법이 있고, 이들을 입력 영상의 개수(single vs. multiple) 관점에서 살펴보았다. 기존 visual odometry/SLAM에서 많이 사용하는 이미지 시퀀스를 이용하는 방법은 기존 결과를 누적하여 활용하는 순차적(sequential) 측위은 높은 성능을 위해 camera pose estimation에도 많이 적용되고 있다. Visual localization의 중요 기반 기술 중 하나는 이미지 매칭으로, Image Matching Challenge에서 사용되는 주요 기법들을 리뷰하였다. 이미지 매칭이나 측위 관점에서 전체적인 영상의 특징을 활용하는 대략적인(coarse) 매칭 단계와 영상의 부분적인 특징을 활용하는 세밀한(fine) 매칭 단계로 구성된 계층적(hierarchical) 측위가 많이 활용되고 있다. 또 visual localization 기법을 공간 (또는 지도)를 표현하는 방법에 따라 구분하였다. Visual localization이 적용되는 공간의 크기가 커지면, 이미지 데이터베이스나 특징 descriptor를 포함한 점군 형태의 공간 표현은 너무 큰 용량으로 운용이 어려워진다. 따라서 보다 가벼운 공간 표현 방법(예: 폴리곤 메시)이나 기존(prior)에 이미 구축된 공간 정보(예: 위성영상)

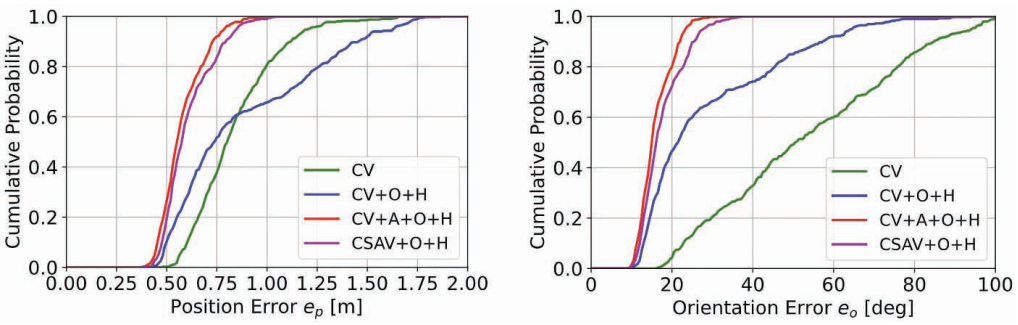

그림 27. 여러 위치/회전 오차를 모두 표현하는 누적확률분포 그래프 [50].

Methods	Measure Types	Point Types	Check Points
NetVLAD [2]	Recall	Top-K	1 / 5 / 10
Satellite v1 [26]	Recall	Top-K	1 / 5 / 10 / 1%
Satellite v2 [27] OrienterNet [33]	Ratio	〈 Position Error 〈 Orientation Error	1m / 3m / 5m 1° / 3° / 5°
InLoc [4]	Ratio	〈 (Position, Orientation) Error	(0.25m, 10°) / (0.5m, 10°) / (1m, 10°)
Visual Localization Benchmark [47] HLoc [12] MeshLoc [29]	Ratio	〈 (Position, Orientation) Error	(0.25m, 2°) / (0.5m, 5°) / (5m, 10°)
F3Loc [32]	Ratio	〈 Position Error 〈 (Position, Orientation) Error	0.1m / 0.5m / 1m (1m, 30°)
DSAC* [43]	Indoor: Ratio Outdoor: Error	〈 (Position, Orientation) Error 〈 Percentile	(0.01m, 1°) / (0.02m, 2°) / (0.05m, 5°) 50% (median)
ACE [45]	Indoor: Ratio Outdoor: Error	〈 (Position, Orientation) Error 〈 Percentile	(0.05m, 5°) 50% (median)

표 1. (누적확률분표 대신) 특정 체크 지점을 이용한 visual localization 기법들.

를 활용한 기법들도 실용적인 관점에서 많이 연구가 되고 있다. 최근 NeRF나 3D Gaussian splatting과 같이 암시적(implicit) 공간 표현 기법에 대한 관심이 뜨겁고 이를 활용한 방법이나, SCR과 같이 신경망을 통해 특징점 추출과 매칭이 아닌 각 픽셀의 3차원 좌표를 바로 얻어 활용하는 연구도 많이 이뤄지고 있다.

본 기고에서는 visual localization의 성능 평가 방법에 대해도 살펴보았다. 다양한 공개 데이터셋과 벤치마크를 리뷰하였고, 성능 평가를 위한 다양한 평가 지표들을 소개하였다. 여러 영상에 대한 visual localization 성능 지표를 단순히 평균이나 중앙값을 표현하지 않고, 누적확률분포나 다양한 체크 지점을 통해 표현하는 방법도 살펴보았다.

Acknowledgements

본 기고는 2024년도 한국전자통신연구원(ETRI)의 "DNA 기반 국가지능화 핵심 기술사업" (과제번호: 24ZR1210)의 지원을 받아 수행되었습니다.

References

[1] Miao et al., **"A Survey on Monocular Re-Localization: From the Perspective of Scene Map Representation"**, IEEE Transactions on Intelligent Vehicles, 2024 DOI

[2] Arandjelovic et al., **"NetVLAD: CNN Architecture for Weakly Supervised Place Recognition"**, CVPR, 2016 DOI (PAMI 2018)

[3] Zeisl et al., **"Camera Pose Voting for Large-Scale Image-Based Localization"**, ICCV, 2015 DOI

[4] Taira et al., **"InLoc: Indoor Visual Localization with Dense Matching and View Synthesis"**, CVPR, 2018 DOI (PAMI 2021)

[5] Kendall et al., **"PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization"**, ICCV, 2015 CVF (Caffe), (PyTorch)

[6] Kitt et al., **"Visual Odometry based on Stereo Image Sequences with RANSAC-based Outlier Rejection Scheme"**, IV, 2010 DOI

[7] Campos et al., **"ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM"**, IEEE Transactions on Robotics, Vol. 37, No. 6, 2021 DOI

[8] Schonberger et al., **"Structure-from-Motion Revisited"**, CVPR, 2016 DOI

[9] Newcombe et al., **"DTAM: Dense Tracking and Mapping in Real-time"**, ICCV, 2011 DOI

[10] Engel et al., **"Direct Sparse Odometry"**, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 40, No. 3, 2018 DOI

[11] **"Image Matching: Local Features & Beyond"** Homepage

[12] Sarlin et al., **"From Coarse to Fine: Robust Hierarchical Localization at Large Scale"**, CVPR, 2019 DOI

[13] Edstedt et al., **"RoMa: Robust Dense Feature Matching"**, CVPR, 2024 CVF

[14] Caron et al., **"Emerging Properties in Self-Supervised Vision Transformers"**, ICCV, 2021 DOI

[15] Oquab et al., **"DINOv2: Learning Robust Visual Features without Supervision"**, arXiv, 2023 arXiv

[16] Vincent Qin, **"Image Matching WebUI"** Github

[17] DeTone et al., **"SuperPoint: Self-supervised Interest Point Detection and Description"**, CVPRW, 2018 DOI

[18] Zhao et al., **"ALIKED: A Lighter Keypoint and Descriptor Extraction**

Network via Deformable Transformation", IEEE Transactions on Instrumentation and Measurement, Vo. 72, 2023 DOI arXiv Github

[19] Edstedt et al., **"DeDoDe: Detect, Don't Describe - Describe, Don't Detect for Local Feature Matching"**, 3DV, 2024 DOI

[20] Edstedt et al., **"DeDoDe v2: Analyzing and Improving the DeDoDe Keypoint Detector"**, CVPRW, 2024 CVF

[21] Potje et al., **"XFeat: Accelerated Features for Lightweight Image Matching"**, CVPR, 2024 CVF

[22] Sarlin et al., **"SuperGlue: Learning Feature Matching with Graph Neural Networks"**, CVPR, 2020, arXiv

[23] Lindenberger et al., **"LightGlue: Local Feature Matching at Light Speed"**, ICCV, 2023 DOI

[24] Sun et al., **"LoFTR: Detector-Free Local Feature Matching with Transformers"**, CVPR, 2021 arXiv

[25] Jiang et al., **"OmniGlue: Generalizable Feature Matching with Foundation Model Guidance"**, CVPR, 2024 CVF

[26] Shi et al., **"Accurate 3-DoF Camera Geo-localization via Ground-to-satellite Image Matching"**, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 45, No. 3, 2023 DOI

[27] Shi et al., **"Beyond Cross-view Image Retrieval: Highly Accurate Vehicle Localization using Satellite Image"**, CVPR, 2022 DOI

[28] Li et al., **"DeepI2P: Image-to-Point Cloud Registration via Deep Classification"**, CVPR, 2021 DOI

[29] Panek et al., "MeshLoc: Mesh-based Visual Localization", ECCV, 2022 DOI

[30] Panek et al., **"Visual Localization using Imperfect 3D Models from the Internet"**, CVPR, 2023 DOI

[31] Min et al., **"LASER: LAtent SpacE Rendering for 2D Visual Localization"**, CVPR, 2022 DOI

[32] Changan Chen et al., "**F3Loc: Fusion and Filtering for Floorplan Localization"**, CVPR, 2024 CVF

[33] Sarlin et al., **"OrienterNet: Visual Localization in 2D Public Maps with Neural Matching"**, CVPR, 2023 arXiv

[34] Yu et al., "**Monocular Camera Localization in Prior LiDAR Maps with 2D-3D Line Correspondences"**, IROS, 2020 DOI

[35] Qin et al., **"VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator"**, IEEE Transactions on Robotics, Vol. 34, No. 4, 2018 DOI

[36] Gao et al., **"Pose Refinement with Joint Optimization of Visual Points and Lines"**, IROS, 2022 DOI

[37] Liu et al., **"Lightweight Structured Line Map Based VIsual Localization"**, IEEE Robotics and Automation Letter, Vol 9, No 6, 2024 DOI

[38] Maggio et al., **"Loc-NeRF: Monte Carlo Localization using Neural Radiance Fields"**, ICRA, 2023 DOI

[39] Moreau et al., **"CROSSFIRE: Camera Relocalization On Self-Supervised Features from an Implicit Representation"**, ICCV, 2023 DOI

[40] Shotton et al., **"Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images"**, CVPR, 2013 DOI

[41] Brachmann et al., "**DSAC - Differentiable RANSAC for Camera Localization"**, CVPR, 2017 CVF

[42] Brachmann et al., **"Learning Less is More - 6D Camera Localization via 3D Surface Regression"**, CVPR, 2018 CVF

[43] Brachmann et al., **"Visual Camera Re-Localization from RGB and RGB-D Images Using DSAC"**, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 44, No. 9, 2022 DOI

[44] Tang et al., **"Learning Camera Localization via Dense Scene Matching"**, CVPR, 2021 DOI

[45] Brachmann et al., "**Accelerated Coordinate Encoding: Learning to Relocalize in Minutes using RGB and Poses"**, CVPR, 2023 arXiv

[46] Geiger et al., **"Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite"**, CVPR, 2012 Homepage

[47] Toft et al., "**Long-Term Visual Localization Revisited"**, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 44, Vol. 4, 2022 DOI

[48] MINT Lab, **"Awesome Robotics Datasets"** Github

[49] Younggun Cho, **"Awesome SLAM Datasets"** Github

[50] Cho et al., **"Accurate and Resilient GPS-only Localization with Velocity Constraints"**, IEEE Access, Vol. 12, 2024 DOI

[51] Geiger et al., **"Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite"**, CVPR, 2012 DOI

[52] Shi et al., **"Where Am I Looking At? Joint Location and Orientation Estimation by Cross-view Matching"**, CVPR, 2020 Github

최 진 원

2016년~2023년 서울과학기술대학교 기계시스템디자인공학과 (공학사)
2023년~현 재 서울과학기술대학교 컴퓨터공학과 (공학석사)

허 동 욱

2014년~2018년 명지전문대학 기계과 (전문학사)
2019년~2022년 서울과학기술대학교 기계시스템디자인공학과 (공학사)
2024년~현 재 서울과학기술대학교 컴퓨터공학과 (공학석사)

Nguyen Cong Quy

2016년~2021년 다낭과학기술대학교 전자통신부 (공학사)
2023년~현 재 서울과학기술대학교 컴퓨터공학과 (공학석사)

서 찬 호

2017년~2023년 서울과학기술대학교 전기정보공학과 (공학사)
2023년~현 재 서울과학기술대학교 컴퓨터공학과 (공학석사)

장 인 성

1992년~1999년 부산대학교 전자계산학과 (이학사)
1999년~2001년 부산대학교 전자계산학과 (이학석사)
2006년~2008년 부산대학교 컴퓨터공학부 (공학박사수료)
2018년~2022년 실감형공간정보연구단 (단장, 국토교통부)
2001년~현 재 한국전자통신연구원 모빌리티인프라 연구실 (기술총괄)

최 성 록

2001년~2006년 서울대학교 기계항공공학과 (공학사)
2006년~2008년 KAIST 로봇공학학제전공 (공학석사)
2014년~2019년 KAIST 로봇공학학제전공 (공학박사)
2008년~2020년 ETRI 지능로보틱스연구본부 (선임연구원)
2021년~현 재 서울과학기술대학교 컴퓨터공학과 (조교수)