

개선된 비문(鼻紋) 기반의 개 개체인식 시스템

이민정, *박종근, **정제창^{†1}

한양대학교

aeca2002@gmail.com, *parkjk14@hanyang.ac.kr, **jjeong@hanyang.ac.kr[†]

An improved system of Dog Identification based on Muzzle Pattern

Minjeong Lee *Jonggeun Park **Jechang Jeong[†]

Hanyang University

요 약

본 논문에서는 비문(鼻紋) 이미지를 이용한 개의 개체인식 시스템에서 정확도를 높이는 방법을 논하고자 한다. 비문 이미지에는 개 크의 특성상 반사로 인한 다수의 잡음이 생기게 되는데 이는 전체 시스템의 인식률을 낮추는 주요 원인이다. 본 알고리즘에서는 반사광으로 인해 비문 이미지에서 탐지된 잘못된 특징점을 제거하고 RANSAC 알고리즘을 적용하여 정확도를 높이는 방법을 제안한다. 실험 결과 본 알고리즘은 종이에 비문을 찍어내지 않고도 충분한 정확도를 가지는 시스템을 구축 할 수 있음을 보였다.

1. 서론

최근 전국적으로 유기견이 급격히 늘어나면서 사회문제가 되었다. 2011 년에서 2013 년까지 유기동물 발생규모는 연간 약 10 만건에 이르고 있으나 이들을 보호할 시설은 전국에 361 개소로 턱없이 부족한 실정이다[3]. 유기동물은 그들의 보호와 안락사 문제 같은 윤리적 문제, 사회적 문제뿐만 아니라 공중보건의 문제로도 연결된다. 공중보건학적으로 중요한 광견병, 브루셀라병, 심장사상충 감염증 등과 같은 중요 질병의 매개체가 되기 때문이다[3]. 이러한 문제를 해결하기 위해 정부에서도 동물등록제, 유기동물 보호소 운영과 같은 정책을 실시하고 있지만 효율적인 방법이라고 보기 어렵다. 잃어 버린 애견을 보호하며 견주가 찾아가기를 기다리는 것 보다는 잃어 버리더라도 쉽게 찾을 수 있는 근본적인 방향의 접근이 필요한 시점이다.

비문이란(鼻紋) 동물의 코에서 발견할 수 있는 고유한 무늬(Biometric marker)이다[1]. 사람의 지문과 같은 역할을 할 수 있어, 최근 비문을 도입한 동물의 개체인식 시스템이 품종협회(breed association), 식품제조이력 관리 시스템 등에서 주목을 받고 있다[1]. 소 사육 분야를 예로 들면 기존에는 태깅(Tagging), RFID(Radio-frequency identification)등이 사용되어 오고 있었는데 태그가 탈락되거나 RFID chip 의 결함 등으로 인한 문제가 발생했다. 이러한 한계를 극복하기 위해 최근 시도되고 있는 새로운 시스템이 컴퓨터 비전의 한 분야인 객체인식(Object recognition) 알고리즘과 소의 비문을 이용한 개체인식 시스템이다.

비문을 이용한 개체인식 방법에는 크게 두 가지가 있다. 첫 번째는 종이에 비문을 찍어내어 일반화된(generalized) 데이터로 만들어 내는 것이다. 이 방법의 장점은

어파인(Affine) 과 같은 일반적인 객체인식 알고리즘이 기본적으로 가지고 있는 문제에서 자유로울 수 있다는 것이다. 그러나 종이에 비문을 찍어내는 과정에서 작업자의 숙련도가 요구되고 데이터화 과정이 추가로 필요하기 때문에 시간과 비용이 추가로 발생한다는 단점이 있다[2]. 두 번째는 비문 이미지에서 특징을 추출(feature detection)하여 개체인식에 사용하는 방법이다[1]. 이 방법은 개체인식을 위한 데이터 확보가 쉽고 실시간 서비스를 제공하기에 적합하다는 장점을 가지고 있다. 단점은 빛에 민감하다는 것이다. 빛과 카메라가 이루는 각도에 따라 반사광이 발생하고, 이로 인해 원래 특정 지점이 갖는 값이 아닌 이미지 센서의 최대 값으로 수렴되어 잘못된 특징점이 추출될 확률이 높아진다. 본 논문에서는 비문 이미지로 개체인식 시스템을 구축 할 때, 빛으로 인한 잘못된 특징점 추출을 방지하여 정확도를 높이는 방안을 제안한다.

본 논문의 구성은 다음과 같다. 2 절에서 기존의 알고리즘인 SURF 와 FREAK, RANSAC 을 이용한 일반적인 이미지 정합 과정을 알아본 후 3 절에서 비문 이미지 전처리 과정과 제안 개체인식 알고리즘을 설명한다. 마지막으로 4 장에서 알고리즘의 정합 결과와 한계점을 알아본다.

2. 기존의 알고리즘

I. SURF(Speeded-Up Robust Features) 특징점 추출

SUFT 는 영상에서 특징점을 찾아 추출하는 SIFT(Scale Invariant Feature Transform) 알고리즘을 고속화 시킨 알고리즘이다[6] [7]. 여기서 말하는 특징점이란 명도(intensity) 값의 변화가 큰 부분을 말한다. SURF 알고리즘은 적분영상(integral image)에서 근사 고속 헤이시안

[†] Corresponding author : 정제창(Jechang Jeong)

TEL : +82-2-2220-4370

행렬(Fast-Hessian Detector)의 행렬식(Determinant)을 이용하여 특징점을 찾고, 비 최대치 억제(non-maximum suppression) 기법을 통해 최종적으로 특징점이 되는지를 판별하는 과정을 거친다[4].

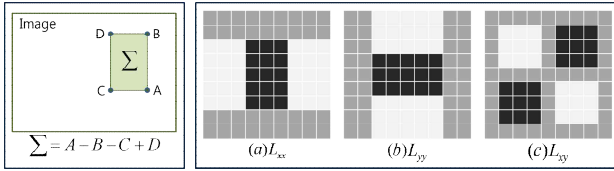


그림 1. 적분 영상(좌)과 근사 헤이시안 행렬의 박스 필터(우)

적분영상은 원래 픽셀의 값 대신 현재 픽셀까지의 합을 저장한 영상이다. 이를 이용하면 그림 1의 (좌)와 같이 ABCD 영역의 합을 덧셈과 뺄셈 연산만으로 빠르게 구할 수 있다. 적분영상에 그림 1의 (우)와 같이 근사화한 박스 필터를 적용하여, σ^2 의 분산을 취하는 가우시안 2차 미분 필터 컨볼루션 연산을 행하는 것을 헤이시안 행렬이라 하고 식은 다음과 같다:

$$H(x, y, \sigma) = \begin{bmatrix} L_{xx}(x, y, \sigma) & L_{xy}(x, y, \sigma) \\ L_{xy}(x, y, \sigma) & L_{yy}(x, y, \sigma) \end{bmatrix} \quad (1)$$

이 때, L_{xx} , L_{yy} , L_{xy} 는 σ^2 의 분산을 취하는 x 방향, y 방향, x와 y 방향으로의 2차 편도 함수(Second Order Partial Derivative)의 근사식을 의미한다.

이렇게 구한 근사 헤이시안 행렬의 행렬식이 지정한 임계치 보다 클 때 특징점 후보가 된다. 마지막으로 비 최대치 억제 기법에 적용하여, 인접한 옥타브(Octave)의 3x3x3 영역에서 후보 특징점의 헤이시안 행렬식 값이 최대 또는 최소일 때 특징점으로 확정한다.

II. FREAK(Fast Retina Keypoint) 특징 기술자와 정합(Match)

특징 기술자(Feature descriptor)는 이 전 단계에서 찾아낸 특징점의 집합이다. 보통은 특징점만으로 기술자를 만들지 않고 특징점 주위의 영역을 함께 이용하여 만들게 된다. FREAK 특징 기술자는 특징점 주위에 정해진 영역의 정보를 모두 저장하지 않고 필요한 부분만 sampling 해서 이용하는 binary string 방식을 사용한다[8]. 필요한 부분은 Sampling pattern이라 불리며 여러 가지가 있는데, FREAK 특징 기술자는 인간의 망막 움직임을 모사한 Retina sampling pattern을 사용한다.

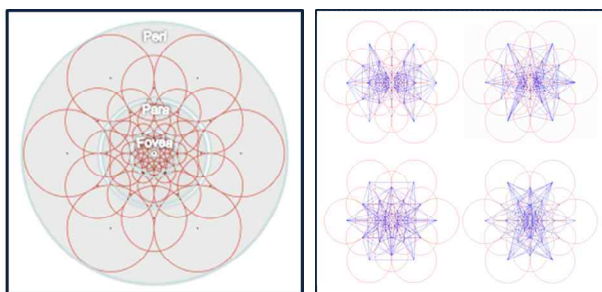


그림 2. Retina sampling pattern(좌)과 Saccadic search(우) [8]

인간의 망막은 중심부에서 바깥으로 foveal, fovea, parafoveal, perifovea의 4가지 영역을 갖는다[8]. 각 영역은

서로 다른 해상도에 반응하며 중심 영역이 고해상도의 영역을 감지하고 바깥쪽으로 갈수록 점점 더 저해상도의 영역을 감지하게 된다. 그 때문에 그림 2의 좌측 Retina sampling pattern 처럼 중심의 고해상도 영역은 조밀한 sampling이 필요하고 바깥 영역에 대해서는 성근 sampling을 해도 된다. 그림 2 좌측에서 붉은색 점으로 표시된 원의 중심 점이 해당 영역을 대표하는 값이 된다.

Saccadic search는 인간의 눈동자가 단속적, 이상적인 빠른 운동을 하여 보고자 한 영역을 빠르게 탐색하는 행위를 말한다[5]. 뇌에서는 탐색한 영상을 재조합 하여 우리가 보고 있는 장면을 만든다. FREAK은 Saccadic search의 움직임을 모사하여 Sampling pattern에서 획득한 점들의 쌍을 만들고 크기를 비교하여 1 또는 0의 binary string을 만든다. Binary string을 정합할 때는 XOR 연산을 통해서 한다.

III. RANSAC(RANDOM Sampling Consensus) 알고리즘

특징점 기술자를 정합한 결과가 모두 맞는 것은 아니다. 추정하고자 하는 모델이 있을 때, 찾아낸 특징들 중 추정 모델의 정상 분포에 속하는 데이터는 inlier, 반대로 정상 분포에서 크게 벗어난 데이터를 outlier라고 한다. RANSAC은 무작위로 정해진 수의 데이터를 뽑아 변형 행렬을 측정하여 모델을 만든 다음, 가장 많은 데이터가 동의한 모델을 찾아낸다[9].

먼저 어떤 모델을 추정하는 지에 따라 뽑을 데이터 표본 개수를 정한다. 점의 경우 1개, 선의 경우 2개, 포물선의 경우 3개와 같은 방식이다. 앞에서 찾아낸 특징점을 추정하고자 하는 모델에 따라 개수를 정해 무작위로 뽑는다. 뽑힌 점을 기반으로 새로운 모델을 만들고, 모든 데이터와 모델 사이의 오류를 측정한다. 오류가 특정 임계치 이하이면, 새로운 모델에 동의하는 데이터의 개수를 센다. 이 과정을 반복해 가장 많은 데이터가 동의한 모델을 찾아낸다. 여기서 동의 한다라는 표현은 추정 모델의 정상 분포 안에 속한다는 뜻이다.

P \ N	1	2	3	4	5	6
0.9	1.1	1.2	1.4	1.5	1.7	1.9
0.8	1.3	1.6	2.0	2.4	3.0	3.8
0.7	1.4	2.0	2.9	4.2	5.9	8.5
0.6	1.7	2.8	4.6	7.7	13	21
0.5	2.0	4.0	8.0	16	32	64
0.4	2.5	6.3	16	39	98	244
0.3	3.3	11	37	123	412	-
0.2	5.0	25	125	625	-	-

표 1. RANSAC 반복 횟수 [9]

RANSAC 알고리즘의 저자는 inlier의 비율과 표본 개수에 따라 반복 횟수를 계산하는 표를 만들어 발표했다. 표 1의 N이 표본 개수, P가 inlier의 비율을 나타낸다.

3. 제안 알고리즘

I. ROI(Region Of Interest) 영역

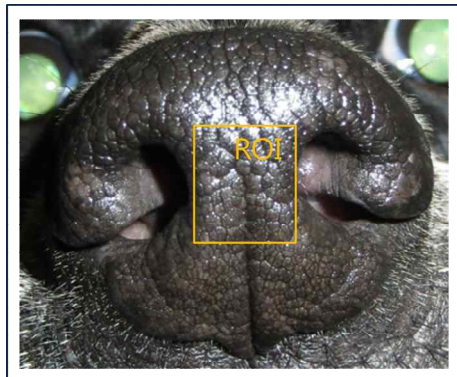


그림 3. 개의 ROI(Region Of Interest) 영역

일반적인 객체인식 알고리즘에서 전형적인(Typical) 이미지로 사용하는 물체는 주로 강체이다. 동물의 코는 작은 압력에 의해서도 변형이 발생 할 수 있기 때문에 엄밀한 의미에서의 강체는 아니지만 근육으로 움직이기 어려운 콧구멍 사이의 부분을 아무런 압력이 가해지지 않았다는 가정 하에 ROI 영역으로 선정 했다.

II. 알고리즘의 순서도

제안하는 알고리즘의 순서도는 그림 4 와 같다. 먼저 비문을 입력 받아서 최대 수렴값의 좌표를 검출 한다. 비문 이미지의 잡음 제거를 위해 중간값 필터(median filter)를 씌우고 잃어버린 에지(edge)를 보강해 주기 위해서 언샤프 마스크(unsharp mask)를 씌워서 이미지 전처리 과정을 마친다. 전처리 과정을 거친 두 장의 이미지에서 특징점을 추출하고 앞에서 검출한 최대 수렴값 특징점을 제거한다. 찾아낸 특징점 기반으로 특징점 기술 과정을 거쳐 두 장의 이미지를 데이터화 시키고 XOR 연산을 이용하여 정합을 한다. 이 정합의 결과물에는 outlier 도 포함되어 있기 때문에 이를 제거해 줄 RANSAC 알고리즘이 필요하다. RANSAC 사용은 4 개의 점을 이용하여 호모그래피(homography)를 추정하는 방식으로 수행했다. 다만 강체의 특성상 뒤틀림(twist)과 오목함(concave)을 제외했다.

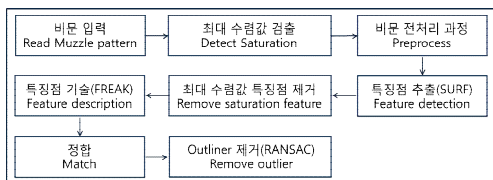


그림 4. 제안 알고리즘 순서도

근사 고속 헤이시안 행렬(Fast-Hessian Detector)은 특정 영역의 기울기(Gradient)가 얼마나 변했는가를 나타낸다. 빛 반사로 인해 원래의 값과 다르게 최대 값으로 수렴된 좌표는 급격한 기울기 변화의 원인이 되고, 그 주위에서 잘못된 특징점이 다수 검출 된다. 이러한 특징점을 제거함으로써 정확도 향상을 모색할 수 있다. 이 과정의 목표는 RANSAC 알고리즘의 inlier 비율 50% 확보이다. 2 장 말미에 설명 했듯이 RANSAC 알고리즘을 사용하면 inlier 의 비율에 따라 반복 횟수가 결정 되는데, 연산 량이 많아져 전체 알고리즘 속도가 현저히 느려질 수 있다.

III. 최대 수렴값 검출 방법과 비문 전처리 과정

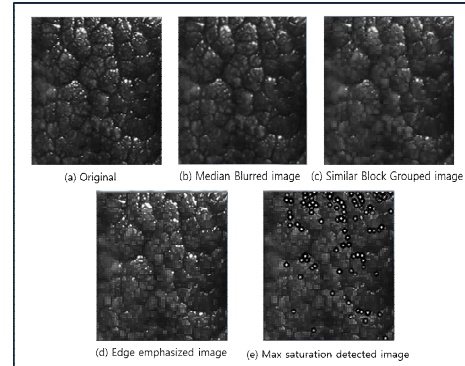


그림 5. 최대 수렴값 검출 단계

그림 5는 최대 수렴값 검출 과정의 각 단계별 결과 영상을 보여주는 그림이다. 원본 이미지에 (b)와 같이 중간값 필터를 씌우면 소금-후추(salt and pepper) 잡음 같이 작은 영역을 차지하는 최대 수렴값은 없어지고 큰 영역을 차지하는 최대 수렴값은 한쪽으로 모이게 된다. 한쪽으로 큰 영역의 최대 수렴값이 모이는 현상을 극대화 하기 위해 다음과 같은 두 가지 식을 사용하게 된다:

$$DSum = \frac{1}{9} \sum_{i=0}^{i<3} \sum_{j=0}^{j<3} |I(1,1) - I(i,j)| \quad (2)$$

$$I(i,j) = \max \text{ (if } DSum < \text{Threshold)}. \quad (3)$$

식(2)의 i 와 j 는 각각 가로 축, 세로 축의 좌표를 나타낸다. DSum 은 3x3 영역 안에서 중앙 픽셀(pixel) 값과 나머지 픽셀 값들의 차를 절대값을 씌워 모두 합한 후 평균 낸 것이다. 만약 DSum 이 특정 임계치 보다 작은 경우 변화가 적은 평탄한 영역으로 인식하고 식 (3)과 같이 3x3 영역 안의 최대값(max)으로 모든 값을 치환한다.

이런 방식으로 비교적 평탄한 영역의 값을 최대 값으로 치환하면 그림 5 의 (c)와 같은 영상을 얻을 수 있다. 최대 수렴값이 큰 영역으로 한데 모인 것을 관찰할 수 있을 것이다. (d)는 최대 수렴값이 더욱 도드라지게 하기 위해 (c)영상에 언샤프 마스크를 씌운 영상이다. (d)의 모든 좌표의 값을 검사하여 이미지 값 범위(Range)의 상위 20%에 해당하면 반사에 의한 최대 수렴값으로 취급한다. 이러한 픽셀을 표시한 그림이 (e)에 해당한다.

4. 실험 결과 및 분석

Parameter	Values
Determinant of Fast-Hessian Detector	0
Median filter size	3
Distance between a point and a Saturated point	3.0
RANSAC inlier threshold	3.0

표 2. 실험 조건(Parameter)과 값(value)

본 실험은 OpenCV 2.4.9 version 에서 36 개체의 비문 이미지를 대상으로 진행 되었다. 이 중 4 장만이 입력으로 들어 올 전형적인 비문 이미지와 동일 개체에서 찍힌 것이다. 목적은 동일 개체의 판별이다. 실험에 쓰인 조건(parameter)은 표

2와 같다.

ROI 영역이 굉장히 작기 때문에 최대한 많은 특징점을 찾아내는 것이 중요하다. 이를 위하여 근사 고속 헤이시안 행렬의 행렬식 판별값을 0으로, median filter size도 최대한 작은 3으로 설정했다. 최대 수렴값 좌표와 특징점과의 거리는 두 점이 동일한 것인지 판별해 줄 거리 값이다. 설정한 값보다 작으면 동일한 점으로 판단되어 제거 될 것이다. RANSAC inlier 임계치는 일반적인 설정을 따랐다.

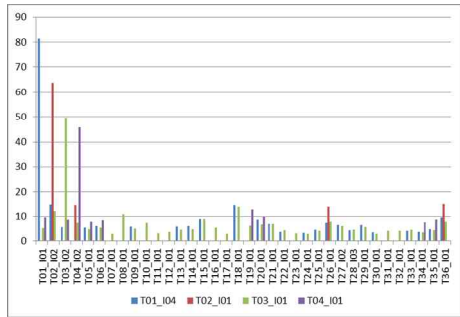


그림 6. 36 마리 개체의 정합 결과

그림 6은 정합 결과를 나타낸다. 가로 축과 범례에 비문 이미지의 이름이 TXX_IYY와 같은 형식으로 표시되어 있다. XX가 개체의 고유번호, YY가 같은 개체의 몇 번째 사진인지에 대한 정보이다. 각 막대그래프의 최대 값이 동일한 개체번호의 다른 사진에서 관찰되는 것을 확인할 수 있다. 일치하지 않는 경우 결과 값이 0이거나 최대 값에 비해 매우 낮다.

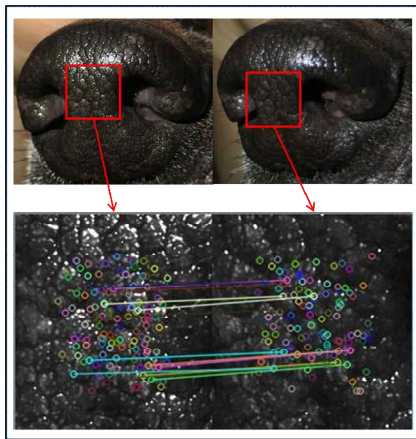


그림 7. 어파인(Affine) 변환된 이미지와 정합 결과

본 알고리즘은 그림 7에서 보이듯 낮은 정도의 어파인 변환된 이미지와의 정합에서 outlier 없이 정확한 호모그래피를 찾아낸다. 다만 여타 다른 객체인식 알고리즘과 같이, 어파인의 정도와 정확도가 반비례 한다는 한계점을 가지고 있다.

5. 결론

본 논문에서는 비문 이미지로 개체인식 시스템을 구축할 때, 빛으로 인해 추출되는 잘못된 특징점을 제거함으로써 정확도를 확보하는 방법을 제안했다. 정합 결과 제안한 알고리즘이 모든 개체를 정확히 찾아냈으며 일치하지 않는

개체에 대해서는 상당히 낮은 결과를 도출했다. 본 알고리즘은 중이에 비문을 찍어내지 않고도 충분한 정확도를 가지는 시스템을 구축할 수 있음을 보였다.

감사의 글

"본 연구는 미래창조과학부 및 정보통신기술진흥센터의 대학 ICT 연구센터 육성지원사업의 연구결과로 수행되었음" (IITP-2015-H8501-15-1005)

참 고 문 헌

- [1] A. Noviyanto, and A. Murni Arymurthy, "Automatic cattle identification based on muzzle photo using speed-up robust features approach," *Proceedings of the 3rd European Conference of Computer Science*, Vol. 110, pp. 110-114, 2012.
- [2] A. I. Awad, H. M. Zawbaa, H. A. Mahmoud, E. H. H. A. Nabi, R. H. Fayed, and A. E. Hassanien, "A Robust Cattle Identification Scheme Using Muzzle Print Images," *Computer Science and Information Systems (FedCSIS), 2013 Federated Conference on. IEEE*, pp. 529-534, 2013.
- [3] 윤가리, 정은하, 라도경, 정철, 이경민, 채현우, 이정구, 이성모, "인천지역 유기동물 현황 및 개 질병 조사," *한국가축위생학회지*, vol. 37(4), pp. 297-305, 2014.
- [4] A. Neubeck, and L. V. Gool, "Efficient non-maximum suppression," *International Conference on Pattern Recognition 18th International Conference on.*, Vol. 3, pp. 850-855, 2006.
- [5] M. Hogan and J. J. Weddell, "Histology of the human eye : and atlas and textbook," 1971.
- [6] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (SURF)," *Computer vision and image understanding*, vol. 110, no.3, pp. 346-359, 2008.
- [7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no.2, pp. 91-110, 2004.
- [8] A. Alahi, R. Ortiz, and P. Vanderghenst, "Freak: Fast retina keypoint," *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE*, pp. 510-517, 2012.
- [9] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24(6), pp. 381-395, 1981.