

김만동의 석사 학위논문을 인준함.

지도 교수: 김 대은

김 은태

이 원기

연세대학교 대학원

2019년 06월

Acknowledgements

처음 연구실 인원들과 안산을 올라가서 일출을 보며 식사 생활을 잘하리라는 마음을 먹었던 게 생생하게 기억이 납니다. 그 이후 연구실에서 여러 가지 일을 하며 힘들지만 즐겁게 식사 생활을 하였습니다. 여러 가지 과정을 버티며 서로에게 버팀목이 되어주며 생활하여 한 계단씩 밟아 나아갈 수 있던 거 같습니다. 때로는 연구가 꿈이 나올 정도로 연구 진행에 대한 강박을 가졌던 적도 있고 편하게 밥을 먹기가 어려울 정도로 부담을 가졌던 적도 있습니다. 지금 생각해보면 제 인생의 최고의 걱정이었던 것 같습니다. 그리고 지나 보니 그러한 과정들이 지금의 연구성과와 저를 만들어준 것 같습니다. 길다면 긴 식사 생활이지만 지난날들을 생각해보면 짧게만 느껴집니다. 연구실에 와서 가장 좋았던 것은 평생을 함께할 스승님과 형제들이 생겼다는 것입니다. 부디 연구실에 졸업하여도 각자 흩어져서 살아도 같은 시간을 공유했던 것을 잊지 않고 서로의 안부를 묻고 서로의 행사를 챙기며 지냈으면 좋겠습니다.

가장 먼저 저를 잘 만들어주신 교수님께 감사드립니다. 처음 연구실에 왔을 때부터 지금까지 어리광부리는 모습만 보여드린 것 같아 죄송한 마음도 있습니다. 하지만 지금의 강한 저를 만들어 주셔서 감사합니다. 다음은 저의 동기 경래형과 병문이 형에게 고맙습니다. 힘든 식사 생활에 버팀목이 되었고 때로는 서로의 방패가 되며 서로를 감싸주었기 때문에 연구실 생활을 더 잘 할 수 있었던 거 같습니다. 저와 스페인에 함께 갔던 민철이, 진수는 서로의 성격이 너무 달라서 함께하며 에피소드가 많았지만, 평생을 살면서 잊지 못할 기억이 될 거 같고, 소중한 함께한 감사합니다. 방장으로 항상 우리를 가이드해주고 우리가 한다는 것은 무조건 오케이 해주는 슬기형, 정말 힘들었을 때 힘이 되어주는 말과 용기를 북돋아 주고 같이 놀러 다닌 재현 이형, 군 위탁생인데도 불구하고 큰 형으로 중심을 잡아주신 종하형, 나와 취향이 잘맞아 이것저것 많이 공유했던 지승 그리고 재우, 그리고 마지막으로 연구를 물려준 이창민 박사에게 감사드립니다. 마지막으로 처음 석사를 하게 되었을 때부터 지금까지 저를 전폭적으로 지지해주신 어머니, 아버지 그리고 누나에게 감사합니다. 가족이라는 울타리가 없었더라면 용기조차 못 낼 일을 하게 되었습니다.

제가 석사를 하면서 가장 크게 얻은 것은 지식이 아닙니다. 용기만 있으면 많은 일을 할 수 있다는 마음가짐입니다. 운동과 학업을 병행하며 두 마리 토끼를 놓치고 싶지 않았습니다. 용기가 없었더라면 할 수 없었던 일이라고 생각합니다. 걱정의 시간을 함께한 형제들이 부디 무사히 졸업하길 바라며 글을 마칩니다.

p.s 마지막으로 코드와 논문을 검수해준 김민철 연구생에게 감사의 말을 전합니다.

차례

그림 차례	iv
표 차례	x
제 1 장 도입	1
제 1 절 생체 모방을 이용한 여러가지 내비게이션	2
제 2 절 연구 동기	4
제 3 절 연구 목적	5
제 4 절 논문의 구성	5
제 2 장 시각 내비게이션에 대한 동향	7
제 1 절 AI 내비게이션 알고리즘	8
1.1 AI 알고리즘을 이용한 내비게이션: SLAM	9
제 2 절 인식 알고리즘	9
제 3 절 동물들의 다양한 내비게이션	10
3.1 PI (Path Integration) 방법	11
3.2 RF (Route Following) 방법	12
제 4 절 다양한 지역 귀소 내비게이션	13
4.1 Snapshot 모델	14
4.2 ALV (Average Landmark Vector) 모델	16
4.3 ACV(Average Correctional Vector) 모델	17
4.4 DELV(Distance Estimated Landmark Vector) 모델	18
4.5 여러가지 Warping 방법	20
제 5 절 최근 연구동향	22
5.1 전체론적 관점	22

5.2	Descent in Image Distance (DID) 방법	23
5.3	Gradient-based Descent in image Distance 방법	24
5.4	Newton Descent in image Distance 방법	25
5.5	Haar-like Feature Landmark Vector (HFLV) 방법	25
제 6 절	스냅샷 모델의 문제점	26
6.1	문제 1 : 현지화 (Localization)	26
6.2	문제 2 : 폐색 (Occlusion)	27
6.3	문제 3 : 필드 시야	28
6.4	문제 4: 정렬	29
6.5	문제 5: Landmark 의 특징 선택	30
제 7 절	요약	31
제 3 장	경사하강법을 적용한 HFLV (Haar-like Feature LandMark Vector)	32
제 1 절	Haar-like Fraecture LandMark Vector 모델	33
1.1	실험환경 및 데이터 획득	33
1.2	Haar-like Feature 생성	34
1.3	매칭스코어 계산법	36
1.4	랜드마크 벡터생성	37
1.5	타입별 매칭스코어의 분포도	38
1.6	HFLV(Haar-like Features Landmark Vector) 결과	38
제 2 절	실험1: 경사 하강법	39
2.1	경사하강법 공식	41
2.2	실험 셋팅: 파라미터	43
2.3	경사하강법 적용:타입1	43
제 3 절	실험2: 내림차순을 적용한 경사하강법	44
3.1	내림차순을 이용한 마스크선별	46
3.2	여러가지 형태의 매칭스코어	47
3.3	경사하강법 과 랜드마크벡터 비교	48
제 4 절	가중치 관계 그래프	50
4.1	관계 그래프1: 가중치 매칭스코어	50
제 5 절	통합 그래프: 매칭스코어에 따른 내림차순 수행능력	51
5.1	다른 환경에서 경사하강법 적용	54
제 6 절	요약	55

제 4 장	HFLV-GD의 다양한 파라미터 학습	57
제 1 절	정보의 양에 따른 수행능력 평가	58
1.1	교차 검증	58
1.2	마스크 갯수별 수행능력 평가	59
1.3	제한된 위치별 학습과 테스트	60
1.4	포지션의 갯수별 수행능력 평가	61
1.5	집 주변 포지션 학습 : 4EA , 8EA	64
제 2 절	여러가지 환경환경에 적용한 경사하강법	66
2.1	장애물 (Obstacle) 이 있는 환경: 'arboreal', 'screen'	68
2.2	빛이 적은 환경: Hall	70
2.3	포지션의 갯수에 따른 환경별 수행능력 : arboreal, screen, hall	72
2.4	제한된 포지션에 따른 환경별 수행능력 : 4EA,8EA	73
제 3 절	여러가지 타입의 HF 를 적용한 HFLV-GD	74
제 4 절	요약	78
제 5 장	모바일 로봇에 적용한 HFLV-GD	81
제 1 절	목적	81
제 2 절	실험환경	82
2.1	이미지 기반 실험 : Ourlab	85
제 3 절	여러가지 학습 방법과 모바일 로봇 실험	85
3.1	이미지 정렬 문제: Visual Compass 사용	87
3.2	방법1 : Odometry 기반 학습	88
3.3	방법2 : Landscape 기반 학습	89
3.4	토의	91
제 4 절	요약	93
제 6 장	결론 및 향후 과제	94
제 1 절	결론	94
제 2 절	향후 과제	96
2.1	알고리즘의 강인함 테스트	96
2.2	시각 내비게이션의 어플리케이션	96
2.3	다양한 모바일 로봇 실험	96

그림 차례

2.1	Simultaneous Localization and Mapping (SLAM) 알고리즘의 모식도	10
2.2	Haar-like Features의 여러가지 형태	10
2.3	스냅샷 모델의 스냅샷 이미지. (a)현재 위치 (b)집 위치 ([80]의 논문에서 발췌)	16
2.4	ALV(Average Landmark Vector)를 이용한 방법.	17
2.5	시각 탐색 분야의 다양한 기술 분류. ([78]의 논문에서 발췌)	23
2.6	수평선 라인을 이용하는 예시. ([62]의 논문에서 발췌)	29
3.1	모바일 로봇과 데이터. (a) 모바일 로봇 (P3-DX) (b) 옴니 방향 이미지 . .	34
3.2	Haar-like Feature의 여러가지 형태. 본 논문에서는 가장 왼쪽편의 HF (Haar-like Feature) 를 타입1로 정의한다.	34
3.3	이미지상의 Horizontal Line 표시방법. (a) 임의의 위치에 생성된 HF (Haar-like Feature) (b) 마스크를 매칭할때 Horizontal 라인 주변에 생성된 HF (Haar-like Feature)	36
3.4	매칭스코어 계산 방법. 위 그림은 생성된 HF와 각 포지션에서 획득한 이미지의 매칭스코어 계산 방법이다.	36
3.5	임의의 5000개 마스크의 매칭스코어 분포도. 실험환경은 'a1original' 이고 6개의Type과 마스크의 모양 및 위치가 임의로 생성되었다. 검정 X 는 Type1,파랑 별은 Type2,마젠다 +는 Type3 , 빨강 동그라미는 Type4 ,노랑 다리아는 Type5 , 크레온 육각별은 Type6 이다.	39
3.6	HF의 갯수별 수행능력 그래프. 이미지상의 랜덤한 위치에서 생성되는 타입 1 형태의 HF (Haar-like Feature) 를 사용하여 HFLV (Haar-like Features Landmark Vector) 구현하였다. Vardy의 'a1original' 환경에 적용하였다. HF를 1개 사용했을때 부터 500개 사용했을 때의 오차율을 나타내었다. .	40

3.7	경사 하강법 원리에 대한 모식도. 경사 하강법은 지역적인 최솟값보다 작은 글로벌한 최솟값을 찾기 위한 방법이다.	41
3.8	AE (Angular Error) 그래프. Type1 형태의 HF (Haar-like Feature) 를 100개의 사용하여 10000번 학습을 시켰고 러닝레이트를 0.01로 고정하였다. 업데이트가 지날수록 수행능력이 향상되는것을 관찰할 수 있다.	45
3.9	'a1original' 실험환경의 벡터맵. 가운데 빨간색 마크는 집을 나타낸다. (a) HFLV (b) HFLV-GD를 각각 적용한 결과이다. 가중치는 총 10000번 업데이트를 진행하였고 러닝레이트는 0.01로 고정하고 랜덤하게 생성한 HF (Haar-like Features)는 100개를 사용하였다.	46
3.10	매칭스코어별 HF를 선별하는 방법 모식도. 각 형태의 매칭스코어를 내림차순 정리를 통해 선별된 마스크를 이용하여 집 이미지와 현재 위치의 이미지에서 다시 매칭스코어를 구하고 방향결정을 하게 된다.	48
3.11	세개의 매칭스코어 식에 따른 내림차순 정렬 적용. (a) 매칭 스코어1 (b) 매칭 스코어2 (c) 매칭 스코어3 을 각각 나타낸다. 각각의 방법 마다 100개의 HF를 사용하였고 10000회 반복 학습하였고 러닝레이트를 0.01로 고정하였다.	49
3.12	형태별로 매칭스코어와 가중치의 분포도를 나타낸 그래프. (a) 매칭 스코어1 (b) 매칭 스코어2 (c) 매칭 스코어3을 이용하여 실험한 결과이다. 각각의 점은 실험에서 지정된 마스크의 번호이다. 가운데 빨간 색 마크는 각각의 실험에 사용된 모든 HF의 매칭스코어 와 가중치의 평균을 나타낸다. 타입1의 HF를 총 100개 사용하였고 10000회 반복 학습하였고 러닝레이트를 0.01로 고정하였다.	50
3.13	세 가지 형태의 매칭 스코어를 이용하여 나타낸 벡터 맵. (a) 집 기준으로 내림차순 스코어 (b) 전체 포지션에서 대하여 누적된 스코어의 절대값 (c) 전체 스코어의 절댓값 형태의 스코어를 누적이 값이다. 각 형태별 매칭 스코어의 평균값보다 큰 HF를 선별하여 벡터 맵을 각각 그리고 있다. (c) 형태의 매칭 스코어가 자장 좋은 결과를 나타내고 있다.	51
3.14	세 가지 형태의 매칭 스코어를 이용하여 나타낸 벡터 맵. (a) 매칭 스코어1 (b) 매칭 스코어2 (c) 매칭 스코어3 을 각각 이용하여 실험한 결과이다. 각각의 가중치의 평균값보다 큰 HF를 선별하여 벡터 맵을 각각 그리고 있다.	52

3.15	여러가지 내림차순 방법 별 수행능력 그래프. 그림에서 그래프는 위 에서 제시한 세개의 형태의 매칭스코어를 사용하고 있다. 각각의 그래프에서 의 스코어는 매칭스코어를 뜻한다. 여러가지 방법으로 내림차순 정렬을 하였을때 마스크 갯수별로 각도의 평균 에러를 한눈에 볼 수 있다.	52
3.16	여러가지 실환경에서 HFLV-GD를 적용한 벡터맵. 각각 실험환경 (a) ar-boreal (b) screen (c) Hall 실험환경에서 각각 실험을 진행하였다. Type 1 사용, HF (Haar-like Feature) 100개 사용, 5000회 반복 , 러닝레이트 0.01 로 고정하였다.	54
4.1	k-fold 교차검증에 대한 모식도. 데이터 사이즈에 따라 k값을 다르게 실험 을 진행한다.	59
4.2	HF의 갯수별 수행능력 그래프. (a) 100EA (b) 500EA (c) 1500EA (d) 3000EA 의 HF를 이용하여 학습을 진행하였고 평균과 표준편차로 에러바 그래프 를 표현하였다. 전부 동일하게 'a1original'환경에서 타입1을 사용하였고 5000회 업데이트 통해 학습을 수행하였다.	60
4.3	'a1original'환경에서 절반의 포지션을 사용했을때 벡터맵. 타입1의 HF 100개 사용하였고 5000회 업데이트 통해 학습을 수행하였다. 위 그림은 학습데이터와 시험데이터를 각각 빨간색 점선의 벡터와 검정색 실선의 벡터로 나타낸 결과이다. (a), (b), (c), (d) 는 전체 포지션의 절반을 학습데 이터로 사용하였다. (e), (f), (g), (h)는 전체 포지션의 사분의 일을 학습데 이터 사용하였다.	62
4.4	'a1original' 환경에서 포지션의 절반을 사용했을때 수행능력 그래프. 타 입1의 HF 100개 사용하였고 5000회 업데이트 통해 학습을 수행하였다. (a) 는 절반에 해당되는 지역들을 학습하였고 (b) 는 사분의일에 해당되는 지역들을 각각 학습한 결과이다. 상황별 수행한 결과를 평균과 표준편차 로 에러바 그래프를 표현하였다.	63
4.5	'a1original' 환경에서 포지션의 갯수별 벡터맵. 타입1의 HF 100개 사용 하였고 5000회 업데이트 통해 학습을 수행하였다. 학습데이터의 갯수별 테스트 데이터의 벡터맵 빨간색 점선은 학습포지션 ,검정색 실선은 테스 트 포지션 각각 (a) 100퍼센트 (d) 50퍼센트 (c) 25퍼센트(d) 10퍼센트 의 포지션을 임의로 선정하여 학습결과의 벡터맵이다.	63

4.6	'a1original' 환경에서 포지션의 갯수별 수행능력 그래프. (a) 100퍼센트 (d) 50퍼센트 (c) 25퍼센트(d) 10퍼센트 의 포지션을 임의로 선정하여 학습한 결과를 에리바 그래프로 나타내었다. 타입1의 HF 100개 사용하였고 5000회 업데이트 통해 학습을 수행하였다. 학습데이터의 갯수별 테스트 데이터의 벡터맵 빨간색 점선은 학습포지션,검정색 실선은 테스트 포지션 각각 나타내었다.	64
4.7	집을 중심으로 동서남북 모식도. 북동,북서,남동,남서인 각각 4방향과 8 방향을 학습 데이터로 사용하였다.	65
4.8	'a1original' 환경에서 동서남북으로 학습 결과와 랜덤학습 결과 벡터맵. 각각 (a) 고정된 4개의 포지션 학습결과 벡터맵 (b) 임의로 지정된 4개의 포지션 학습결과 를 나타낸다. 100개의 HF를 사용했으며 5000회의 반복 학습을 진행하고 러닝레이트는 0.01로 고정하였다.	66
4.9	'a1original' 환경에서 동서남북으로 학습 결과와 랜덤학습 결과 수행능력 그래프. (a) 4개의 고정된 포지션을 학습한 결과를 나타내는 AE(Angular Error) 그래프 (b) 4개의 임의로 선택된 포지션을 학습한 결과를 나타내 AE(Angular Error)그래프	67
4.10	'a1original' 환경에서 집 주변 8개 위치 학습 결과와 랜덤학습 결과 벡터 맵. 100개의 HF를 사용했으며 5000회의 반복학습을 진행하고 러닝레이트는 0.01로 고정하였다. 각각 (a)고정된8개의 포지션 학습결과 벡터맵 (b) 임의로 지정된 8개의 포지션 학습결과 를 나타낸다.	68
4.11	'a1original' 환경에서 8개의 위치 학습 결과와 랜덤학습 결과 수행능력 그래프. (a) 8개의 고정된 포지션을 학습한 결과를 나타내는 AE (Angular Error) 그래프 (b) 8개의 임의로 선택된 포지션을 학습한 결과를 나타내 AE (Angular Error) 그래프	69
4.12	실험 환경별 파노라마 이미지. varyd의 데이터셋 환경중 'a1original'이외의 여러가지 실험 환경이 있다. (a) aboreal 과 (b) screen 환경에는 각각 장애물을 가지고 있다. (c) hall환경의 경우 빛이 적은 환경이어서 낮은 해상도의 이미지가 수집된 환경이다.	70
4.13	각 실험환경에서 상황별 벡터맵. 모두 타입1 의 HF 100개 사용하였고 모든 포지션에서 5000회 업데이트 통해 학습을 수행하였다. (a),(b)는 arbo-real의 환경에서 실험한 결과이다. (c)(d) screen 환경에서 실험한 결과이다. (a),(c)는 HFLV (Haarlike Feature Landmark Vector)알고리즘,(b),(d)는 HFLV-GD (Haarlike Feature Landmark Vector Gradient Descent)알고리즘을 각각 적용한 결과를 순서대로 나열하였다.	71

- 4.14 각 실험환경에서 상황별 수행능력 그래프. 모두 타입1 의 HF 100개 사용하였고 모든 포지션에서 5000회 업데이트 통해 학습을 수행하였다.(a) arboreal 환경,(b) screen 환경에서 여러번 반복실험을 진행한 뒤 평균값과 표준편차를 이용하여 에러바를 그렸다. 71
- 4.15 'Hall' 실험환경에서 백터맵과 수행능력 그래프. 타입1 의 HF를 100개 사용하였고 모든 포지션에서 5000회 업데이트 통해 학습을 수행하였다. (a)는 HFLV 알고리즘을 'hall' 환경에 적용한 결과이고 (b)는 HFLV-GD 알고리즘을 'hall' 환경에 적용한 결과이다. (c)의 결과는 여러번 반복실험을 진행한 뒤 평균값과 표준편차를 이용하여 에러바를 그렸다. 72
- 4.16 각 실험환경에서 상황별 수행능력 그래프. 모두 타입1 의 HF를 100개 사용하였고 5000회 업데이트 통해 학습을 수행하였다. (a), (b), (c)는 전체 포지션의 50퍼센트를 랜덤하게 선택하여 학습한 결과이다. 순서대로 'arboreal', 'screen', 'hall' 환경이다.(d), (e), (f)는 전체 포지션의 10퍼센트를 랜덤하게 선택하여 학습한 결과이다. 순서대로 'arboreal', 'screen', 'hall' 환경이다. 73
- 4.17 각 실험환경에서 상황별 백터맵. 모두 타입1 의 HF를 100개 사용하였고 5000회 업데이트 통해 학습을 수행하였다. (a), (b), (c) 집 주변 4개의 포지션을 학습한 결과의 백터맵이다. 순서대로 'arboreal', 'screen', 'hall' 환경이다.(d), (e), (f)는 집 주변 8개의 포지션을 학습한 결과의 백터맵이다. 순서대로 'arboreal', 'screen', 'hall' 환경이다. 75
- 4.18 각 실험환경에서 상황별 수행능력 그래프. 두 환경 모두 타입1 의 HF를 100개 사용하였고 5000회 업데이트 통해 학습을 수행하였다. (a), (b), (c) 집 주변 4개의 포지션을 학습한 결과의 AE 그래프이다. 순서대로 'arboreal', 'screen', 'hall' 환경이다. (d), (e), (f)는 집 주변 8개의 포지션을 학습한 결과의 AE 그래프이다. 순서대로 'arboreal', 'screen', 'hall' 환경이다. . 76
- 4.19 각 실험환경에서 상황별 백터맵. 랜덤하게 생성된 모든 타입의 HF를 100개 사용하였고 모든 포지션의 정보를 이용하여 5000회 업데이트 통해 학습을 수행하였다. (a)'a1original', (b)'arboreal', (c)'screen', (d)'hall' 환경의 순서대로 HFLV 알고리즘만 적용한 백터맵이고 (e)'a1original', (f)'arboreal', (g)'screen', (h)'hall' 는 환경의 순서대로 HFLV-GD알고리즘을 적용하였다. 77
- 4.20 각 실험환경에서 상황별 수행능력 그래프. 랜덤하게 생성된 모든 타입의 HF를 100개 사용하였고 모든 포지션의 정보를 이용하여 5000회 업데이트 통해 학습을 수행하였다. (a), (b), (c), (d)는 각각 'a1original', 'arboreal', 'screen', 'hall' 환경 순서대로 에러그래프를 나타내었다. 78

5.1	실험 로봇. (a) Pioneer (P3-DX) (b) Turtlebot (Kobuki)	82
5.2	Mobile Robot의 실험환경. 로봇의 정면을 기준으로 포스터(Landmark)가 있다. 측면과 후면에 책상과 연구실의 물품이 있는 연구실 환경이다. 창문으로 들어오는 빛은 전부 차단하였고 밝기는 일반 실내의 형광등 빛의 밝기 정도로 볼 수있다.	83
5.3	'Ourlab'환경에서 각 알고리즘 별 벡터맵. (a) 는 HFLV 방법이고 (b) 는 학습을 이용한 HFLV-GD 방법이다. (6.6)의 빨간점은 로봇이 도달해야 할 목표지점이다. HF는 Type1만 사용하였고 두 방법 모두 100개의 HF를 사용하였다.(b)의 경우 10000번의 반복학습을 통해 결과를 얻었다.	84
5.4	학습 횟수에 따른 이미지 기반 실험의 오차의 평균값 그래프. 학습이 진행됨에 따라 오차의 평균값이 낮아지는 것을 볼 수 있다.	84
5.5	같은 포지션에 각도가 다른 파노라마 이미지. 두 이미지는 같은 포지션에서 로봇의 머리가 각각 Compass를 Desired 하게 설정한 이미지와 오차를 포함한 이미지이다. 위쪽의 이미지가 Desired 헤드의 위치의 이미지이고 아래의 이미지는 Desired Compass에서 살짝 벗어난 이미지이다. 두 이미지의 각도는 약 9도정도 차이가 나게 되는데 수치상으로 비교해 볼 이미지상의 차이나는 픽셀의 비율과 차이나는 각도 비율이 일치한다. . .	86
5.6	방법1 을 설명해 놓은 그림. (a) 는 집에서의 방향을 결정하는 이미지이고 (b) 는 이후 스텝에서 로봇이 랜덤하게 방향을 설정하는 이미지이다. 위 방법처럼 탐사를 시행하고 매 스텝이 끝나게 될때마다 탐사 이미지를 수집하게 된다.	88
5.7	Landscape 기반 학습. (a) 의 빨간 점은 미리 탐사가 이루어진 위치. (b) Landmark 빨간 점이 있는 위치에서 선행 탐사로 수집된 이미지 데이터를 학습의 테스트 데이터로 이용하여 학습을 미리 진행하게 된다. 이후 (b) 와 같이 로봇이 랜덤한 방향으로 탐색을 한다. 이후에 방향 결정이 필요한 시점에서 이전에 학습된 가중치를 이용하여 방향 결정을 하게 된다. . . .	90
5.8	두 가지 방법과 PI 방법을 비교한 그래프. 방법1의 경우 오차의 범위가 방법2에 비해 크다.	91
6.1	여러가지 센서를 장착한 모바일 로봇. 여러가지 센서의 융합을 통해 발전된 형태의 내비게이션을 구현할 수 있을것으로 보인다.	97

표 차례

3.1	상황별 수행능력 테이블. 총 100개의 HF를 사용하였고 10000회 반복 학습하였고 러닝 레이트를 0.01로 고정하였다.	51
3.2	환경별 수행능력 테이블. 총 100개의 HF를 사용하였고 10000회 반복 학습하였고 러닝 레이트를 0.01로 고정하였다.	54
4.1	본 표는 여러가지 파라미터에서 환경별 수행능력 테이블. 각 환경별 수행능력을 알고리즘별로 비교 하였다.	79
5.1	방법1의 결과 테이블. 스텝은 총 10회로 제한하였고 오차 각도는 실제 로봇의 위치에서 측정한 각도값과의 오차를 나타낸다.	89
5.2	방법2 의 결과 테이블. 스텝은 총 10회로 제한하였고 Viusal Compass를 사용하여 이미지 정렬을 하였을때와 정확한 Compass를 측정하여 이미지 정렬했을때의 결과를 각각 나타내었다.	91
5.3	로봇이 실험환경에서 떨어졌을 때 결과. 반복 실험 중 로봇이 랜덤하게 움직일 때 실험환경 내에 학습이 시행된 범위안에서 의사 결정을 하게 되면 이상적인 실험 결과가 나오지만, 실험 환경에 벗어나게 되면 위와 같은 결과가 나온다.	91

요약

Haar-like Feature의 학습을 이용한 시각 내비게이션

연세대학교 대학원
전기전자공학과
김만동

모바일 로봇의 자율주행 알고리즘은 과거부터 현재까지 다양한 방식으로 연구 및 발전되고 있다. 이전의 많은 연구를 통해 실내에서의 모바일 로봇의 자율주행은 간단한 문제가 아님이 증명되었고 여러 세대에 걸쳐 모바일 로봇의 내비게이션에 대한 문제를 해결하기 위한 연구들이 발전했다. 그리하여 발전하게 된 이전의 연구중 성공적인 알고리즘은 많은 연산량과 고가의 장비들을 통해 비교적 복잡한 모바일 로봇 내비게이션을 구현하게 되었다. 하지만 인간이 풀기 힘든 형태의 문제들을 적은 세포와 간단한 뉴런 구조를 가진 곤충들은 문제에 대하여 매우 효율적인 방향으로 해결하고 이를 바탕으로 생존하며 진화해 왔다. 본 연구진은 이러한 곤충들의 성공적인 내비게이션에 영감을 얻었고 이전의 연구로 제시된 알고리즘보다 비교적 간단한 구조의 알고리즘을 통해 모바일 로봇의 성공적인 내비게이션을 구현하고 더욱더 강인한 알고리즘을 만들기 위한 과정을 연구하였다.

첫 번째로는 기존의 모델에 머신러닝 알고리즘인 경사 하강법(Gradient Descent) 알

고리즘 적용하는 실험을 진행하였다. 기존의 호밍 내비게이션을 구현했던 Snapshot 모델을 바탕으로 Haar-like Feature를 이용하여 이미지상의 랜드마크를 형성하고 이를 기반으로 시각 내비게이션으로 HFLV(Haar-like Feature Landmark Vector) 모델에 경사 하강법을 적용하였고, 이를 통해 경사 하강법으로 학습된 가중치와 Haar-like Feature를 랜드마크로 만들 때 사용되는 매칭 스코어의 관계 파악 통해 각각의 파라미터에 대한 영향력을 분석적으로 연구하였다. 또한 내림차순을 통한 매칭 스코어의 형태별 비교를 통해 매칭 스코어의 본연의 의미를 분석하였다.

이후 경사 하강법을 통한 학습을 할 때 사용되는 파라미터를 변형하며 실험을 진행하였다. 알고리즘을 구현할 때 최소로 필요한 Haar-like Feature의 개수, 포지션의 정보, 포지션의 위치등 다양한 실험 셋팅을 통해 최적의 파라미터를 찾고 이를 통해 연산량이 적고 강인한 시각 내비게이션 알고리즘을 제시하였다. 또한 학습을 통한 이미지 기반 실험으로 기존에 시각 내비게이션의 적용이 어려웠던 환경들에 적용할 수 있는 새로운 시각 내비게이션 알고리즘을 구현하였다. 이후 실제 일반적인 실험 환경에서 모바일 로봇에 알고리즘을 적용하여 실내 모바일 로봇 내비게이션을 구현하였다.

핵심되는 말 : 자율주행, 시각 내비게이션, 모바일 로봇, 머신 러닝, Haar-like Features, 경사 하강법(Gradient Descent)

제 1 장

도입

모바일 로봇의 자율주행 알고리즘은 과거부터 현재까지 다양한 형태로 연구 및 발전되고 있다. 실내에서의 모바일 로봇의 자율주행은 간단한 문제가 아님을 이전 연구들을 통해 증명되었다. 인공지능을 접목한 자율주행은 모바일 로봇뿐만 아니라 다양한 분야에서 미래 과학의 핵심 기술이다. 최근 인공지능은 의료, 사물인터넷, 금융 등 다양한 분야에 접목되어 인간의 서비스에 도움을 주고 있다. 다양한 이점을 가진 인공지능과 자율주행 방법의 융합은 많은 장점과 넓은 활용 범위를 가지고 모바일 로봇기술 분야 발전에 기여할 전망으로 보인다. 역동적으로 변화하는 주변 환경을 빠르게 인식하여 대응할 수 있다면 실시간으로 사용할 수 있는 자율주행 알고리즘이 된다. 또한 장애물을 실시간으로 감지하여 대응하면 로봇을 목표지점으로 정확하게 이동시킬 수 있을 것이다.

본 연구진은 이러한 복잡한 문제를 비교적 간단하게 해결하는 동물들의 내비게이션에 영감을 얻었다. 동물들은 음식을 구하거나 둥지로 돌아가는 등의 생존을 위하여 본능적으로 내비게이션을 행한다. 개미와 벌 같은 벌목 곤충의 경우 낮은 해상도의 시각 능력과 간단한 뉴런의 구조를 가졌음에도 불구하고 그들의 생존을 목적으로 하는 귀소 능력은 내비게이션으로서 충분히 견고하다.

본 연구진은 높은 복잡도로 이루어져 있는 기존의 탐색 알고리즘을 대체하기 위해 곤충들의 내비게이션을 모방하여 적은 연산량을 가졌지만 강인한 탐색 알고리즘에 대해

연구하였다. 강인한 탐색 알고리즘 만들기 위하여 머신러닝 알고리즘을 활용하여 동물, 곤충과 같이 환경의 변화에 따라 학습하는 능력을 기존의 시각 내비게이션 알고리즘에 적용하는 연구를 진행하였다. 본 논문에서는 이러한 시각적인 데이터를 학습하는 과정을 통해 더욱 좋은 수행능력의 새로운 알고리즘을 제안하였다.

이 장에서는 학습을 바탕으로 시각적인 내비게이션의 생체 모방 내비게이션을 소개하고 본 논문의 동기와 목적을 밝힌다.

제 1 절 생체 모방을 이용한 여러가지 내비게이션

인간의 역사에서 동물의 생체에 영감을 받아 여러 가지 문제가 해결되는 사례들이 있었다. 여러 가지 사례로 상어의 피부 돌기 모방은 작은 소용돌이를 튕겨내 추진력 손실을 줄여주는 수영선수들의 수영복을 만들어졌고, 게코 도마뱀의 강력한 접착력을 가지고 있는 나노 섬모를 모방하여 등산에 적합한 바이오 슈즈들을 만들었다. 모르포나비는 반사광을 가진 나노구조의 날개를 가지고 있다. 이러한 나노구조를 모방하여 빛의 회절 강도에 따라 색이 변하는 몰 포텍스를 만들었다. 이 밖에도 흰개미 집의 자연 냉방 시스템을 모방하여 건축된 건물 등 생체 메커니즘을 모방한 사례는 다양한 분야에서 개발되어 활용되고 있다. 이러한 생체모방 메커니즘을 이용한 해법을 'Biomimetic'이라 불리며 이는 생체모방 메커니즘을 이용하여 만들어진 장치 및 물질 시스템을 말한다[92].

'Biomimetic'의 여러 연구 분야 중 하나로 생체 모방을 이용한 탐색 알고리즘이 연구되고 있는데 이는 주변 환경을 파악하고 현재 에이전트의 위치를 파악할 때 복잡한 연산과 많은 장비가 필요한 모바일 로봇의 내비게이션의 어려움을 극복하는 데 도움이 되고 있다. 동물들의 세계에서는 인간이 풀지 못하는 문제들을 동물들의 간단한 생체 구조를 이용하여 비교적 쉽게 해결한다. 계절마다 수천킬로 비행하여 이동하는 철새와 바다를 넘어 다시 돌아오는 물고기와 같은 사례가 많이 있다[73].

개미와 같은 곤충의 경우 다른 동물들보다 적은 세포 수와 간단한 뉴런의 구조에도 불과하고도 먹이 탐사 등 외부 탐사 이후 성공적으로 둥지로 돌아온다. 개미가 음식을 구

하고 동지를 옮기는 등 외부 활동을 하고 다시 동지로 돌아오는 과정은 생존에 필수적인 요소이다. 개미뿐만 아니라 곤충들의 이러한 능력은 인간의 만드는 공학적인 방법들의 한계를 뛰어넘는 경우가 많다. 기존의 연구원들은 모바일 로봇에 적용할 수 있는 생체모방 기반의 효율적인 귀소 내비게이션을 구현하였다. 초기의 내비게이션들은 주로 주변 환경을 측정하고 주변 환경에 대한 정보를 맵핑하는 것에 중심을 두었다. 그런 다음 해당 환경에 대한 지도를 참조하여 에이전트가 현재 자신의 위치를 파악하고 목표 위치로 이동하도록 하였다. 하지만 이러한 과정에서 정확한 기하학적인 지도를 만들어 내려면 고가의 장비와 복잡한 알고리즘들이 필요하다[104].

최근 연산의 복잡도가 비교적 낮고 간단한 프로세싱으로 이루어져 있는 내비게이션을 구현하기 위하여 다양한 생체 모방 내비게이션 알고리즘이 연구되었다. 그중 이러한 고가의 장비를 사용하지 않고 내비게이션을 하는 방법중 PI (Path Integration) 를 하는 방법이 있다. PI는 사막같이 주변이 빠르게 변하는 환경에서 사막 개미가 사용하는 내비게이션이다. 하지만 PI 알고리즘 같은 경우 로봇에 적용할 때 스텝이 늘어날수록 오차가 누적되는 문제를 가지고 있다[17].

이러한 치명적인 문제를 가진 물리적인 측정 방법에 의한 내비게이션의 한계를 극복하기 위해 여러가지 내비게이션 알고리즘이 제시되었다. 그중 시각적인 정보를 이용한 호밍 내비게이션은 모바일로봇이 집의 위치에서 저장된 이미지를 이용하여 집이 아닌 위치로 탐색을 하였을 때 다시 집으로 돌아올 수 있게 하는 알고리즘이다. 이러한 알고리즘의 전제는 곤충이 공간의 위상학 적인 표현을 사용한다는 것과 두뇌에 저장 공간이 적을 것이라는 점이다. 이러한 시각적인 정보를 이용하는 호밍내비게이션 연구 중 꿀벌의 내비게이션을 스냅샷 모델로 모델링이 제안되었다[9]. 스냅샷 모델 알고리즘에서는 동지의 이미지와 현재 위치의 이미지의 유사도 측정으로 서로 비교하여 원점 복귀 방향을 결정하는 방법이다.

스냅샷 모델은 단 2개의 스냅샷 이미지만 가지고 있으면 원점 복귀 방향을 정할 수 있는 특징이 있는데 이러한 특징은 적은 연산으로 호밍 내비게이션의 가능성을 제시한다. 스냅샷 모델은 두 이미지 간의 시각적인 차이를 줄여나가며 방향을 탐색하게 된다. 또한

두 이미지 사이에 특징적인 공통 랜드마크가 있다면 효율적으로 에이전트를 안내할 수 있다. 만일 집과 현재 위치 사이에 어느 정도 거리 차이가 있다면 두 이미지는 위상차이에 의해 신호적인 차이가 생긴다. 반대로 두 이미지 사이에 거리 차이가 없다면 서로 비슷할 것인데 본 연구진은 스냅샷 모델과 같은 생체모방을 알고리즘을 기반으로 하여 시각적인 단서와 신호의 차이를 이용한 여러 가지 귀소 내비게이션들을 연구할 수 있었다.

제 2 절 연구 동기

본 논문은 여러 가지 내비게이션 연구를 바탕으로 진행되었다. 이전의 내비게이션 관련 연구에서 여러 가지 방법의 알고리즘들을 제시하였다. 하지만 대부분의 모바일로봇 내비게이션은 비교적 복잡한 연산량과 값비싼 장비들이 필요하였다. 이에 따라 조금 더 적은 연산량과 복잡도가 낮은 시스템의 내비게이션 알고리즘이 필요로 하였다.

본 연구진은 이러한 문제들을 해결할 알고리즘으로 벌과개미와 같은 적은 세포 수와 간단한 뉴런 구조를 가진 곤충들에게 영감을 얻었다. 본 논문에서는 개미가 사용하는 여러 가지 내비게이션을 구현하고 비교하여 더 높은 정확도와 효율적인 알고리즘을 가지는 정확도를 높이는 내비게이션을 제시한다. 정통적으로 고가의 장비와 복잡한 알고리즘에 비해 구현에 있어 가격이 저렴하고 효율적인 내비게이션은 대표적으로 PI (Path Integration) 방법과 시각 정보 (Visual Information) 를 이용한 내비게이션이 있다. 전통적으로 사용되는 PI 방법은 여러 로봇과 이미지 기반 실험에 적용하여 사용되어 왔다. 하지만 실제 적용하면서 PI의 경우 로봇의 이동경로가 복잡해 짐에 따라 누적되는 에러의 총량이 늘어나는 문제가 있다. 기존의 이러한 문제들을 해결하기 위하여 다양한 형태의 알고리즘을 제시하였다. 그중 시각 정보를 이용한 내비게이션은 두 개의 스냅샷만 비교하여 집으로 가는 방향을 찾기 때문에 기존의 에러가 누적되는 문제들의 해결점이 되었고, 본 연구진은 최적화 알고리즘을 통학 학습을 적용하여 시각 내비게이션을 강화할 수 있는 가능성을 찾게 되었다. 이후 강화된 알고리즘을 여러가지 환경에 적용하여 실험하고 본 알고리즘을 실제 모바일로봇에 적용 수 있도록 연구를 진행하였다.

제 3 절 연구 목적

생체모방 (Bio-inspired) 에 영감을 받아 인간이 풀지 못한 어려운 문제들을 풀은 사례들이 많다. 성공적인 능력을 가진 동물들에 영감을 얻어 그것을 모방하여 구현하는 것을 최종 목적으로 가지고 있다. 하지만 자연의 동물들을 그대로 모방하여 알고리즘을 개발하고 이를 실제 로봇에 적용하는 것은 쉽지 않은 일이다. 그 이유는 오랜 세월 진화해온 곤충의 능력을 우리의 공학적인 메커니즘으로 재현하는 것은 결코 쉬운 일이 아니기 때문이다. 한가지 예로 개미가 시각적인 단서를 이용하여 집으로 향하는 방향을 추정하는 것을 공학적으로 구현하려면 개미의 감각기관을 대변하는 여러 가지 센서, 기계적인 요소 그리고 학습이 필요하기 때문에 생체모방의 장점들을 간단하고 쉽게 따라 하는 건 어느 정도 한계를 가지고 있다.

하지만 최근에 더욱더 발전된 최적화 알고리즘과 모바일로봇의 알고리즘을 적용하여 이전에 비해 새로운 형태의 알고리즘 제시할 수 있었다. 본 연구진은 이전에 연구된 환경적 제한을 가지고 있던 알고리즘을 최적화시키고 여러 가지 분석적인 검증을 통해 여러가지 환경에서도 적용할 수 있는 강인한 알고리즘으로 발전시켰다. 이를 통해 이전의 연구된 알고리즘들의 한계점을 극복하여 실제로 모바일로봇에 적용하고 여러 가지 상황에서도 높은 정확도를 가진 내비게이션 구현을 목표로 하였다.

제 4 절 논문의 구성

다음과 같은 내용으로 생체 모방에 기반을 둔 내비게이션 알고리즘에 대한 연구를 진행하였다.

챕터1: 도입 및 동기,목적 제시 본 논문 도입으로서 전체적인 소개를 통해 연구의 동기와 목적을 밝힌다.

챕터2: 다양한 생체모방 알고리즘과 내비게이션에 관해서 소개 여러 가지 생체모방 알고리즘과 본 연구진이 연구한 알고리즘과 유사한 종류의 시각 내비게이션 알고

리즘에 대해서 소개한다. 시각적 내비게이션 알고리즘들이 가지는 문제와 문제를 해결하기 위한 여러 가지 방법들을 제시한다.

챕터3: 경사 하강법 (Gradient descent) 을 이용하여 마스크를 학습 스냅샷 모델을 기반으로 Haar-like Feature를 이용하여 랜드마크를 생성하여 신호의 차이를 분석하며 포지션별로 목표지점을 찾아가는 방법에 경사 하강법을 적용하여 이미지 기반 실험을 하였다.

챕터 4: 여러 가지 조건의 환경과 파라미터에 대하여 알고리즘 검증 경사 하강법을 제시하여 만든 알고리즘이 여러 가지 환경과 조건에서 잘 동작하는지 실험하고 수행능력별 검증을 하였다. 경사하강법 가지는 이점과 이전의 알고리즘들의 문제를 경사 하강법을 통해 돌파하였는데 이를 파라미터와 환경별로 적용이 가능한지 검증하였다.

챕터 5: 실제 환경과 모바일 로봇에 적용 기존에 있는 데이터셋 뿐만 아니라 본 연구진들의 실제 환경에서도 알고리즘을 모바일로봇에 적용하고 여러가지 알고리즘과 비교 분석한다.

챕터 6: 결론 및 Future work 본 논문의 총결론과 이후에 진행해야 할 Future works들에 대해서 제시하고 여러 가지 확장주제에 대해서 논한다.

제 2 장

시각 내비게이션에 대한 동향

시각적 단서를 기반으로 원정 항법을 이용하는 많은 동물과 곤충들이 있다. 그중에서 개미와 벌은 집을 떠나 탐색을 하고 다시 성공적으로 집으로 돌아오는 벌목 곤충이다. 개미와 벌은 간단한 뉴런의 구조와 적은 숫자의 세포를 가졌음에도 불구하고 성공적으로 집으로 돌아오는 성과를 보여주었다. 현대에 연구되고 있는 모바일 로봇의 내비게이션은 간단하게 풀 수 있는 문제 아니라는 것은 여러 가지 연구를 통해 증명되었다[67].

본 연구진은 내비게이션 분야에서 기존의 복잡도가 높은 알고리즘을 개선하는 방안으로 곤충들의 귀소 내비게이션을 연구한다. 본 장에서는 곤충들의 귀소 내비게이션뿐만 아니라 내비게이션에 필요한 인공지능을 이용한 인식 알고리즘 및 실내에서의 모바일 이동로봇의 자율주행 알고리즘과 내비게이션 알고리즘의 전반적인 동향을 소개하고 곤충들의 내비게이션 중 대표적인 예로 꿀벌의 Snapshot 모델을 주로 소개하려 한다[9].

많은 동물과 곤충들은 다양한 방법을 이용하여 내비게이션 한다. 또한 곤충들은 다양한 이유로 내비게이션이 중요하게 여겨진다. 동물들과 곤충들은 음식을 구하거나 계절에 따라 서식지를 옮기는 과정에서 다양한 내비게이션을 사용한다. 또한 내비게이션을 사용하는 것이 생존에 큰 영향을 끼치게 된다. 따라서 동지를 떠나서 여러 가지 활동을 마치고 다시 동지로 돌아가는 항법은 직접적으로 생존과 연결된다[73]. 여러 가지 내비게이션 방법중 많은 연구가 진행되고 있는 방법은 시각을 이용하는 방법이다. 그 중 고안된

모델 중에 하나인 스냅샷 모델은 시각을 이용한 방법이다. Snapshot 모델은 꿀벌이 귀소를 할 때 사용하는 방법으로 연구되었는데 꿀벌은 등지와 현재의 위치의 신호 차이를 최소화하고 이를 이용하여 목표 지점으로 복귀하는 방향을 찾는다[86, 112].

제 1 절 AI 네비게이션 알고리즘

최근 여러 가지 연구 분야에서 인공지능이 사용되고 있다. 인공지능은 점점 더 다양하고 복잡한 문제들을 해결하기에 적합하도록 발전하였다. 더 다양한 문제들을 해결하기 위해 AI 알고리즘들은 복잡하고 견고한 알고리즘으로 발전하고 있다. 초기 모델로 제시되는 Perceptron은 XOR 및 비선형적인 분류는 할 수 없었다. 하지만 이후 제시된 다중 Perceptron은 XOR뿐만 아니라 비선형적인 분류까지 할 수 있게 되었다. 이후에 Support Vector Machine (SVM) 같은 분류기가 제시되었는데 SVM은 전통적인 분류기들 보다 상대적으로 높은 성능을 보였다[34]. 하지만 신경망 이론을 근거로 하여 만들어진 복잡한 네트워크들은 이러한 SVM보다 높은 성능을 보였다.

최근 특징 검출과 같은 다양한 분야에 AI 알고리즘들이 활용되고 있다. AI 알고리즘을 통해 더 높은 정확도를 얻을 수 있다. 대표적인 예로 얼굴 인식 알고리즘이 있는데 센서에서 획득한 이미지 데이터를 AI 알고리즘을 통해 특징 검출을 통해 더 정확한 얼굴을 인식할 수 있다. 비슷한 방법으로 번호판, 표지판, 자동차 등을 감지하는 애플리케이션에도 활용된다[15, 4, 85, 103]. AI 알고리즘은 시각적인 신호뿐만 아니라 음성 신호에 대해서도 비슷한 방법으로 활용된다. 인공지능은 지속해서 발전하고 있으며 소프트웨어뿐만 아니라 하드웨어에도 적용된다. 최근 이러한 인공지능 기술들이 시각을 이용한 네비게이션에도 이용되고 있다[38, 14, 93, 48, 13].

최근 모바일 로봇에 대한 여러가지 연구가 진행되고 있다. 곤충들은 시각적으로 획득한 이미지 안에서 지정해 놓은 랜드마크를 이용하여 집으로 귀소하는 네비게이션을 행한다. 이러한 귀소 네비게이션을 할 때 지속해서 랜드마크를 추적하고 현재 이미지와 비교한다. 이미지를 비교하고 피쳐를 뽑아내는 과정에서 AI 알고리즘을 사용한다[36,

23, 38, 24].

1.1 AI 알고리즘을 이용한 내비게이션 : SLAM

Simultaneous Localization and Mapping (SLAM) 은 가장 잘 알려진 연구 중 하나이다. 이동하는 경로를 기억하고 이동 위치에 따라 업데이트하고 지도화한다. 초기의 SLAM은 칼만 필터를 기반으로 만들어졌다. 칼만 필터는 가우스 분포에 따른 문제들이 제기되었다. 이를 해결하기 위해 확장 칼만 필터 및 무감각 칼만 필터가 연구되었다. 이후 Monte Carlo 등을 이용한 맵핑 알고리즘들이 연구되었고 이후 이러한 알고리즘들이 SLAM을 더욱 발전시켰다. (예 : Fast SLAM, Fast SLAM, SLAM, SE SLAM)

하지만 Simultaneous Localization and Mapping (SLAM) 은 곤충들이 사용하는 방법에 비해 너무 복잡하다. 또한 SLAM을 구현할 때 사용되는 데이터 수집 장비는 비교적 가격이 높다. 한가지 예로 'Velodyne'은 실시간 3D 데이터를 취득할 수 있는 고가의 3차원 거리 센서이다. 소형 모바일 로봇뿐만 아니라 자동차에서도 여러 가지 애플리케이션에 이용될 수 있다. 좋은 성능의 센서를 사용하지 않는 경우에 복잡성이 높은 SLAM 알고리즘을 컴퓨터에 적용하기에는 여러 가지 문제가 발생한다. 우리는 매우 복잡하고 높은 비용의 센서를 사용하는 대신 간단한 알고리즘을 곤충의 귀소 알고리즘으로 사용하는 것을 연구했다.

제 2 절 인식 알고리즘

객체검출 및 인식에 사용되는 알고리즘들은 영상 검색 과 얼굴인식등 여러분야에서 응용되고 지속적으로 연구되어 왔다. 객체 검출 및 인식 알고리즘들은 실시간 으로 사용되기 위하여 빠른 처리속도와 간단한 연산이 중요하데 이러한 조건은 생체모방 연구에서도 중요하게 다루어진다. 이러한 조건에 부합한 분류기로 Haar-like feature가 있다[109].

Haar-like feature는 여러가지 부스팅 알고리즘과 함께 다양한 인식 알고리즘에서 사용되는 분류기 이다[75]. 또한 생체모방 내비게이션 에도 부스팅 알고리즘과 함께 연구되

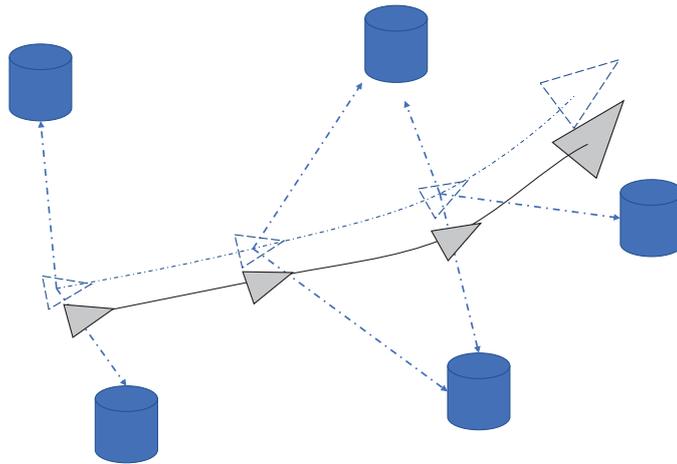


그림 2.1: Simultaneous Localization and Mapping (SLAM) 알고리즘의 모식도



그림 2.2: Haar-like Features의 여러가지 형태

었다[6]. Haar-like Feature는 시각의 양극세포를 모델링한 형태이다. 그림3.2은 여러가지 Haar-like Feature의 형태이다. 또한 Haar-like Feature는 약한 분류기이다. 본 연구진은 Haar-like feature의 간단한 연산 방법과 빠른 처리 속도를 이용하여 생체모방 내비게이션에 이용 하였다.

제 3 절 동물들의 다양한 내비게이션

동물들은 생존을 위해 여러 가지 활동을 한다. 동지를 만들고 먹이를 찾기 위해 탐험을 한다. 또한 계절별로 다른 서식지를 찾기 위해 떠나고 돌아오기를 반복한다. 동물들의 활동은 생존을 위하여 필수적이며 집을 떠나고 다시 돌아오는 능력 또한 생존을 위해 필수적인 능력이다. 동물들은 시력, 후각, 청각 등 여러 가지 감각을 사용하여 집에 가는 방향을 찾는다.

동물들은 사람에 비하여 작은 두뇌로 집을 돌아오는 능력을 갖췄다. 동물들의 귀소 내비게이션은 로봇에도 적용하여 자율주행 알고리즘에 대해 활발한 연구가 진행되고 있다. 기존의 모바일로봇의 알고리즘들은 복잡성이 크다. 하지만 여러 동물의 내비게이션은 기존의 알고리즘에 비해 단순한 특징이 있다. 예를 들어 페로몬을 활용하여 집을 돌아오는 개미가 있다. 이러한 강점 덕분에 biomimetics에 대해 많은 연구가 이루어지고 있다[17, 72, 69, 106, 98, 66].

최근에 2017년에 연구된 내용 중 시각적 단서에 기반한 개미 탐색에 관한 연구는 수집된 음식을 운반하는 과정에서 개미가 자리를 옮길 때 시각적인 신호가 미치는 영향을 연구하였다 [122, 97, 65, 95]. 다른 예로 말벌은 집의 위치에서 태양의 광선을 이용하여 시각적 정보를 획득한다. 말벌은 시각적으로 제한이 생길 경우 복귀 경로에 대한 오차가 생긴다. 이러한 동물들이 가진 귀소 내비게이션의 대표적으로 방법들이 있다. 본 연구진들은 이러한 생체능력에 영감을 받았다.

3.1 PI (Path Integration) 방법

사하라 사막에서 사막 개미는 몸의 크기에 비하여 장거리 여행을 한다. 하지만 이런 종류의 개미는 일반적으로 다른 개미들이 사용하는 페로몬을 사용할 수 없다. 사하라 사막은 땅의 온도가 페로몬을 증발시키기 때문이다. 또한 사막의 가변적 특성은 짧은 시간 안에 시각 정보가 달라지게 할 수 있다. 예) 모래폭풍 등 사막 개미는 PI (Path Integration) 를 이용하여 집으로부터 이동 경로를 추적하고 축적된 데이터를 사용하여 집으로 돌아온다. 이러한 방법으로 위치 및 방향을 찾은 다음 출발 위치를 기준으로 이동한 이전 위치부터 방향을 고려한 거리를 누적하여 목표 위치까지 도달한다. 이처럼 사막 개미는 이동

중에 위치와 방향을 모두 갱신한다. 그런 다음 마지막 순간에 누적된 총량을 확인하고 역방향을 찾는다.

PI를 사용하는 다양한 동물에 관한 여러 가지 연구가 있다[52, 108, 30, 66, 76, 123]. 또한 이 방법은 방위각 변화에만 초점을 두기 때문에 수직 변화에 취약하다. 따라서 3차원 Path Integration 와 관련된 연구는 한계를 극복하기 위한 방법으로 제안된다[115]. 다른 이들은 신경망 구조를 사용하여 귀소 내비게이션의 기본적인 원리를 증명하려고 시도하였다[33, 90, 105, 47, 116, 12, 54].

그러나 PI는 오류를 누적시킬 수 있고, Path Integration 지점의 출력은 축적된 오류로 인해 에이전트를 매우 다른 방향으로 이끌 수 있기 때문에 경로 통합은 매우 정확한 측정 모델을 요구한다. 따라서 PI의 한계를 극복하는 방법이 많이 연구되고 있다. 다양한 방법으로 오류가 축적되는 것을 줄이기 위한 연구가 수행되었다[91, 22, 74].

경로 통합과 시각적인 정보를 이용하는 것이 서로 다른 장점을 가지고 있고 서로를 보완할 수 있기 때문에 일부 연구는 관련 거리에 따라 두 가지 방법을 결합한다. 비전 기반 경로 통합에 대한 연구가 최근에 실시되었다. 예를 들어 [113]는 개미가 PI를 만들기 위해 초기 궤도에서 시각적 단서를 사용한다고 가정한다[19, 21]. 벌 등 다양한 종에 대해서도 유사한 연구가 진행되었다.

$$\begin{aligned}
 P_{k+1} &= P_k + \delta = (v+, \delta v, r+, \delta r) \\
 direction &= \tan^{-1}(v+, \delta v, r+, \delta r)
 \end{aligned}
 \tag{2.1}$$

3.2 RF (Route Following) 방법

RF (Route Following) 는 이동 중에 캡처된 이미지를 사용하여 경로를 학습하는 것을 말한다. 경로 통합과는 달리 RF 방법은 에이전트가 경로에서 장면을 반복적으로 학습할 수 있도록 한다. 그런 다음 학습된 결과에 따라 시각적 단서를 사용하여 학습된 경로를 따라 집으로 돌아오게 된다. 과거에 지나온 경로에 대한 학습을 통해 귀소 내비게이션을 한다. 경로 통합과는 반대로 이 방법은 학습 단계 [94]에 시간을 투자해야 한다. 따라서 학습

단계에 오류나 누락이 있는 경우 정확한 귀소 내비게이션을 수행할 수 없다. 하지만 RF는 학습하는 부분에서 개미나 벌의 실제 모형을 만들어내기 위한 알고리즘을 사용하는 실험에는 적합하다[5, 114, 6, 118, 41, 64].

최근 연구에서는 RF의 여러 가지 단점을 극복하기 위하여 '스카이라인' 정보(이미지 내 랜드마크와 하늘 사이의 경계)를 경로마다 적용하는 연구가 있었다[117, 89]. RF와 PI는 장거리에서도 사용할 수 있지만 방법마다 짧은 거리에서도 사용할 수 있다. 그러나 PI는 주변을 관찰하고 축적하기 위해 여러 개의 감각기관을 사용해야 한다는 단점이 있다. RF의 경우 비슷한 단점이 있다. 에이전트가 이전에 학습되지 않은 새 경로를 발견하면 에이전트는 목표 경로에서 벗어날 수 있다는 점이다.

위 방법의 공통적인 단점 및 여러 내비게이션 알고리즘들이 가지는 단점 중 하나는 에이전트가 경로를 완전히 익히지 않으면 그 장소에 되돌아갈 수 없다는 것이다. 최근 연구에서는 RF 및 다른 내비게이션 알고리즘들의 이러한 단점을 극복하기 위하여 새로운 방법들이 제시되어왔고 시각적인 정보를 이용한 경로추적에 대한 다른 연구가 수행되었다. 예를 들어 날아다니는 곤충의 경로를 비행 로봇에 적용하는 연구는 다양한 높이의 스냅샷을 이용하여 이미지 사이의 거리를 계산하여 경로에 대한 최적의 경로를 찾는 방법 등이 있다[35].

제 4 절 다양한 지역 귀소 내비게이션

지역 귀소 내비게이션은 시각적인 정보를 활용한 내비게이션 중 가장 단순한 형태로 만들어진다. 시각적 정보만 사용하여 현재 위치에서 목표 위치로 가는 방향을 찾는다. 따라서 지역 귀소 내비게이션 방법은 시각적 단서만 사용하여 귀소 내비게이션 하는 것을 목표로 한다. 위에서 언급한 것과 같이 우리는 곤충이 작은 뇌와 비교적 간단한 뉴런의 구조 그리고 적은 세포 수를 이용하여 시각적인 정보 처리를 한다는 것을 고려해야 한다. 우리는 개미가 효과적인 데이터 처리와 효과적인 귀소 내비게이션 능력을 갖추고 있다고 가정한다. 곤충의 효율적인 시각적 내비게이션 분석을 위한 여러 가지 연구가 있

는데 각 모델의 생물학적 근거를 찾는 데 초점을 맞춘 연구들과 그것을 실제 내비게이션 애플리케이션에 적용하는 두 가지 범주로 나누어진다[1, 18, 124].

본 논문에서 소개하는 지역적인 시각 정보를 이용한 내비게이션의 스냅샷 모델이다[9]. 스냅샷 모델은 벌의 귀소 내비게이션에 영감을 받았다. 귀소할 집 위치의 이미지와 현재 위치에서의 이미지의 시각적 단서 2개만 사용하여 귀소 방향을 쉽게 찾을 수 있다. 벌은 이 두 스냅샷을 비교하여 귀소 방향을 계산하기 위한 두 스냅샷 사이의 차이를 찾는다. 집 이미지와 현재 이미지에서 물체 위치의 차이는 광학을 통한 측정으로 이루어지고 귀소 방향을 찾는 데 사용할 수 있다. 이 방법은 대상과 현재 위치의 스냅샷 모두 서로 일치시킬 수 있는 공통의 시각적 정보를 가지고 있으므로 고정 영역 내에서 이 방법을 사용할 수 있다는 전제를 따른다. 따라서 이론적으로는 스냅샷 사이의 거리가 한계 이상으로 너무 멀어지게 되면 에이전트(bee)는 귀소 방향을 찾을 수 없다 [107].

우리는 학습이나 시각 이외의 다른 센서 정보가 필요 없고 귀소 위치와 현재 위치의 두 스냅샷만을 이용하여 귀소 내비게이션을 구현할 수 있다. SLAM과 같은 다른 내비게이션들과 달리 본 연구진의 방법은 다른 센서의 정보 없이 오직 시각정보에만 의존한다. 본 장에서 우리는 오직 시각적인 정보만을 이용하여 귀소를 하는 내비게이션을 구현한다.

4.1 Snapshot 모델

스냅샷 모델은 꿀벌의 항해에 적용할 수 있는 생체모방 탐색 알고리즘 중 하나이다[9, 10, 77, 114]. 스냅샷 모델은 시각적인 정보를 이용한 탐색의 시초이기 때문에 다수의 유사한 모바일로봇의 시각적 호밍 내비게이션 알고리즘이 스냅샷 방법을 기반으로 개발되었다. 스냅샷 모델의 핵심적인 기술은 에이전트가 귀소 내비게이션을 수행할 때 사용하는 시각적인 정보는 첫 번째 스냅샷 과 가장 마지막 스냅샷만 기억하고 사용한다는 것이다. 현재 위치의 시각적인 정보를 귀소할 집 위치의 스냅샷과 유사성을 비교하여 에이전트는 귀소할 집 위치에 도달하기 위해 어떤 방향으로 이동할지 결정할 수 있다. 스냅샷은 파노라마 이지미 형식이고 파노라마 이지미상의 방위각에 현재 위치를 포함한다[84]. 스

냅샷 모델에서 파노라마 이미지를 사용하는 이유는 일부 연구자들은 실제 생물체들이 파노라마 같은 정보를 사용하여 내비게이션 할 것을 주장한다[124, 53, 16].

만일 집 위치와 현재 위치의 사이의 거리가 한계 이상으로 멀지 않다고 가정할 때 두 스냅샷 간에 공통점이 있는 경우 간단한 비교로 집 방향을 가리키는 홈 벡터를 생성할 수 있다. 그러나 간단한 스냅샷에는 랜드마크의 깊이 정보가 없고 컬러 픽셀만 있으므로 홈 벡터의 길이는 현재 위치와 함 사이의 실제 거리와 같지 않다는 문제 있어 거리정보를 반영할 수 가없다. 스냅샷 모델은 몇 가지 간과하면 안 되는 몇 가지 유의할 단점이 있다. 스냅샷 간의 변동이 심각하게 크지 않을 경우 에이전트가 성공적인 호밍 탐색을 수행하기 위해 첫 번째와 마지막 장면 외에는 사이의 다른 장면을 기억할 필요가 없다는 것이다.

하지만 스냅샷 모델은 주변 환경이 역동적이게 달라짐에 대한 약점을 가지고 있다. 첫 번째는 외부 요인에 의해 생기는 환경의 변화 때문이다. 이 경우 스냅샷 사이의 심각한 오류가 발생할 수 있다. 기본 스냅샷 방법을 사용하기 위해서는 환경에ダイナミック 변화가 없다고 가정할 필요가 있다. 두번째는 움직임에 의해 생기는 환경의 변화와 관련이 있는데 에이전트의 이동이 작으면 스냅샷 간에 큰 변화가 없기때문에 스냅샷간의 비교를 위해 어느정도 이동에의한 변화가 있어야 한다. 하지만 에이전트가 집에서 한계 이상으로 먼 거리를 이동하거나 특정 지역에서만 관찰되는 특수한 랜드마크 배치가 있는 경우 스냅샷 사이에 공통된 특징이 없게 되고 성공적인 귀소 내비게이션을 할 수 없다.

따라서 기본 스냅샷 모델을 사용하려면 홈 주변으로부터의 거리에 제한이 있어야 하는데 두 사이에 시각적 정보의 허용 가능한 변화가 있는 범위 내로 이동의 한계를 정해야 한다. 또한 장애물에 의한 음영지역이 없고 이동 중 역동적인 변화가 없는 경우 에이전트는 두 스냅샷 사이의 공통 요인을 찾아 스냅샷 모델을 사용하여 성공적인 호밍 내비게이션을 수행할 수 있다. 생체모방을 기반으로 하는 스냅샷 모델은 자연 속의 생물들과 유사한 적은 저장공간과 연산이 필요하다. 이러한 점에서 스냅샷 모델은 단순하고 명확한 특징을 가지며 이것은 관련된 여러 가지 연구의 사용되었다. 최근 스냅샷 모델을 이용하여 다양한 유형의 애플리케이션에 적용된 연구가 있다. 우리는 이 스냅샷



(a)



(b)

그림 2.3: 스냅샷 모델의 스냅샷 이미지. (a)현재 위치 (b)집 위치 ([80]의 논문에서 발췌

알고리즘을 우리의 방법에 적용하고 개선할 예정이다[39, 88, 26].

4.2 ALV (Average Landmark Vector) 모델

스냅샷 모델을 이용하여 랜드마크를 생성하는 알고리즘 중 하나인 ALV (Average Landmark Vector metho) 는 여러 알고리즘의 기반이 되었다. ALV는 시각적인 정보를 이용한 내비게이션 방법을 모델링한 기반으로 한 가장 잘 알려진 방법이다[60]. ALV방법은 단계별로 진행된다. 첫 번째는 정렬이다. 방향에 대한 절대적인 기준을 세울 필요가 있기 때문에 정렬을 가장 먼저 한다. 곤충은 편광 광선을 사용하여 방향에 대한 정렬하는 것으로 알려져 있다.

곤충의 비전은 해상도가 낮기 때문에 우리가 쉽게 따라 할 수 있는 그룹화 메커니즘을 가지고 있다. 어떤 알고리즘은 하나의 픽셀을 랜드마크로써 사용한다. 추출 후 최종 벡터는 랜드마크 벡터와 일치하는 벡터가 된다. 기본적인 전제는 각 랜드마크와의 거리를 알 수 없다는 것이다. 그러한 이유로 길이가 동일한 각 랜드마크를 가리키는 단위 벡터를 생산하는 것이다. 그다음에는 단위 벡터의 합이 ALV이고 호밍 방향은 두 ALV 사이의 뺄셈으로 계산할 수 있다.

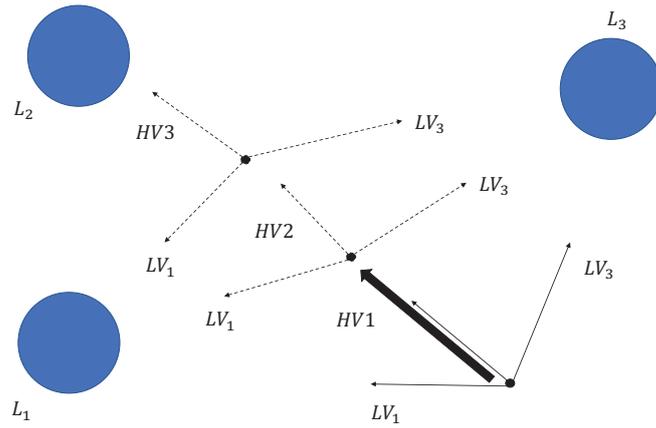


그림 2.4: ALV(Average Landmark Vector)를 이용한 방법.

식2.2은 대치어 없음 알고리즘을 요약하는 공식이다. 랜드마크 벡터는 망막 이미지의 랜드마크를 가리키는 벡터로 정의된다. 각 벡터에는 동일한 거리의 가정하에 단위 거리가 있다. 거리 정보가 없기 때문에 랜드마크의 각도 위치만 고려된다. 랜드마크 벡터는 평균 랜드마크 벡터(ALV: Averaged Landmark Vector)라고 불리는 랜드마크 벡터를 형성하고 현재 위치의 스냅샷에서 생성된 ALV와 집 스냅샷에 생성된 ALV를 빼면 호밍 방향을 결정할 수 있다. 더 발전된 모델들은 로봇 실험으로 테스트 되었다. [96, 39] ALV의 단순한 특성으로 인해 지역에 한정되는 시각 정보를 이용한 내비게이션으로 가장 잘 알려진 알고리즘이 되었다.

$$\overrightarrow{HV} = \overrightarrow{ALV_H} - \overrightarrow{ALV_C} = \frac{1}{N} \sum_{i=1}^N \overrightarrow{LV_{Hi}} - \frac{1}{N} \sum_{i=1}^N \overrightarrow{LV_{Ci}} \quad (2.2)$$

4.3 ACV(Average Correctional Vector) 모델

여러 가지 내비게이션 알고리즘 중 ALV를 기반으로하는 ACV (Average Correctional

Vector method) 가 있다. ACV 방법은 ALV의 수정된 버전이다. [96, 111] 이 방법 또한 랜드 마크를 사용하여 귀소 내비게이션을 구현한다. 그러나 단위 벡터와 랜드 마크 벡터는 홈위치 에서 사용하지 않습니다. 대신 각도 차이를 사용하여 현재 위치에서 교정 벡터를 만든다. 교정 벡터의 방향은 각도 차이를 감소시키는 각이고. 보정 벡터의 길이는 각도 차이의 크기와 같다. 현재 위치에서 생성된 벡터의 크기는 가중치의 기능을 가지며, 보정된 각도는 개정의 기능을 갖는다. 이 수정 된 방법은 본질적으로 스냅 샷 모델과 유사하며 아래 정의다.

$$\begin{aligned}
 |\overrightarrow{CV}_i| &= |\theta_i - \alpha_j| \\
 \angle \overrightarrow{CV}_i &= \begin{cases} \alpha_i + 90^\circ & \text{if } \theta_i < \alpha_i \\ \alpha_i - 90^\circ & \text{if } \theta_i \geq \alpha_i \end{cases} \\
 \overrightarrow{HV} &= \sum_{i=1}^N \overrightarrow{CV}_i
 \end{aligned} \tag{2.3}$$

4.4 DELV(Distance Estimated Landmark Vector) 모델

거리 추정 랜드 마크 벡터 DELV (Distance Estimated Landmark Vector) 모델은 다른 랜드 마크 기반 원점 탐색 알고리즘이다. 다른 두 가지 방법과 달리 랜드 마크 사이의 각도 차이를 직접 사용하지 않는다. 그러나 하나의 이미지만 사용하여 견적을 수행할 수는 없다. 따라서 그것은 본질적으로 하나의 위치에서 2개의 이미지를 사용하여 방위각 변화를 관측하기 위해 진행 방향을 따라 추가 이동을 한다. 이후 다음 삼각 함수를 사용하여 각 표식에 대한 예상 거리를 계산할 수 있다. 왜냐하면 두 그림은 직선을 따라 움직이는 추가 운동에 의해 정렬되기 때문이고 이 기능을 사용하면 추정된 거리와 함께 각도 위치를 갖는 랜드 마크 벡터를 결정할 수 있다.

다음 단계는 참조 나침반 없이 랜드 마크의 거리 추정과 함께 랜드 마크 탐색 방법을 시연하다. 나침반 정보를 랜드 마크 정렬 순서로 바꾸면 에이전트의 탐색 방법이 성공적인 원점 수행 성능을 나타낸다. DELV 방법은 우리의 이전 연구에서 제안되었으며,

DELV에 적용된 양자화 된 거리의 효과가 기술되었다[120]. DELV는 아래의 식과 같이 지정된다. DELV는 거리 팩터를 사용하여 방향을 보다 정확하게 만든다. 그리고 이것은 이동 방향을 계산을 수행하는 랜드 마크 벡터 계산과 잘 맞습니다. 또한 각 측정에서 두 번 측정 오류가 발생할 수 있다. 측정 오류를 줄이기 위해 양자화 된 정보를 사용하는 연구가 있다[120].

현재 색 정보를 사용하여 피쳐 추출 및 추가 이동을 사용하지 않고 DELV를 개선하기 위한 연구를 수행한다. DELV 모델은 처음에는 비전 센서를 사용하여 테스트되었으므로 선택된 랜드 마크에 대해 추가 거리 추정 프로세스가 필요하다[120, 119]. 시각 정보를 기반으로, 일련의 로봇 위치에 대해 획득 된 스냅 샷 간의 랜드 마크 위치 변동은 랜드 마크 거리 계산을 용이하게한다. 깊이 센서의 정보를 활용하여 각 랜드 마크까지의 거리 정보를 랜드 마크 방향과 함께 제공 할 수 있으므로 견고한 탐색 방법으로 제공됩니다.

$$\begin{aligned}
 \overrightarrow{LV_{H_i}} &= (R_i, \theta_i) \text{ and } \overrightarrow{LV_i} = (d_i, \alpha_i) \\
 \overrightarrow{PV_i} &= \overrightarrow{LV_j} - \overrightarrow{LV_{H_i}} \\
 \overrightarrow{HV} &= \frac{1}{N} \sum_{i=1}^N \overrightarrow{PV_i}
 \end{aligned} \tag{2.4}$$

4.4.1 최근 연구 동향

ALV 기반의 방법은 기본적으로 표식의 각 위치를 찾기 위해 객체 추출이 필요하다. 그러나 단순한 이미지에서 객체를 찾는 것이 쉬운 문제는 아니다. 따라서 초기 연구에서는 지정된 마커를 사용하고 다양한 특징점 알고리즘을 사용했다. 이러한 유형의 방법은 추출 된 특징점을 대상으로 쉽게 정의할 수 있으며 피쳐 매칭을 기반으로 폐색 (Occlusion) 효과를 감소시킨다. 최근에 SIFT와 함께 ALV를 사용하여 폐색(Occlusion)에 대한 연구가 개선되었다[68, 43].

스냅 샷 가정에 따라 두 이미지에서 SIFT 피쳐를 추출하고 이상 치를 필터링하는 데 나머지 피쳐는 서로 연결되고 원점 복귀 방향은 매개 변수 방법으로 쉽게 결정된다.

비슷한 관찰자의 상대적인 움직임에서 시작된 시각적 단서에서 특징점의 광학적 움직임 패턴을 기반으로 하는 옵티컬 플로우를 사용한다[99]. 또한 단순한 계산을 나타내지는 않지만 반복을 사용하여 강력한 매칭을 생성하는 방법을 사용한다[71]. 여기서는 지역 일치 보존 (LPM) 및 반복 매칭을 사용한다. 따라서 매개 변수 기반 방법은 로봇 공학에서 광범위하게 연구되었다.

4.5 여러가지 Warping 방법

또다른 지역 원정 내비게이션 모델중 하나는 워핑(Warping) 방법입니다[32, 31, 78, 126]. 다른 이미지를 근거로 한 접근법과 달리 워핑(Warping) 방법은 모바일 에이전트의 모든 가능한 움직임에 따라 이미지 픽셀의 모든 가능한 변경을 평가하고 각 픽셀에 대한 각 움직임의 효과를 계산한다. 따라서 대상 스냅 샷 이미지와 가장 잘 일치하도록 현재 스냅 샷 이미지의 신호를 변형하게 된다. 에이전트는 스냅샷 이미지를 왜곡함으로써 현재 위치에서 이동할 수 있는 모든 가능한 방향에 대해 새 이미지를 예측한다. 뒤틀린 각 이미지는 불일치가 가장 작은 이미지를 색출하여 대상 이미지와 비교한다. 가장 작은 차이가 있는 변형된 이미지는 대상 위치에 도달하기 위해 이동할 방향을 나타낸다. 예측 워핑 방법의 성능은 환경의 특성과 기준의 상대적 방향이나 절대 방향의 존재에 크게 영향을 받는다. 그러나 이 방법은 간단하고 스냅 샷 이미지에서 피쳐 추출의 사전 처리가 필요하지 않다.

이 워핑 방법은 또한 스냅 샷 모델을 기반으로 하지만 가능한 픽셀 일치 중에서 최적의 픽셀 이동을 찾는다. 다음 섹션에서 설명할 다른 픽셀 매칭을 사용하는 네 가지 알고리즘이 있습니다. 즉, Compass의 변형, 픽셀 거리 및 픽셀의 각도 차이를 사용하여 모든 경우를 계산한다. 이 방법은 정확성은 견고하지만 많은 양의 계산이 필요하다.

4.5.1 1D-Warping 방법

여러 가지 Warping 방법 중 첫 번째는 1D-Warping이다[32, 31, 44]. 이 메소드는 워핑 메소드의 기본 양식입니다. 먼저 이 부분에서는 컴퍼스가 있으면 이미지를 정렬하고 에이전트는 수평 링의 단일 픽셀에 포커스를 맞추는 것으로 가정한다. 1D-Warping은 이미

지의 픽셀에 대한 등거리 가설을 적용할 수 있습니다. 그런 다음 원점 복귀 방향은 워핑 함수와 각도 차이와 상대 거리를 삼각법 문제로 사용하여 방향 결정할 수 있다. 각도 차이는 집과 현재 위치 사이의 회전 각도이며 상대 거리는 랜드마크까지의 거리와 현재 위치 간의 비율이다. 그러나 이 방법은 수평 픽셀만을 고려하므로 수직 방향의 변화에 따른 픽셀을 변화를 Warping 시키지 못한다. 따라서 이 방법의 성능은 다른 것보다 낮은 특징이 있다.

4.5.2 2D-Warping, min-Warping

두 번째는 2D-Warping이고 세 번째는 Min-Warping이다[78, 59, 101]. 1차원의 1D-Warping과 달리 이 방법은 2차원 정보를 사용한다. 즉, 방위각 및 수직 변화 모두를 나타낸다. 방위각 변화의 경우 1D 뒤틀림 함수와 동등한 뒤틀림 함수를 사용하고 세로 방향의 경우 배율이 적용된 파노라마 이미지는 수직 배율 인수를 사용하여 배율이 조정합니다. 그런 다음 모든 가능한 경우에 대한 스케일 평면을 파생하고 합하여 최적의 경사 곡선을 생성한다. 마지막으로 원점 복귀 방향과 각도 차이는 워프 커브의 최적점에서 결정된다. 1D 워핑과 달리 이 방법은 가능한 모든 위치에서 최적의 방향을 결정하므로 Compass를 이용한 절대 방향 인지가 필요 없다. 또한 수직 구성 요소를 사용하면 견고성이 향상된다. 하지만 차원의 증가는 계산량을 증가시키고 계산 시간이 추가시킨다. 간단히 말하면, 2D Warping은 상당히 느리다. 전체론적 로컬 시각적 원점 복귀 방법을 기반으로 하는 세 번째 방법 (min-warping)은 이미지 열을 픽셀 단위로 비교하여 변환으로 인한 왜곡을 고려하는 방법이다.

첫 번째 단계에서는 이미지 열 사이의 모든 쌍방향 거리를 계산하고 이미지 사이의 상대 방향을 추정하려고 시도한다. 수직 왜곡을 만드는 다양한 스케일 평면을 사용하여 객체와의 거리의 변형됨을 찾습니다. 또한 스케일 평면에서 스케일 변형과 수평으로 이동한 조합을 모두 포함하여 일치하는 모든 오류를 계산합니다. 그런 다음 전체 평면 스택에서 최소 케이스를 수집하고 최적 경로를 선택하여 원점 복귀 방향을 찾는다. 그러나 위의 기법은 많은 2D 뒤틀림 변수를 필요로 하며 등가 거리 가정을 부분적으로 높이는

가속 과정을 포함한다. MinWarping은 나침반 추정, 사전 계산 된 워프 및 평면 템플릿 등 스케일 평면 내 계산량을 감소시켜 사용한다. 이 부분에서 에이전트의 집과 현재 위치 사이의 각도 차이를 찾고 시각 나침반의 예상 각도가 포함 된 30퍼센트 섹션을 추출한다. 이 30퍼센트는 시각 나침반의 오류를 허용하는 안전 계수이다. 2D 뒤틀림은 30 퍼센트 예상 각도 단면에서만 수행되며 계산 요구 사항은 70퍼센트 줄이는데 이 부분에서는 등거리 가설을 사용하는 세 가지 Warping방법 (1D, 2D 및 최소 Warping) 이 제공된다.

제 5 절 최근 연구동향

Warping 모델은 다양한 포즈 변경에서 픽셀 매칭을 기반으로 하므로 로봇 애플리케이션에서 광범위하게 연구되었다. 일반적으로 min-warping 이외에 많은 연구가 확장되어왔다 [81, 83, 82]. 예를 들어, 일부 연구에서는 신호의 뒤틀림에 기반한 로봇 실험에 적용된 SURF 및 SIFT와 같은 다양한 기능을 사용한다. 기존의 연구들은 토폴로지 맵에 청소 로봇을 지역화(Localization) 하기 위해 옵티컬 플로우 필터 또는 입자 필터와 병행하여 최소 차원이 Warping 방법을 사용하여 병합을 시도하였다. 이러한 Warping 기반 접근 방식은 다른 모델과 비교하여 빛에 대한 문제에 대한 견고성이 뛰어나며 다양한 모바일 로봇용 탐색 응용 프로그램에 Warping 모델을 적용된다.

5.1 전체론적 관점

전체론적 관점은 파노라마 이미지의 모든 픽셀을 사용하는 방법이다. 이 방법은 픽셀 일치에 유사도 측정을 통해 사용된다. 이 방법은 피쳐 추출에서 지정된 픽셀을 사용하지 않고 이미지 사이의 모든 픽셀을 일치시켜 신뢰할 수 있는 데이터를 얻는다. 즉 모든 픽셀을 비교하게 된다. 그런데도 여전히 이미지에 불필요한 부분이 있다. 따라서 일반적으로 이 방법은 각 이미지에서 바닥이나 하늘과 같은 일부 영역을 잘라낸다. 그리고 절단의 양은 출력의 정확도에 영향을 줄 수 있습니다. 또한 개미의 시각적 탐색에 관한 또 다른 연구에서 개미는 스카이라인뿐만 아니라 도형을 사용한다고 주장한다[8]. 기본

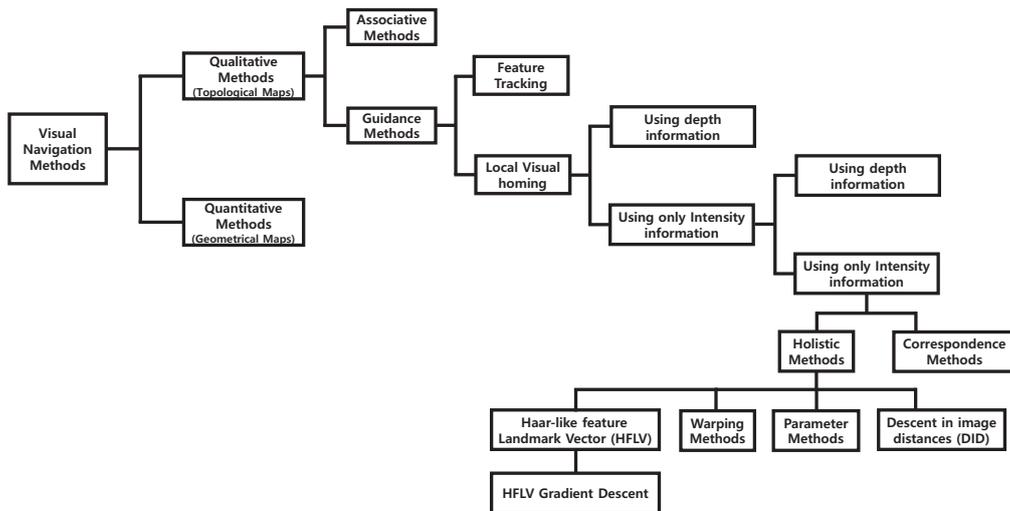


그림 2.5: 시각 탐색 분야의 다양한 기술 분류. ([78]의 논문에서 발췌)

전체는 이미지에서 적절한 참조 부분을 지정하는 것이다. 즉 필요 없는 부분은 제거하고 필요한 부분만 비교하게 된다. 비교하는 부분이 클수록 더 많은 계산이 필요하기 때문에 적절한 처리를 통한 연산량 감소에도 목적이 있다. 하지만 전체 비교하는 부분이 너무 작으면 중요한 정보가 손실된다. 따라서 파노라마 이미지에서 제거할 부분을 선택할 때 많은 기법을 필요로 한다.

5.2 Descent in Image Distance (DID) 방법

이미지 깊이 하강 (DID) 방법은 워핑 방법의 개념이 확장된 방법이다[122, 79]. 원점 위치 주변의 인접 위치에서 수집된 여러 참조 이미지를 현재 시각적 큐와 비교하여 원점 복귀 방향을 찾는다. 원래의 이미지는 가정과 가장 유사한 이미지를 수동으로 선택 했으므로 이 문서에서는 집 위치 $(x(x, y))$ 및 (y) 에 수집된다. 그런 다음 현재 참조와 참조 사이의 이미지 차이를 계산하고 원점 복귀 방향을 찾기 위해 비율을 찾는다. 1차원 워핑 방법은 후보

이미지들 중에서 홈 이미지와의 최소 차이만을 탐색하는 반면, DID 방법은 한 쌍의 위치들간의 이미지 차이의 변화를 모니터링한다. 현재 위치가 원점에 접근함에 따라 스냅 샷간의 차이가 줄어 든다.

경사 하강법 방법을 적용하면서 탐색 알고리즘이 목표 위치와 방향을 성공적으로 찾는다[102, 59]. 즉, DID 방법은 픽셀 차이를 사용하는 스냅 샷 기반 방법이다. DID는 나침반이 이미지를 정렬해야한다. 그런 다음 에이전트에 나침반 센서가 있고 모든 이미지가 정렬되어 있다고 가정한다. 그런 다음 에이전트는 가정의 스냅 샷 이미지 픽셀값과 현재 이미지 픽셀값을 비교한다. 즉, DID는 두 개의 파노라마 사이의 전체 픽셀값의 차이를 계산하고 가장 낮은 차이를 기준으로 최적의 방향을 찾아간다. 기본 DID 방법은 현재 위치에서 약간 떨어진 새 스냅 샷을 만드는 것이다. (Visual Compass 모델을 기반으로 한다[122].) 먼저 에이전트가 집에서 스냅 샷을 찍어 저장한다. 어떤 다른 위치에서, 에이전트는 측정 각도를 따라 고정 된 거리를 움직여 다른 스냅 샷을 찍고 위치와 집 사이의 픽셀 차이를 측정한다. 픽셀 차이는 두 이미지의 모든 픽셀 사이의 뺄셈에 의해 계산되는데 현재 위치로 돌아간 후에 에이전트는 다른 각도로 다른 위치를 사용하여 프로세스를 반복한다. 최적의 방향은 경사하강법의 경사가 현재 위치보다 낮고 감속 량이 가장 큰 방향이다. 하지만 계산 시간이 매우 길기 때문에 이 문제를 줄이기위한 연구가 있다.

5.3 Gradient-based Descent in image Distance 방법

이 방법은 DID 기록의 기본 응용 프로그램이다[79]. 위에서 언급했듯이 우리는 집 위치에서 근처를 방문하여 방향을 확인해야 하며 홈 방향을 찾기 위해 많은 계산을 해야 한다. 하지만 그래디언트 기반 방법의 핵심 아이디어는 이동 횟수를 줄이는 것이다. 간단히 말해 홈 스냅 샷과 세 개의 이미지만 사용하여 원점 복귀 방향을 찾으려고 한다. (원래 방법은 $360 / N$ 과 같은 수의 움직임 사용했고 N 은 각도 분해능이다.) 첫 번째 이미지는 머리 방향 (북쪽)에 있고 다른 이미지는 머리 방향 (동쪽)에 수직이며, 마지막 이미지는 현재 위치에 있다.

이동량이 크게 줄어들고 집 방향 스냅 샷을 포함한 4개의 이미지가 기준 Compass와

같이 정렬이 되는데 다음 단계는 집과 다른 이미지 간의 픽셀 차이를 계산하는 것이다. 홈 스냅 샷과 이미지 간의 픽셀 차이는 각 방향의 전환을 나타낼 수 있다. 즉, 진행 방향 이미지의 출력은 진행 방향의 차이를 나타내고 수직 이미지는 수직 변경을 이미지를 캡처한다. 원점 복귀 방향은 두 원점 간의 비율을 계산하여 결정된다. 경우에 따라 앞뒤로 움직이는 두 방향의 이미지가 사용된다.

$$\theta_{H,DID} = \tan^{-1} \frac{\sum_{i,j} *H_Y(i,j) - C(i,j) - \sum_{i,j} *H(i,j) - C(i,j)}{\sum_{i,j} *H_X(i,j) - C(i,j) - \sum_{i,j} *H(i,j) - C(i,j)} \quad (2.5)$$

5.4 Newton Descent in image Distance 방법

그라디언트 기반 방법을 사용하면 계산량이 줄어들지만 정보 손실과 일부 오류가 절충안이 되어야 해서 이 방법을 강화해야한다. Newton 방법은 이 알고리즘에 적용된다. 간단히 말해서, 우리는 대각선 차이점에 몇 가지 참조를 추가하는데 이는 계산량은 약간 증가하지만, 정확도는 향상되는데 일부 버전에서는 머리 방향, 수직 방향 두 개, 대각선 위치에서 앞뒤로 4개 (총 8개 이미지는 현재 위치 제외)의 두 이미지를 사용한다. 이것을 Newton 기반 DID 방법이라고 한다. Gradient 기반 DID 및 Newton 기반 DID 방법은 계산량을 혁신적으로 줄이는 반면에 다른 지역의 원거리 항법과 비교할 때 움직이는 데 약간의 시간 딜레이가 있다.

5.5 Haar-like Feature Landmark Vector (HFLV) 방법

HFLV는 시각 기반의 내비게이션 중 스냅샷 모델을 이용하는 방법이다. 현재 위치의 이미지와 목표 위치 각각의 스냅샷 이미지에 Haar-like Feature를 이용하여 랜드마크를 생성하고 Haar-like Feature의 타입별로 달라지는 매칭 스코어를 이미지와 계산하여 이미지 간의 유사도를 측정하여 집으로 향하는 방향벡터를 만드는 방법이다. 하지만 이미지의 픽셀만으로 거리를 측정할 수 없는 문제가 있고 기준이 되는 Compass 가 있어야 실제 알고리즘을 적용할 수 있다. 거리에 대한 정보를 얻을 수 없는데 이러한 부분을 추가적으로

극복할 수 있는 알고리즘 또한 제시되었다. Haarlike Feature는 모양과 주변 환경에 영향을 많이 받는다. 기존의 연구에서는 여러 가지 환경에서도 HFLV를 성공적으로 사용하는 연구를 제시하였다[62].

제 6 절 스냅샷 모델의 문제점

위에서 논의된 스냅샷 매칭 알고리즘은 몇 가지 공통적인 문제가 있다. 첫 번째는 거리 정보가 없어서 발생하는 문제이다. 두 번째 이미지는 두 개만 사용하면 생길 수 있는 교합 문제이다. 세 번째는 시각적 저장공간에 입력이 되어 사용되는 시야의 선택입니다. 넷째는 홈 스냅 샷과 현재 보기 간의 정렬이다. 다섯째, 적절한 랜드 마크를 선택하는 기준을 정하는 것이다. 이 다섯 가지 문제는 현지 시각적 탐색을 시행할 시에 고려해야 할 문제입니다. 다음 절에서는 이러한 문제점의 세부 사항과 해결 방법에 관해 설명한다.

6.1 문제 1 : 현지화 (Localization)

현대화 Simultaneous Localization and Mapping (SLAM) 기술에서도 모바일 로봇 내 현지화 문제는 까다로운 문제다. 현지화는 모바일로봇의 중요한 과정이다. 현재 로봇이 절대적 또는 상대적으로 어디에 위치해 있는지 아는 것은 내비게이션의 수행능력에 중요한 판단이 된다. 환경에 대한 정보는 여러 유형의 센서에서 얻을 수 있다. 전역 위치 파악 시스템 (GPS) 및 레이저 거리 측정 센서로 전역 위치 파악하여 맵을 얻을 수 있다. 레이저 거리 센서는 인간과 로봇의 상호 작용 또는 산업 응용 분야에서 적용할 수 있습니다[13, 51]. 환경에 대한 거리 정보 외에도 공기 및 지면에서 흔히 볼 수 있는 객체 또는 입체 기능의 고도지도가 있는 Monte Carlo 접근 방식을 사용하여보다 나은 현지화 정보를 제공할 수 있다[57, 56]. 만일 레이저 등의 범위 센서가 없을 경우에 시각 정보는 위치, 밝기, 색상 및 객체 모양에 대한 많은 정보를 제공할 수 있으므로 시각 내비게이션 알고리즘은 탐색 알고리즘에 가장 널리 적용된 복잡한 탐색 프로세스를 사용하여 비전을 사용하여 피쳐 추출을 적용하는 많은 탐색 알고리즘이 개발되었다[11, 125, 2, 3].

만일 시각적인 방법으로 내비게이션을 구현하려면 센서가 필요하다. ALV 모델을 포함한 간단한 랜드 마크 기반 탐색 방법조차도 비전 센서가 있는 이동 로봇에 적용되었고, 랜드 마크를 식별하는 특징 추출 프로세스도 필요하다[125]. 대조적으로 DID 방법은 한 쌍의 스냅 샷 이미지를 픽셀 단위로 비교하는 총체적 접근 방식이고 이미지의 거리를 계산한다 [122, 79, 59]. 그러므로 이 알고리즘은 특징 추출 프로세스가 필요하지 않을 것이다. 원점 복귀 방향을 결정하려면 이미지 사이의 거리를 계산해야한다. 이러한 방법은 성공적인 원점 복귀 탐색을 나타내지 만 일부 정확성을 잃어 버리고 장거리에서 최적 경로를 찾을 수 없다. 거리 요소를 극복하기 위해 더 많은 고급 모델이 개발되었다. 비전이있는 거리 센서를 사용하면 위치 정보를 쉽게 제공 하는 이점이 있다. 픽셀의 세기 정보에 범위 정보를 추가하는 다른 연구도 있는데, 이전의 연구에서 거리 센서를 사용하는 방법을 적용하여 원점 복귀 탐색의 정렬 및 정확도에서 높은 성능을 쉽게 달성한다[121, 63]. 그런 다음 추가 이미지 프로세스, 추가 이미지 수집 또는 추가 계산과 같은 높은 복잡성을 추가하는 방법이 있다.

6.2 문제 2 : 폐색 (Occlusion)

랜드 마크 폐색은 위치 이미지에서 하나 이상의 랜드 마크가 누락되거나 방해받는 것을 나타낸다. 기본적으로 예상되는 물체가 폐색에 의해 이미지상에 나타나지 않으며 에이전트는 비교에 의해 수행하는 물체의 예상 위치 감지를 할 수 없게 된다. 네 개의 랜드 마크가 있고 에이전트가 이미지에서 가려져서 세 개만 찾을 수 있으면 폐색에 의한 문제가 도출된다. 이러한 경우에 에이전트가 사용 가능한 시각적 정보 외의 로컬 홈 내비게이션이 적용되지 않는 다른 기술을 통해 추정하게 된다. 폐색을 극복하기 위해서는 다른 센서 또는 시퀀스 추적과 같은 보정 도구를 적용한다. DID 및 뒤틀립과 같은 메소드는 랜드 마크를 추출하지 않는다. 따라서 폐색 문제는 스냅 샷 모델을 기반으로 하는 매개 변수 방법에 따라 다릅니다. 폐색이없는 경우 ALV는 가장 높은 정확도를 나타낸다. 그러나 폐색이 있는 경우 ALV는 성능이 크게 저하된다. ALV와 달리 DELV 방법은 폐색에 강인하다. 그러므로 알고리즘과 환경에 따라 폐색 효과가 다를 수 있다. 또한 이미지 재구성을

사용하여 폐색 효과를 필터링하려고 하는 연구가 시도되고 있다[61].

6.3 문제 3 : 필드 시야

먼저 새로운 알고리즘으로 스냅 샷 모델을 사용한다고 가정하면 입력 시각적 신호로서의 파노라마 이미지는 바닥, 하늘, 명소, 벽, 로봇의 일부와 같은 주변에 대한 모든 정보를 갖습니다. 그러면 파노라마 이미지상에서 사진의 모든 부분이 실제로 필요한지 여부가 문제가 됩니다. 바닥, 옴니- 카메라 플랫폼 또는 천장 등에 의해 생기는 이슈를 시야 (FOV) 야기 한다.

6.3.1 스카이 라인

시각적인 단서를 이용한 내비게이션에서 스카이라인을 활용한 방법들이 많다. 스카이라인은 하늘과 지표 또는 경계표를 포함한 다른 지형 간의 경계이다. 경계는 하늘과 다른 물체 사이의 색상 차이를 기반으로 한다. 그러면 데이터의 양이 각도 분해능의 수로 줄어듭니다. 스카이라인은 거대한 나무 나 산과 같은 탁월한 랜드 마크를 포함하고 장거리에서 강력한 성능을 발휘할 수 있다. [117, 89]. 스카이라인을 이용한 경계표는 단거리 여행에서 눈에 띄는 차이점을 나타내지 않지만 가까운 대상물은 큰 차이를 만들어낸다. 따라서 이 방법은 단거리와 관련된 약점이 있다. 따라서 우리는 이 모델을 적절한 환경에 선택적으로 사용해야 한다. 이 약점에도 불구하고 시각적인 내비게이션 용 스카이라인을 사용하는 곤충에 관한 많은 연구가 있다[42].

6.3.2 수평선 라인

그러나 수평선을 따라 한 줄의 픽셀 만 사용하면 빛에 의한 왜곡과 같은 이유로 인해 랜드 마크가 누락되는 문제가 발생할 수 있다. 따라서 가로 줄 주위에 픽셀 줄 집합을 사용하는 것이 가장 좋다.

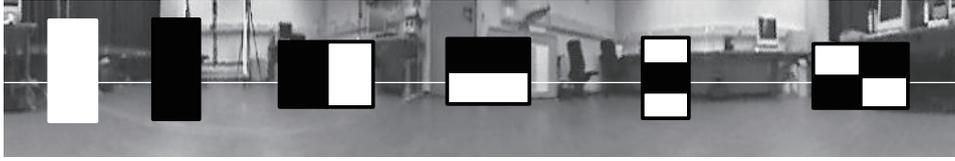


그림 2.6: 수평선 라인을 이용하는 예시. ([62]의 논문에서 발췌)

6.4 문제 4: 정렬

모바일로봇의 내비게이션에서 나침반이 있는지는 중요한 문제입니다. 이 과정에서 나침반은 절대 방향으로 기준 될 수 있습니다. 시각적 정로를 이용하는 내비게이션은 스냅샷을 통해 이미지를 수집하고 해당 이미지 정보에 의해 랜드마크의 위치가 결정된다. 만일 나침반 센서가 방향의 절대 표준을 제공한다고 하면 환경에 큰 변화가 없는 경우 스냅샷의 랜드마크 순서는 서로 다른 위치에서 찍은 스냅샷에 대해 거의 동일하다. 따라서 랜드마크 위치의 변화를 쉽게 관찰할 수 있습니다. 그러나 로봇에 절대 방향을 기록할 나침반을 가지고 있지 않다면 기준 이미지와 비교하여 비틀어진 각을 먼저 계산해야 한다. 따라서 많은 응용 분야에서 외부 자기 센서를 나침반으로 사용하는 경우가 있지만 컴퍼스 센서를 사용하여 나침반을 생략하고 방향을 찾는 다른 여러 가지 방법들이 있다. 아래에 몇 가지 예를 설명한다.

6.4.1 Visual Compass

Visual Compass는 대표적인 정렬 방법이다. Visual Compass를 구현하기 위해서는 정렬의 기준이 되는 이미지가 필요하다. 기준 이미지와 시프트된 이미지 간의 신호적인 차이를 에러로 가정하고 최소 에러가 결정될 때까지 다른 이미지가 이동된다. 그러면 최소 오차를 갖는 이동된 이미지는 정렬된 기준 이미지와 비교적 방향이 같다. 대부분의 데이터 획득과정에서는 이미지의 자기 북쪽을 알 수 없기 때문에 홈 이미지를 기반으로 모든 입력 이미지를 정렬했으므로 필요하지 않다.

6.4.2 랜드마크의 재정렬

랜드마크 위치의 차이를 사용하는 방법의 경우, 로봇에 내장된 나침반이 있는 경우 랜드마크 사이를 일치시켜 집을 찾고 현재 위치를 각 랜드마크로 추정한다. 따라서 시각 나침반과 마찬가지로 모든 가능한 각도를 매칭하여 각 랜드마크에 대한 예상 위치를 찾는다. 그리고 추정된 위치의 분포를 고정된 각도 차이로 확인할 수 있으며 또한 순서와 각도의 차이가 있을 때 분포의 변화가 가장 작은 경우를 찾을 수 있습니다. 이 경우 각도 차이는 나침반 값이 된다. 시각적 나침반과 마찬가지로 두 스냅샷 모두 공통된 요인이 충분하면 정확한 방향을 찾을 수 있다. 그러나 이 방법은 모든 픽셀을 사용하는 것이 아니라 추출된 랜드마크를 사용하는 점이 다르다.

6.4.3 다양한 정렬 알고리즘

다양한 정렬 모델 중 시각적 나침반을 사용할 시 알고리즘에 전반적인 성능에 개선에 큰 영향을 미친다. 그 이유는 시각을 내비게이션 모델에서는 이미지의 정렬이 중요한 사항이기 때문이다. 첫 번째는 [58]에 설명된 시각 나침반(Visual Compass)은 신호를 이용하는 방법이다. 기준 이미지에서 한 열식 회전되고 절대 차이를 계산하게 된다. 그리고 가장 작은 차이를 기준으로 각도 재정렬을 하는 방법이다. 다양한 크기의 이미지를 처리하기 위해 정규화된 NSAD를 사용한다[45]. 또 다른 방법은 행 합계 방법과 위상 상관을 사용한다[55, 100]. 회전 성분이 수평 이미지 시프트를 야기하기 때문에, 시간 영역에서 시간 시프트를 찾는 데 사용되는 위상 상관 방법이 사용될 수 있다.

6.5 문제 5: Landmark의 특징 선택

가장 큰 문제 중 하나는 랜드마크를 선택하는 과정에서 생기는 문제이다. 랜드마크는 내비게이션을 할 때 시각적 특수 정보를 나타내게 되는데 랜드마크가 선택될 때 센서의 특성에 따라 특성이 다른 랜드마크들이 선택된다. 센서기술의 발전과 더불어 랜드마크 기술들은 다양한 분야에 적용 가능한데 그중 모바일 로봇에 적용되고 있다. 다양한 내비

게이션 기술이 제시되고 있는데. 수중 차량 및 무인 공중 비행체 및 우주선에 적용되고 있다. 또한 다양한 센서를 이용한 추가 연구가 발표되었다[70, 25, 37, 110, 87]. 따라서 센서 유형에 따라 피처를 선택하는 것이 매우 중요하다. 우리는 생물에 영감을 받은 탐색 기능을 기반으로 하였기 때문에 비전에 중점을 두고 있다. 그러나 시각 데이터를 이용하여 최적의 수행능력 갖추려면 다양한 시각적 탐색 방법을 활용할 수 있다. 하지만 환경이 매우 복잡하다면 랜드마크를 사용해야 한다. 따라서 시각을 기반으로한 내비게이션의 선 처리 부분으로 좋은 랜드마크를 추출하는 것은 성능에 직접적인 영향을 주기 때문에 랜드마크를 뽑을 때 특징을 나누는 기준은 매우 중요한 파라미터로서 문제가 된다.

제 7 절 요약

모바일로봇의 내비게이션 연구는 여러 방향의 알고리즘들로 발전하였다. 하지만 대부분 공통적인 문제를 가지고 있는데 그것은 바로 'Localization' 문제이다. 또한 모바일로봇을 내비게이션으로 이동시킬때 기준이 되는 Compass를 통해 이미지 가공의 선 처리가 달라진다. 대부분의 알고리즘은 이러한 문제들을 우회하거나 추가적인 알고리즘을 통해 문제들을 돌파하였다. 본 논문에서 다루는 HFLV 또한 이러한 여러 가지 문제들을 가지고 있다. 이후 챕터에서 경사 하강법 알고리즘을 적용하면서 이러한 문제들의 돌파구를 찾는 실험들을 진행할 것이다. 본 연구진이 연구하는 알고리즘이 최신동향에 비해 어떤 수행능력을 갖추는지 비교분석을 할 수 있을 것이다.

제 3 장

경사하강법을 적용한 HFLV (Haar-like Feature LandMark Vector)

스냅샷 모델은 곤충들의 시각 정보를 이용한 내비게이션에 영감을 받아 만든 모델이다. 본 챕터에서 우리는 스냅샷 모델을 바탕으로 HFLV (Haar-like Features Landmark Vector) 와 여러가지 방법을 적용한 모델을 구현하고 실험을 통해 수행능력을 비교하게 된다[29, 62]. 또한 이전의 방법들과 비교하여 적은 오차율을 보이는 방법을 여러 가지 형태에 따라 분석하고 여러 가지 실험을 통해 오차율을 줄일 수 있는 방안을 제시한다. 본 챕터에서는 정확도를 높이고 파라미터 수를 줄일 수 있는 세가지 형태의 매칭 스코어를 제시하게 된다.

우리는 매칭 스코어마다 새로 소개된 방법들을 각각 적용하고 가장 좋은 수행능력을 가지는 방법들을 분석하는 실험들을 진행하여 2가지 목표에 대한 실험 결과들을 얻고자 하였다. 첫 번째 목표는 임의로 생성된 수많은 HF (Haar-like Feature) 중 좋은 HF를 선별하는 방법이다. 두 번째 목표는 마스크의 스코어와 학습된 가중치의 관계를 분석하는 것이다. 이는 마스크의 매칭 스코어 형태와 가중치의 관계 분석을 통하여 Haar-like

Features를 선별하고 조금 더 최적화된 학습을 구현할 수 있을 것으로 예상되기 때문이다. 우리는 방향 결정의 중요한 역할을 하는 마스크를 선별하고 학습에 최적화시키기 위하여 매칭 스코어의 형태를 고려하는 여러 가지 실험을 진행하였다. 본 챕터에서는 여러가지 실험을 통해 좋은 수행능력을 가진 알고리즘을 제시하고 다양한 환경에 적용하여 알고리즘의 강인함을 입증하는 실험을 진행한다. 또한 본 연구의 일부 또는 확장된 내용에 대해서는 다음 논문지에 게재될 예정이다[50].

제 1 절 Haar-like Fraecture LandMark Vector 모델

벌이나 개미 같은 곤충들은 등지를 떠나 먹이를 사냥하고 다시 등지로 돌아올 때 시각적인 정보를 바탕으로 내비게이션을 하여 집으로 돌아오게 된다[20]. 이때 가장 첫 번째 장면과 마지막 장면을 기억하게 되는데 이를 바탕으로 제안된 스냅샷 모델은 집에서의 이미지와 현재 위치에서의 이미지를 비교하고 어떤 방향으로 이동할지 결정하는 방법이다[29]. 본 섹션에서는 이전의 연구진이 제안한 스냅샷 모델을 바탕으로 HFLV (Haar-like Features Landmark Vector) 을 구현하는 과정과 실험 환경들을 설명한다[62].

1.1 실험환경 및 데이터 획득

본 실험에서는 HFLV를 이용한 내비게이션을 구현하기 위하여 특정 환경을 이용하였다. 우리는 기본적으로 Vardy의 데이터 중 면적이 약 2.7m * 4.3m 환경에서 170개의 포지션 별로 Omni directional 이미지로 되어있는 여러 환경의 데이터를 사용하였다[107]. Vardy의 데이터셋을 비롯하여 본 논문에서 사용되는 이미지 데이터는 Omni directional 형태로 수집된다. 그림3.1 (b) 는 Omni directional 이미지의 예시이다.

본 논문에서는 HF (Haar-like feature) 를 이용하여 랜드마크를 만들기 위하여 Omni directional 이미지를 panoramic 이미지 형태로 변환하여 사용한다[107]. 본 논문에서 사용되는 이미지 데이터들은 그림3.1 (a) 의 모바일 로봇 P3-DX 이용하여 획득하였다. 또한 실험 환경구성은 전부 실내에서 진행되었다. 본 논문에서 주로 이용된 Vardy의 여러 환

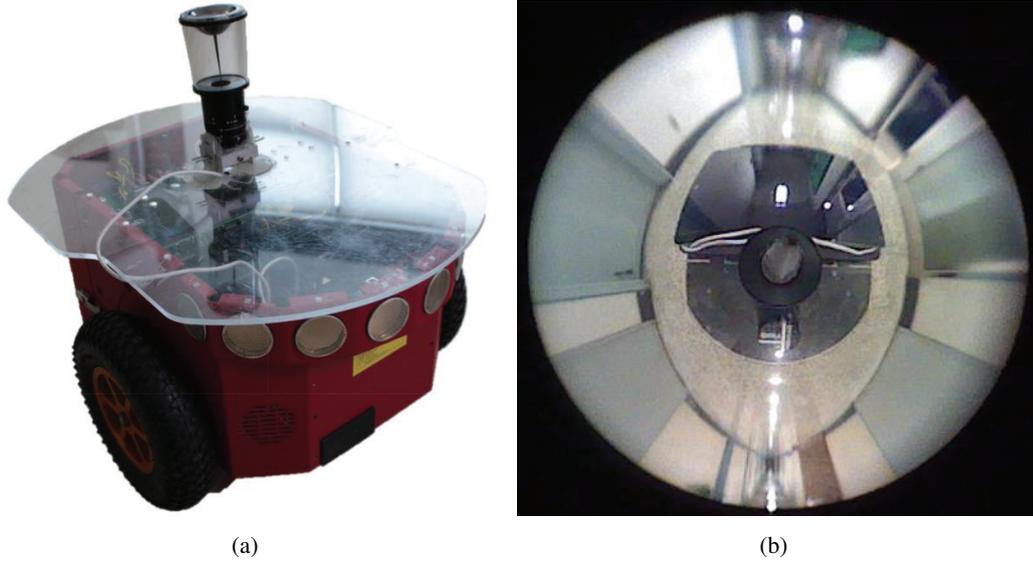


그림 3.1: 모바일 로봇과 데이터. (a) 모바일 로봇 (P3-DX) (b) 옴니 방향 이미지



그림 3.2: Haar-like Feature의 여러가지 형태. 본 논문에서는 가장 왼쪽편의 HF (Haar-like Feature) 를 타입1로 정의한다.

경의 데이터셋 중 가장 기본적인 환경은 'a1original' 이다. a1original 환경의 경우 빛의 세기의 영향이 골고루 분포된 실내의 실험 환경이다. 본 논문에서는 'a1original' 이외의 빛이 적은 환경과 장애물에 의해 가려진 환경 등을 사용한다.

1.2 Haar-like Feature 생성

본 논문에서는 HFLV (Haar-like Feature Landmark Vector) 경사 하강법을 적용하기 이전에 랜드마크를 생성하기 위하여 HF (Haar-like features) 를 사용한다[119]. HF은 주로 인식 알고리즘에 사용된다. HF는 시각의 양극 세포를 여러 가지 형태로 모델링 되었다.

각각의 이미지가 세포를 통과한 이후 가지게 되는 전달을 자극을 HF의 스코어로 표현하였는데 이러한 이유로 시각인식 시스템에 많이 이용되고 있다. HF가 생성된 영역을 살펴보게 되면, 단일 또는 여러 개의 직사각형들끼리 이웃하여 생성된다. 직사각형의 영역의 픽셀의 합 또는 두 개 영역의 각각의 평균 명암 차를 구하여 스코어를 계산한다.

그림3.2는 HF의 대표적인 여러 가지 형태이다. 가장 기본적인 형태는 맨 좌측 검은 색과 흰색 HF 이다. 본 논문에서는 가장 기본적인 형태의 마스크를 타입1 로 정의한다. 타입1 의 마스크는 생성 위치의 평균 명암을 나타내는 마스크이다. 좌측부터 형태별 타입으로 나누었다. 타입은 총 6개가 있고 두 번째 와 세 번째 타입의 마스크를 통해 좌우 또는 상하 명암비를 알 수 있다. 박스 모양의 마스크 등 여러 가지 마스크들이 있고 해당 마스크들이 각각의 모양에 따라 다른 수행능력을 보인다.

HF (Haar-like Feature) 는 약한 분류기이기 때문에 단일 마스크로는 비교적 강한 분류를 하기 어렵다[109]. 하지만 단일 마스크인 약한 분류기를 여러 개를 모으면 비교적 강한 능력의 분류기를 만들 수 있다. 약한 분류기로 강한 분류기를 만드는 방법들이 있다. 그 방법을 앙상블 (Ensemble) 이라고 하는데 이러한 이유로 여러 개의 HF를 동시에 생성하여 사용하게 된다[27]. HFLV (Haar-like Features Lanmark Vector) 두 개의 이미지에 랜드마크를 생성하여 유사도를 비교하고 생성된 벡터들의 수학적 계산을 통해 이동해야 할 방향을 도출한다. 따라서 현재 위치의 이미지와 집 위치의 이미지에서 각각 같은 자리에 HF를 생성하게 된다. 여기서 HF는 이미지에서 랜드마크 역할을 하게 된다. HF의 스코어를 계산한 뒤 두 스코어의 차를 통해 방향벡터를 생성하게 되고 생성된 방향벡터들을 모두 합하여 최종 방향을 결정하게 된다[62].

마스크는 생성될때 이미지상의 수평선근처에서 생성되는데 그림3.3 (b) 보게되면 수평선HFLV에서 HF는 수평선 근처에서 생성되게 설정한다. 그 이유는 상대적인 정보의 양이 수평선의 아래보다 수평선의 윗부분이 많기 때문이다[7].

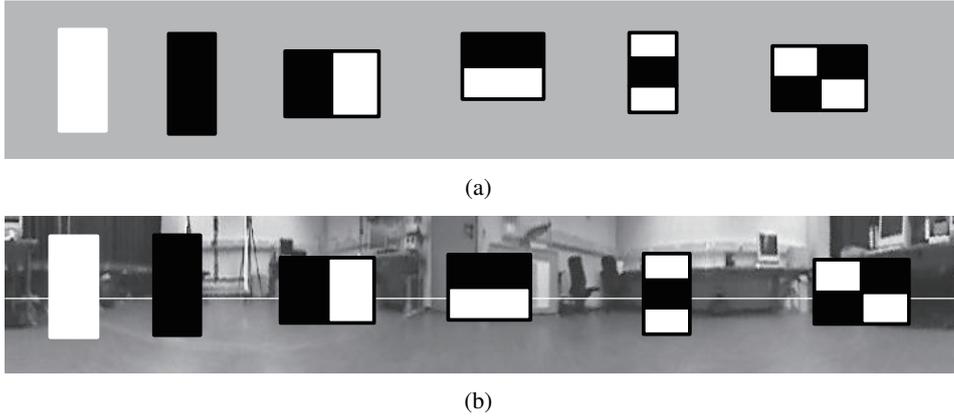


그림 3.3: 이미지상의 Horizontal Line 표시방법. (a) 임의의 위치에 생성된 HF (Haar-like Feature) (b) 마스크를 매칭할때 Horizontal 라인 주변에 생성된 HF (Haar-like Feature)

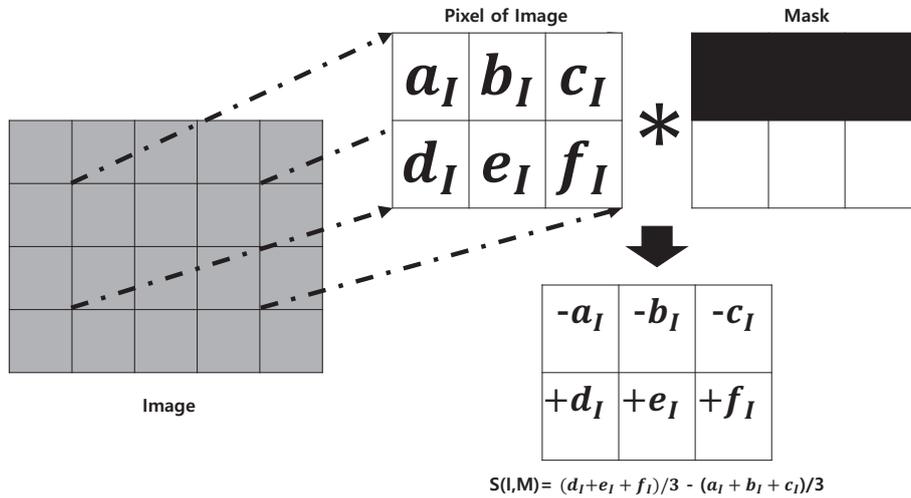


그림 3.4: 매칭스코어 계산 방법. 위 그림은 생성된 HF와 각 포지션에서 획득한 이미지의 매칭스코어 계산 방법이다.

1.3 매칭스코어 계산법

스냅샷 모델은 기반으로 한 HFLV (Haar-like Features Landmark Vector) 집 이미지와 현재 위치의 이미지인 두 스냅샷의 유사도를 비교하여 방향 결정을 하고 내비게이션을

구현한다. 두 스냅샷 이미지에서 각각 HF (Haar-like Feature) 를 생성하게 되고 생성된 HF가 랜드마크의 역할을 하게 된다. 이때 HF는 두 개의 스냅샷 이미지에서 같은 위치에 생성된다. 그림3.4은 이미지와 마스크의 합성 방법에 대한 그림이다. 그림3.4와 같이 HF와 이미지 간의 합성을 통해 계산하여 나온 값을 매칭 스코어라고 하고 매칭 스코어의 누적 합을 벡터와 합성하여 방향을 결정하게 된다.

본 섹션에서 매칭 스코어를 계산하는 방법을 자세하게 설명하려고 한다. 먼저 그림3.3 (a)와 같이 HF (Haar-like features) 를 임의의 위치에 랜덤하게 생성하게 된다. 생성된 이미지를 그림3.3 (b) 와 같이 집 이미지와 현재 위치의 이미지에 같은 위치에 생성한다. 매칭이 된 이미지와 HF는 각각의 위치에서 스코어 계산을 하게 되는데 식3.1 와 같이 마스크의 백색 부분은 +1, 마스크의 흑색 부분은 -1로 정의하여 이미지상 마스크의 위치에서의 평균 밝기 강도 값의 차이를 구하게 된다. 그림3.4를 보게 되면 이미지가 가지는 픽셀값과 마스크에 모양에 따라 얻을 수 있는 스코어가 달라지기 때문에 이렇게 계산된 매칭 스코어를 통해 마스크 모양에 따라 생성된 랜드마크에 의해 해당 위치의 특성을 알 수 있다. 대부분의 스코어의 크기는 마스크의 모양과 관계 크다.

$$S(I, M) = \left| \sum_{i \in m_w+} I_i/m_w - \sum_{i \in m_b-} I_i/m_b \right| \quad (3.1)$$

1.4 랜드마크 벡터생성

HFLV (Haar-like Features Landmark Vector) 하나의 HF (Haar-like features) 는 하나의 랜드마크 역할을 한다. HFLV는 랜드마크 역할을 하는 마스크들은 매칭스코어에 비례하여 각각의 위치별 방향벡터로 만들어준다. 매칭스코어 계산을 통해 얻은 $S(I, M_j)$ 는 식3.2를 통해 랜드마크 벡터로 생성된다. 매칭스코어는 \hat{u}_j 와 곱해지고 방향 벡터 형태인 $\vec{L}_j(I)$ 를 얻을 수 있다. 각각의 마스크와 이미지별로 방향벡터를 만들 수 있다. 집 이미지에서 HF에 의해 만들어진 매칭스코어의 총합과 현재 위치의 이미지에서 HF에 의해 만들어진 총합의 차를 구하여 두 이미지 간의 랜드마크 벡터 비교를 하게 된다.

이후 방향추정 위하여 식3.2을 통하여 각각의 매칭스코어인 $S(I_{home}, M_j)$ 과 $S(I_{Current}, M_j)$

를 구한다. 임의의 모양과 위치에 생성했던 마스크들의 스코어를 계산하여 각각의 마스크의 각도와 매칭 스코어를 고려하여 LV (Landmark Vector) 를 $\vec{L}_j(I)$ 를 생성하게 된다. 랜드마크 벡터 $\vec{L}_j(I)$ 은 $S(I, M)$ 와 \hat{u}_j 를 곱하게 된다. \hat{u}_j 는 $(\cos \Theta, \sin \Theta)$ 이다.

$$\vec{L}_j(I) = S(I, M_j)\hat{u}_j = (S(I, M_j) \cos \Theta, S(I, M_j) \sin \Theta) \quad (3.2)$$

$$\begin{aligned} \vec{H} &= \sum_{j=1}^N \vec{h}_j = \sum_{j=1}^N (Current_j - home_j) \cdot \hat{u}_j \\ &= \sum_{j=1}^N \Delta L \cdot \hat{u}_j \end{aligned} \quad (3.3)$$

1.5 타입별 매칭스코어의 분포도

그림3.5는 매칭 스코어의 타입별 분포 그래프이다. 가장 평범한 실내 환경인 'a1original' 을 실험 데이터로 사용하였다. 그림3.5의 HF (Haar-like Features) 의 매칭스코어는 현재 위치와 집 위치에서의 스코어의 차이이다. 모든 형태의 마스크들이 임의로 생성되었고 그중 타입 1의 마스크가 일정한 높은 스코어 분포도를 가지는 것을 볼 수 있다.

마스크 타입 1은 형태의 특성상 매칭스코어가 다른 마스크보다 높기 때문에 매칭스코어의 크기가 크면 방향 결정을 할 때 많은 기여를 하게 된다. 그 이유는 방향벡터의 크기 마스크 각각의 매칭 스코어에 비례하기 때문이다. 따라서 매칭 스코어가 크면 클수록 더 많이 기여를 한다고 추측할 수 있다. 이후 실험에서는 매칭 스코어의 크기와 형태별 어떠한 영향력을 가졌는지 알아보는 실험을 진행하였다.

1.6 HFLV(Haar-like Features Landmark Vector) 결과

그림3.6는 HFLV (Haar-like Features Landmark Vector) 알고리즘을 적용한 결과이다. HF 의 개수에 따라 얼마만큼 AE(Angular Error)를 나타내는지 그래프로 표현하였다. AE 는 집을 제외한 169개의 포지션의 오차의 평균값이다. 그림3.6은 일정 개수 이상의 HF

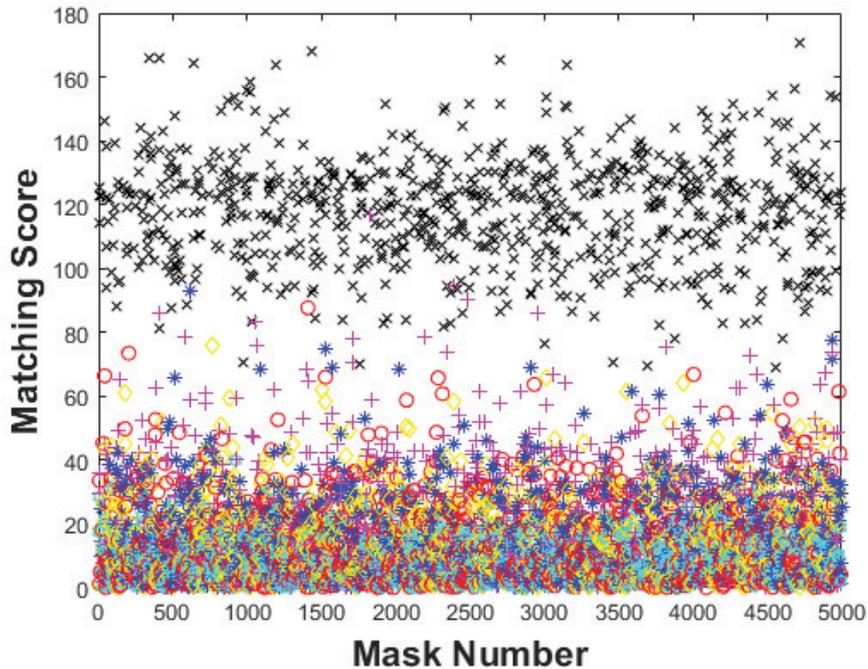


그림 3.5: 임의의 5000개 마스크의 매칭스코어 분포도. 실험환경은 'a1original' 이고 6개의 Type과 마스크의 모양 및 위치가 임의로 생성되었다. 검정 X 는 Type1, 파랑 별은 Type2, 마젠타 +는 Type3, 빨강 동그라미는 Type4, 노랑 다이아는 Type5, 크레온 육각별은 Type6 이다.

(Haar-like Features) 를 사용하게 되면 AE가 포화하기 시작한다. 이후 실험에서는 포화점의 안정화 단계에 있는 거로 관찰되는 100개의 HF를 이용하여 실험을 하게 된다.

제 2 절 실험1 : 경사 하강법

본 논문에서는 이러한 벡터의 합으로 생성된 최종 벡터를 집으로 향하는 방향으로 만들기 위해 경사 하강법을 사용하였다. 경사 하강법은 머신러닝의 최적화 알고리즘 중 하나이다. 가장 기본이 되는 아이디어는 함수의 기울기를 이용하여 지속적인 피드백을 주고 목표값을 찾는 것이다. 여기서 기울기를 구하는 함수는 에러를 구할 때 사용하는

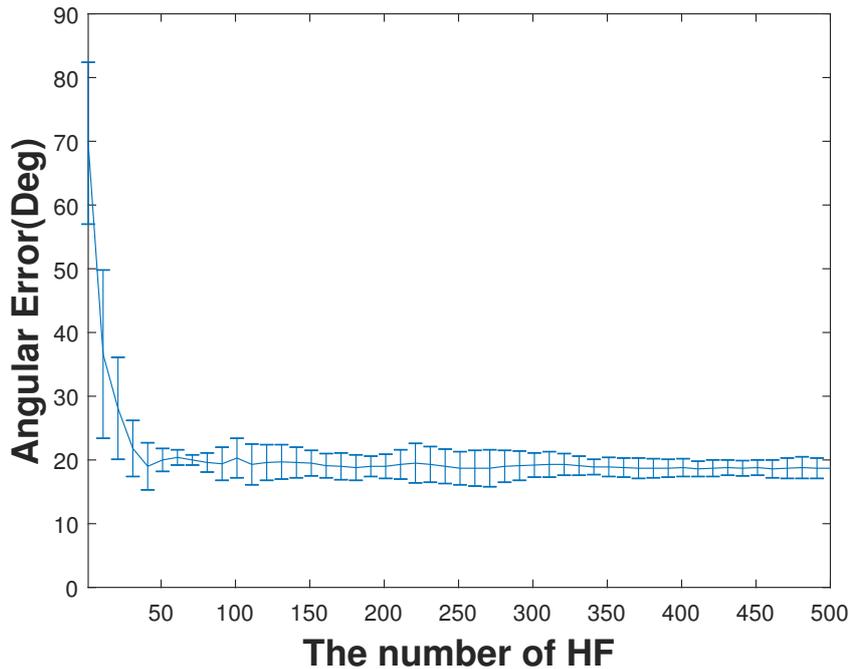


그림 3.6: HF의 갯수별 수행능력 그래프. 이미지상의 랜덤한 위치에서 생성되는 타입 1 형태의 HF (Haar-like Feature) 를 사용하여 HFLV (Haar-like Features Landmark Vector) 구현하였다. Vardy의 'a1original' 환경에 적용하였다. HF를 1개 사용했을 때 부터 500개 사용했을 때의 오차율을 나타내었다.

합수가 된다. 경사 하강법은 피드백을 통해 극값을 이룰 때까지 반복 학습하는 방법이다.

그림3.7 와 같이 경사 하강법의 특징은 차원과 관계없이 사용할 수 있다는 점이다. 최종적으로 경사 하강법을 통해 구하게 되는 값은 글로벌한 최솟값이 된다. 이러한 글로벌 최솟값에 도달하기 위해서는 여러 가지 접근이 필요한데 여러 가지 파라미터의 조정 에 따라 다르게 목표한 최솟값에 도달하기 때문이다. 최종적으로 Cost Function이 최소로 되는 값을 찾게 되면 목적을 달성하게 된다.

본 고에서는 이러한 경사 하강법을 이용하여 벡터의 합으로 이루어져 있는 최종 벡 터가 최적의 방향을 향하게 만든다. 이를 통해 성공적인 모바일 로봇이 성공적인 내비 게이션을 할 수 있도록 더욱 강인한 알고리즘을 제시한다. 또한 본 우리는 기본적으로

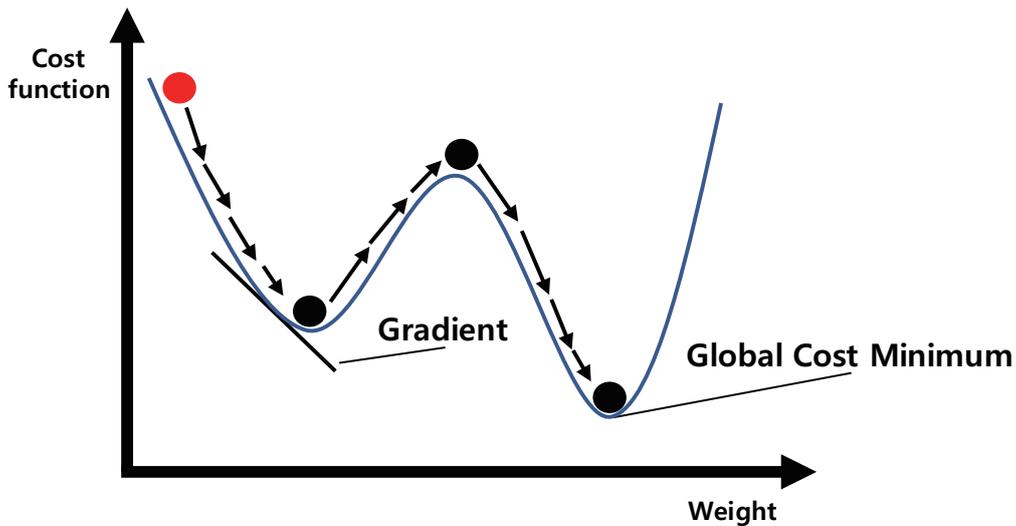


그림 3.7: 경사 하강법 원리에 대한 모식도. 경사 하강법은 지역적인 최솟값보다 작은 글로벌한 최솟값을 찾기 위한 방법이다.

HFLV(Haar-like Features Landmark Vector) 경사 하강법을 적용하게 되므로 경사 하강법을 적용한 HFLV를 본문에서 HFLV-GD (Haar-like Features Landmark Vector-Gradient Descent) 라고 표기한다.

2.1 경사하강법 공식

경사 하강법은 Cost Function이 극점인 지점에서의 가중치를 구하고 이를 식3.3에 적용하는 방법이다. 위의 그림3.7에서와 같이 지역적인 최솟값에 빠질 수도 있다. 하지만 최종 목표는 글로벌한 최솟값에서의 가중치를 찾는 것이다. 업데이트가 반복될 수록가중치 또한 수렴할 것이다. 아래 식3.4와 같이 Cost function을 사용할 수 있다. 이러한 Cost Function을 이용하여 HFLV에 적용한 최종식은 식3.5과 같다. 최종 \vec{H} 를 구하기 위해서는 업데이트된 가중치를 구해야한다. 아래 식들을 통해 CostFunction의 기울기를 구하는

방법을 자세하게 설명한다.

$$Cost(W) = 1/2m \sum_{i=1}^m (Wx_i - y_i)^2 \quad (3.4)$$

$$\begin{aligned} \vec{H} &= \sum_{j=1}^N \vec{h}_j \cdot W_{i+1j} = \sum_{j=1}^N (Current_j - home_j) \cdot \hat{u}_j \cdot W_{i+1j} \\ &= \sum_{j=1}^N \Delta L \cdot \hat{u}_j \cdot W_{i+1j} \end{aligned} \quad (3.5)$$

Cost Function을 이용하여 반복 학습이 진행됨에 따라 총 에러가 줄어들고 Weight가 변화하게 된다. Weight 같은 경우 조절된 Learning Rate만큼 변화하게 된다. 식3.7은 업데이트 지점의 오차의 기울기에 따라 변화하게 된다. 식3.7을 보게 되면 에러율의 기울기가 0으로 수렴하게 되는 순간 Weight의 업데이트가 느려진다. 이때 수렴하게 된 Weight 값이 Cost Function이 최솟값을 가지게 되는 Weight가 된다.

$$W_{i+1} \approx W_i - LearningRate \cdot \frac{d}{dx} Cost(W) \quad (3.6)$$

식3.7의 가중치 업데이트식 이다. Y^e 는 집으로 향해 갈 수 있는 우리가 원하는 각도이고 계산을 통해 나온 각도는 Y 이다. $Y^e - Y$ 의 차이만큼 가중치를 업데이트하게 되는데 업데이트를 진행하기 위해 사용되는 Y 의 미분 값을 곱하여 기울기를 통해 방향값을 정하게 된다.

$$Weight(i + 1, M_j) = Weight(i, M_j) + \eta \cdot (Y_p^e - Y_p) \cdot \frac{dY_p}{dw} \quad (3.7)$$

본 식에서 Cost Function의 기울기를 구하기 위해서는 여러 가지 요소들이 필요하다. 위 실험에서 우리는 파노라마 방향의 이미지에 각각의 랜드마크를 맵핑하기 위하여 Cost-

Function이 아크탄젠트 형식으로 사용된다.

$$\frac{dY_p}{dw} = \frac{1}{1+z^2} \cdot \frac{dz}{dw} \quad (3.8)$$

$$z = \frac{\sum Weight_{ij} \cdot Score \cdot X_j}{\sum Weight_{ij} \cdot Score \cdot Y_j} \quad (3.9)$$

$\frac{dY}{dw}$ 의 미분값은 아크탄젠트의 미분이기 때문에 식3.8과 같이 표현되는데 미분에 의해 $\frac{dz}{dw}$ 또한 미분하여야 한다. z 에 관하여 풀게 되면 식3.9와 같은데 이를 미분한 값은 식 3.10이 된다. 식 3.10을 이용하여 최종식인 식3.11을 구하게 된다.

$$\begin{aligned} \frac{dz}{dw} &= \frac{(Score \cdot Y_j) \cdot (\sum Weight_{ij} \cdot Score \cdot X_j)}{(\sum Weight_{ij} \cdot Score \cdot Y_j)^2} \\ &+ \frac{(\sum Weight_{ij} \cdot Score \cdot Y_j) \cdot (Score \cdot X_j)}{(\sum Weight_{ij} \cdot Score \cdot Y_j)^2} \end{aligned} \quad (3.10)$$

$$\therefore \frac{dY}{dw} = \frac{1}{1 + \left(\frac{\sum Weight_{ij} \cdot Score \cdot X_j}{\sum Weight_{ij} \cdot Score \cdot Y_j}\right)^2} \cdot \frac{dz}{dw} \quad (3.11)$$

2.2 실험 셋팅 : 파라미터

HFLV (Haar-like Features Landmark Vector) 경사 하강법을 적용하는 실험을 시행할 때 실험의 조건을 간단하게 설명하면 기본적으로 위치와 크기가 랜덤하게 생성되는 HF (Haar-like Features) 를 100개 사용하였고 Learning Rate는 0.01로 고정하였으며, 가중치의 초깃값은 0과1 사이로 랜덤하게 설정하였다. 업데이트 횟수는 실험의 종류에 따라 5000번에서 10000번을 기본으로 설정하였다.

2.3 경사하강법 적용:타입1

HFLV (Haar-like Features Landmark Vector) 경사 하강법을 적용하는 실험은 기본적으로 HF를 100개 사용하였고 가장 간단한 마스크인 타입 1 형태의 마스크를 사용하였다. 업

데이트 속도 및 파라미터의 개수 등을 고려하였을 때 100개의 마스크를 사용할 때 최대 효율을 얻을 수 있었다.

본 실험에서 경사 하강법을 이용하여 HF (Haar-like Feature) 숫자를 얼마나 줄일 수 있는지 검증하는 실험을 진행하고 가중치와 매칭 스코어 간의 관계 등을 파악하여 좋은 마스크를 선별할 예정이다. 경사 하강법은 기존의 식3.3 에서 가중치 추가된 새로운 식3.12을 따르게 된다. 경사 하강법을 통해 얼마나 개선될 수 있는지 식3.3과 식3.12을 비교하여 실험하였다.

매칭 스코어가 가중치의 효과를 얼마만큼 받아서 업데이트되는지에 대하여 실험하였고 매칭 스코어의 형태마다 분석하여 반복이 진행될 때 경사 하강법이 더 강인하게 작용하는 매칭 스코어 형태를 찾는 실험을 하였다. 그림3.9의 (a)는 HFLV를 적용한 벡터 맵이다. (a)에 경사 하강법을 적용하였을 때 그림3.9의(b)는 HFLV-GD (Haar-like Features Landmark Vector Gradient Descent) 결과를 얻을 수 있었다. 시각적으로도 오차율이 가장 적어진 것을 볼 수 있다.

$$\sum_{j=1}^N \Delta L \cdot W_{i+1j} \cdot \hat{u}_j \quad (3.12)$$

제 3 절 실험2 : 내림차순을 적용한 경사하강법

실험2에서는 경사 하강법을 통하여 구한 결과 또한 내림차순 정렬을 이용한다. 내림차순 정렬을 이용하는 이유는 항목별로 높은값을 가질 때 얼마나 영향력을 가지는지에 대한 평가 실험이다. 최종적으로 그림3.15을 통해 반복 횟수에 따른 각도의 평균 오차 감소 폭을 볼 수 있다.

내림차순 정렬을 적용해보는 항목은 총 4가지 항목이다. 첫 번째 항목은 매칭 스코어를 정렬하여 각각 적용하여 보는 방법이다. 총 3가지 형태의 매칭 스코어가 큰 스칼라값을 가질 때 방향벡터를 만들 때 더 많은 영향력이 있을 것이라고 생각된다. 다음 항목은 업데이트가 끝나고 난 뒤의 가중치 값이다.

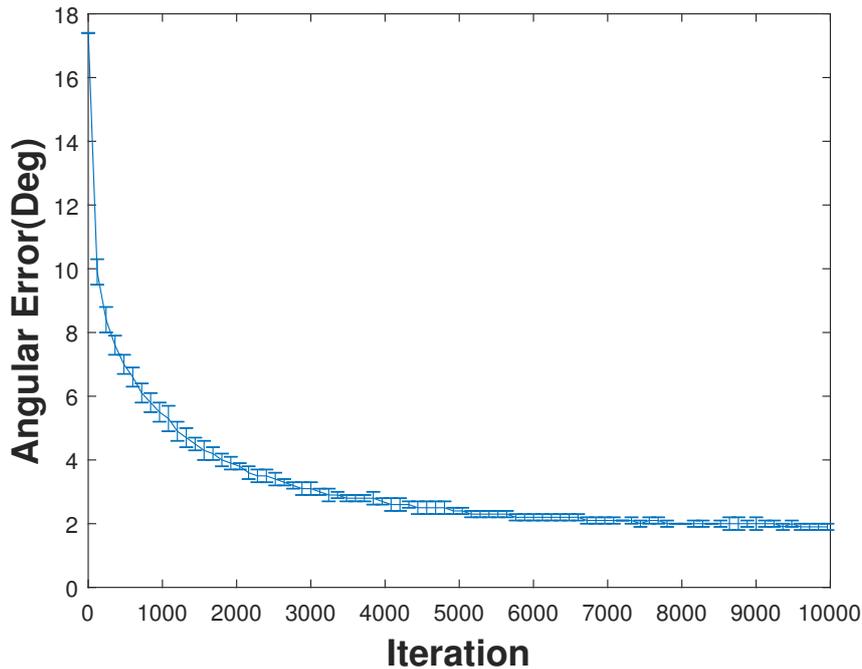
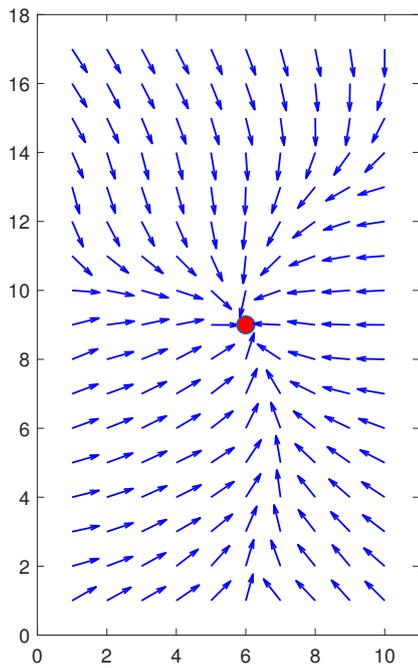


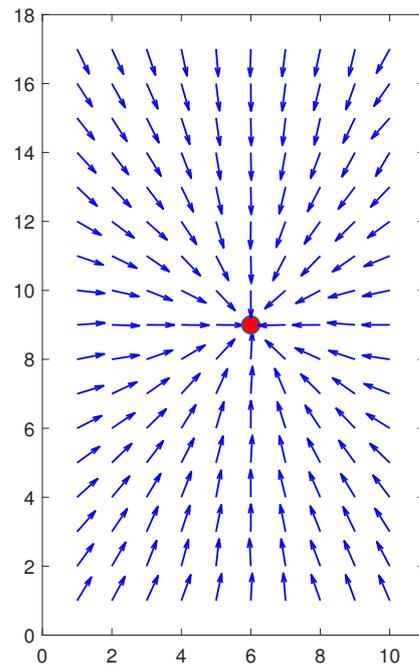
그림 3.8: AE (Angular Error) 그래프. Type1 형태의 HF (Haar-like Feature) 를 100개의 사용하여 10000번 학습을 시켰고 러닝레이트를 0.01로 고정하였다. 업데이트가 지날수록 수행능력이 향상되는것을 관찰할 수 있다.

일반적으로 마스크가 여러 포지션에서 업데이트가 될 때 좋은 마스크일수록 가중치가 높아질 가능성이 있으며 집으로 향하는 벡터를 만들 때 도움이 적거나 방해가 되는 마스크의 가중치는 작아질 것으로 판단된다. 마지막으로 가중치와 매칭 스코어를 곱한 형태의 값을 내림차순 정렬해본다. 본 값을 내림차순 정렬해 보는 이유는 경사 하강법을 사용할 때 매칭 스코어와 가중치의 곱으로 설정되어있고 값을 잘 활용하는지 평가하기 위함이다.

본 실험의 목적은 최대한 적은 마스크를 사용하고 높은 정확도로 성공적인 귀소를 하기 위함이다. 매칭 스코어 파트에서 다루었던 형태별 매칭 스코어가 어떠한 효과를 가졌는지와 어떤 매칭 스코어를 사용해야 안정적으로 귀소 내비게이션을 구현할 수 있는지 실험하였다. 이후 섹션에서 내림차순을 이용하는 방법과 실험을 자세하게 설명한다.



(a)



(b)

그림 3.9: 'a1original' 실험환경의 벡터맵. 가운데 빨간색 마크는 집을 나타낸다. (a) HFLV (b) HFLV-GD를 각각 적용한 결과이다. 가중치는 총 10000번 업데이트를 진행하였고 러닝레이트는 0.01로 고정하고 랜덤하게 생성한 HF (Haar-like Features)는 100개를 사용하였다.

3.1 내림차순을 이용한 마스크선별

우리는 HF (Haar-like Features)가 가지는 매칭 스코어를 내림차순 방법을 이용하여 정렬하고 마스크를 선별할 예정이다. 내림차순을 이용하여 마스크를 선별하는 이유는 식3.3에 의하여 HF의 매칭 스코어의 크기에 비례하여 방향 결정에 영향을 미치기 때문이다. 식에 의하면 매칭 스코어의 크기가 큰 HF일수록 방향 결정에 영향을 많이 미치고, 매칭 스코어의 크기가 작은 HF일수록 방향 결정에 영향을 적게 미친다. 내림차순을 이용하여 매칭 스코어가 큰 HF를 선별하면 영향력이 미미한 HF를 제외하여 사용되는 HF의

개수를 줄일 수 있다고 예상된다.

결론적으로 내림차순을 이용하여 HF를 선별하는 이유는 영향력이 큰 HF만 선별하여 사용되는 HF의 개수를 줄이고 더 적은 정보로 빠르게 방향 벡터를 얻고자 함에 있다.

3.2 여러가지 형태의 매칭스코어

본 섹션에서는 내림차순 정리를 하기 위해서 총 3개의 매칭 스코어 형태를 제시한다. 각각의 매칭 스코어들을 자세하게 설명하면 식3.13은 집 이미지의 매칭스코어이다. 식3.13의 크기가 큰 HF (Haar-like Feature) 들은 집 이미지에서 매칭 스코어가 높은 랜드마크를 생성했을 것이다. 바로 집 이미지에서 밝은 부분을 찾은 HF일 것이다.

다음은 식3.14의 형태이다. HFLV의 기본 계산식과 같은 홈 이미지와 현재 위치의 이미지의 유사도를 나타내는 가장 기본적인 형태의 식이다. 크기가 가장 큰 HF는 집 이미지와 포지션별 현재 위치의 차이가 큰 HF일 것이다. 두개의 차이가 큰 HF의 경우 집 이미지와 다른 포지션에서의 이미지 간의 차이를 분별하는 능력이 뛰어난 HF일 것이다. 하지만 음의 방향을 크기가 커질 수 있고 각각의 HF마다 모든 위치의 매칭 스코어 총합을 사용하게 되면, 세 번째로 식3.15의 형태로 내림차순 된 HF이다. 식3.15은 식3.14와 같이 현재 위치와 집 위치의 매칭 스코어의 차이를 기본으로 하게 되는데 절대값을 취해주는 특징을 가지고 있다. 절댓값을 통해 음의 방향으로 커진 값까지 고려할 수 있고, 절댓값을 취한 뒤 제곱 형태를 통해 큰 값들을 더욱 돋보이게 만드는 특징을 가진 매칭 스코어의 형태이다.

여러 가지 형태별 매칭 스코어의 내림차순을 통해 방향 결정에 더 효율적인 영향을 주는 마스크를 선별할 수 있을 것으로 보인다. 이후 실험을 통해 어떤 스코어 형태가 HF를 더 잘 선별하는지 알아볼 예정이다.

$$MatchingScore1 : |S(I_{home}, M_j)| \quad (3.13)$$

$$MatchingScore2 : |(\sum S(I_{current}, M_j) - S(I_{home}, M_j))| \quad (3.14)$$

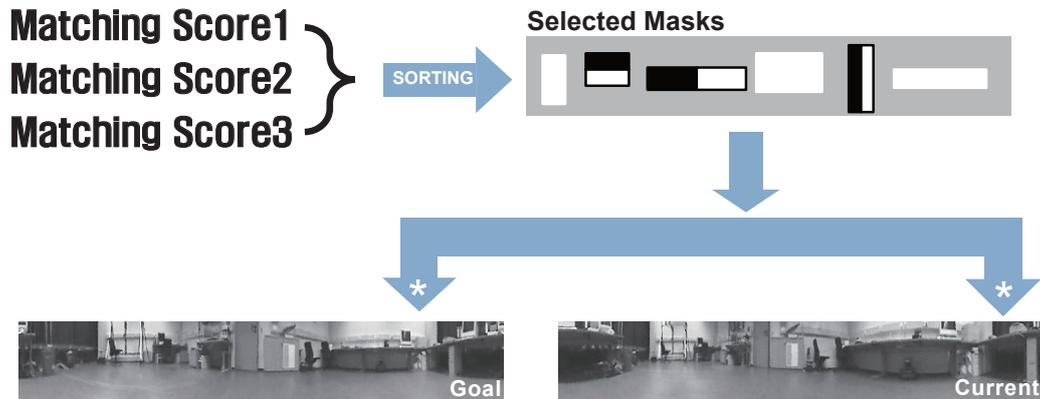


그림 3.10: 매칭스코어별 HF를 선별하는 방법 모식도. 각 형태의 매칭스코어를 내림차순 정리를 통해 선별된 마스크를 이용하여 집 이미지와 현재 위치의 이미지에서 다시 매칭스코어를 구하고 방향결정을 하게 된다.

$$MatchingScore3 : \left(\sum |S(I_{current}, M_j) - S(I_{home}, M_j)| \right)^2 \quad (3.15)$$

그림3.10는선별한 마스크를 이용하여 새롭게 매칭스코어를 계산하는 과정을 도식화 하였다. 여러가지 형태별 매칭스코어를 내림차순하여 HF를 선별한 뒤 새롭게 매칭스코어를 계산하는 과정을 나타낸 그림이다.

3.3 경사하강법 과 랜드마크벡터 비교

그림3.11은 가중치 학습을 사용한 결과는 청록색 육각형을 나타내고 가중치 학습을 사용하지 않은 결과는 검은색 다이아로 표현된다. (a),(b),(c) 각각 다른 형태의 매칭 스코어의 내림차순정렬을 나타낸다.

두 방법 모두 마스크 1개에서부터 100개를 사용하였을 때의 결과이다. 그림3.11을 보게 되면 경사 하강법을 이용하여 학습을 진행한 청록색의 식(9)는 어떤 형태의 매칭

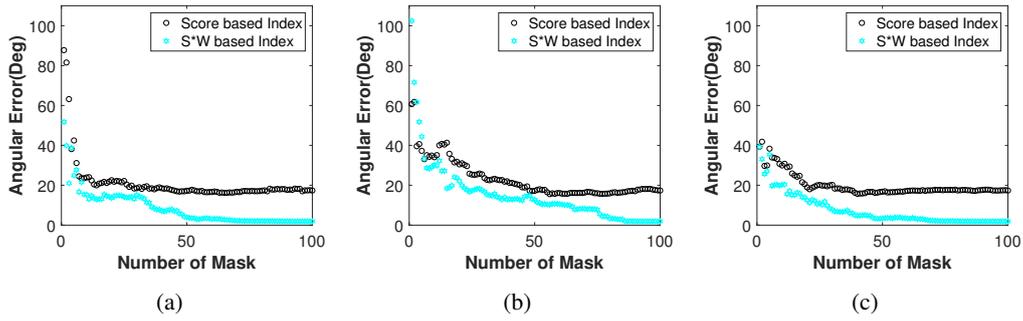


그림 3.11: 세개의 매칭스코어 식에 따른 내림차순 정렬 적용. (a) 매칭 스코어1 (b) 매칭 스코어2 (c) 매칭 스코어3 을 각각 나타낸다. 각각의 방법 마다 100개의 HF를 사용하였고 10000회 반복 학습하였고 러닝레이트를 0.01로 고정하였다.

스코어와 마스크 수가 늘어날수록 모든 위치에의 각도의 평균 오차가 안정적으로 줄어드는 것을 볼 수 있다. 또 하나 확인할 수 있는 것은 매칭 스코어가 높아도 큰 변화폭을 갖지 않는다는 것이다. 같은 내림차순 정렬을 사용하였을 때 경사 하강법을 사용한 방법이 더 효과적인 것을 볼 수 있다. 다른 특징으로 (a) 는 Home 기준으로 만들어진 매칭 포인트를 내림차순 정렬한 것이다. 매칭 스코어가 높은 부분에서 검은색 그래프가 큰 폭으로 평균 오차가 감소하는 것을 볼 수 있다. 하지만 이후 마스크가 늘어남에 따라서 오차율을 늘어나지만, 안정적으로 변화하는 모습을 볼 수 있다.

이전의 가설에 의하여 매칭 스코어가 높으면 방향 결정에 많은 영향을 미칠 것이다. 하지만 높은 마스크 스코어가 항상 좋은 방향으로 가이드 하는지에 대한 의문은 그림12의 실험으로 어느 정도 증명된다. 그림3.11의 (c) 의 경우 가장 단조로운 마스크의 변화를 볼 수 있었다. 하지만 가장 안정적이라고 할 수 있다.

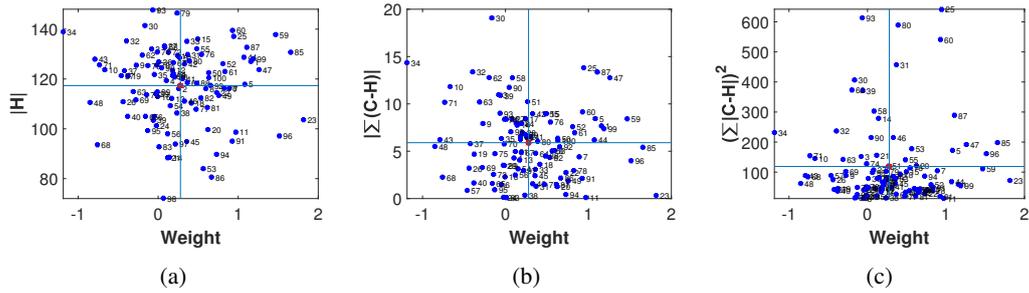


그림 3.12: 형태별로 매칭스코어와 가중치의 분포도를 나타낸 그래프. (a) 매칭 스코어1 (b) 매칭 스코어2 (c) 매칭 스코어3을 이용하여 실험한 결과이다. 각각의 점은 실험에서 지정된 마스크의 번호이다. 가운데 빨간 색 마크는 각각의 실험에 사용된 모든 HF의 매칭 스코어와 가중치의 평균을 나타낸다. 타입1의 HF를 총 100개 사용하였고 10000회 반복 학습하였고 러닝레이트를 0.01로 고정하였다.

제 4 절 가중치 관계 그래프

4.1 관계 그래프1: 가중치 매칭스코어

우리는 가중치와 스코어의 관계를 알아보기 위해 학습이 완료된 마스크에 대해 스코어와 가중치에 대한 그래프를 제시하였다. 각 매칭 스코어의 형태에 따라 분포도가 다르게 나왔다. 우리는 각 가중치의 평균값과 스코어의 평균값으로 임이의 분면을 나누었는데 그림 3.12을 보게되면 분면별 다양한 특징을 가지고 있다. 매칭 스코어에 따라 분면을 나누었지만 각각의 값을 크게 가지고 있던 마스크들은 매칭스코어가 달라져도 속해있던 분면의 변화가 크지 않았다. 이후 세부적인 실험을통해 가중치의 영향을 알아보려한다. 그림3.13는 가중치 평균보다 작은 마스크들을 사용한 결과이고그림3.14 는 가중치 평균보다 큰 분면에 위치해 있는 마스크들을 사용했을때 결과 이다. 본 실험 결과에서 그림3.13 (d)의 결과를 나타낸 가중치의 평균보다 높은 마스크들의 갯수는 대략 32개 정도인데 기존의 100개를 학습하여 만든 결과인 그림3.9의 (b)와 비교하여 각도의 평균 오차가 그다지 큰 차이가 없는것을 관찰 할 수 있었다. 결론적으로 본 실험을 통해 가중치가 업데이트된 마스크를 사용하게 되면 적은 마스크를 사용하더라도 그림3.9의 (b)

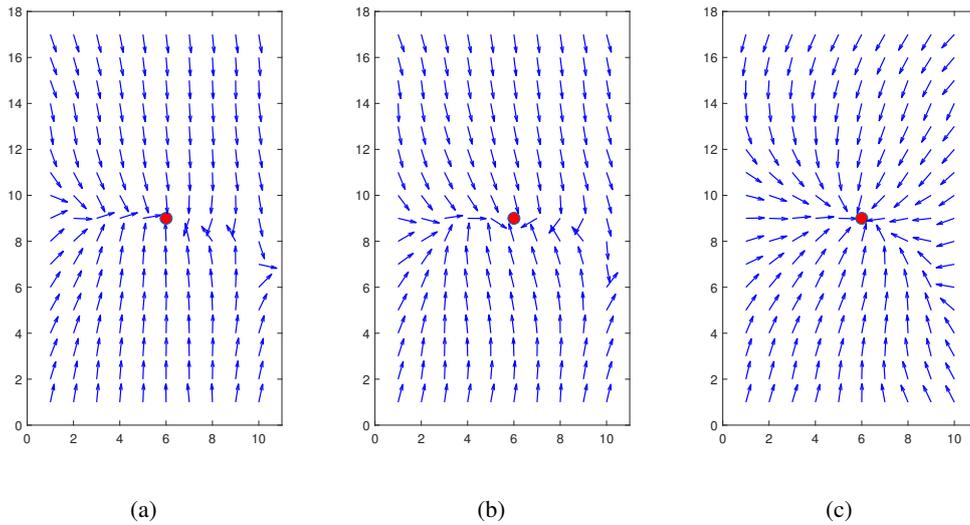


그림 3.13: 세 가지 형태의 매칭 스코어를 이용하여 나타낸 벡터 맵. (a) 집 기준으로 내림차순 스코어 (b) 전체 포지션에서 대하여 누적된 스코어의 절대값 (c) 전체 스코어의 절댓값 형태의 스코어를 누적인 값이다. 각 형태별 매칭스코어의 평균값보다 큰 HF를 선별하여 벡터 맵을 각각 그리고 있다. (c) 형태의 매칭 스코어가 가장 좋은 결과를 나타내고 있다.

t!

표 3.1: 상황별 수행능력 테이블. 총 100개의 HF를 사용하였고 10000회 반복 학습하였고 러닝 레이트를 0.01로 고정하였다.

와 비슷한 결과를 낼 수 있었고 그림3.10의 방법을 통하여 가중치를 통한 분류 작업을 이용하면 마스크를 적게 사용할 방법을 찾을 수 있을 것으로 보여진다.

제 5 절 통합 그래프: 매칭스코어에 따른 내림차순 수행능력

그림3.15은 여러가지 방법들을 내림차순 정렬한 인덱스를 순차적으로 적용한 결과 그래프이다. 특징적으로 볼 수 있는 결과는 Weight 값이 속해있는 방법들이 가장 에러율이 낮았다. 그림3.15에서는 가중치를 이용하여 내림차순 정렬한 초록색과 가중치와스코어의 곱하기 형태를 내림차순 정렬한 마젠타색이다.이후 실험에서 Weight의 효과와 Score

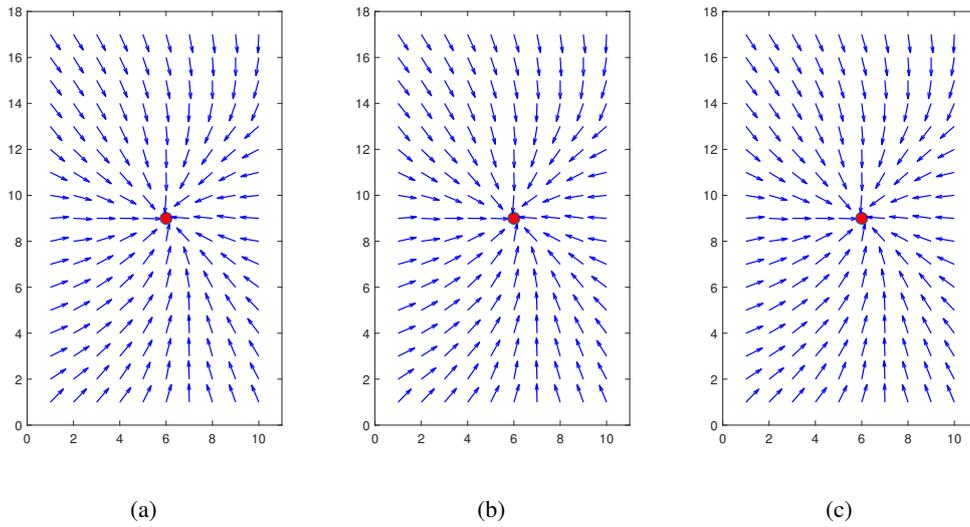


그림 3.14: 세 가지 형태의 매칭 스코어를 이용하여 나타낸 벡터 맵. (a) 매칭 스코어1 (b) 매칭 스코어2 (c) 매칭 스코어3 을 각각 이용하여 실험한 결과이다. 각각의 가중치의 평균값보다 큰 HF를 선별하여 벡터 맵을 각각 그리고 있다.

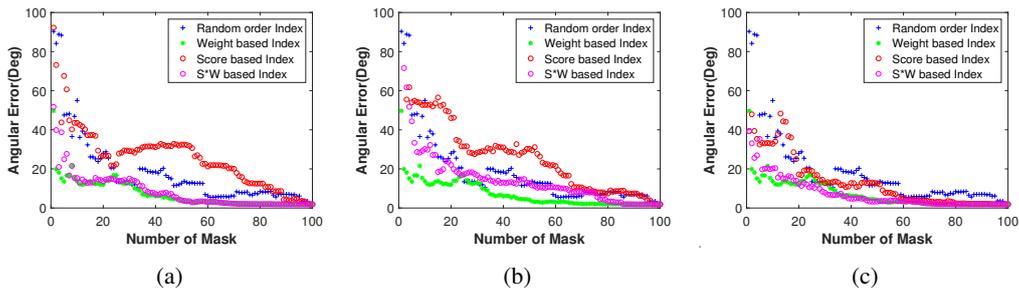


그림 3.15: 여러가지 내림차순 방법 별 수행능력 그래프. 그림에서 그래프는 위 에서 제시 한 세개의 형태의 매칭스코어를 사용하고 있다. 각각의 그래프에서의 스코어는 매칭스 코어를 뜻한다. 여러가지 방법으로 내림차순 정렬을 하였을때 마스크 갯수별로 각도의 평균 에러를 한눈에 볼 수 있다.

에 따른 Weight의 변화를 관찰할 예정이다.

Weight가 높에 업데이트된 마스크 경우 대부분의 포지션에서 좋은 역할을 할것이라고 추측된다.이전에 가설에 따라 매칭스코어를 형태에 따라서 다른 효과를 보았다. 이

후 실험에서는 조금더 적은 정보를 가진 상태에서도 목표 위치를 결정짓는 방향벡터가 잘 형성되는지에 대해서 실험하였다. 그림3.15는 100개의 마스크를 여러 가지 방법으로 내림차순 정렬하여 마스크의 개수 별로 각도의 평균에러를 나타낸 그래프이다. 해당 그래프에서 내림차순 방법을 이용하는 이유는 각각의 항목에 대해 마스크가 큰 값을 가질 경우 해당 항목에 대한 영향력을 많이 가지고 있는 것이고 마스크를 전부 사용하였을 때 보다 영향력을 많이 가지고 있는 마스크를 먼저 사용하였을 때 결괏값을 알아보기 위해서다.

대부분의 실험 결과에서는 내림차순 정리를 하여 해당 방법에 대해서 큰 값을 가지면 가질수록 빠르게 결과에 도달함을 보였다. 이전 챕터와 본 챕터에서는 3가지 형태의 매칭 스코어에 대해서 실험을 진행하였기 때문에 그림3.15에서도 3가지 형태의 매칭 스코어에 대해서 각각 그래프를 제시하였다.

제시된 그래프를 보게 되면 마스크의 매칭 스코어를 내림차순인덱스 같은 경우는 매칭 스코어의 형태별로 차이를 보이게 되는데 가장 안정적인 모습을 보이는 것이 (c) 그래프이다. (c) 에서 사용된 매칭 스코어의 공식은 식3.15인데 이전의 실험 결과에서도 집 위치의 이미지와 현재 위치의 이미지간의 변화량을 기하학적으로 표현하였다. 그림3.15에서 (c) 그래프에서 쓰인 형태의 매칭 스코어를 높게 가지기 위해서는 마스크를 통해 여러 포지션에서 집 이미지와 비교하였을 때 많은 변화를 관찰 후 있어야만 한다. 많은 실험을 바탕으로 결론을 내리면 많은 변화를 관찰할 수 있는 마스크가 가장 좋은 퍼포먼스를 내는 마스크이다.

따라서 그림3.15의 (c) 가 세 개의 매칭 스코어 중 가장 빨리 오차를 감소시키는 것을 볼 수 있다. 세 개의 그래프를 비교해 보게 되면 가중치가 곱해져있는 방법들은 성능에 큰 차이가 없고 오차율의 변화가 비교적 평탄한 것을 볼 수 있는데 이는 가중치가 각도의 평균 에러를 감소 시는 것에 큰 영향을 미친다는 것을 볼 수 있다.

세개의 그래프 전부에서 빨간색 동그라미인 가중치만으로 생성된 각도의 평균 오차와 마젠타색 동그라미인 스코어와가중치의 곱 형태로 생성된 각도의 평균 오차를 비교해 보게 되면 세개의 그래프 전부 큰 차이를 보이지 않는다. 이는 앞선 실험들과 함께 학습이

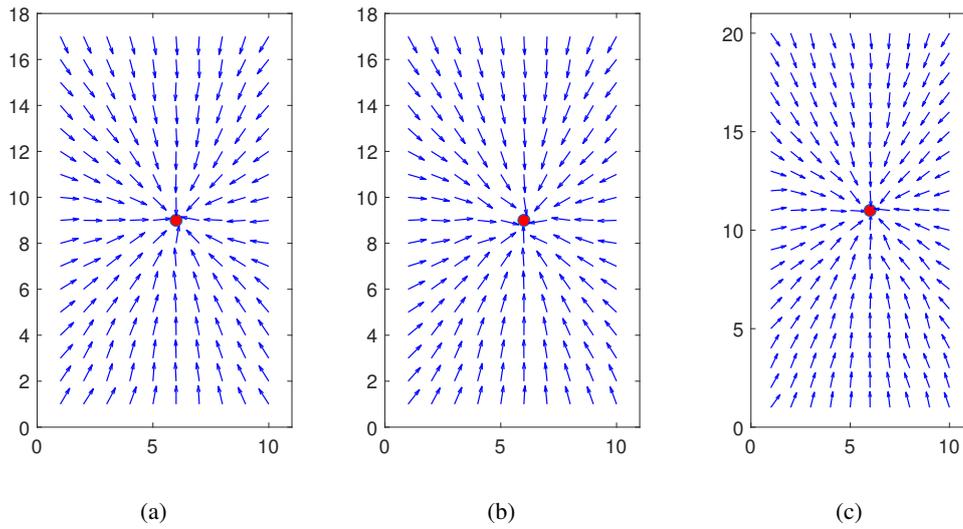


그림 3.16: 여러가지 실환경에서 HFLV-GD를 적용한 벡터맵. 각각 실험환경 (a) arboreal (b) screen (c) Hall 실험환경에서 각각 실험을 진행하였다. Type 1 사용, HF (Haar-like Feature) 100개 사용, 5000회 반복, 러닝레이트 0.01로 고정하였다.

Type 1 100EA		a1original	Arboreal	Screen	Hall
HFLV	Angular Error Deg(std)	21.4(3.2)	46.3(5.8)	32.5(5.6)	69.1(11.2)
HFLV-GD		2.3(0.1)	3.1(0.3)	4.3(1.1)	4.1(0.3)

표 3.2: 환경별 수행능력 테이블. 총 100개의 HF를 사용하였고 10000회 반복 학습하였고 러닝 레이트를 0.01로 고정하였다.

된 이후에는 매칭스코어보다 가중치가 더 큰 역향력을 가지고 있다는것을 전적으로 보여준다.

5.1 다른 환경에서 경사하강법 적용

이전 섹션의 연구는 장애물이 없는 가장 기본적인 환경인 Vardy의 'a1origin' 환경에서 실험하였다. 'a1origin'의 환경 같은 경우에는 가장 기본이 되는 환경이기 때문에 가장 좋은 수행능력 볼 수 있었다. 가장 수행능력이 좋지 못했던 환경은 Hall 환경이었다. Hall 환경 같은 경우 이미지의 해상도가 가장 낮고 어두운 환경이었으며 이전의 HFLV또한 가장

낮은 퍼포먼스를 보였다. Screen과 Arboreal은 장애물이 있는 환경이다. 각각 그림3.16 기준으로 (a)와 (b)이다. (a)는 벡터 맵 기준으로 왼쪽에 스크린이 설치되어 있고 (b)는 중간 부분에 나무가 있다. 하지만 해당 부분에 장애물을 문제없이 통과하고 있다. 모든 위치를 경사 하강법으로 학습하였을 때는 비교적 평탄하게 AE가 감소하는 것을 보였다.

제 6 절 요약

본 챕터에서는 매칭스코어의 형태별로 마스크를 내림차순으로 정리하였을 때 매칭스코어의 크기와 형태에 따른 수행능력의 변화를 알아보는 실험을 하였다. HF (Haar-like Feature)를 이용하는 방법은 간단한 유사도 판단에 적절한 방법이다. 그 이유는 계산이 간단하고 비교적 사용하기 쉬운 방법이기 때문이다. HF는 이미지간의 간단한 화소의 평균 명암을 비교하여 간단하게 판단 할 수 있다고 본다. 이는 본 논문에서 제시한 HF를 이용한 귀소 내비게이션에도 적용된다.

이전의 연구진들은 HF를 랜덤하게 생성하여 랜드마크를 만들어서 이미지간의 유사도를 비교하여 내비게이션을 하는 HFLV (Haar-like Landmark Vector)에 적용하였고 다른 알고리즘들과 비교하여 성과를 얻었다. 본 논문에서는 간단한 머신러닝 알고리즘을 적용하여 HF를 학습시킨다. GD (Gradient Descent)알고리즘을 HFLV에 적용하여서 간단한 피드백을 통해 수행능력을 향상시킨 HFLV-GD (Haar like Feature Landmark Vector-Gradient Descent) 알고리즘을 제시하였다.

본 논문에서 실험을 통해 세가지 사실을 밝힐 수 있었다. 첫번째는 머신러닝 알고리즘인 GD (Gradient Descent) 알고리즘을 통해 학습한 결과 더 좋은 수행능력의 HFLV-GD 알고리즘을 얻을 수 있었다. 두번째 학습의 이후의 가중치의 변화를 관찰 하였을 때 HF의 크기와 상관 없이 가중치가 높은 값을 가지는 HF이후 수행능력에 많은 영향력을 가졌다는 것을 알 수 있었다. 세번째 HF가 높은 수행능력을 가질 때 HF의 생성 위치를 분석해본 결과 높은 가중치를 가지는 HF를 모아서 정렬하였을 때 HF가 위치하는 위치는 특정 지역의 유사도 측정에 유리한 위치가 아닌 전 지역에 쓰여도 유사도 측정이 잘 동작하는

HF라는 것을 관찰 할 수 있었다.

본 논문에서 이러한 분석결과가 가지는 의미는 특정 HF를 생성할 때 위의 분석결과를 고려하여 생성한다면 적은 연산량으로 질 좋은 HF를 선별하고 높은 수행능력의 내비게이션을 구현할 수 있을것이라고 생각된다. 따라서분석을 통해 HF의 특징을 잘 파악하게 되면 가장 간단하게 사용할 수 있는 랜덤한 생성 형태에서 벗어나 마스크를 생성하는 사람이 직접 조절하여 좋은 마스크를 생성할 수 있을 것으로 보인다. 이후 챗터에서 학습의 효과와 영향을 밝힐 예정이다.

제 4 장

HFLV-GD의 다양한 파라미터 학습

4장에서는 반복실험을 통한 교차검증 시행하고, 여러가지 환경에 HFLV-GD (Haarlike Feature Landmark-Gradient Descent)를 적용하여 파라미터별 가중치 업데이트의 결과를 평가하였다. HFLV-GD을 구현할때 집 위치를 제외한 나머지 포지션에서 학습이 된다. 본 챕터에서는 학습을 할 때 사용되는 정보의 량이 얼마만큼 필요한지를 분석하는 실험을 진행하였다. 만일 적은 정보를 이용하였지만 목표 위치에 잘 도달 한다면 좋은 좋은 내비게이션이라고 할 수 있다.

본 챕터에서 진행되는 여러가지 실험에서 두개의 파라미터를 중점적으로 관찰 하였다. 첫번째는 학습되는 HF (Haar-like Feature)의 갯수이다. 이전에 실험된 HFLV의 경우 HF를 많이 사용하게 되면 오차가 줄었는데, HFLV-GD를 구현 하였을때 HF의 갯수가 수행능력에 얼마나 영향을 미치는지 알아보았다. 두번째로 관찰하는 파라미터는 포지션의 위치이다.

일반적인 결과로 학습을 시행할때 많은 포지션의 정보를 이용할 수록 수행능력이 좋은것으로 추측하였는데, 포지션의 위치와 갯수별 실험을 통해 상황별 수행능력들 간의 상관관계를 분석하였다. 또한 실험환경별 상황에 맞추어 학습데이터와 검증 데이터를 나누어 교차검증 방식을 택하여 실험을 진행하였다[40]. 위 연구의 일부 또는 확장된 내용에 대해서는 다음 논문지에 게재될 예정이다[50].

제 1 절 정보의 양에 따른 수행능력 평가

이전의 실험들은 충분한 마스크의 갯수와 모든 포지션을 정보들을 고려하여 학습을 시행하였다. 본 섹션에서는 2가지 파라미터에 대해서 고려해본다. 첫번째 파라미터는 사용되는 HF (Haar-like Feature)의 갯수이다. 적은 HF를 사용할 수록 연산량이 적어지게 된다. HF의 갯수별 수행능력을 비교해 보고 가장 좋은 효율적인 수행능력을 나타내는 HF의 갯수를 찾아보는 실험을 하였다. 수행능력에서 가장 크게 고려해야할 부분은 수행되는 총 반복 횟수 동안에 줄어드는 각도의 평균 오차값이다.

본 논문에서는 마스크의 평균 오차값을 AE (Angular Error)로 표현한다. 같은 조건에서 시행하였을때 마스크의 갯수별로 포화가 시작되는 반복횟수가 다를것으로 예상되고 이것을 알아보는 실험을 진행 하였다. 두번째 파라미터는 학습이 되는 포지션의 갯수이다. 많은 포지션에서 정보를 얻고 학습을 하게 되면 정확도가 올라가게 된다. 하지만 실제 로봇은 모든 포지션의 정보를 알수 없을것이므로 최소 포지션 또는 적절한 갯수의 포지션 만으로도 학습이 가능하지 실험을 통하여 분석하였다.

포지션의 선택적인 학습을 하더라도 좋은 퍼포먼스를 구현할 수 있는지 실험하였다. 이후 교차검증을 통해 해당 부분을 검증하는 과정을 진행하였다[40].

1.1 교차 검증

모델에 학습을 적용하다 보면 파라미터를 셋팅하는 일은 쉬운일이 아니다. 또한 트레이닝 데이터와 테스트 데이터를 나눌때 과도적합(Overfitting)이 생길수 있게 되어 정확한 평가와 학습이 어려울때가 있다. 우리는 교차 검증을 이용하여 학습을 더욱더 강인하게 만들어주는 HyperParameter를 찾을 수 있다. 또한 학습데이터와 테스트데이터가 적절하게 설정되어 있는지 교차검증을 통해 평가할 수 있다[40].

본 연구진은 일반적인 교차검증중 k-Fold 교차검증을 시행하였다. 그림 4.1의 모식도를 보게되면 데이터를 총 K 개로 나누게 된다. 실험마다 데이터를 상황에 맞춰 섞어 가면서 실험하였고 실험의 종류에 따라 실험의 횟수또한 다르게 시행하였다. 포지션의

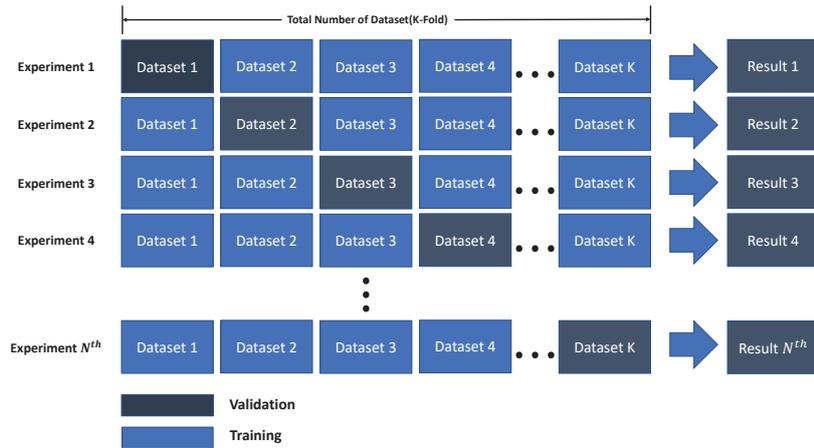


그림 4.1: k-fold 교차검증에 대한 모식도. 데이터 사이즈에 따라 k값을 다르게 실험을 진행한다.

갯수별 학습을 할때 Example의 갯수도 달라지며k값 또한 달라진다.

1.2 마스크 갯수별 수행능력 평가

본 실험 마스크의 갯수별 수행능력에 대하여 여러가지 실험을 통해 평가하였다. 이전에 연구된 HFLV (Haar-like Feature Landmark Vector) 모델에서는 많은 마스크를 사용할수록 상대적 좋은 결과를 얻을 수 있었다. 하지만 경사하강법으로 학습을할때는 마스크의 갯수가 늘어남에 따라 연산의 복잡도 또한 늘어난다[62]. 본 실험을 통해 마스크의 갯수에 따라 보여지는 퍼포먼스를 보도록 한다. 위 실험에서 보게 되면 마스크가 늘어나게 되면 학습이 전반적으로 포화지점이 늦게 형성되는것을 볼 수 있다. 하지만 마스크가 너무 적어지게 되면 학습의 결과가 좋지 못하거나 제대로 학습되지 않는다.

본 파트를 통해서 실험할때 가장 이상적인 마스크의 갯수를 알 수 있었는데 이후 가장 좋은 조건을 사용하여 실험을 하였다. 마스크의 숫자에 대비하여 업데이트 속도를 고려 하였을때 항상 마스크가 많을수록 좋은 결과는 가져오진 않았다. 알맞은 갯수의 HF를

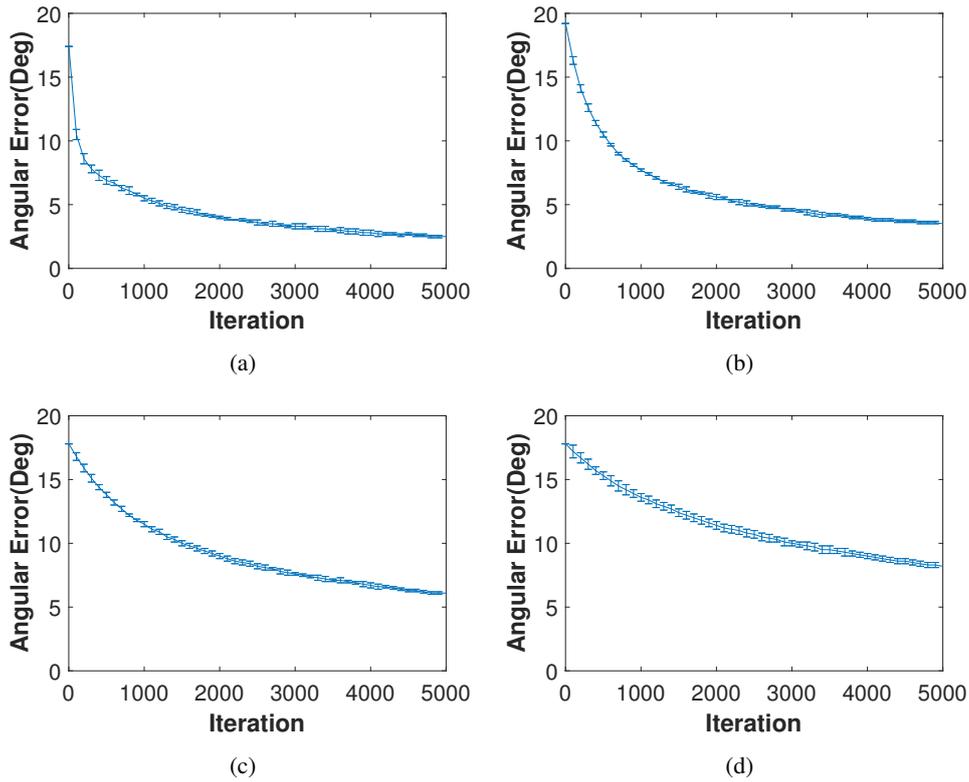


그림 4.2: HF의 갯수별 수행능력 그래프. (a) 100EA (b) 500EA (c) 1500EA (d) 3000EA의 HF를 이용하여 학습을 진행하였고 평균과 표준편차로 에러바 그래프를 표현하였다. 전부 동일하게 'a1original' 환경에서 타입1을 사용하였고 5000회 업데이트 통해 학습을 수행하였다.

가중치 업데이트를 실행하였을때 좋은 결과를 얻을 수 있었다. 그림 4.2을 기준으로 실험에서 사용할 수 있는 가장 효율적인 HF의 갯수는 100개로 판단되며 이후 실험에서는 HF를 100개 사용하는것을 기준으로 실험들을 진행 하였다.

1.3 제한된 위치별 학습과 테스트

경사하강법을 이용하여 학습을 할때 학습을 하는 위치는 결과에 중요한 영향을 미치게 한다. 경우에 따라서 학습 지역과 테스트 지역의 거리가 멀거나 정 반대에 있을때 등

여러가지 위치를 제한시켜 학습하는 실험을 하였다.

본 실험을 통해 집 위치를 제외한 169개의 위치가 어떤식으로 선택되어야 좋은 학습과 테스트결과를 얻을 수 있는지 알 수 있다. 그림4.3 의 결과를 보게 되면 벡터맵에서 빨간색 점선 벡터는 학습에 이용된 위치이고 검은색 실선 벡터는 테스트에 이용된 위치이다. 그림4.4는 트레이닝셋에 대한 각도의 평균 오차는 낮아지는 반면에 테스트셋의 각도의 평균 에러는 많이 개선되지 않는다.

그림4.4은 지역적 한계를 가진 학습방법을 여러번 시행하여 에러바를 나타냈다. 학습 데이터의 경우 AE가 감소함을 볼 수 있었다. 테스트 데이터의 경우는 두드러지는 변화를 나타내지 않았다. 하지만 전체적으로 AE가 감소하는 경향을 보였는데 이후 진행된 포지션의 갯수별 실험과 비교분석을 해보게 되면 지역적인 한계를 가진 학습이 상대적으로 수행능력이 떨어지는것을 볼 수 있다. 그림4.4에서 두 그래프의 공통점은 특정한 지역에 한정된 학습을 시행하였고 학습데이터의 AE가 줄어드는 반면에 테스트 데이터는 AE가 눈에 띄게 줄어들지 않는다.

본 실험을 통해 학습을 할때 주의할 점은 지역적인 컨디션을 고려하여 학습을 해야한다는 것을 알 수 있다. 이후에 지역적인 구분을 짓지 않고 포지션의 갯수에 따라서 어떤 학습이 되는지 알아보는 실험을 시행하여 비교하였다.

1.4 포지션의 갯수별 수행능력 평가

이전 실험들은 170개의 포지션중 집을 제외한 169개의 포지션에 대해서 중복없이 랜덤하게 학습하여 각 포지션 마다 가중치가 학습되게 하였다. 우리는 제한된 갯수의 포지션에서 학습을 진행 한 뒤에도 모든 포지션에서 잘 작용하는지 실험하였다. 이 실험의 목적은 목표는 효율적인 수행능력의 최소갯수를 찾는 것이다. 적은 갯수의 포지션만을 참고 하여 학습하여도 얼마나 잘 목표 위치로 이동할 수 있는지 실험하였다. 가중치가 학습되는 포지션의 갯수는 각각 50퍼센트,25퍼센트,10퍼센트,5퍼센트로 랜덤하게 포지션을 선택하여 실험하였다.

본 실험은 가장 적은 갯수의 포지션으로 업데이트한 가중치를 이용하여 최대한 많은

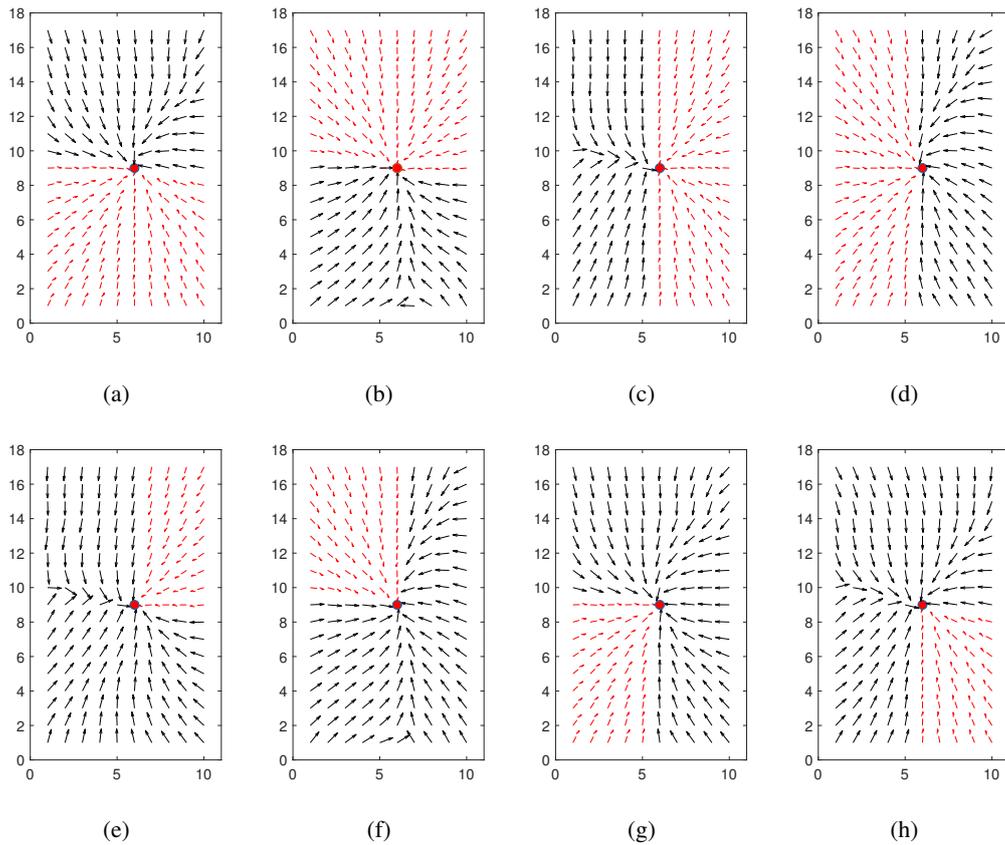


그림 4.3: 'aloriginal'환경에서 절반의 포지션을 사용했을때 벡터맵. 타입1의 HF 100개 사용하였고 5000회 업데이트 통해 학습을 수행하였다. 위 그림은 학습데이터와 시험데이터를 각각 빨간색 점선의 벡터와 검정색 실선의 벡터로 나타낸 결과이다. (a), (b), (c), (d)는 전체 포지션의 절반을 학습데이터로 사용하였다. (e), (f), (g), (h)는 전체 포지션의 사분의 일을 학습데이터 사용하였다.

포지션의 방향벡터의 정확성을 높이는 것이 목적인데 실험을 통한 결과로 가장 효율적인 갯수를 찾을 수 있었다. 그림4.5은 임의의 선택된 포지션의 갯수별 벡터맵의 결과이다. 가장 선택이 많이된 (a)와 (d)의 변화폭이 제일 크다. 이것은 (d)와(f)를 비교해 보았을때 가장 크게 느낄 수 있다. 그림4.6의 (a)의 결과가 가장 좋다.그림4.6의 (a)는 모든 포지션의 정보를 이용하여 실험한 결과이고 (a)와(d)를 비교해보면 그 차이를 비교할 수 있다. 그 이유는 모든 포지션에 부합하는 마스크들을 가중치를 통해 업데이트 했기 때문이다.

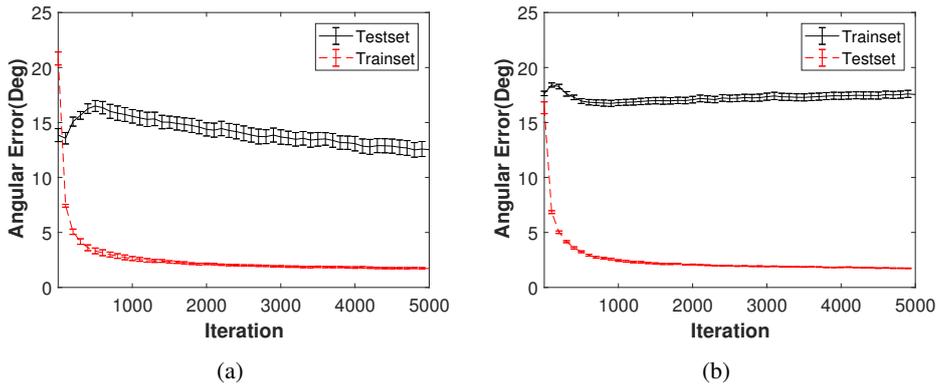


그림 4.4: 'aloriginal' 환경에서 포지션의 절반을 사용했을때 수행능력 그래프. 타입1의 HF 100개 사용하였고 5000회 업데이트 통해 학습을 수행하였다. (a) 는 절반에 해당되는 지역들을 학습하였고 (b) 는 사분의일에 해당되는 지역들을 각각 학습한 결과이다. 상황별 수행한 결과를 평균과 표준편차로 에러바 그래프를 표현하였다.

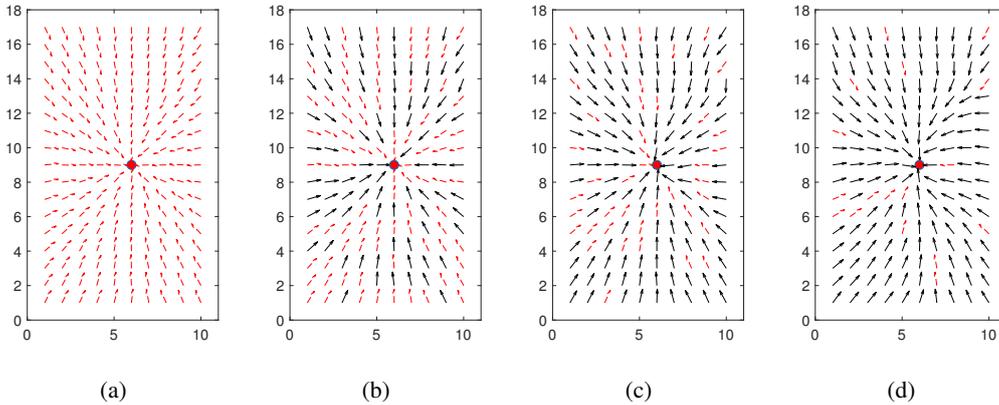


그림 4.5: 'aloriginal' 환경에서 포지션의 갯수별 벡터맵. 타입1의 HF 100개 사용하였고 5000회 업데이트 통해 학습을 수행하였다. 학습데이터의 갯수별 테스트 데이터의 벡터맵 빨간색 점선은 학습포지션 ,검정색 실선은 테스트 포지션 각각 (a) 100퍼센트 (d) 50퍼센트 (c) 25퍼센트(d) 10퍼센트 의 포지션을 임의로 선정하여 학습결과의 벡터맵이다.

여기서 우리는 정보가 부족한 상황 과 정보 획득이 제한된 상황에서도 귀소 내비게이션을 구현할 수 있는지 실험을 통해 평가할 수 있게 된다. 그림4.6의 다른 결과들을 보게 되면학습하는 포지션이 반씩 줄어들때 마다 AE는 약 2도 정도의 최종 수행능력이

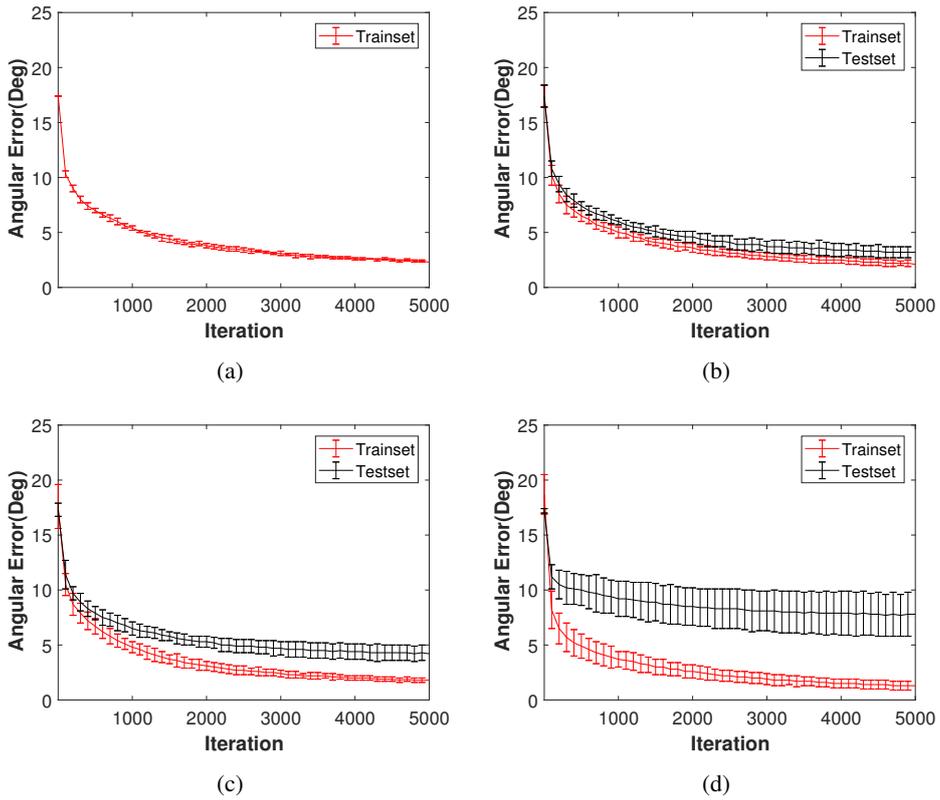


그림 4.6: 'aloriginal' 환경에서 포지션의 갯수별 수행능력 그래프. (a) 100퍼센트 (d) 50 퍼센트 (c) 25퍼센트(d) 10퍼센트 의 포지션을 임의로 선정하여 학습한 결과를 에러바 그래프로 나타내었다. 타입1의 HF 100개 사용하였고 5000회 업데이트 통해 학습을 수행하였다. 학습데이터의 갯수별 테스트 데이터의 벡터맵 빨간색 점선은 학습포지션 ,검정색 실선은 테스트 포지션 각각 나타내었다.

줄어드는 것을 알 수 있다.

1.5 집 주변 포지션 학습 : 4EA , 8EA

본 논문에서 개미의 귀소능력에 영감을 받아 연구를 진행하였다. 만일 개미가 효율적인 귀소 능력을 가졌다고 한다면 집을 떠나 먹이사냥을 시작하기 이전에 집의 가장 가까운 주변을 학습한 뒤 그 정보를 바탕으로 다시 집으로 돌아올수 있을 것이다. 또한 이러한

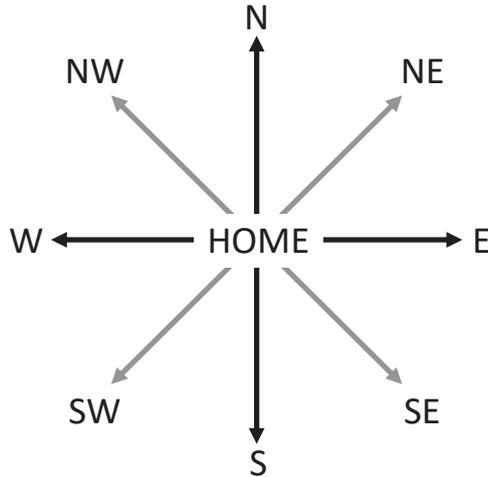


그림 4.7: 집을 중심으로 동서남북 모식도. 북동,북서,남동,남서인 각각 4방향과 8방향을 학습 데이터로 사용하였다.

성질을 모바일로봇으로 구현하게 된다면 그림4.7과 같이 집 기준으로 적게는 동서남북 많게는 필요에 따라서 북동,북서,남동,남서를 학습하여 집으로 잘 돌아올 수 있게 된다면 효율적인 내비게이션 방법이라고 할 수 있다. 우리는 먼저 그림4.8와 같이 동서남북 4방향의 정보만을 학습하여 얼마나 좋은 정확도를 유지하거나 개선할 수 있는지에 실험하였다. 그림4.8에서 실험들은 전부 학습한 포지션에 대한 결과는 빨간색 점선벡터로 나타냈고 테스트 포지션에 대해서는 검은색 벡터로 나타냈다. 이후 실험들도 동일하게 표기하였다.

본 실험에서는 집 근처의 포지션을 학습한 결과를 상대적 비교 분석을 하기 위해 임의의 포지션 선택하여 학습한 결과를 이용하였다. 그림4.8의 (a)를 보게되면 집 주변에 학습데이터는 목표지점에 비교적 정확하게 도달하는것을 볼 수 있다. 반면에 집 포지션에서 멀어질 수록 정확도가 조금씩 떨어지게 된다. 그림4.8에서 (a)와(b)를 비교해 보면 (b)의 랜덤하게 포지션이 선택된 결과가 집으로 향하는 방향 결정력이 비교적 좋은 것을 알 수 있다. 그림4.9의 (a)와 (b)의 결과를 비교해 보면 조금더 명확하게 알 수 있다. 그림4.9 (b)의 결과가 조금더 좋은것을 알 수 있다. (b)의 에러바의 크기로 볼때 적절한

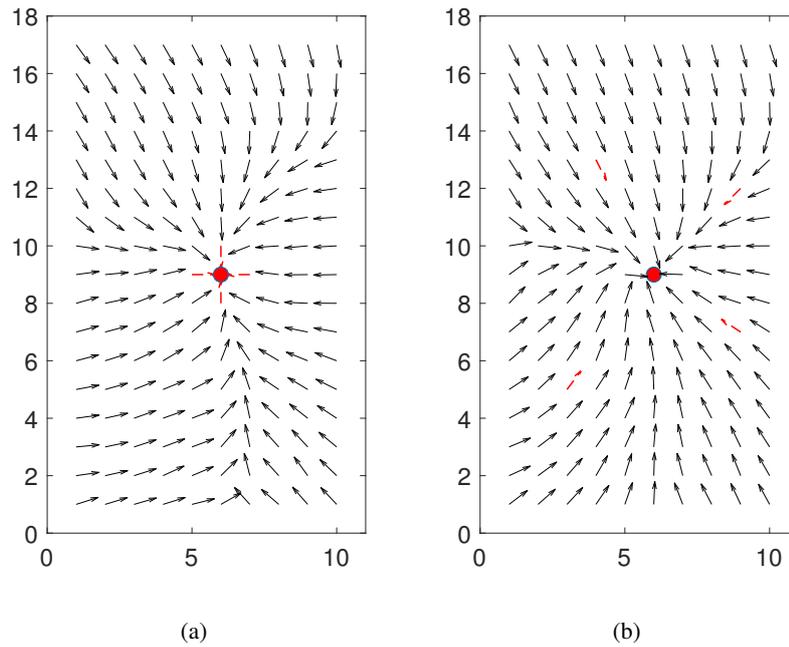


그림 4.8: 'a1original' 환경에서 동서남북으로 학습 결과와 랜덤학습 결과 벡터맵. 각각 (a) 고정된 4개의 포지션 학습결과 벡터맵 (b) 임의로 지정된 4개의 포지션 학습결과를 나타낸다. 100개의 HF를 사용했으며 5000회의 반복학습을 진행하고 러닝레이트는 0.01로 고정하였다.

포지션이 학습되었을때 우리는 그림4.9 (a)에 비하여 명확하게 더 좋은 결과를 얻을 수 있다.

이와같은 결과는 그림4.10에도 동일한 양상을 보인다. 그림4.11을 보게되면 (b)의 결과가(a)에 비하여 좋은것을 볼 수 있다. 그림4.11 (a)의 에러바가 큰 이유는 이전 실험에서 증명한것과 같이 포지션이 선택될때 지역적인 분포에 영향을 받을것으로 추측된다.

제 2 절 여러가지 환경환경에 적용한 경사하강법

본 논문에서 우리는 Vardy의 데이터셋을 사용하였다[107]. 앞서 진행한 대부분의 실험들은 'a1original' 에서 실험을 진행하였다. 'a1original' 환경은 빛과 장애물에 대해서 가장

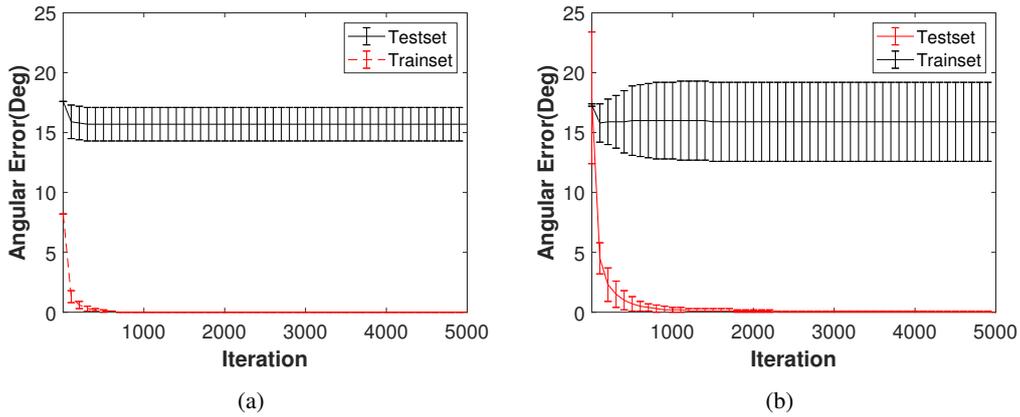


그림 4.9: 'a1original' 환경에서 동서남북으로 학습 결과와 랜덤학습 결과 수행능력 그래프. (a) 4개의 고정된 포지션을 학습한 결과를 나타내는 AE(Angular Error) 그래프 (b) 4개의 임의로 선택된 포지션을 학습한 결과를 나타내 AE(Angular Error)그래프

보편적인 실험환경이다. 본 섹션에서는 빛,장애물,저해상도 등의 환경적 요소를 다르게 적용하여 실험환경별 HFLV-GD 알고리즘을 테스트해볼 예정이다.

이전의 연구진부터 스냅샷 모델과 HF(haar-like Feature)는 여러가지 돌파해야할 문제를 가지고 있었다. 스냅샷 모델의 대표적으로 거론된 문제는 음영지역에 의한 한계였다. 스냅샷모델은 스냅샷 이미지상의 시야에서 장애물에 의한 음영 과 이미지 변형이 수행능력에 영향을 주는 경우가 있었다. HF의 경우 빛에 대해서 민감한 실험결과들을 나타냈다. 때문에 얼굴인식 또는 물체인식을 할때 강한 빛에 의해 구별이 어려울 정도의 밝은 환경이거나 빛이 적어 너무 어두운 환경에서는 수행능력이 떨어지는 경향을 보였다. 본 연구진이 제시한 알고리즘의 성능 테스트를 위하여 앞서 제시한 문제들을 가진 실험환경에서 테스트를 진행하였다. 우리는 Vardy의 데이터셋을 기준으로 'a1original' 이외의 환경들을 이용하게 된다. 장애물이 있는 환경으로는 'arboreal'과'screen'환경이 있다. 두 환경은 모두 장애물에 대한 환경이고 환경마다 장애물의 종류가 다르다. 'screen'의 벡터맵 기준으로 왼쪽편의 스크린 장애물이 존재한다. 'arboreal'은 벡터맵의 중앙부분에 나무 장애물이 있다. 또 다른 환경으로는 'hall'환경이 있다. 'hall'환경의 경우 복도

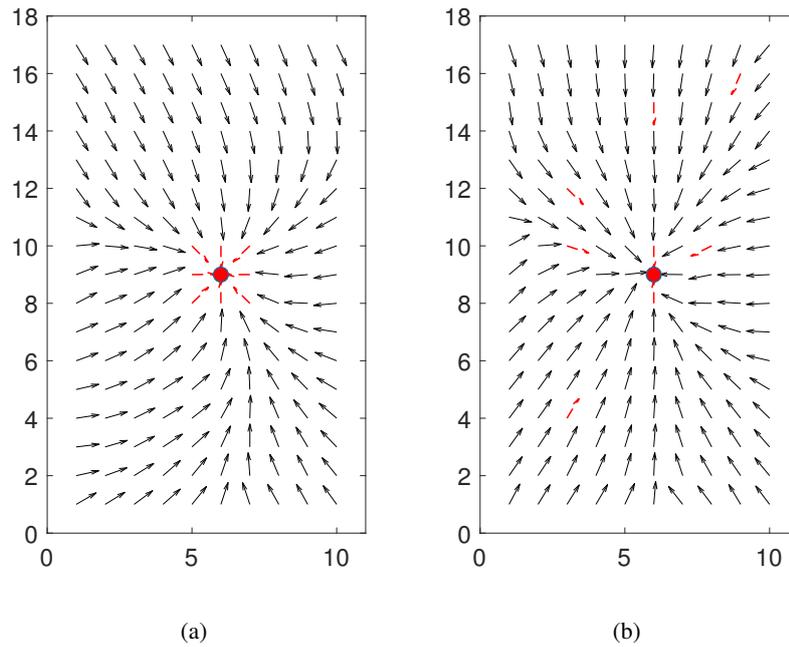


그림 4.10: 'a1original' 환경에서 집 주변 8개 위치 학습 결과와 랜덤학습 결과 벡터맵. 100개의 HF를 사용했으며 5000회의 반복학습을 진행하고 러닝레이트는 0.01로 고정하였다. 각각 (a)고정된8개의 포지션 학습결과 벡터맵 (b) 임의로 지정된 8개의 포지션 학습결과를 나타낸다.

의 어두운 환경이고 이미지의 해상도가 낮은 환경이다.

우리는 HFLV-GD(Haarlike Feature Landmark Vector -Gradient Descent) 알고리즘이 다른 환경에서도 적용이 잘 되는지 실험해볼 예정이다. 그림4.12은 다른 실험환경에 대한 Panoramic 이미지다. 그림4.12 의 (a)와(b)의 빨간 마킹은 이미지마다 가지고 있는 장애물을 마킹하였다. 이번 섹션에서는 여러가지 환경에서 다양한 조건으로 실험한다.

2.1 장애물 (Obstacle) 이 있는 환경: 'arboreal','screen'

가장 먼저 장애물이 있는 'arboreal'환경 과'screen'환경에 대해 실험하였다. 두개의 환경에서 기본적인 알고리즘인 HFLV(Haarlike Feature Landmark Vector) 과 GD(Gradient

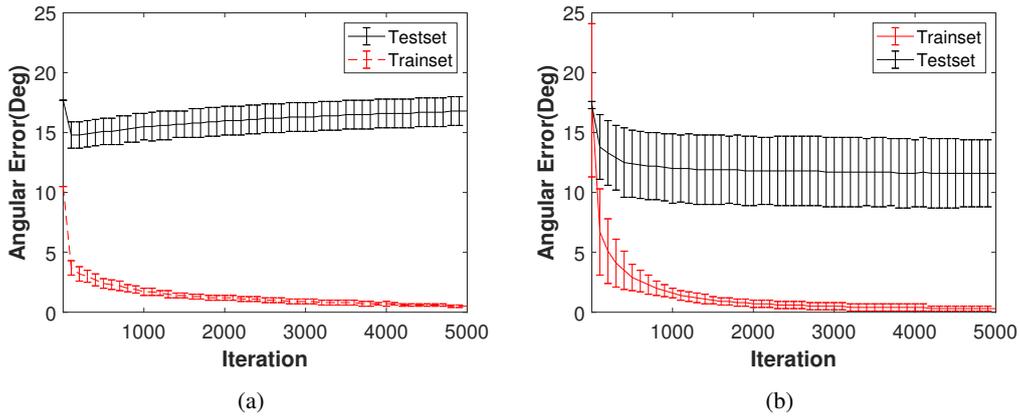


그림 4.11: 'a1original' 환경에서 8개의 위치 학습 결과와 랜덤학습 결과 수행능력 그래프. (a) 8개의 고정된 포지션을 학습한 결과를 나타내는 AE (Angular Error) 그래프 (b) 8개의 임의로 선택된 포지션을 학습한 결과를 나타내 AE (Angular Error) 그래프

Descent)을 적용한 HFLV-GD(Haarlike Feature Landmark Vector Gradient Descent)알고리즘을 적용하여 비교분석하는 실험을 진행하였다. 우리는 HFLV 알고리즘만 적용했을 때 보다 GD를 사용하였을때 얼마나 장애물(Obstacle)을 피해갈 수 있는지 알아보는 실험을 진행하였다. 그림4.13을 보게 되면 HFLV(Haarlike Feature Landmark Vector)알고리즘만 적용한 (a),(c)보다 HFLV-GD(Haarlike Feature Landmark Vector Gradient Descent) 알고리즘을 적용한 (b),(d)의 결과가 확연하게 좋아지는것을 알 수 있다. 그림4.14의 실험결과는 모든 포지션의 정보를 통해 학습하였을 때 결과이다. (a)는 'arboreal'환경의 실험 결과인데 처음 HFLV(Haarlike Feature Landmark Vector)알고리즘을 적용하였을때는 AE(Angular Error)가 높지만 GD(Gradient Descent)를 적용한 후에는 확연하게 AE가 줄어드는것을 볼 수 있다. (b)의 'screen'환경 또한 GD를 적용할때 현저하게 AE가 감소한다. 이번 실험을 통해 HFLV-GD를 적용하 되면 스냅샷 모델의 가장 큰 문제였던 장애물에 의한 음영 현상까지 극복할 수 있다는 것을 알 수 있다.



(a)



(b)



(c)

그림 4.12: 실험 환경별 파노라마 이미지. vardy의 데이터셋 환경중 'a1original'이외의 여러가지 실험 환경이 있다. (a) aboreal 과 (b) screen 환경에는 각각 장애물을 가지고 있다. (c) hall환경의 경우 빛이 적은 환경이어서 낮은 해상도의 이미지가 수집된 환경이다.

2.2 빛이 적은 환경: Hall

우리는 장애물이 있는 환경에서 HFLV-GD (Haarlike Feature Landmark Vector-Gradient Descent) 알고리즘의 수행능력을 테스트 해보았다. HF (Haar-like Feature)의 경우 빛의 변화에 따라 수행능력이 달라진다. 'Hall'환경은 빛이 제한되어있는 환경이고 이로 인해 해상도또한 좋지 않다. 이러한 환경에서도 GD (Gradient Descent) 알고리즘이 정확하게 작동하는지 실험하였는데 그림4.15의 경우를 보게 되면 GD를 적용하기 이전인 (a)의 HFLV (Haarlike Feature Landmark Vector)알고리즘을 적용했을 경우 모든 포지션에서 AE가 높은 것으로 보인다. 하지만 GD 알고리즘을 적용한 (b)의 경우 거의 정확하게 집으로 돌아오는 것을 볼 수 있다. 그림4.15의 (c)를 보게 되면 GD를 적용하였을때 문제없이

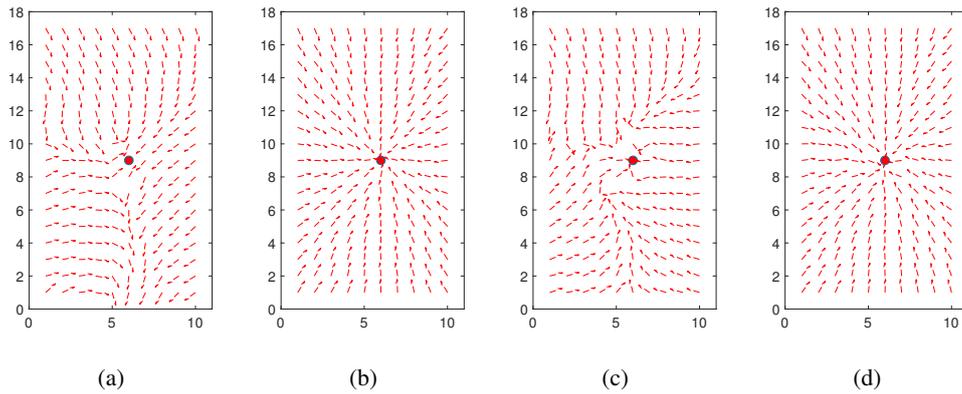


그림 4.13: 각 실험환경에서 상황별 벡터맵. 모두 타입1 의 HF 100개 사용하였고 모든 포지션에서 5000회 업데이트 통해 학습을 수행하였다. (a),(b)는 arboreal의 환경에서 실험한 결과이다. (c)(d) screen 환경에서 실험한 결과이다. (a),(c)는 HFLV (Haarlike Feature Landmark Vector)알고리즘,(b),(d)는 HFLV-GD (Haarlike Feature Landmark Vector Gradient Descent)알고리즘을 각각 적용한 결과를 순서대로 나열하였다.

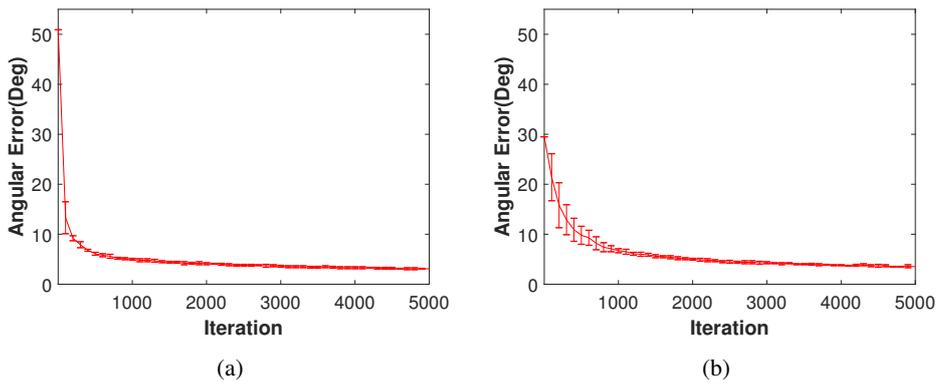


그림 4.14: 각 실험환경에서 상황별 수행능력 그래프. 모두 타입1 의 HF 100개 사용하였고 모든 포지션에서 5000회 업데이트 통해 학습을 수행하였다.(a) arboreal 환경,(b) screen 환경에서 여러번 반복실험을 진행한 뒤 평균값과 표준편차를 이용하여 에러바를 그렸다.

AE가 감소하는것을 볼 수 있다. HFLV-GD 알고리즘을 적용하게 되면 빛에 민감한 HF의 단점까지 극복할 수 있는것을 알 수 있다.

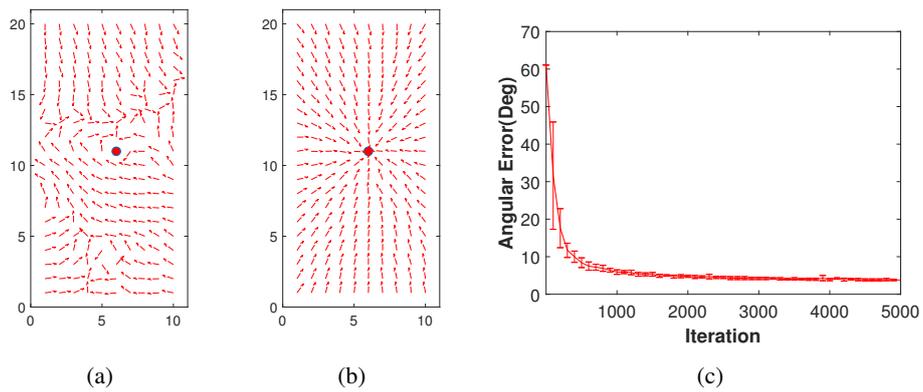


그림 4.15: 'Hall' 실험환경에서 벡터맵과 수행능력 그래프. 타입1의 HF를 100개 사용하였고 모든 포지션에서 5000회 업데이트 통해 학습을 수행하였다. (a)는 HFLV 알고리즘을 'hall' 환경에 적용한 결과이고 (b)는 HFLV-GD 알고리즘을 'hall' 환경에 적용한 결과이다. (c)의 결과는 여러번 반복실험을 진행한 뒤 평균값과 표준편차를 이용하여 에러바를 그렸다.

2.3 포지션의 갯수에 따른 환경별 수행능력 : arboreal, screen, hall

이전 섹션에서는 'a1original' 환경과 여러 가지 형태의 제한된 상황의 정보를 포함한 여러 가지 환경들에서 실험을 진행하였다. 하지만 모든 위치의 정보를 이용하여 제한된 상황들을 극복하였다. 본 연구진은 얼마만큼 적은 정보를 이용할 때 제한된 상황을 극복하지 못하여 수행능력에 영향을 가지는지 실험하였다.

본 실험을 통해 경사 하강법이 얼마나 강한 알고리즘인지 평가할 수 있다. 그림 4.16(a), (b), (c)는 전체 포지션의 50퍼센트를 랜덤하게 선택하여 학습한 결과이다. 순서대로 'arboreal', 'screen', 'hall' 환경이다. 포지션을 100퍼센트 학습한 상위 결과들과 수치로 크게 차이 나지 않는 것을 알 수 있다. 그림 4.16 (d), (e), (f)는 전체 포지션의 10퍼센트를 랜덤하게 선택하여 학습한 결과이다. 순서대로 'arboreal', 'screen', 'hall' 환경이다. 수치적으로 최종 AE가 늘어났지만, 전체적으로 학습이 순조롭게 진행되는 결과가 나왔다.

앞선 실험들과 같이 많은 포지션을 참고하여 많은 정보량을 바탕으로 학습을 하면 좋은 수행능력을 얻을 수 있다. 하지만 약 10퍼센트의 포지션을 참고하게 되더라도 학습이

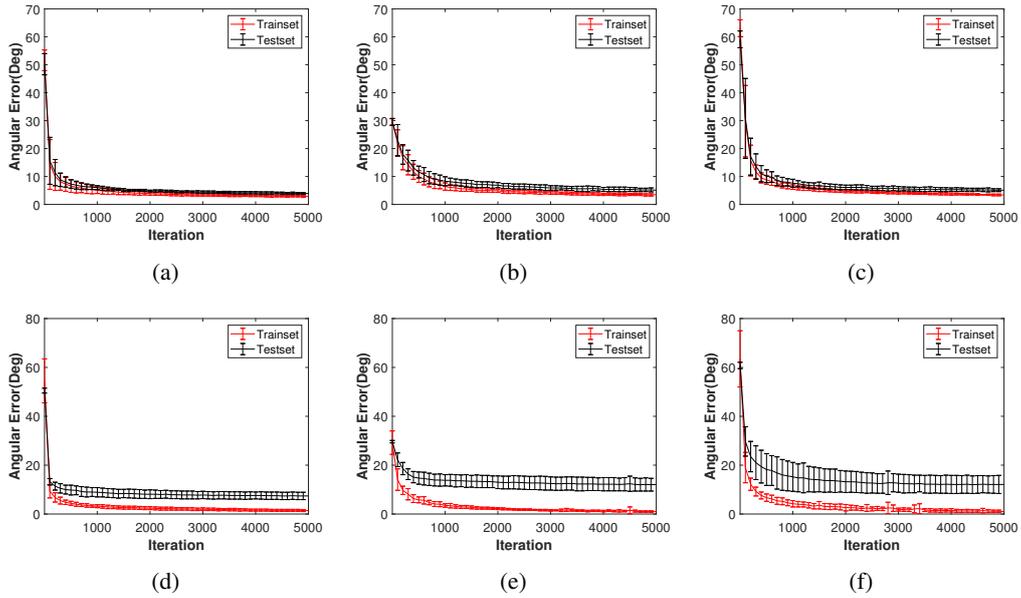


그림 4.16: 각 실험환경에서 상황별 수행능력 그래프. 모두 타입1의 HF를 100개 사용하였고 5000회 업데이트 통해 학습을 수행하였다. (a), (b), (c)는 전체 포지션의 50퍼센트를 랜덤하게 선택하여 학습한 결과이다. 순서대로 'arboreal', 'screen', 'hall' 환경이다. (d), (e), (f)는 전체 포지션의 10퍼센트를 랜덤하게 선택하여 학습한 결과이다. 순서대로 'arboreal', 'screen', 'hall' 환경이다.

가능했고 GD를 이용하여 한계를 가진 실험환경을 극복하는 결과를 도출할 수 있었다. (a), (b), (c)의 결과와 (e), (f), (g)의 결과 모두 학습 데이터만 현저하게 낮은 AE로 수렴하였고 많은 포지션을 참고할수록 테스트 데이터 또한 낮은 AE로 수렴하였다. 많은 포지션을 참고하여 학습하는 것이 최종 AE를 낮추는 방법이지만 본 실험을 통해 전체 10퍼센트 정도의 포지션의 정보만 참고 하더라도 GD를 통한 성과를 얻을 수 있다고 기대된다.

2.4 제한된 포지션에 따른 환경별 수행능력 : 4EA, 8EA

우리는 'original' 환경에서 실험한 것과 동일하게 다른 환경에서도 같은 실험들을 진행하였다. 그림 4.17의 (a), (b), (c)는 각각 'arboreal', 'screen', 'hall' 환경에서 집주변 4개의

포지션만 참고하여 학습한 결과이다. 그림 4.18의 (a), (b), (c) 를 함께 참고하여 보면 4개의 포지션의 정보를 참고하였을 때 학습데이터 이외의 데이터에서는 AE의 큰 변화폭을 관찰하기 힘들었다. 하지만 학습데이터들은 전부 0으로 수렴하는 공통점이 관찰되었다. 이는 학습데이터의 숫자가 적기 때문일 것이다.

4.17의 (d), (e), (f)는 각각 'arboreal', 'screen', 'hall' 환경에서 집주변 8개의 포지션의 정보를 참고하여 학습한 벡터 맵인데 (a), (b), (c)보다는 조금 더 나은 결과를 얻을 수 있었다. 4.18의 (d), (e), (f) 참고하게 되면 5개의 포지션을 참고한 결과보다 AE의 감소 폭이 조금 더 늘어난 것을 볼 수 있다. 포지션 간의 상대적 거리는 멀지 않지만, 상대적으로 더 많은 포지션의 정보를 참고하게 되어 이와 같은 결과를 얻었다고 보인다.

제 3 절 여러가지 타입의 HF 를 적용한 HFLV-GD

기존의 실험들은 가장 간단한 타입 1만 사용하여 실험을 진행하였다. 타입 1의 경우는 HF (Haar-like Features) 가 맵핑되는 위치의 평균 명암비를 나타내는 마스크이다. 타입 1의 특징은 빛이 적절하게 투과되어 조성된 환경에서는 수행능력이 극대화된다. 하지만 빛이 적어지거나 빛이 너무 많은 환경에서는 제대로 작동하지 않을때가 있다. 이전에 3장의 실험 결과와 'hall' 환경에서의 실험들은 GD (Gradient Descent) 알고리즘을 사용하게 되면 학습을 통한 가중치 업데이트로 HF가 가지는 빛에 대한 한계를 극복하고 정확도를 높일 수 있었다.

본 섹션에서는 Type 1뿐만 아니라 Type2등 수평 및 수직 엣지를 추출하는 성질 등 여러 가지 형태 및 성질을 가지는 마스크를 이용하여 포지션의 갯수별 환경별 퍼포먼스를 비교하였다. Type1의 경우 스코어의 크기가 다른 형태의 HF보다 크므로 Type 1 이외의 마스크를 사용할 때는 가중치의 업데이트가 비교적 스코어의 영향을 받지 않도록 스코어의 표준화 작업을 진행하였다. 그림4.19의 (a), (b), (c), (d) 는 'a1original', 'arboreal', 'screen', 'hall' 환경의 순서대로 HFLV 알고리즘만 적용한 벡터 맵이다. 6개의 타입의 HF를 사용하였지만, HF를 단 100개만 사용했으므로 좋은 결과를 얻지는 못하였다. 그

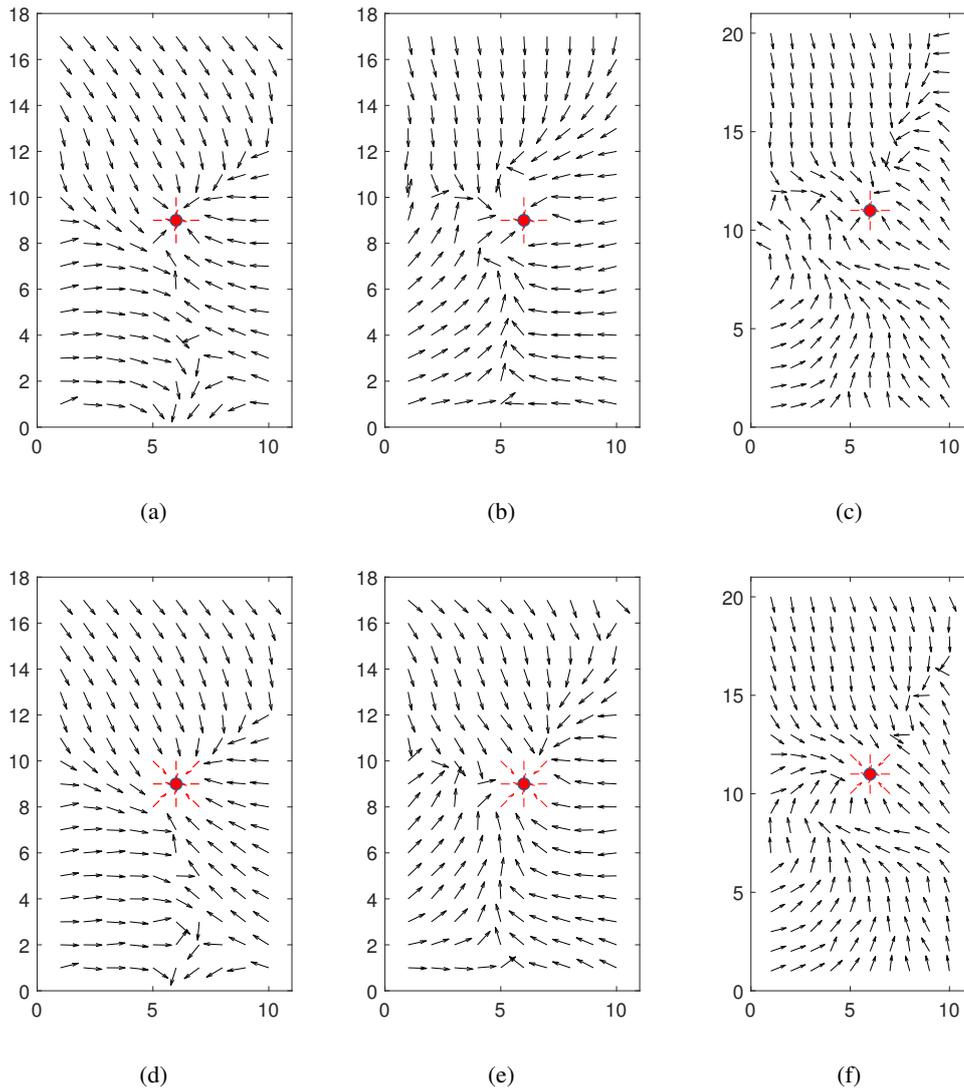


그림 4.17: 각 실험환경에서 상황별 벡터맵. 모두 타입1의 HF를 100개 사용하였고 5000회 업데이트 통해 학습을 수행하였다. (a), (b), (c) 집 주변 4개의 포지션을 학습한 결과의 벡터맵이다. 순서대로 'arboreal', 'screen', 'hall' 환경이다.(d), (e), (f)는 집 주변 8개의 포지션을 학습한 결과의 벡터맵이다. 순서대로 'arboreal', 'screen', 'hall' 환경이다.

이유는 3장에서 밝힌 것과 같이 크기가 큰 매칭 스코어로 벡터의 방향을 리드하는 Type 1의 분포 비율이 낮아졌기 때문에 이와 같은 결과가 나왔다. 우리는 학습을 통해 더 좋

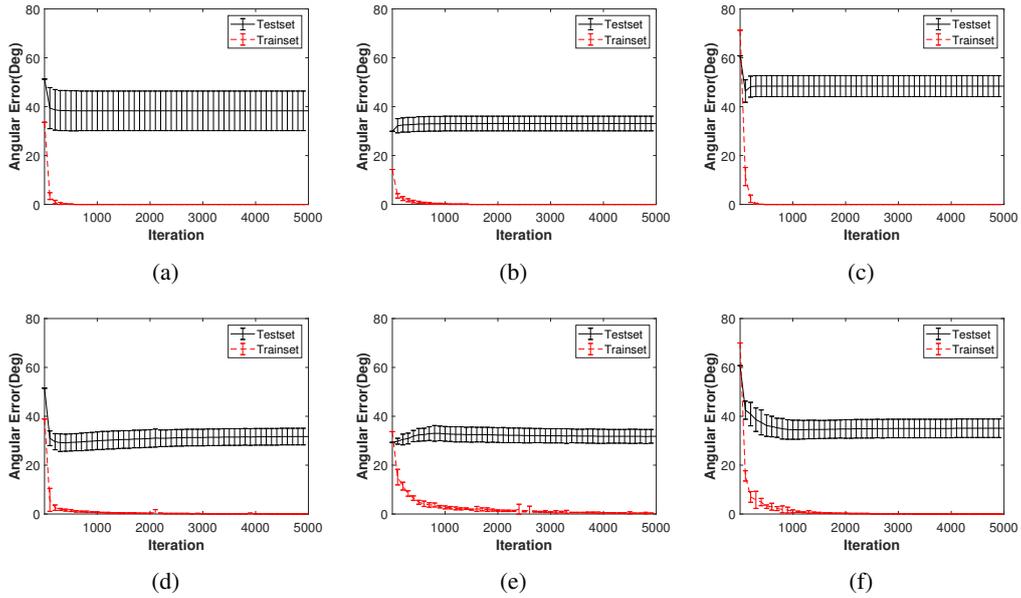


그림 4.18: 각 실험환경에서 상황별 수행능력 그래프. 두 환경 모두 타입1의 HF를 100개 사용하였고 5000회 업데이트 통해 학습을 수행하였다. (a), (b), (c) 집 주변 4개의 포지션을 학습한 결과의 AE 그래프이다. 순서대로 'arboreal', 'screen', 'hall' 환경이다. (d), (e), (f)는 집 주변 8개의 포지션을 학습한 결과의 AE 그래프이다. 순서대로 'arboreal', 'screen', 'hall' 환경이다.

은 결과를 도출해 냈는데 그림4.19의 (e), (f), (g), (h) 를 보면 알 수 있는데 'aloriginal', 'arboreal', 'screen', 'hall' 환경의 순서대로 HFLV-GD 알고리즘을 적용한 결과이다. 높은 정확도의 벡터 맵을 확인할 수 있다. 그림4.20을 보게 되면 환경에 상관없이 AE가 아주 낮은 수치까지 떨어지는 것을 볼 수 있다. 여러 가지 타입의 HF를 사용했을 때도 마찬가지로 좋은 수행능력을 얻을 수 있었다. 하지만 수치적으로 비교하였을 때 타입1만 사용하여 학습할 때와 큰 차이를 찾기는 어려웠다. 오히려 HF의 갯수를 적게 사용할 때 HFLV의 AE는 Type1만 사용했을 때가 더 적게 나타난다.

본 연구진은 이번 실험결과를 통해 타입을 목적에 맞게 적절하게 사용하는 것이 좋다는 것을 밝혀 냈다. 더 많은 갯수의 HF를 사용할 수 있을 때는 여러가지 타입을 섞어서 사용하고 효율적으로 적은 갯수의 HF를 사용할 때는 단일 타입의 마스크만 사용하는 것

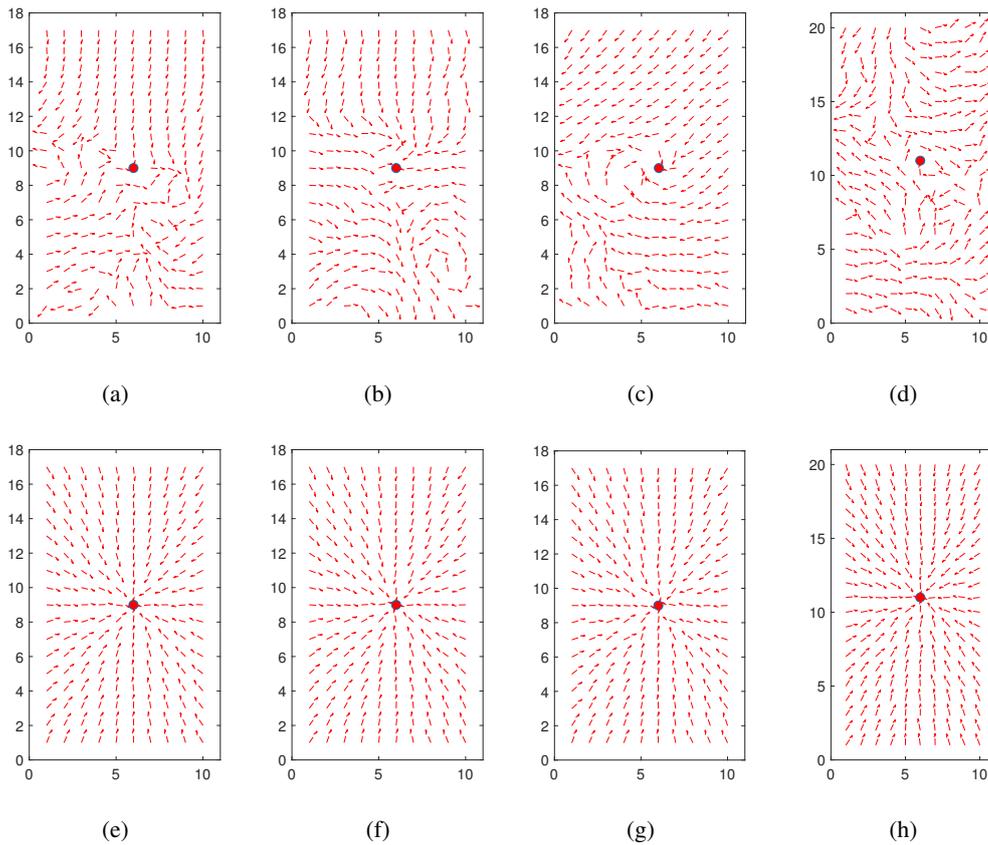


그림 4.19: 각 실험환경에서 상황별 벡터맵. 랜덤하게 생성된 모든 타입의 HF를 100개 사용하였고 모든 포지션의 정보를 이용하여 5000회 업데이트 통해 학습을 수행하였다. (a)'a1original', (b)'arboreal', (c)'screen', (d)'hall' 환경의 순서대로 HFLV 알고리즘만 적용한 벡터맵이고 (e)'a1original', (f)'arboreal', (g)'screen', (h)'hall' 는 환경의 순서대로 HFLV-GD알고리즘을 적용하였다.

이 좋을 것으로 판단된다. 이후 실험에는 타입1과 2의 HF를 섞어서 사용할때 여러가지 상황에서의 수행능력을 판단해 보았다. 타입 1과 타입 2의 HF를 섞어서 사용하게 되면 타입 1의 HF가 생성되는 위치의 평균 명암비와 타입 2가 생성되는 부분의 수직 엣지 정보를 사용할 수 있게 되는데 이러한 성질이 학습에 적용될 수 있다.

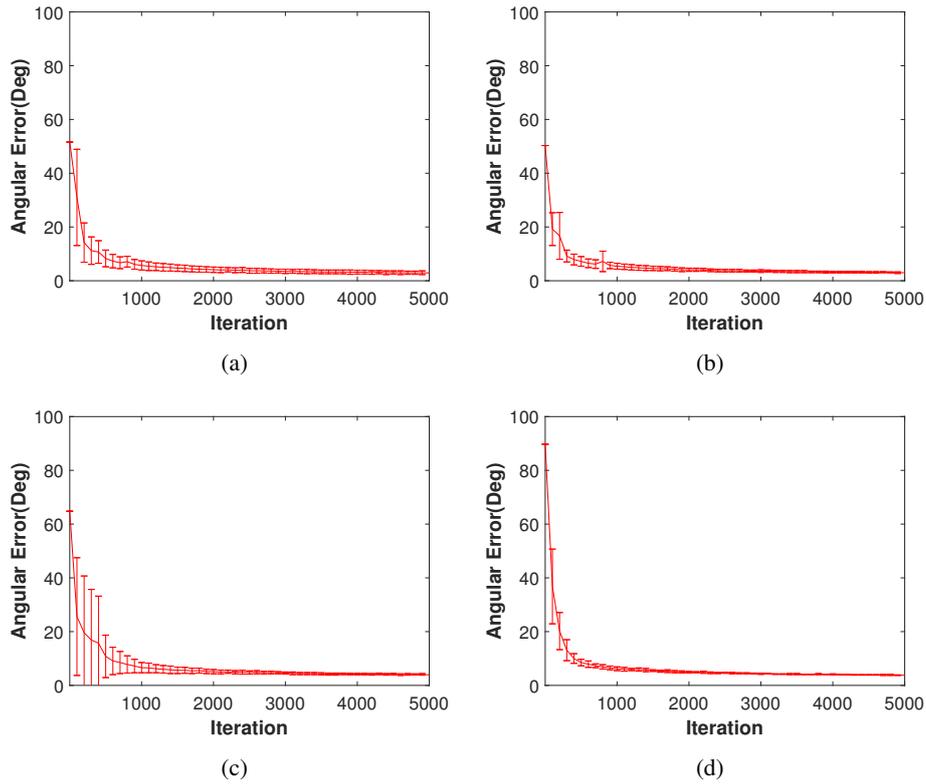


그림 4.20: 각 실험환경에서 상황별 수행능력 그래프. 랜덤하게 생성된 모든 타입의 HF를 100개 사용하였고 모든 포지션의 정보를 이용하여 5000회 업데이트 통해 학습을 수행하였다. (a), (b), (c), (d)는 각각 'a1original', 'arboreal', 'screen', 'hall' 환경 순서대로 에러그래프를 나타내었다.

제 4 절 요약

경사하강법을 이용하여 포지션 또는 마스크의 갯수별 또는 마스크의 형태별 수행능력을 비교하는 실험들을 진행하였다. 본 실험에서 사용한 Vardy의 Dataset은 환경별로 여러 가지 특성을가지고 있다. 기본적으로 Haar-like features는 빛에 강도에 따라 수행능력 및 활성정도가 달라졌다. 이러한 특징을 경사하강법을 통해 극복할 수 있는지 알아보았고 여러환경에도 적용이 가능하지 실험하였다.

모서리를 찾는 마스크와 그외의 모양의 마스크등을 이용하여 여러가지 형태의 마스크

Various performance tables		Environment	alorigianl	Arboreal	Screen	Hall1
		N	169	169	169	199
COMALV	-	Error(std)	19(2.0)	44(6.3)	34(4.4)	49(6.6)
HFLV (HF 100EA)	Type1	Error(std)	21.4(3.2)	46.3(5.8)	32.5(5.6)	69.1(11.2)
	Type1&2		39.5(10.1)	64(11)	52.3(10.8)	105.6(19.7)
	TypeAll		50.8(12.4)	73.4(11.2)	57.4(11.1)	119.5(20.5)
HFLV-GD (Position 10%)	Type1	Error(std)	7.1(1.8)	6.8(1.3)	11.0(2.4)	11.0(3.3)
	Type1&2		10.6(4.5)	10(3.3)	11.8(4.8)	14.1(3.3)
	Type All		7.4(2.4)	8.3(1.2)	12.8(3.7)	15.6(4.5)
HFLV-GD (Position 25%)	Type1	Error(std)	3.6(0.6)	4.1(0.5)	5.0(0.9)	6.1(0.5)
	Type1&2		5.3(0.8)	4.9(0.6)	5.7(0.6)	7.9(1.4)
	Type All		3.7(0.6)	4.9(0.7)	6.4(0.7)	7.3(1.2)
HFLV-GD (Position 50%)	Type1	Error(std)	2.7(0.3)	3.4(0.2)	4.3(0.5)	4.2(0.4)
	Type1&2		4.2(1.1)	4.0(1.1)	4.7(0.6)	5.8(1.3)
	Type All		3.1(0.6)	3.9(0.6)	4.8(0.3)	4.7(0.4)
HFLV-GD (Position 100%)	Type1	Error(std)	2.3(0.1)	3.1(0.1)	3.5(0.2)	3.8(0.2)
	Type1&2		3.3(0.8)	2.8(0.1)	4.4(1.3)	4.7(0.3)
	Type All		2.9(0.7)	3.0(0.2)	3.9(0.3)	3.8(0.1)
HFLV-GD (Fixed 4 Position)	Type1	Error(std)	15.4(1.4)	37.4(8)	32.3(2.9)	47.4(4.2)
	Type1&2		20.9(2.5)	48(4.8)	33.1(5.7)	45.9(4.8)
	Type All		27.1(2.8)	33.3(4.9)	46.8(4.5)	61.1(6.2)
HFLV-GD (Random Position 4EA)	Type1	Error(std)	15.5(3.2)	25.7(10.9)	27.4(3.2)	41(7.4)
	Type1&2		28.6(9.5)	36.9(9.3)	33(10.9)	47.2(7.6)
	Type All		26.6(7.4)	32.6(5.2)	34.4(7)	56.5(11.1)
HFLV-GD (Fixed 8 Position)	Type1	Error(std)	16(1.2)	30.2(3.2)	30.3(2.7)	33.7(3.6)
	Type1&2		13.6(2.4)	21.7(2.2)	15.8(1.5)	34.2(4.7)
	Type All		24.5(2.5)	20.4(5.1)	24.1(5.8)	50.1(5.4)
HFLV-GD (Random Position 8EA)	Type1	Error(std)	11(2.7)	15(6.1)	19.9(5.4)	25.3(8.6)
	Type1&2		17.8(6.5)	18.9(7)	22.4(8.8)	29.5(7.9)
	Type All		19.3(8)	20.4(11.7)	23.1(8.6)	30.2(11.9)

표 4.1: 본 표는 여러가지 파라미터에서 환경별 수행능력 테이블. 각 환경별 수행능력을 알고리즘별로 비교 하였다.

크를 사용할때의 결과도 살펴보았다. 가중치의 업데이트를 통해 여러가지 환경에서 HF의 수행능력이 이전 결과들에 비해 얼마나 향상되었는지 평가하였는데 가장 좋은 퍼포먼스를 나타내는 마스크타입의 조합과 제한적인 상황을 극복하고 높은 수행능력을 가지는 마스크의 형태와 특징들을 찾을 수 있었다. 결론적으로 많은 포지션의 정보를 사용하게 되면 지역적인 한계를 극복할 수 있었다. 여러가지 환경과 파라미터에 따라 실험을 진행하였다. 결론적으로 정보가 많으면 내비게이션이 된다. 그리고 정보는 질적으로 다양해야 더 좋은 결과가 나타났다. 표4.1를 보게되면 여러가지 상황에 대해서 수행능력을 비교해볼 수 있다.

가장 먼저 포지션의 셋수별 수행능력을 비교해보게 되면 랜덤하게 위치를 선택하였을 때 많은 포지션을 선택할 때가 적게 선택한 경우보다 비교적 수행능력이 좋았고 포지션을 전체 포지션에서 25퍼센트 정도 사용하게 되면 전체 포지션을 이용한 것에 비해 크게 차이 나지 않는 결과를 얻을 수 있었다. 포지션을 고정해서 학습을 진행하는 것 보다 임의로 선택하여 학습을 진행하는 것이 더 좋은 결과를 가져다주었다. 고정된 포지션 학습은 집주변의 포지션에서 학습하는 것이다. 그림4.7를 참고하게 되면 4개의 고정된 포지션은 집주변의 동서남북 방향을 학습하는 것이고 8개의 고정된 포지션은 북동, 북서, 남동, 남서 까지 학습하는 결과이다. 해당 결과에서는 4개보다 8개의 포지션을 학습하는 것이 더 좋은 결과를 얻을 수 있었고 4개의 고정된 포지션을 학습한 결과는 학습을 하지 않은 HFLV에 비하여 아주 적은 변화를 보였다. 하지만 환경에 따라 어느정도 개선된 결과를 얻을 수 있었다.

본 챕터에서는 여러가지 HF의 형태에 따라서 학습의 결과도 달라지는 것을 알 수 있었다. 기존의 연구에서 제시된 이론들을 다양한 형태의 HF를 사용할 때 더 좋은 결과를 얻을 수 있었는데 본 실험에서는 꼭 그렇지만은 않다는 것이 밝혀졌다. 그 이유는 이전의 실험들에서는 HF를 충분히 사용하는 반면에 본 실험에서는 제한적인 갯수의 HF를 사용하게 되어서 표4.1 와 같은 결과가 나왔다고 보여진다. 가장 심플한 형태의 타입 1의 결과가 대부분의 실험에서 좋게 나왔으며 일부 실험에서는 여러 형태의 타입을 사용한 결과가 좋게 나온 경우도 있었다. 고정된 8개의 포지션과 4개의 포지션에서의 결과는 여러 타입을 사용했을 때 좋게 나오는 경우도 있었다. 이는 제한적인 정보를 가진 상황에서는 다양한 형태의 마스크를 사용했을 때 학습의 선택폭이 더 넓어져 좋은 결과가 나왔다고 할 수 있는데 이것에 대해서는 차후 실험이 더 필요하다고 판단된다.

본 논문에서 제시하는 HFLV-GD라는 알고리즘은 비록 많은 학습데이터를 사용할 수 없는 실제 상황에서는 구현함에 따라 많은 제한 상황들이 생기겠지만 여러 가지 이미지 기반 실험을 통해 그 가능성을 입증하였다. 여러가지 센서를 이용하여 융합단계의 로봇알고리즘을 구현한다면 충분히 사용할 수 있을 것으로 보인다.

제 5 장

모바일 로봇에 적용한 HFLV-GD

실내에서의 모바일로봇의 귀소 내비게이션은 여러 분야에서 연구되고 있는데 모바일로봇의 귀소 내비게이션은 여러 분야에서 쉽게 풀리지 않는 내비게이션임이 증명되었다. 근래에 연구를 통해 SLAM등의 모델이 제시되었지만 높은 가격의 장비들과 복잡한 센싱 시스템이 포함된 알고리즘 등 여러 가지 문제들이 발생했다[28, 46].

본 연구진은 이미지 기반실험과 실시간 로봇 적용실험을 하였다. 이미지 기반 실험을 할 때는 그림6.1 (a) Pioneer (P3-DX) 를 이용하여 데이터를 획득하여 실험하였고, 실시간 로봇 실험을 할 때는 그림6.1 (b) Tuttlebot(Kobuki) 을 이용하여 실험하였다. 두 실험 모두 HFLV-GD (Haar like Features Landmark Vector-Gradient Descent) 를 적용하여 실험하였고 일반 실내 환경에서 미리 지정해 놓은 환경에 적용 실험을 하였다. 위 연구의 일부 또는 확장된 내용에 대해서는 다음 논문지에 게재될 예정이다[49].

제 1 절 목적

본 장에서 두 가지 실험 목표를 가진다. 첫 번째 목표는 실제 환경에서 실시간으로 적용할 수 있는지에 대한 여부를 알기 위해서이다. 앞장의 이미지 기반 실험에서는 HFLV-GD (Haar-like Features Landmark Vector Gradient Descent) 알고리즘을 여러 가지 환경에 적



그림 5.1: 실험 로봇. (a) Pioneer (P3-DX) (b) Turtlebot (Kobuki)

용하여 얻은 성과들이 충분히 증명되었다. 장애물과 제한 사항을 돌파하는 강인한 알고리즘으로 평가되었기 때문에 실제 상황에서도 비슷한 평가가 이루어질 수 있는지 본 장에서 실제 로봇에 적용하여 알아보도록 한다. 두번째 목표는 다양한 조건에 의한 학습 수행능력을 평가하는 것이다.

실시간으로 학습을 진행할 때 두 가지 조건에 의해 다른 학습시스템을 구현할 수 있다. 첫 번째는 로봇이 이동할 때 얻은 정보들을 종합하여 학습을 진행하는 것이다. 가장 일반적인 학습 방법이고 본 장에서 첫 번째 방법으로 제시한다. 두번째는 특정 지역의 탐색 정보를 가지고 학습을 진행하는 방법이다. 특정 지역을 먼저 탐사했다는 가정하에 이전에 탐색한 정보들을 바탕으로 학습된 가중치를 실시간 상황에 맞게 적용하는 것이다. 본 장에서는 두 방법의 장단점을 밝히고 수행능력을 비교하게 된다. 또한 실험을 진행할 때 Path Integration (PI) 방법과 수행능력을 비교하여 본다.

제 2 절 실험환경

본 실험을 진행하기 위하여 이미지 기반 실험의 환경과 유사한 환경을 새롭게 구성하였다. 우리는 기본적으로 면적이 약 3m * 3m 환경에서 41개의 데이터를 사용한다. 그림5.2은 실험환경이다. 가장 중심이 되는 점을 (6.6) 으로 지정하고 30cm 간격으로 집을

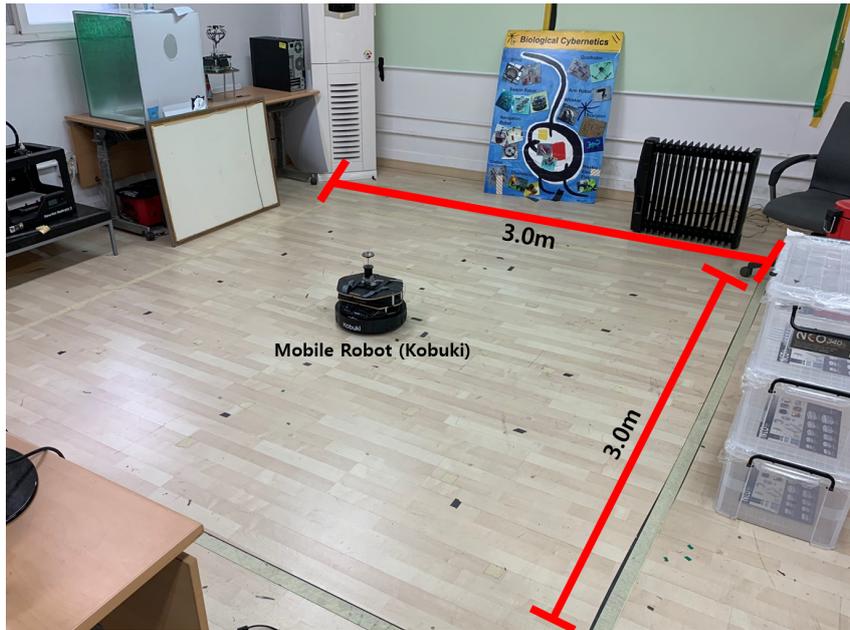


그림 5.2: Mobile Robot의 실험환경. 로봇의 정면을 기준으로 포스터(Landmark)가 있다. 측면과 후면에 책상과 연구실의 물품이 있는 연구실 환경이다. 창문으로 들어오는 빛은 전부 차단하였고 밝기는 일반 실내의 형광등 빛의 밝기 정도로 볼 수 있다.

제외한 40개의 포지션을 지정하였다. 귀소 내비게이션을 재현한 모바일 로봇 P3-DX 을 이용하였으며 실험환경은 실내의 조명이 있는 평범한 연구실 환경으로 선택하였다. 앞선 과정들과 같이 이미지의 변형 과정을 진행하였다.

최초의 이미지는 640*480 해상도의 Omni Direction 이미지이다. 우리는 이미지를 사용할 때 정보의 변형을 최대한 지양하면서 정보의 활용도가 높은 부분만 Panoramic 이미지로 가공하였다. 손실을 최소화하고 중요한 정보를 극대화를 고려한 최종 변경된 Panoramic 이미지의 크기는 1099*114의 크기의 이미지이다. 우리는 Panoramic Image를 만들 때 필요 없는 정보는 최대한 축소하고, 이미지의 변형이 생기지 않도록 여러 가지 방법을 이용하여 처리하였다.

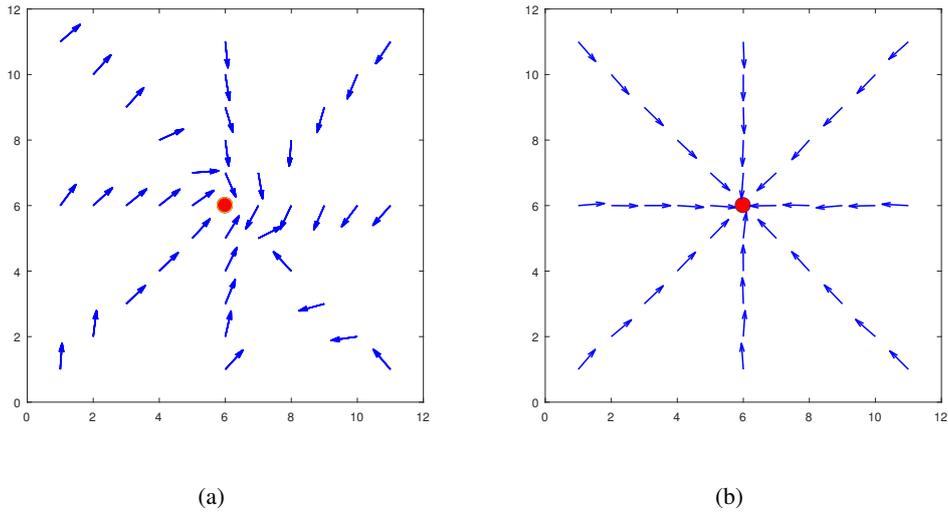


그림 5.3: 'Ourlab' 환경에서 각 알고리즘 별 벡터맵. (a) 는 HFLV 방법이고 (b) 는 학습을 이용한 HFLV-GD 방법이다. (6.6)의 빨간점은 로봇이 도달해야 할 목표지점이다. HF 는 Type1만 사용하였고 두 방법 모두 100개의 HF를 사용하였다.(b)의 경우 10000번의 반복학습을 통해 결과를 얻었다.

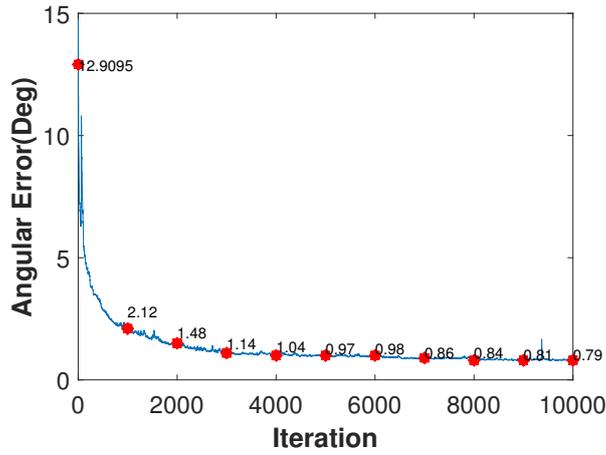


그림 5.4: 학습 횟수에 따른 이미지 기반 실험의 오차의 평균값 그래프. 학습이 진행됨에 따라 오차의 평균값이 낮아지는 것을 볼 수 있다.

2.1 이미지 기반 실험 : Ourlab

먼저 모바일 로봇에 실시간 환경에 알고리즘을 적용하기 이전에 이미지 기반 실험을 먼저 진행하였다. 이미지 기반 실험은 본 실험환경에서 실시간으로 모바일 로봇에 적용하기 위해 알고리즘을 테스트하는 단계이다. 이전의 HFLV (Haar-like Features Landmark Vector)는 비교적 낮은 수행능력을 보였다. 우리는 경사 하강법을 적용하여 개선된 HFLV-GD (Haar-like Features Landmark Vector Gradient Descent)를 이미지 기반 실험에 적용하여 더 나은 수행능력을 증명하였다. 그림5.3은 각각 알고리즘을 적용한 결과이다. 그림5.3의 (a)와(b) 비교해보게 되면 정확도 면에서 확연하게 차이가 있다는 것을 알 수 있다. 이후 로봇실험에서 두 알고리즘의 차이를 증명할 예정이다.

두 알고리즘을 적용하기 위해서는 HFLV-GD (Haar-like Features landmark Vector Gradient Descent)를 사용하여 그림5.4과 같은 결과를 얻었다. 10000번의 학습을 통해 AE(Angular Error)가 거의 0에 가깝게 도달하였다. 우리는 이러한 그림5.3의 (b) 와 같은 벡터맵 이용하여 이미지 기반 실험을 시각화하였다. 이후 각각의 포지션에서 목표 위치에 이동하기 위해 벡터로 방향을 추정하여 모바일 로봇을 이동 시키기 때문에 벡터 맵으로 표현하였다. 각각의 포지션을 이동하기 위하여 스텝당 30cm를 이동하였다. 이동 거리를 정해놓는 이유는 스냅샷모델에서 픽셀 정보만 이용하기 때문에 거리를 알 수 없기 때문이다. 또한 모든 위치는 목표 위치와 관계를 가지는 각도의 벡터값을 알 수 있지만, 포지션별 각도는 알 수 없기 때문에 이러한 점을 고려하여 실험을 진행하였다.

제 3 절 여러가지 학습 방법과 모바일 로봇 실험

실시간 환경에서 모바일 로봇(Kobuki)에게 알고리즘을 적용하는 것은 여러 가지 제한 사항을 고려해야 한다. 실시간 적용을 할 때 생기는 여러 가지 제한사항이 있는데 첫 번째는 알고리즘의 속도이다. 로봇이 움직이는 상황에서 알고리즘이 바로바로 적용될 수 있는지 테스트해 보아야 한다. HFLV-GD의 경우 실시간에서 적용해도 문제없을 만큼 속도가 지연되지 않았다. 두 번째로 실시간으로 수집되는 이미지 데이터를 형태에 맞

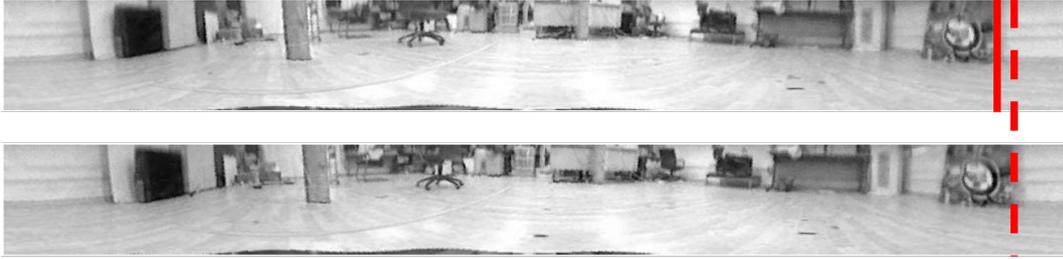


그림 5.5: 같은 포지션에 각도가 다른 파노라마 이미지. 두 이미지는 같은 포지션에서 로봇의 머리가 각각 Compass를 Desired 하게 설정한 이미지와 오차를 포함한 이미지이다. 위쪽의 이미지가 Desired 헤드의 위치의 이미지 이고 아래의 이미지는 Desired Compass에서 살짝 벗어난 이미지이다. 두 이미지의 각도는 약 9도정도 차이가 나게 되는데 수치상으로 비교해 볼 이지미상의 차이나는 픽셀의 비율과 차이나는 각도 비율이 일치한다.

게 처리해야 한다. 로봇의 헤드가 돌아가 있기 때문에 집 이미지를 기준으로 이미지의 정렬을 시행해야 한다.

본 실시간 로봇 실험에서는 수집한 이미지를 정해진 크기와 모양으로 변경후 Visual Compass를 이용하여 이미지 정렬하는 방법을 제시하였다. 이러한 제한사항을 고려한 실험과 로봇의 학습에 따른 실험을 시행하였다. 본 챕터에서는 로봇실험에 대하여 총 두 가지 방법을 제시한다. 첫 번째 방법은 학습하기 이전에 탐사를 진행하고 집으로 돌아가는 방향을 추정하기 이전에 한 번에 처리하는 방법이다. 이 방법의 경우 탐사를 하면서 실험을 진행하기 때문에 탐사의 이동 경로와 PI (Path Integration) 의 오차가 수행능력에 큰 영향을 미친다.

PI의 오차가 수행능력에 큰 영향을 미치는 이유는 학습을 시킬 때 Desired 각도를 Path Integration에서 추정한 각도로 사용하기 때문이다. 두 번째 방법은 이전에 미리 탐사해 놓은 학습데이터를 이용하여 학습하고 학습된 가중치를 이용하여 방향 추정이 필요할 때 바로 적용하는 방법이다. 이후 섹션에서 각각의 방법에 관해서 설명한다.

3.1 이미지 정렬 문제: Visual Compass 사용

우리는 그림 5.2과 같은 환경에서 Omni direction 카메라를 이용하여 포지션별 이미지 데이터를 획득하여 실험을 진행하였다. 각각의 포지션 사이의 거리는 가장 가까운 포지션에서부터 30cm고, 데이터를 획득할 때는 로봇이 같은 방향을 향하도록 실험하였다. 그 이유는 스냅샷 모델의 문제점에서도 언급되었던 이유 때문이다. 스냅샷 모델은 두 이미지상에서 같은 각도 선상에 HF (Haar like Feature) 를 배치해야 비교해야 한다는 점이다. 로봇의 실제 움직임에서는 헤드의 방향이 바뀌게 되면 Omni 방향의 이미지까지 각도가 변화하게 된다. 이러한 문제는 실제로 알고리즘으로 적용하기에 중요한 문제가 된다. 따라서 로봇이 움직이는 각도를 전부 기록하여야 한다. 하지만 본 과정은 매우 번거롭고 정확하지 않을 수 있다.

본 연구진은 기준이 되는 이미지와 각도를 정렬하려는 이미지의 신호 차이를 분석하여 각도를 정렬하는 방법인 Visual Compass를 사용한다[58]. Visual Compass의 경우 SLAM과 같은 값비싼 장비로 구현하는 알고리즘에도 적용하여 사용할 정도로 비교적 신뢰도가 높다. 그림 5.5는 Visual Compass를 통해 같은 방향으로 정렬을 한 결과이다. 그림 5.5는 같은 포지션에서 찍은 사진인데 두 사진은 사진상의 1cm 정도 이미지상의 차이를 보인다. 하지만 실제 알고리즘을 사용하여 판단한 각도는 8도에서 12도 정도 차이가 나게 된다. 이는 로봇의 움직임에서 나오는 Odometry 에 의한 오차 이거나 Visual Compass를 사용하여 이미지를 재생성하였을 때도 마찬가지이다. 두 이미지는 위치에 따른 사물의 크기 차이와 방향의 차이는 미세하다고 볼 수 있다. 하지만 Visual Compass를 이용하였을 때 항상 정렬이 완벽하지는 않다.

해당 실험환경 기준으로 만일 Compass 정렬의 기준이 되는 이미지와 정렬시킬 이미지가 실제 환경에서 멀리 떨어져 있는 경우 정렬의 정확도가 현저하게 낮아지는 것을 관찰할 수 있다. Visual Compass의 이러한 신뢰도를 바탕으로 실험을 진행하였고 Visual Compass로 추정된 값과 실제로 직접 이미지를 정렬시킨 값의 차이를 비교하는 실험을 진행하였다.



그림 5.6: 방법1 을 설명해 놓은 그림. (a) 는 집에서의 방향을 결정하는 이미지이고 (b) 는 이후 스텝에서 로봇이 랜덤하게 방향을 설정하는 이미지이다. 위 방법처럼 탐사를 시행하고 매 스텝이 끝나게 될때마다 탐사 이미지를 수집하게 된다.

3.2 방법1 : Odometry 기반 학습

방법1은 탐험을 진행하면서 데이터를 수집하고 수집한 데이터를 이용하여 학습을 진행하는 방법이다. 본 방법은 생체 모방적으로 생각해 볼 때 개미가 탐사를 나갈 때마다 탐사하는 주변 환경에 맞춰 학습하게 되고 이를 바탕으로 집으로 돌아오는 것을 로봇으로 나타낸 것이다. 모바일 로봇은 스텝마다 랜덤하게 방향을 설정하고 스텝을 이동할 때마다 데이터를 수집하게 된다. 매번 수집한 데이터를 이용하여 마지막 스텝전에 학습을 진행한다. 학습을 진행하여 업데이트된 가중치를 얻고 본 가중치를 이용하여 마지막 순간의 방향을 판단하게 된다. 그림 5.6의 (a) 를 보면 로봇이 시작 지점에서부터 이동할 방향을 랜덤하게 선택하게 되고 이동을 진행 중인 그림 5.6 (b) 경우에도 마찬가지로 랜덤하게 방향 결정을 하며 이동하고 이미지를 수집하게 된다. 지정된 스텝만큼 이러한 과정을 거치게 되고 마지막 스텝에서 수집한 데이터를 전부 학습을 하게 된다. 학습을 진행할 때 수집한 데이터의 정답지가 필요하다. 본 상황에서 Path Integration을 통해 측정된 각도의 정답지를 이용하여 학습을 진행하게 된다. 따라서 본 방법을 시행할 때 Path Integration의 정확도가 중요하게 된다.

Method 1		1	2	3	4	5	6	7	8	9	10
Path Integration	Error(Deg)	1.5412	1.8475	11.6337	5	2.9059	8.7462	2.6542	1.7899	3.3468	6.5198
HFLV-GD		20.864	9.5904	2.8774	12.7216	9.7059	6.1844	6.3466	6.124	24.5167	29.524

표 5.1: 방법1의 결과 테이블. 스텝은 총 10회로 제한하였고 오차 각도는 실제 로봇의 위치에서 측정된 각도값과의 오차를 나타낸다.

표 5.1를 보게 되면 Path Integration과 HFLV-GD의 오차 각도를 각각 나타내었다. 오차 각도는 실제 로봇이 있는 위치에서 측정된 값과 각 항목의 값을 비교하여 도출해 낸 결과이다. 스텝이 10번이었을때 PI의 결과가 더 좋은 경우가 많았는데 이는 스텝의 수가 적어서 Odometric한 오차가 적었기 때문이다.

본 방법에서는 핵심적인 문제가 있다. 스텝이 늘어나게 되면 PI의 오차 각도 또한 늘어나게 된다. PI를 통하여 학습을 진행하게 되기 때문에 PI의 오차 각도가 늘어나는 것과 마찬가지로 HFLV-GD 알고리즘 또한 오차를 가진 값을 학습하게 되기 때문에 이동 거리에 따른 PI의 오차 각도를 고려해야 한다.

3.3 방법2 : Landscape 기반 학습

3.3.1 Visual Compass 의 오차 측정 실험

방법2는 탐사를 이전에 진행했다는 가정하에 미리 탐사를 진행하여 수집한 이미지로 학습을 진행한다. 미리 탐사가 진행된 포지션은 그림 5.7 의(a)의 빨간 점의 영역으로 볼 수 있다. 빨간 점인 영역을 선정한 기준은 이미지 기반 실험을 기준으로 실험을 진행할 때 환경 내에서 집을 제외하고 지역적인 한계를 극복하기 위하여 집 주변의 영역에서 일정 거리만큼 측정하여 지역 15개를 선정하였다. 선정된 포지션을 먼저 학습하고 해당 학습된 포지션을 기준으로 업데이트된 가중치를 로봇의 방향을 결정할 때 현재 위치에서의 이미지에 적용하고 의사결정을 한다.

이 방법의 장점은 위에서 방법1의 문제점으로 언급된 PI의 오차 증가에 따른 학습 오차가 증가하는 것과 달리 본 실험에서는 PI의 각도 오차와 방법2의 수행능력이 무관하다 점이다. 또한 학습에 사용되는 테스트 데이터가 많아질수록 정확도가 올라가는 경향이



그림 5.7: Landscape 기반 학습. (a)의 빨간 점은 미리 탐사가 이루어진 위치. (b) Landmark 빨간 점이 있는 위치에서 선행 탐사로 수집된 이미지 데이터를 학습의 테스트 데이터로 이용하여 학습을 미리 진행하게 된다. 이후 (b)와 같이 로봇이 랜덤한 방향으로 탐색을 한다. 이후에 방향 결정이 필요한 시점에서 이전에 학습된 가중치를 이용하여 방향 결정을 하게 된다.

있는데 이는 학습 데이터를 선정하는 기준이 얼마나 중요한지를 반증하는 결과이다. 또한 이후 실험에서 Visual Compass를 통해 알고리즘을 사용하여 생기는 에러가 얼마큼인지 같이 측정하였다.

본 실험은 Visual Compass 알고리즘을 이용하여 이미지 정렬을 하였을 때 오차를 측정하는 방법을 제시한다. 표 5.2를 보게 되면 Visual Compass를 사용하였을 때와 로봇을 이미지 정렬을 위한 이상적인 각도에 배치했을 때 추정하는 각도가 다른 것을 관찰할 수 있다. 이상적인 각도에 배치하여 이미지를 획득하여 각도 추정을 하게 되면 각도의 오차가 현저하게 줄어든다.

본 실험을 통해 Visual Compass에 대한 오차가 얼마만큼 결과에 큰 영향을 미치는지 알 수 있고 통상적으로 Visual Compass의 정확도는 기준 이미지로부터 멀어질 때 점점 증가하는 것을 볼 수 있다. 우리는 본 실험을 통해 시각 내비게이션을 이용할 때 이미지 정렬이 얼마나 중요한지와 Visual Compass를 실제 적용하여 사용할 때 신뢰도를 알 수 있었다. 또한 이상적인 환경인 이미지 기반 실험 환경과 실제 로봇에 적용할 때의 수행능력

Method 2		1	2	3	4	5	6	7	8	9	10
Path Integration	Error(Deg)	2.0395	1.2601	1.8039	4.1849	4.3987	15.2552	7.7458	1.7709	2.9274	0.6987
Visual Compass		10.5091	15.2968	2.0182	5.3594	16.3954	21.5970	21.1993	28.3489	19.0435	22.0240
Desired Compass		4.3582	2.2972	2.0301	1.7823	6.3575	4.7652	6.3444	7.6854	0.5062	4.3648

표 5.2: 방법2 의 결과 테이블. 스텝은 총 10회로 제한하였고 Visual Compass를 사용하여 이미지 정렬을 하였을때와 정확한 Compass를 측정하여 이미지 정렬했을때의 결과를 각각 나타내었다.

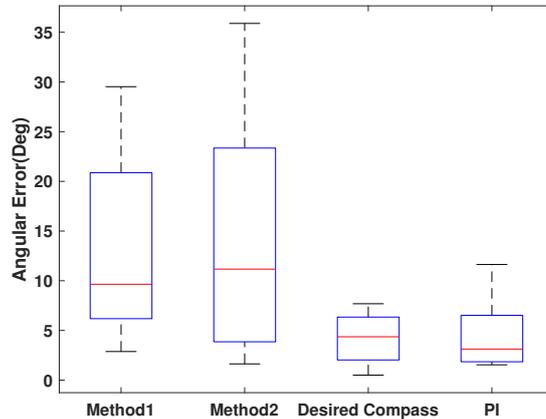


그림 5.8: 두 가지 방법과 PI 방법을 비교한 그래프. 방법1의 경우 오차의 범위가 방법2에 비해 크다.

Method 2		1	2
Path Integration	Error(Deg)	1.5482	6.2420
Visual Compass		43.3847	40.0016
Desired Compass		28.1518	13.1166

표 5.3: 로봇이 실험환경에서 멀어졌을 때 결과. 반복 실험 중 로봇이 랜덤하게 움직일 때 실험환경 내에 학습이 시행된 범위안에서 의사 결정을 하게 되면 이상적인 실험 결과가 나오지만, 실험 환경에 벗어나게 되면 위와 같은 결과가 나온다.

차이가 어떤 문제에서 생기는지 판단할 수 있다.

3.4 토의

본 실험에서는 여러 가지 토의할만한 내용이 있다. 먼저 우리는 결과 분석 및 평가를 위해 PI (Path Integration) 알고리즘과 비교 하였다. 가장 먼저 비교해 볼 수 있는 특징으

로는 PI 알고리즘의 경우 적은 수의 스텝을 이동하였을 때 적은 오차율을 보였다. 반면에 HFLV-GD 알고리즘은 적은 수의 스텝과 많은 수의 스텝을 비교하였을 때 수행능력의 기록은 없었다. 하지만 HFLV-GD 알고리즘의 경우 학습데이터의 질이 결과에 영향을 많이 미쳤고 Visual Compass의 정확도에 따라 수행능력이 달라졌다.

기본적으로 실제 로봇에 실시간으로 알고리즘을 적용할 때 생기는 오차는 크게 두 가지가 있다. 첫 번째 중요한 오차는 방법 1에서 생기는 오차이다. PI를 통하여 생성된 Desired 값을 통해 학습을 진행하게 되는데 스텝이 늘어남에 따라 Desired 값에도 오차가 생기게 되어 오차까지 그대로 학습되는 경향이 있었다. 그림 5.1에 방법1의 경우 로봇의 주행거리가 늘어남에 따라 PI의 오차가 축적되고 HFLV-GD의 학습 경향도 이를 따라가는 것을 관찰하였다. 하지만 방법2의 경우 주행거리에 상관없이 일정한 퍼포먼스를 유지하는 것을 볼 수 있었다.

본 실험에서 우리는 이미지 기반 실험을 통해 강인함을 확인했던 HFLV-GD에 실제 적용해 보았는데 실제 로봇에 적용할 때 이미지 기반 실험과는 달리 여러 가지 문제점이 있었다. 가장 중요한 문제는 스냅샷 모델을 사용하는 HFLV-GD 모델의 경우 비교하는 이미지의 정렬이 필요하다. 기준이 되는 이미지와 비교하는 이미지상의 정렬이 되어야 유사도를 측정할 수 있기 때문이다. 정확한 Compass를 알아야 하고 그러기 위하여 Visual Compass 알고리즘을 사용하게 되는데 Visual Compass 알고리즘은 기준 이미지와의 거리에 따라 수행능력에 영향을 많이 받게 된다. 그림 5.2의 결과를 보게 되면 Desired 한 Compass를 예측할 수 있을 때 우리는 향상된 결과를 얻을 수 있었다. Desired Compass의 경우 정확한 이미지 정렬을 이용한 결과인데 이처럼 만약 Compass를 정확하게 예측할 수 있으면 좋은 수행능력의 알고리즘을 로봇에 적용할 수 있을 것으로 보인다.

본 로봇 실험에서 우리는 표 5.3를 통해 특징적인 결과를 관찰할 수 있다. 표 5.3의 경우 로봇이 학습된 데이터의 위치와 거리가 멀어질 때 얻을 수 있는 결과이다. 수행능력에 영향을 미치는 조건은 Compass의 영향도 있지만, 마지막으로 방향을 결정하게 될 때 로봇의 위치가 어딘지에 따라 결과가 바뀐다는 것이다. 본 실험에서는 학습을 진행한 데이터와 위치상으로 멀어지게 된다면 에러율이 높아진다는 결과가 나왔다. 이러한

결과로 볼 때 데이터의 질적인 부분이 중요하다는 결론을 얻었다.

본 챕터에서 로봇 실험을 통해 우리는 여러 가지 결론에 도달하였다. 그림 5.8을 보게 되면 전반적으로 PI 알고리즘이 더 높은 수행능력을 갖췄다는 것을 볼 수 있다. 하지만 스텝이 늘어남에 따라 Path Integration의 오차 또한 늘어나기 때문에 본 논문에서 제시하는 HFLV-GD 알고리즘과 함께 사용하는 것이 좋다고 판단된다. PI의 특성적인 부분으로 고려하여 이를 HFLV-GD와 융합하여 사용하면 더 나은 수행능력을 얻을 것으로 판단이 된다. 또한 그림 5.8의 Desired Compass의 결과를 보면 알고리즘을 실시간으로 적용할 때 생기는 여러 가지 문제를 해결하게 되면 좋은 수행능력을 얻을 것으로 기대된다.

제 4 절 요약

이미지 기반 실험은 대체로 좋은 결과를 얻었다. 하지만 본 환경에서 알고리즘을 실시간으로 로봇에 적용할 때 여러 가지 문제를 가지게 되었다. 하지만 Compass 등의 중요한 문제를 해결하면 좋은 수행능력을 얻을 수 있다는 결과를 얻게 되었다. 차후에 여러 가지 파라미터와 조건의 변화를 이용하여 여러 가지 실험을 해볼 필요가 있다. 이미지 기반 실험상의 수행능력에서는 장애물이 있는 환경이라든지 여러 가지 경우에서 HFLV가 강인함을 가졌다. 이를 로봇실험으로 증명해 볼 필요가 있다.

이후 연구에서는 실제 환경에서의 가장 적합한 파라미터를 찾는 연구나 알고리즘의 특징이 분석되는 연구가 필요할 것이다. 또한 역동적인 환경에서 얼마나 적용이 가능한지에 대한 연구도 필요할 것이다. 우리는 앞서 연구된 여러 가지 센서의 결합을 통해 알고리즘을 조금 더 강인하게 만들 수 있을 것으로 판단된다. 따라서 향후 연구로 여러 가지 센서와 알고리즘과의 융합 연구가 필요할 것으로 보인다.

제 6 장

결론 및 향후 과제

제 1 절 결론

HF (Haar-like Features) 는 기존부터 귀소 내비게이션 또는 간단한 유사도 판단 및 탐지에 사용된 기법이다. 그 이유는 계산이 간단하고 비교적 사용하기 쉬운 방법이기 때문이다. 두 이미지 또는 한 개의 기준점이 되는 이미지와 다른 이미지를 비교할 때 HF (Haar-like Features) 의 간단한 화소의 평균 명암을 비교하여 간단하게 판단할 수 있기 때문에 적은 연산능력과 비용으로 내비게이션을 구현할 수 있다. 최근 여러 가지 내비게이션 알고리즘들이 제시되고 있다. 그리고 최근 연구된 알고리즘들의 좋은 수행능력은 부정할 수 없다. 하지만 값비싼 센서와 복잡한 알고리즘에 의한 비용의 증가는 간과할 수 없는 중요한 부분이다.

실제 환경에 적용할 때 간단한 컴퓨팅으로 모바일 로봇의 내비게이션을 구현하는 것은 여러 가지로 경쟁력 있는 이점이 될 수 있고 본 연구진은 그러한 알고리즘을 구현하기 위하여 HF (Haar-like Features) 를 사용하였고 다른 알고리즘에 비해 좋은 성과를 얻었다. 또한 이미지 기반 실험에 적용한 알고리즘을 모바일 로봇에 적용하여 알고리즘의 수행능력을 평가하였다.

우리는 본 논문에서 두 가지 방식으로 실험을 하게 되는데 그 방식은 이미지 기반 실

험 실험과 실제 모바일로봇에 적용하는 실험이다. 이미지 기반 실험 같은 경우는 Vardy의 데이터셋을 사용하게 된다. Vardy의 데이터셋은 여러 가지 환경을 제시하고 있는데 모바일 로봇을 이용하여 포지션마다 데이터를 수집한 실제 환경의 데이터이다. 본 논문에서 구현하는 이미지 기반 실험은 실제 환경의 데이터를 바탕으로 구현하였다. 본 논문에 이미지 기반 실험을 진행할 때 기존에 제시되었던 HFLV (Haar-like Features Landmark Vector)을 개선하기 위해 머신러닝 알고리즘으로 학습을 하였고 머신러닝이 적용된 새로운 알고리즘으로 HFLV-GD (Haar-like Features Landmark Vector)를 제시하였다. 이미지 기반 실험에서 GD 알고리즘을 적용할 때 우리는 가장 이상적인 실험환경을 세팅하였고 그리하여 조금 더 이상적인 결과와 가까운 성과를 얻을 수 있었다.

이미지 기반 실험상에 본 논문에서의 실험들이 도출해 내는 의의와 성과는 다양하다. 가장 첫 번째로 학습을 통해 HF에 대한 분석이 가능해졌다. HF의 분석이 가능해졌다는 이야기는 환경 또는 파라미터별 가장 유용하게 사용될 HF를 알아낼 수 있는 것이고 조건에 따라 좋은 HF를 준비하여 연산량을 더욱 낮출 수 있을 것이다. 이는 내비게이션의 수행능력이나 로봇의 컴퓨팅 시간에 직접적인 영향을 미칠 수 있는 중요한 요소이기 때문에 주목할만한 성과라고 제시한다. 두 번째로 학습을 통해 여러 가지 환경에서 적용 가능한 내비게이션 알고리즘을 구현하였다는 점이다. 환경에 대해서 학습이 가능하다는 점은 생각보다 큰 의미가 있다. 모바일로봇의 실내 내비게이션을 구현하면서 기존의 연구에서 중요한 이슈는 역동적인 환경에 대해서 어떤 식으로 적응하고 대응할 수 있는지가 관건이었다. 본 논문에서 제시한 실험 결과들로 볼 때 학습을 통해 이러한 부분을 어느 정도 돌파할 수 있을 것으로 예상된다.

실제 모바일 로봇에 적용하는 실험은 적용 과정에서 현실적인 성과를 얻었다. 이미지 기반 실험 상황과 다르게 실제 환경에서 알고리즘을 적용하는 것은 여러 가지 제한사항이 생길 수 있다. 하지만 문제들을 파악하여 제한사항에 대한 리스크를 최소화하였고 이는 향후 실험에도 생길 수 있는 문제에 대해서 어느 정도 돌파할 수 있는 실마리가 될 수 있다고 보인다. 여러 가지 알고리즘 또는 센서들과 융합하여 사용할 때 더 좋은 수행능력을 기대할 수 있다.

제 2 절 향후 과제

2.1 알고리즘의 강인함 테스트

본 논문에서 제시한 시각 내비게이션 알고리즘은 로봇에 적용할 때 평범한 실험실 환경에서 적용하였다. 이미지 기반 실험에서는 장애물에 의해 폐쇄이 생긴 지역과 빛의 양이 적은 환경과 같은 여러 가지 환경에도 적용하여 실험하였다. 이미지 기반 실험에서는 여러 가지 환경에서도 알고리즘의 강인함을 증명하였다. 이처럼 차후 연구에서는 로봇 실험도 다양한 실험환경에서의 테스트를 통해 강인함을 증명해야 할 것이다. 또한 위에서 제시한 환경 이외에 실제 숲과 같은 나무가 있는 환경 등에서도 연구가 필요하다.

이전 연구는 대부분 지역적인 한계를 가지고 있었다. 학습된 지역이나 스냅샷을 이용할 때 거리가 멀어지게 되면 수행능력이 낮아졌는데 이러한 지역적인 한계를 뛰어넘을 연구가 필요하다. 이전 연구진은 두 개의 이미지를 이용하는 스냅샷을 세 개의 이미지를 이용하여 좀 더 정확도를 높이는 방법을 택하였다[62]. 세 개의 이미지를 이용하여 X축 좌표의 변화와 Y축 좌표의 변화를 조금 더 정확하게 예측하는 방법이다. 본 논문에서 제시한 경사 하강법과 세 개의 이미지를 사용하는 이전 알고리즘과 융합하여 더욱 강인한 수행능력을 가진 알고리즘을 만들 수 있을 것으로 보인다.

2.2 시각 내비게이션의 어플리케이션

2.3 다양한 모바일 로봇 실험

본 논문에서는 간단한 형태의 머신러닝 알고리즘을 이용하여 학습을 진행하였다. 향후 실험 과제로 다양한 실험에 앞서 이전의 간단한 머신러닝이 아닌 조금 더 향상된 네트워크 구조에 의한 학습을 연구해 볼 필요가 있다. 또한 본 논문에서 진행한 모바일 로봇 실험은 평범한 환경에서 진행하였다. 하지만 4장에서 제시한 것과 같이 여러 가지 환경에서의 수행능력을 체크해볼 필요가 있다. 저 해상도의 환경과 장애물로 인해 음영지역이 생기는 환경 등을 구성하여 알고리즘을 테스트해 볼 필요가 있다.

4장의 성과에 의하면 환경이 변하고 장애물이 있을 때 알고리즘이 잘 작동하는 것으로

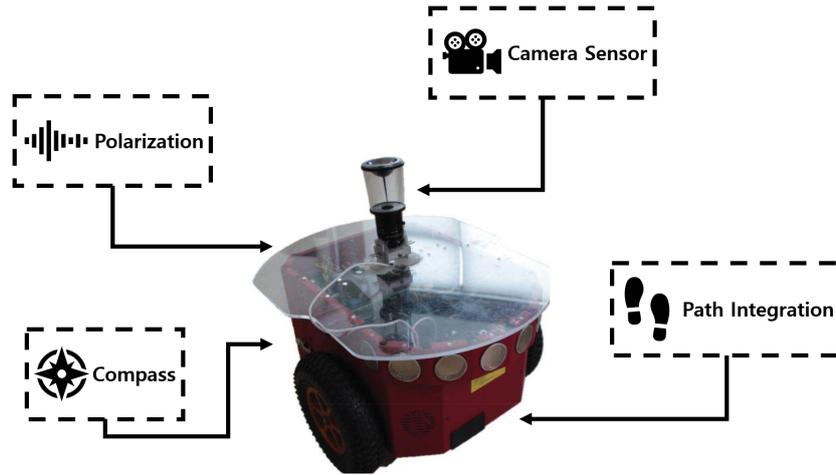


그림 6.1: 여러가지 센서를 장착한 모바일 로봇. 여러가지 센서의 융합을 통해 발전된 형태의 내비게이션을 구현할 수 있을것으로 보인다.

로 보였다. 이것이 로봇 알고리즘에 적용할 때도 똑같은 성과를 얻을 수 있는지 확인해 보아야 할 것이다. 또한 로봇 실험에서 이슈로 남은 문제에 대해서 실험환경별 극복 방법을 연구해보아야 할 것이다. 향후 연구를 통해 환경에 구애받지 않는 극복 방법이 생긴다면 더 높은 수행능력을 기대할 수 있을 것으로 보인다.

시각 내비게이션은 다양한 분야에서 활용될 수 있다. 본 논문에서는 본 알고리즘을 모바일 로봇에 한정시켜 적용하였다. 하지만 본 논문에서 제시된 알고리즘을 여러 가지 센서들과 함께 융합하여 Multi-modal system으로 구축할 수 있을 것으로 판단된다. 사전에 여러 가지 환경에 대해서 실험이 필요할 것이고 시각적인 부분으로만 돌파하기 어려운 문제에 대해서 Multi-modal system을 통해 여러 가지 센서의 융합으로 여러 가지 문제에 대해 돌파가 가능할 것으로 보인다.

향후 향상된 학습능력뿐만 아니라 Multi-modal system을 이용하여 만들어진 알고리즘은 육지뿐만 아니라 여러 환경에서도 적용 가능할 것으로 보인다. 여러 가지 사전 연구가 수행된다면 모바일로봇에 한정된 알고리즘이 아닌 드론과 같은 공중 환경과 바다 한가운데 있는 배 또는 잠수정 같은 환경에서도 쓰일 수 있을 것으로 보인다.

References

- [1] C. Angulo and L. Godo. Using the average landmark vector method for robot homing. *Artificial intelligence research and development*, 163:331, 2007.
- [2] M. Aranda, G. Lopez-Nicolas, and C. Sagües. Sinusoidal input-based visual control for nonholonomic vehicles. *Robotica*, 31:811–823, 2013.
- [3] M. Aranda, G. Lopez-Nicolas, and C. Sagües. Angle-based navigation using the 1D trifocal tensor. In *Control of Multiple Robots Using Vision Sensors*, pages 19–51. Springer, 2017.
- [4] Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atilla Baskurt. Sequential deep learning for human action recognition. In *Human Behavior Understanding*, pages 29–39. Springer, 2011.
- [5] B. Baddeley, P. Graham, P. Husbands, and A. Philippides. A model of ant route navigation driven by scene familiarity. *PLoS computational biology*, 8(1):e1002336, 2012.
- [6] Bart Baddeley, Paul Graham, Andrew Philippides, and Philip Husbands. Holistic visual encoding of ant-like routes: Navigation without waypoints. *Adaptive Behavior*, 19(1):3–15, 2011.

- [7] K. Basten and H.A. Mallot. Simulated visual homing in desert ant natural environments: efficiency of skyline cues. *Biological Cybernetics*, 102(5):413–425, 2010.
- [8] Cornelia Buehlmann, Joseph L Woodgate, and Thomas S Collett. On the encoding of panoramic visual scenes in navigating wood ants. *Current Biology*, 26(15):2022–2027, 2016.
- [9] BA Cartwright and TS Collett. Landmark learning in bees. *Journal of Comparative Physiology A*, 151(4):521–543, 1983.
- [10] B.A. Cartwright and T.S. Collett. Landmark maps for honeybees. *Biological Cybernetics*, 57(1):85–93, 1987.
- [11] S. Y. Chen. Kalman filter for robot vision: A survey. *IEEE Transactions on Industrial Electronics*, 59(11):4409–4420, 2012.
- [12] A. Cheung and R. Vickerstaff. Finding the way with a noisy brain. *PLoS computational biology*, 6(11):e1000992, 2010.
- [13] W. Chung, H. Kim, Y. Yoo, C. Moon, and J. Park. The detection and following of human legs through inductive approaches for a mobile robot with a single laser range finder. *IEEE Transactions on Industrial Electronics*, 59(8):3156–3166, 2012.
- [14] W. Chung, S. Kim, M. Choi, H. Kim, C. Moon, and J. Song. Safe navigation of a mobile robot considering visibility of environment. *IEEE Transactions on Industrial Electronics*, 56(10):3941–3950, 2009.
- [15] Dan Cireşan, Ueli Meier, Jonathan Masci, and Jürgen Schmidhuber. Multi-column deep neural network for traffic sign classification. *Neural Networks*, 32:333–338, 2012.

- [16] M. Collett. How desert ants use a visual landmark for guidance along a habitual route. *Proceedings of the National Academy of Sciences*, 107(25):11638, 2010.
- [17] M. Collett and T.S. Collett. How do insects use path integration for their navigation? *Biological Cybernetics*, 83(3):245–259, 2000.
- [18] T. Collett. Insect navigation en route to the goal: multiple strategies for the use of landmarks. *Journal of Experimental Biology*, 199(1):227–235, 1996.
- [19] Thomas S Collett and Paul Graham. Insect navigation: do honeybees learn to follow highways? *Current Biology*, 25(6):R240–R242, 2015.
- [20] Thomas S Collett, Paul Graham, Robert A Harris, and Natalie Hempel-de Ibarra. Navigational memories in ants and bees: memory retrieval when selecting and following routes. *Advances in the Study of Behavior*, 36:123–172, 2006.
- [21] Thomas S Collett, Antoine Wystrach, and Paul Graham. Insect orientation: The travails of going straight. *Current Biology*, 26(11):R461–R463, 2016.
- [22] J. Conklin and C. Eliasmith. A controlled attractor network model of path integration in the rat. *Journal of computational neuroscience*, 18(2):183–203, 2005.
- [23] AJ Davison. Real-time simultaneous localisation and mapping with a single camera. In *Ninth IEEE International Conference on Computer Vision, 2003. Proceedings*, pages 1403–1410, 2003.
- [24] A.J. Davison, I.D. Reid, N.D. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1052–1067, 2007.
- [25] Francisco J Delgado, José M Quero, Juan Garcia, Cristina L Tarrida, Pablo R Ortega, and Sandra Bermejo. Accurate and wide-field-of-view mems-based sun sensor for

- industrial applications. *IEEE Transactions on Industrial Electronics*, 59(12):4871–4880, 2012.
- [26] Aymeric Denuelle, Reuben Strydom, and Mandyam V Srinivasan. Snapshot-based control of uas hover in outdoor environments. In *Robotics and Biomimetics (RO-BIO), 2015 IEEE International Conference on*, pages 1278–1284. IEEE, 2015.
- [27] Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.
- [28] Jingming Dong, Xiaohan Fei, Nikolaos Karianakis, Konstantine Tsotsos, and Stefano Soatto. VL-SLAM: Real-time visual-inertial navigation and semantic mapping. In *IEEE Conference on Computer Vision and Pattern Recognition, Live Demo*, 2016.
- [29] Basil El Jundi, James J Foster, Lana Khaldy, Marcus J Byrne, Marie Dacke, and Emily Baird. A snapshot-based mechanism for celestial orientation. *Current Biology*, 26(11):1456–1462, 2016.
- [30] A.S. Etienne and K.J. Jeffery. Path integration in mammals. *Hippocampus*, 14(2):180–192, 2004.
- [31] M.O. Franz. *Minimalistic visual navigation, Minimalistische visuelle Navigation*. PhD thesis, Universitat Tübingen, 1999.
- [32] M.O. Franz, B. Schölkopf, H.A. Mallot, and H.H. Bülthoff. Where did I take that snapshot? Scene-based homing by image matching. *Biological Cybernetics*, 79(3):191–202, 1998.
- [33] B.J. Frost and H. Mouritsen. The neural mechanisms of long distance animal navigation. *Current opinion in neurobiology*, 16(4):481–488, 2006.

- [34] Terrence S Furey, Nello Cristianini, Nigel Duffy, David W Bednarski, Michel Schummer, and David Haussler. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):906–914, 2000.
- [35] Douglas D Gaffin and Brad P Brayfield. Autonomous visual navigation of an indoor environment using a parsimonious, insect inspired familiarity algorithm. *PloS One*, 11(4):e0153706, 2016.
- [36] A. Gilg and G. Schmidt. Landmar-oriented visual navigation of a mobile robot. *IEEE Transactions on Industrial Electronics*, 41(4):392–397, 1994.
- [37] Anton Gilg and Gunther Schmidt. Landmark-oriented visual navigation of a mobile robot. *IEEE Transactions on Industrial Electronics*, 41(4):392–397, 1994.
- [38] T. Goedemé, T. Tuytelaars, L. Van Gool, G. Vanacker, and M. Nuttin. Feature based omnidirectional sparse visual path following. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005.(IROS 2005)*, pages 1806–1811, 2005.
- [39] A. Goldhoorn, A. Ramisa, R.L. de Mantaras, and R. Toledo. Using the average landmark vector method for robot homing. *Frontiers in artificial intelligence and applications*, 163:331–338, 2007.
- [40] Gene H Golub, Michael Heath, and Grace Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223, 1979.
- [41] P. Graham, A. Philippides, and B. Baddeley. Animal Cognition: Multi-modal Interactions in Ant Learning. *Current Biology*, 20(15):R639–R640, 2010.

- [42] Paul Graham and Andrew Philippides. Vision for navigation: What can we learn from ants? *Arthropod Structure & Development*, 2017.
- [43] Misha Gupta, GK Arunkumar, and Leena Vachhani. Bearing only visual homing: Observer based approach. In *Control and Automation (MED), 2017 25th Mediterranean Conference on*, pages 358–363. IEEE, 2017.
- [44] J. Hong, X. Tan, B. Pinette, R. Weiss, and E.M. Riseman. Image-based homing. *Control Systems Magazine, IEEE*, 12(1):38–45, 1992.
- [45] Michael Horst and Ralf Möller. Visual place recognition for autonomous mobile robots. *Robotics*, 6(2):9, 2017.
- [46] S. Hwang and J. Song. Monocular vision-based slam in indoor environment using corner, lamp, and door features from upward-looking camera. *IEEE Transactions on Industrial Electronics*, 58(10):4804–4812, 2011.
- [47] J.B. Issa and K. Zhang. Universal conditions for exact path integration in neural systems. *Proceedings of the National Academy of Sciences*, 109(17):6716–6720, 2012.
- [48] Y. Kang, H. Kim, S. Ryu, N. Doh, Y. Oh, and B. You. Dependable humanoid navigation system based on bipedal locomotion. *IEEE Transactions on Industrial Electronics*, 59(2):1050–1060, 2012.
- [49] Mandong Kim and DaeEun Kim. Visual homing navigation of mobile robot using learning of haar-like feature (being prepared). *IEEE Transactions on Robotics and Automation*, 2019.
- [50] Mandong Kim and DaeEun Kim. Visual navigation using learning of haar-like feature (being prepared). *IEEE Transactions on Robotics and Automation*, 2019.

- [51] Y.-K. Kim, Y. Kim, Jung Y., G. Jang, K.-S. Kim, S. Kim, and B. Kwak. Developing accurate long-distance 6-dof motion detection with one-dimensional laser sensors: Three-beam detection system. *IEEE Transactions on Industrial Electronics*, 60(8):3386–3395, 2013.
- [52] T. Kimchi, A.S. Etienne, and J. Terkel. A subterranean mammal uses the magnetic compass for path integration. *Proceedings of the National Academy of Sciences of the United States of America*, 101(4):1105, 2004.
- [53] M. Kohler and R. Wehner. Idiosyncratic route-based memories in desert ants, *melophorus bagoti*: How do they interact with path-integration vectors? *Neurobiology of Learning and Memory*, 83(1):1–12, 2005.
- [54] J.L. Kubie and A.A. Fenton. Heading-vector navigation based on head-direction cells and path integration. *Hippocampus*, 19(5):456–479, 2009.
- [55] C. D. Kuglin and D. C. Hines. The phase correlation image alignment method. In *Proceedings International Conference on Cybernetics and Society*, pages 163–165, 1975.
- [56] T. Kwon and J. Song. A new feature commonly observed from air and ground for outdoor localization with elevation map built by aerial mapping system. *Journal of Field Robotics*, 28(2):227–240, 2011.
- [57] T. Kwon, J. Song, and S. Joo. Elevation moment of inertia: A new feature for monte carlo localization in ourdoor environment with elevation map. *Journal of Field Robotics*, 27(3):371–386, 2010.
- [58] F. Labrosse. The visual compass: performance and limitations of an appearance-based method. *Journal of Field Robotics*, 23(10):913–941, 2006.

- [59] F. Labrosse. Short and long-range visual navigation using warped panoramic images. *Robotics and Autonomous Systems*, 55(9):675–684, 2007.
- [60] D. Lambrinos, R. Moller, T. Labhart, R. Pfeifer, and R. Wehner. A mobile robot employing insect strategies for navigation. *Robotics and Autonomous Systems*, 30(1-2):39–64, 2000.
- [61] Changmin Lee and DaeEun Kim. Image reconstruction-based solution for occlusion problems in visual navigation systems. In *ITC-CSCC: International Technical Conference on Circuits Systems, Computers and Communications*, pages 878–879, 2015.
- [62] Changmin Lee and DaeEun Kim. Visual homing navigation with haar-like features in the snapshot. *IEEE Access*, 6:33666–33681, 2018.
- [63] Changmin Lee, Seung-Eun Yu, and Dae Eun Kim. Landmark-based homing navigation using omnidirectional depth information. *Sensors*, 17(8):1928, 2017.
- [64] E.L.G. Legge, M.L. Spetch, and K. Cheng. Not using the obvious: desert ants, *Melophorus bagoti*, learn local vectors but not beacons in an arena. *Animal cognition*, 13(6):849–860, 2010.
- [65] David D Lent, Paul Graham, and Thomas S Collett. Phase-dependent visual control of the zigzag paths of navigating wood ants. *Current Biology*, 23(23):2393–2399, 2013.
- [66] D.D. Lent, P. Graham, and T.S. Collett. Image-matching during ant navigation occurs through saccade-like body turns controlled by learned visual features. *Proceedings of the National Academy of Sciences*, 107(37):16348–16353, 2010.

- [67] Hyon Lim, Ji-Hyuck Yang, Young-Sam Lee, and Jin-Geol Kim. Indoor single camera slam using fiducial markers. *Journal of Institute of Control, Robotics and Systems*, 15(4):353–364, 2009.
- [68] Ming Liu, Cedric Pradalier, and Roland Siegwart. Visual homing from scale with an uncalibrated omnidirectional camera. *IEEE Transactions on Robotics*, 29(6):1353–1365, 2013.
- [69] P. Luschi, F. Papi, HC Liew, EH Chan, and F. Bonadonna. Long-distance migration and homing after displacement in the green turtle (*Chelonia mydas*): a satellite tracking study. *Journal of Comparative Physiology A*, 178(4):447–452, 1996.
- [70] Sergey Edward Lyshevski. Signal processing in cyber-physical mems sensors: Inertial measurement and navigation systems. *IEEE Transactions on Industrial Electronics*, 2017.
- [71] Jiayi Ma, Ji Zhao, Hanqi Guo, Junjun Jiang, Huabing Zhou, and Yuan Gao. Locality preserving matching. 2017.
- [72] J.A. Mather. Navigation by spatial memory and use of visual landmarks in octopuses. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, 168(4):491–497, 1991.
- [73] GVT Matthews. Sun navigation in homing pigeons. *Journal of Experimental Biology*, 30(2):243–267, 1953.
- [74] B.L. McNaughton, F.P. Battaglia, O. Jensen, E.I. Moser, and M.B. Moser. Path integration and the neural basis of the ‘cognitive map’. *Nature Reviews Neuroscience*, 7(8):663–678, 2006.

- [75] Takeshi Mita, Toshimitsu Kaneko, and Osamu Hori. Joint haar-like features for face detection. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1619–1626. IEEE, 2005.
- [76] M.L. Mittelstaedt and H. Mittelstaedt. Homing by path integration in a mammal. *Naturwissenschaften*, 67(11):566–567, 1980.
- [77] R. Möller. Do insects use templates or parameters for landmark navigation? *Journal of Theoretical Biology*, 210(1):33–45, 2001.
- [78] R. Möller, M. Krzykawski, and L. Gerstmayr. Three 2d-warping schemes for visual robot navigation. *Autonomous Robots*, 29(3):253–291, 2010.
- [79] R. Möller and A. Vardy. Local visual homing by matched-filter descent in image distances. *Biological Cybernetics*, 95(5):413–430, 2006.
- [80] R. Möller, A. Vardy, S. Kreft, and S. Ruwisch. Visual homing in environments with anisotropic landmark distribution. *Autonomous Robots*, 23(3):231–245, 2007.
- [81] Ralf Möller. A model of ant navigation based on visual prediction. *Journal of Theoretical Biology*, 305:118–130, 2012.
- [82] Ralf Möller. Column distance measures and their effect on illumination tolerance in minwarping. Technical report, Tech. rep., University of Bielefeld, Faculty of Technology, Computer Engineering Group, 2016 http://www.ti.uni-bielefeld.de/html/people/moeller/tsimd_warpingsimd.html, 2016.
- [83] Ralf Möller. A simd implementation of the minwarping method for local visual homing. Technical report, Technical report, Computer Engineering, Faculty of Technology, Bielefeld University, 2016.

- [84] R.C. Nelson and J. Aloimonos. Finding motion parameters from spherical motion fields (or the advantages of having eyes in the back of your head). *Biological Cybernetics*, 58(4):261–273, 1988.
- [85] Wanli Ouyang and Xiaogang Wang. Joint deep learning for pedestrian detection. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2056–2063. IEEE, 2013.
- [86] C. Pan, H. Deng, X.F. Yin, and J.G. Liu. An optical flow-based integrated navigation system inspired by insect vision. *Biological cybernetics*, pages 1–14, 2011.
- [87] Sunhong Park and Shuji Hashimoto. Autonomous mobile robot navigation using passive rfid in indoor environment. *IEEE Transactions on Industrial Electronics*, 56(7):2366–2373, 2009.
- [88] A. Ramisa, A. Goldhoorn, D. Aldavert, R. Toledo, and R.L. de Mantaras. Combining invariant features and the ALV homing method for autonomous robot navigation based on panoramas. *Journal of Intelligent & Robotic Systems*, pages 1–25, 2011.
- [89] S.F. Reid, A. Narendra, J.M. Hemmi, and J. Zeil. Polarised skylight and the landmark panorama provide night-active bull ants with compass information during route following. *Journal of Experimental Biology*, 214(3):363–370, 2011.
- [90] S.M. Reppert, R.J. Gegear, and C. Merlin. Navigational mechanisms of migrating monarch butterflies. *Trends in neurosciences*, 33(9):399–406, 2010.
- [91] A. Samsonovich and B.L. McNaughton. Path integration and cognitive mapping in a continuous attractor neural network model. *The Journal of Neuroscience*, 17(15):5900–5920, 1997.

- [92] Clément Sanchez, Hervé Arribart, and Marie Madeleine Giraud Guille. Biomimetism and bioinspiration as tools for the design of innovative materials and systems. *Nature Materials*, 4(4):277, 2005.
- [93] T. Sasaki, D. Brscic, and H. Hashimotor. Human-observation-based extraction of path patterns for mobile robot navigation. *IEEE Transactions on Industrial Electronics*, 57(4):1401–1410, 2010.
- [94] S. Schwarz and K. Cheng. Visual associative learning in two desert ant species. *Behavioral Ecology and Sociobiology*, 64(12):2033–2041, 2010.
- [95] Sebastian Schwarz, Michael Mangan, Jochen Zeil, Barbara Webb, and Antoine Wystrach. How ants use vision when homing backward. *Current Biology*, 27(3):401–407, 2017.
- [96] L. Smith, A. Philippides, P. Graham, B. Baddeley, and P. Husbands. Linked local navigation for visual route guidance. *Adaptive Behavior*, 15(3):257–271, 2007.
- [97] Mandyam V Srinivasan. How lost “passenger” ants find their way home. *Learning & Behavior*, pages 1–2, 2017.
- [98] K. Steck, M. Knaden, and B.S. Hansson. Do desert ants smell the scenery in stereo? *Animal Behaviour*, 79(4):939–945, 2010.
- [99] Reuben Strydom, Aymeric Denuelle, and Mandyam V Srinivasan. Bio-inspired principles applied to the guidance, navigation and control of uas. *Aerospace*, 3(3):21, 2016.
- [100] W. Stürzl and HA Mallot. Efficient visual homing based on fourier transformed panoramic images. *Robotics and Autonomous Systems*, 54(4):300–313, 2006.

- [101] W. Stürzl and R. Möller. An insect-inspired active vision approach for orientation estimation with panoramic images. *Bio-inspired Modeling of Cognitive Tasks*, pages 61–70, 2007.
- [102] W. Stürzl and J. Zeil. Depth, contrast and view-based homing in outdoor scenes. *Biological Cybernetics*, 96(5):519–531, 2007.
- [103] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation from predicting 10,000 classes. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1891–1898. IEEE, 2014.
- [104] S. Thrun and M. Montemerlo. The graph SLAM algorithm with applications to large-scale mapping of urban structures. *The International Journal of Robotics Research*, 25(5-6):403–429, 2006.
- [105] O. Trullier, J.A. Meyer, et al. Biomimetic navigation models and strategies in animats. *AI communications*, 10(2):79–92, 1997.
- [106] A. Ugolini, G. Borgioli, G. Galanti, L. Mercatelli, and T. Hariyama. Photoresponses of the compound eye of the sandhopper *Talitrus saltator* (Crustacea, Amphipoda) in the ultraviolet-blue range. *The Biological Bulletin*, 219(1):72–79, 2010.
- [107] A. Vardy. Long-range visual homing. In *Robotics and Biomimetics, 2006. RO-BIO'06. IEEE International Conference on*, pages 220–226. IEEE, 2006.
- [108] R.J. Vickerstaff and T. Merkle. Path integration mediated systematic search: a bayesian model. *Journal of Theoretical Biology*, 2012.
- [109] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

- [110] Wei Wang and Guangming Xie. Online high-precision probabilistic localization of robotic fish using visual and inertial cues. *IEEE Transactions on Industrial Electronics*, 62(2):1113–1124, 2015.
- [111] K. Weber, S. Venkatesh, and M. Srinivasan. Insect-inspired robotic homing. *Adaptive Behavior*, 7(1):65–97, 1999.
- [112] R. Wehner and F. Rüber. Visual spatial memory in desert ants, *Cataglyphis bicolor* (Hymenoptera: Formicidae). *Cellular and Molecular Life Sciences*, 35(12):1569–1571, 1979.
- [113] Rüdiger Wehner. Early ant trajectories: spatial behaviour before behaviourism. *Journal of Comparative Physiology A*, 202(4):247–266, 2016.
- [114] Rüdiger Wehner, Barbara Michel, and Per Antonsen. Visual navigation in insects: coupling of egocentric and geocentric information. *Journal of Experimental Biology*, 199(1):129–140, 1996.
- [115] S. Wintergerst and B. Ronacher. Discrimination of inclined path segments by the desert ant *cataglyphis fortis*. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, pages 1–11, 2012.
- [116] T. Wittmann and H. Schwegler. Path integration—a network model. *Biological Cybernetics*, 73(6):569–575, 1995.
- [117] A. Wystrach, G. Beugnon, and K. Cheng. Ants might use different view-matching strategies on and off the route. *The Journal of Experimental Biology*, 215(1):44–55, 2012.
- [118] A. Wystrach, S. Schwarz, P. Schultheiss, G. Beugnon, and K. Cheng. Views, landmarks, and routes: how do desert ants negotiate an obstacle course? *Journal of Com-*

- parative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, 197(2):167–179, 2011.
- [119] Seung-Eun Yu and DaeEun Kim. Image-based homing navigation with landmark arrangement matching. *Information Sciences*, 181(16):3427–3442, 2011.
- [120] Seung-Eun Yu and DaeEun Kim. Landmark vectors with quantized distance information for homing navigation. *Adaptive Behavior*, 19(2):121–141, 2011.
- [121] Seung-Eun Yu and DaeEun Kim. Analyzing the effect of landmark vectors in homing navigation. *Adaptive Behavior*, 20(5):337–359, 2012.
- [122] J. Zeil, M.I. Hofmann, and J.S. Chahl. Catchment areas of panoramic snapshots in outdoor scenes. *Journal of the Optical society of America A*, 20(3):450–469, 2003.
- [123] J. Zeil and J. Layne. Path integration in fiddler crabs and its relation to habitat and social life. *Crustacean experimental systems in neurobiology*, pages 227–246, 2002.
- [124] Jochen Zeil. Visual homing: an insect perspective. *Current opinion in neurobiology*, 22(2):285–293, 2012.
- [125] Q. Zhu, C. Liu, and C. Cai. A Robot Navigation algorithm based on sparse landmarks. In *Proceedings of the 6th Int. Conf. on Intelligent Human-Machine Systems and Cybernetics*. IEEE, 2014.
- [126] Qidan Zhu, Chuanjia Liu, and Chengtao Cai. A novel robot visual homing method based on sift features. *Sensors*, 15(10):26063–26084, 2015.