

무인항공기를 위한 영상기반의 활주로 인식 및 항법 연구

Vision-based Runway Recognition and Navigation for UAVs

홍경우, 박준우, 구성모, 정우영, 방효충
KAIST 항공우주공학과

초 록

무인항공기에 대한 관심과 연구가 진행됨에 따라 오차가 생기고 값비싼 GPS 대신 영상기반 항법 시스템이 GPS 대안으로서 연구되고 있다. 본 논문은 영상기반의 활주로 인식 및 항법 해 계산에 대한 연구를 서술하고 있다. 실제 비행영상에 활주로 인식을 쉽게 하기위해 여러 가지의 전 처리 과정을 거친 후 호프 변환을 이용하여 활주로를 인식하였다. 또한 실제 영상을 쉽게 구할 수 없는 문제 때문에 비행기 위치 및 자세에 따른 활주로의 위치 및 자세를 계산하고 실제 영상을 대신하는 시뮬레이션을 구현하였다. 그리고 시뮬레이션의 항법 해를 계산하고 실제 값과 비교하였다. 이를 통해 실제 비행영상에도 이용할 수 있을 것으로 기대된다.

ABSTRACT

With increasing interest and research on UAVs, vision-based navigation system is being developed as an alternative to GPS, which is cost-ineffective and less accurate. This study explored growing research on recognition of runways by vision-based system and calculations on navigation. In order to allow vision-based system to easily capture runways, hough transform was used after a series of processing stages. Due to the difficulty of acquiring actual video samples, simulation program, which enabled calculation of runway position with respect to aircraft position, was used as an alternative. In addition, simulation program's navigation was calculated, and compared with the actual value. With this result, it is expected to be used in real-life aircraft videos.

Key Words : runway recognition(활주로 인식), vision-based(영상기반), navigation(항법)

1. 서 론

보편적으로 무인항공기의 항법을 알기위해 사용되는 도구는 Global Positioning System(GPS)이다. GPS를 이용한 무인항공기는 GPS 센서를 통해 비행체의 자세 및 위치를 알아낸다. 그러나 GPS의 오차로 인해 정밀한 항법 정보를 얻지 못하는 것이 GPS의 가장 큰 문제점이다. GPS에 비해 영상기반 항법 시스템은 무인기에 장착된 카메라를 통해 획득한 영상으로부터 비행체의 자세 및 착륙위치와 비행체의 상대위치를 추출해낸다. 카메라로 얻은 영상은 많은 정보를 포함하고 있으며 이 영상을 일련의 공정을 거쳐 추가적으로 필요한 정보를 얻으면 무인기의 자동 이착륙도 가능하게 한다. 이 뿐만 아니라 GPS를 설치하는 비용에 비해 카메라 설치비용이 훨씬 적다는 장점을 가지고 있다.

기존의 영상기반 항법 시스템은 영상에서 활주로를 인식 후 그 좌표를 이용하여 비행체의 자세 및 위치를 계산하는 방법도 있으며¹⁾ 자세 및 위치를 계산하지 않고 영상에서 원하는 지점까지 바로 제어하는 방법도 있다.⁽²⁾

이외에도 실제 비행 영상을 구하기 어렵고 기존의 비행 영상을 통해 비행체의 자세 및 위치 정보를 정밀하게 계산이 불가능하기 때문에 비행체의 자세와 위치가 정해졌을 때 활주로의 어디에 위치할 지 계산하는 방법도 제시되었다.⁽³⁾

본 연구에서는 일반 영상에서 활주로를 인식하고 인식한 데이터를 바탕으로 비행체의 항법 해를 계산하도록 한다. 2절에서는 영상에서 활주로를 인식하는 과정을 소개하며 3절에서는 시뮬레이션 구현과 항법해를 계산하는 방법을 제시한다. 그리고 마지막 4절에서는 결론을 맺도록 한다.

2. Runway Recognition

2.1 전 처리 과정

UAV에서 얻은 영상을 토대로 활주로를 인식한 후 UAV의 안전한 착륙으로 이어지기 위해서는 이미지 형태의 정보를 인식하기 조금 더 쉬운 형태로 바꾸어주는 전 처리 과정이 필요하다.

2.1.1 히스토그램 평활화

모든 이미지는 UAV에 장착된 카메라의 성능, 이미지를 얻은 시간, 그 날의 일조량, UAV의 고도와 속도 등에 영향을 받는다. 여러 가지 요인들에 의해 변할 수 있는 이미지를 표준화하기 위해 히스토그램 평활화를 이용하여 fig. 1의 이미지를 얻는다. 해당 이미지는 흑백 이미지에서 가장 어두운 부분과 가장 밝은 부분을 기준으로 잡은 후 나머지 픽셀들을 0과 1사이에 사상(mapping)한 결과이다.

2.1.2 이진화 (2진화, Binarization)

0과 1사이의 값으로 결정된 각 픽셀의 값을 특정 값(Threshold Value)과 비교해 0과 1 두 개의 값으로 변환하는 과정이다. 보다 뚜렷하게 활주로를 드러나게 함과 동시에 이미지를 더 간단하게 바꿈으로써 처리해야 할 정보의 양을 줄이는 효과를 얻을 수 있다. Fig. 1의 이미지를 이진화 하면 fig. 2의 이미지를 얻을 수 있다. 다만 2.2에서 언급하였듯이 어떠한 이미지라도 다양한 요인에 영향을 받으므로 Threshold Value의 값은 반드시 시간, 날짜, 카메라 모델 등의 함수이어야 한다.



Fig. 1 히스토그램 평활화 후 이미지



Fig. 2 이진화 후 이미지

2.1.3 가장자리 탐색(Edge Detection)

이미지에서 활주로를 탐색하기 위해서는 활주로의 특징을 규명해 경계조건을 지속적으로 추가해야 한다. 이진화한 이미지에서 단순히 0에서 1로 값이 변화하는 부분을 모으면 fig. 3의 이미지를 얻을 수 있으며 이를 이용해 최종적으로 활주로를 결정짓게 된다. 이는 MATLAB의 canny를 이용하였다.

2.2 호프 변환을 이용한 활주로 인식

호프 변환은 주어진 이미지에서 가장 선형 성분에 가까운 부분을 탐색할 수 있는 변환이다.

$$\rho = x \cos\theta + y \sin\theta \quad (1)$$

식 (1)은 주어진 이미지의 좌표에서 직선의 성분을 나타내는 식이다. 이미지의 임의의 점 x, y 를 식 (1)에 대입하면 fig. 4 과 같이Hough plane에서의 (ρ, θ) 관계를 얻을 수 있다. 같은 직선상의 점 1, 2, 3에 대해서는 ρ, θ 가 고정된 값이므로 Hough plane에서 얻은 세계의 관계식은 하나의 점에서 만난다. 주어진 이미지의 모든 Edge위의 점들을 Hough plane의 방정식으로 변환한 후

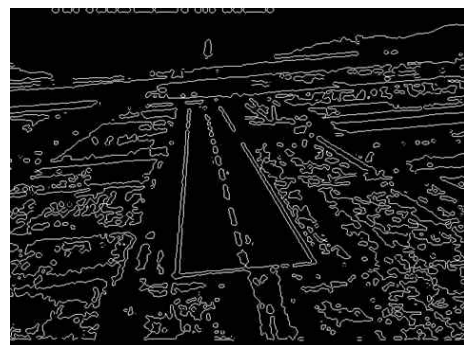


Fig. 3 가장자리가 탐색된 이미지

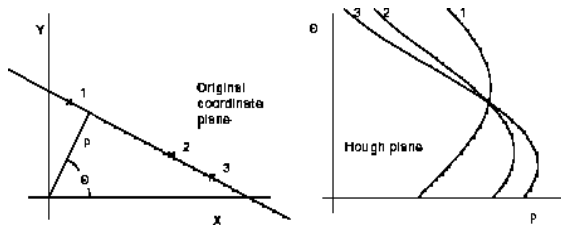


Fig. 4 Hough Transform

가장 밀도가 높은 부분을 탐색하는 방법으로 원래의 이미지에서 가장 선형 성분에 가까운 부분을 인식할 수 있다.

Fig. 3의 이미지에 위의 과정을 처리하면 fig. 5와 같은 이미지를 얻을 수 있다. 여기서 단순히 호프 변환을 시도하면 주의에 활주로 이외의 직선들이 인식될 수 있다. 활주로의 조건을 추가해 활주로가 아닌 부분을 배제시켰다. 또한 전 처리 과정으로 인해 활주로 영상의 정보를 잃어 한 직선이 아닌 직선위의 여러 선들로 분할되어 인식하게 되는 경우도 있다. 이 경우도 각 직선이 한 직선위에 있는지 판단한 뒤 끝점을 찾아내는 과정을 거쳐야 한다. 이러한 과정을 통해서 최종적으로 Fig.5와 같은 활주로를 탐색할 수 있다.

3. Simulation and Navigation

3.1 Simulation

2절에서 실제 영상으로 활주로 인식을 하였지만 서론에서 언급했듯이 실제 비행 영상을 구하기 어렵고 기존의 비행 영상을 통해 비행체의 자세 및 위치 정보를 정밀하게 계산이 불가능하기 때문에 비행체의 자세와 위치가 정해졌을 때 영상에서 활주로는 어디에 위치할 지 보여주는 시뮬레이션을 구현하였다.

무인기 입장에서 활주로는 어디에 위치하는지

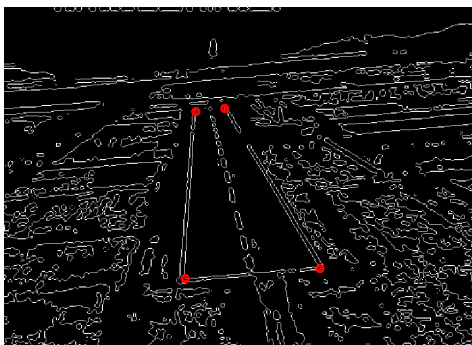


Fig. 5 선형 성분이 탐색된 이미지

를 계산하기 위해 Inertial frame, Vehicle frame, Body frame, 3가지 frame을 설정한다. Inertial frame에서 활주로의 네 점을 $(30,1000,0)$, $(-30,1000,0)$, $(30,0,0)$, $(-30,0,0)$ 으로 설정한다. 무인기의 위치와 자세를 $_{AV}$, Y_{UAV} , Z_{UAV} , θ , ψ 라고 정해지면 아래와 같은 계산을 통해 Inertial frame의 활주로 좌표를 Body frame의 점으로 변환시킬 수 있다.

$$\begin{pmatrix} X_{R,B} \\ Y_{R,B} \\ Z_{R,B} \end{pmatrix} = \begin{pmatrix} X_{UAV} \\ Y_{UAV} \\ Z_{UAV} \end{pmatrix} - \begin{pmatrix} X_{R,I} \\ Y_{R,I} \\ Z_{R,I} \end{pmatrix} \times \begin{pmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ s_\theta s_\psi c_\psi - c_\theta s_\psi & s_\theta s_\psi s_\psi + c_\theta c_\psi & s_\theta c_\psi \\ c_\theta s_\psi c_\psi + s_\theta s_\psi & c_\theta s_\psi s_\psi - s_\theta c_\psi & c_\theta c_\psi \end{pmatrix}^{-1} \quad (2)$$

식 (2)에서 아래 첨자 R,B는 Body frame의 활주로 좌표이며 R,I는 Inertial frame의 활주로 좌표이다. 또한 c는 cos을 뜻하며 s는 sin을 뜻한다.

$$X_{im} = \frac{X_{R,B}}{Z_{R,B}} \times f \quad (3)$$

$$Y_{im} = \frac{Y_{R,B}}{Z_{R,B}} \times f \quad (4)$$

식 (3,4)을 이용하면 3차원으로 표현된 활주로 좌표를 초점거리가 f인 2차원 평면상에 투영시킬 수 있다.

3.2 Navigation

항법 해를 계산하기 위해 먼저 3.1절에서 그린 활주로를 2절의 내용과 같이 호프 변환을 이용하여 네 꼭짓점을 인식한다.

먼저 상대자세를 구하는 수식을 전개한다. $[u_i, v_i]$ 는 활주로의 꼭짓점 픽셀 값이며 아래첨

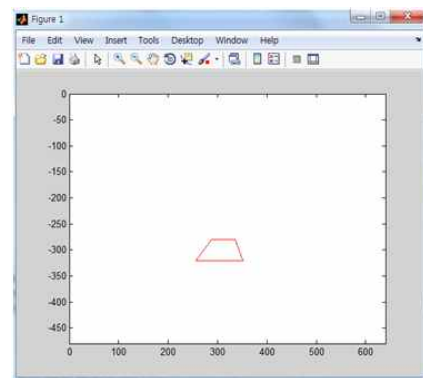


Fig. 6 시뮬레이션으로 그린 활주로

자 0은 활주로 앞쪽 중심의 픽셀 값이다. 여기서 projection perspective equation은 아래와 같다.

$$= f_x \frac{x_i}{z_i} - f_x \cot \theta \frac{y_i}{z_i} + u_0 \quad (5)$$

$$v_i = \frac{f_y}{\sin \theta} \frac{y_i}{z_i} + v_0 \quad (6)$$

여기서 f 는 초점거리이며 θ 는 skew각이다. z 는 -1로 가정하고 풀어도 문제가 없다. 그러므로 네 점을 이용해 x, y 를 계산할 수 있다. 여기 x, y 를 이용해서 광선벡터를 계산하면 아래와 같다.

$$ray_i = \frac{[x_i, y_i, z_i]}{x_i^2 + y_i^2 + z_i^2} \quad (7)$$

4개의 광선벡터들 3과 2, 1과 4를 벡터 곱한 뒤 그 값들을 벡터 곱하면 활주로의 x 축 방향을 알 수 있다. 같은 논리로 3과 1, 2와 4를 벡터 곱한 뒤 그 값들을 벡터 곱하면 y 축 방향을 구할 수 있다. z 방향은 구한 x 벡터와 y 벡터를 벡터 곱하여 구할 수 있다.

$$i = \frac{(v_i - v_0)}{f_y} \sin \theta \quad (8)$$

$$B_i = (y_{rot} - A_i z_{rot_i}) \quad (9)$$

식 (8,9)를 이용해서 A 와 B 를 계산한다. 활주로의 원점에서 각 꼭짓점을 향하는 벡터를 식 (5~7)을 통해 계산하였다. 그리고 아래첨자에 rot 를 붙였다. 이렇게 구한 A 와 B 를 아래 식에 대입하면 $x_{off}, y_{off}, z_{off}$ 를 구할 수 있다.

$$z_{off} = - \frac{B_2 - B_1}{A_2 - A_1} \quad (10)$$

$$y_{off} = A_i z_{off} - B_i \quad (11)$$

$$x_{off} = \cot \theta (y_{rot_i} + y_{off}) \quad (12)$$

$$- \frac{(u_0 - u_1)(z_{rot_i} + z_{off})}{f_x} - x_{rot_i}$$

아래첨자 off 가 붙은 벡터는 활주로의 원점에서 비행체의 중심까지 향하는 벡터이다.

4. 결 론

본 연구에서는 영상기반의 활주로 인식 및 항법 해 계산에 대한 연구를 하였다. 실제 비행영상을

이용하여 활주로 인식을 하였으며 활주로 인식을 쉽게 하기위해 여러 가지의 전 처리 과정을 거친 후 호프 변환을 이용하여 활주로를 인식하는 방법을 제시하였다. 또한 비행기 위치 및 자세에 따른 활주로의 위치 및 자세를 계산하였으며 이를 이용하여 실제 비행 영상을 대신하는 시뮬레이션 구현하였다. 마지막으로 시뮬레이션의 항법 해를 계산하고 실제 값과 비교하였다. GPS 데이터를 이용하지 않고 영상만으로 항법 해를 계산할 수 있었으며, 시뮬레이션 실험 오차가 적은 것을 보면 실제 비행영상에도 이용할 수 있을 것으로 기대된다.

5. 후 기

본 연구는 URP 프로그램 지원으로 수행되었으며 이에 감사의 뜻을 전합니다.

참고문헌

- (1) Richard Warren and Amber Fixcher, 2012, "Determining the position of runways from UAV video". 21st Century Systems, inc. 3375 Koapaka Street, Suite I-540 Honolulu, Hi 96819, pp.1~6
- (2) Ryan S.Holt, Randal W.Beard, 2009, "Vision-Based Road-Following Using Proportional Navigation", J Intell Robot Syst(2010) 57, pp. 193-216
- (3) D.Blake Barber, Joshua D.Redding, Timothy W.McLain, Randal W.Beard, Clark N. Taylor, 2006, "Vision-based Target Geo-location using a Fixed-wing Miniature Air Vehicle", J Intell Robot Syst(2006) 47, pp. 361~382