

## 論文

J. of The Korean Society for Aeronautical and Space Sciences 42(10), 833-841(2014)

DOI:<http://dx.doi.org/10.5139/JKSAS.2014.42.10.833>

## 풍선의 움직임 추적을 이용한 바람 속도 벡터 추정

오혜주, 조성범, 최기영\*

Estimation of Wind Velocity  
Using Motion Tracking of a Balloon

Hyeju Oh, Sungbeom Jo and Keeyoung Choi\*

Department of Aerospace Engineering, Inha University

## ABSTRACT

This paper proposes an algorithm to estimate the wind velocity by tracking free flying balloons. Balloons used in this method are expendable but inexpensive, which increases the usefulness of the method. Also we can obtain accurate 3D information by using multiple cameras and estimate the wind velocity of the local field. The proposed system consists of aerodynamic modeling of the balloon, a tracking algorithm using image processing, and the velocity estimation algorithm. We performed unit tests of each algorithm for the verification. The method is validated using a system simulation and sources of error case identified.

## 초 록

본 논문에서는 공중에 띄운 풍선을 카메라를 이용하여 촬영하고 이를 영상처리를 통하여 추적한 다음 그 움직임 정보를 이용해서 풍속을 추정하는 알고리즘을 제안한다. 여기서 사용되는 풍선은 아무런 장치가 장착되지 않은 것을 사용하므로 소모적이지만 저렴하며, 모든 지상 장비들은 반영구적으로 사용 가능하므로 효율성이 높다. 또한 다수의 카메라를 사용함으로써 정밀한 3차원 정보 획득이 가능하며, 일정 영역의 바람 속도장 추정이 가능하다. 본 논문에서 제안하는 시스템은 풍선의 공기 역학적 모델과 영상처리를 통한 추적 시스템 그리고 속도 추정 알고리즘으로 구성된다. 각각의 알고리즘을 단위 검증한 후에 설계한 검증 시스템과 가시화 프로그램을 통하여 통합 검증 및 시뮬레이션을 수행한다.

**Key Words** : Wind velocity estimation(풍속 추정), Balloon modeling(풍선 모델링), Image processing(영상처리), Camera geometry(카메라 기하학), Camera tracking(카메라 추적), Kalman Filter(칼만 필터)

## 1. 서 론

항공기 이착륙 시 바람의 방향과 세기를 알아야 하는 비행기 조종사는 바람의 상태를 항상 확

인해야 한다. 특히 무인기 등을 비행시험 하거나, 유인 항공기가 비상착륙하는 상황에서는 간이공 항 등에서 비행을 하게 되는데, 여기에는 정밀 기상 측정 장비가 없을 가능성이 크다. 이때 바

† Received: April 10, 2014 Accepted: September 29, 2014

\* Corresponding author, E-mail : [kchoi@inha.ac.kr](mailto:kchoi@inha.ac.kr)<http://journal.ksas.or.kr/>

pISSN 1225-1348 / eISSN 2287-6871

람의 방향과 세기를 확인할 수 있는 간편하면서 유용한 방법이 필요하다.

일반적으로 풍향이나 풍속을 측정하기 위해서 지상에 고정된 기둥에 풍속계를 고정시켜 이를 통해 지면 근처의 풍속을 측정한다. 이는 간단히 구성할 수 있다는 장점이 있지만 지면근처의 풍속만 잴 수 있다는 단점이 있다. 기상 항공기를 사용하는 방법은 비용이 많이 들고[1], 풍속계 및 기타 센서가 장착된 sonde를 부력에 의해 상승 시키며 대기의 상태를 측정해 무선으로 데이터를 전송받는 방법[2]은 측정 후 장비를 회수할 수 없어 많은 비용이 소모된다. 최근 풍속 측정에 주목 받고 있는 장비로 SODAR와 LIDAR가 있다. SODAR는 음파를 이용하고, LIDAR는 레이저를 이용하여 풍향과 풍속을 측정하는 원격 관측 장비이다. 그러나 SODAR는 소음과 오차율이 크고, LIDAR는 정확도가 높으나 고가라는 단점이 있다[3].

본 연구에서 제안하는 풍속 추정 방법은 Fig. 1에서와 같이 공중에 띄운 풍선의 움직임을 다수의 카메라를 이용하여 추적하고 영상처리한 정보를 이용하여 풍선의 위치와 속도를 예측한다. 이는 아무런 장치가 장착되지 않은 소모성 풍선을 사용하여 풍속 측정에 대한 비용을 절감하고, 모든 지상의 기구물을 반영구적으로 사용하므로 효율성이 높다. 또한 기존의 풍속계와 달리 다수의 카메라 정보를 사용함으로써 풍선의 3차원 운동을 통해 바람의 수평세기 뿐 아니라 정밀한 3차원 정보를 얻을 수 있다. 그리고 다수의 풍선을 동시에 운용하거나 또는 풍선을 띄우는 위치를 다양하게 함으로써 일정 영역의 바람의 속도장을 정확하게 추정할 수 있다. 본 연구에서 알고리즘 검증 및 시뮬레이션을 위해 헬륨 풍선 모델링을 수행하였고 이를 통하여 풍선을 공중에 띄우고 OpenGL을 사용하여 영상을 생성하였다. 이 OpenGL은 추적카메라 영상을 모사하며 이것의 이미지 트래킹을 통해 풍선의 움직임을 추적하는 알고리즘에 사용하였다.

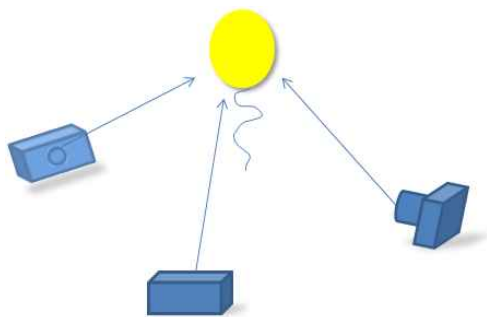


Fig. 1. Conceptual diagram

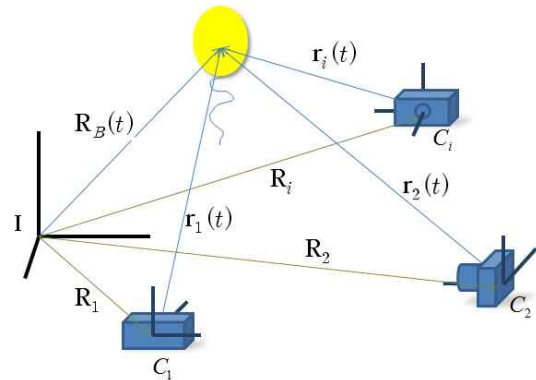


Fig. 2. Coordinate system

## II. 본 론

### 2.1 영상처리를 이용한 풍속추정 알고리즘

#### 2.1.1 풍속추정 알고리즘

본 연구에서는 영상처리 된 정보를 이용하여 풍선의 위치를 추정한다. 이렇게 추정된 풍선의 위치를 가지고 풍속을 추정하는 두 가지 방법을 제안한다. 첫 번째 방법은 영상처리 정보와 추정된 풍선의 위치로부터 풍속을 추정하였다. 두 번째 방법은 추정된 풍선의 위치에 칼만 필터를 사용하여 풍속을 추정하였다.

#### 2.1.1.1 영상처리 정보를 이용한 풍선 위치 추정 알고리즘

풍선 위치를 추정하기에 앞서 Fig. 2와 같이 좌표계를 설정한다.

$\vec{R}_B(t)$ 는 관성좌표계  $I$ 를 기준으로 풍선의 위치를 나타내는 벡터이다.  $\vec{R}_i$ 는  $i$ 번째 카메라의 위치를 나타내며, 이 값은 고정된다.  $\vec{r}_i(t)$ 는  $i$ 번째 카메라와 풍선간의 상대위치를 나타내며 이 값은 카메라를 통해 측정된다. 풍선의 위치를 관성좌표계를 기준으로 다시 정의하면 수식 1과 같다.  $i$ 번째 카메라 좌표계는 관성좌표계  $I$ 에서  $\vec{R}_i$ 로 위치이동을 하고  $T_{i/I}$ 만큼 회전을 한 국소 좌표계인  $C_i$ 로 정의된다. 여기서  $T_{i/I}$ 는 카메라 좌표계에서 관성좌표계로의 좌표변환 행렬이다.

$$\vec{R}_B^I = \vec{R}_i^I + T_{I/i}^i \vec{r}_i \quad (1)$$

여기서 카메라가 정지되어 있으면 풍선이 짧은 시간 내에 영상 밖을 벗어나게 된다. 따라서 높은 고도에서의 풍속 추정을 위해 카메라가 풍선을 추적하도록 해야 한다. 수식 2는 수식 1에

카메라 추적을 추가한 관계식이다.  $T_{i/t}$ 는 추적한 카메라 좌표계에서 추적 전의 좌표계의 좌표 변환 행렬이다.

$$\vec{R}_B^I = \vec{R}_i^I + T_{I/i} T_{i/t} \vec{r}_i^t \quad (2)$$

이를 다시 성분별로 쓰면, 수식 3과 같다.

$$\begin{aligned} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_B^I &= \begin{bmatrix} x \\ y \\ z \end{bmatrix}_1^I + T_{I/1} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_1^I = \begin{bmatrix} x \\ y \\ z \end{bmatrix}_2^I + T_{I/2} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_2^I \\ &= \dots = \begin{bmatrix} x \\ y \\ z \end{bmatrix}_i^I + T_{I/i} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_i^I \end{aligned} \quad (3)$$

Figure 3은 풍선과 카메라 영상의 관계를 나타낸 것이다. (A)는 3차원, (B)는 측면, (C)의 좌측은 풍선의 실제 위치이고 우측은 카메라 영상에서 보여지는 풍선의 위치이다.  $x, y, z$ 는 실제 풍선의 위치이고  $x_c, y_c$ 는 카메라 영상에서 보여지는 풍선의 위치이다.  $f$ 는 카메라의 초점거리이고  $R$ 은 카메라와 풍선 사이의 직선거리이다.

수식 4를 통해 카메라로부터 실제 풍선의 위치를 구할 수 있다.

$$\begin{aligned} x &= \frac{R}{f} x_c \\ y &= \frac{R}{f} y_c \\ z &= R \end{aligned} \quad (4)$$

여기서 카메라로부터 영상처리를 통해 얻게 되는 정보는 픽셀단위이다. 따라서  $x_c, y_c$ 를 수식 5와 같이 변환해야 한다. 여기서  $b$ 는 카메라의 픽셀수이다.

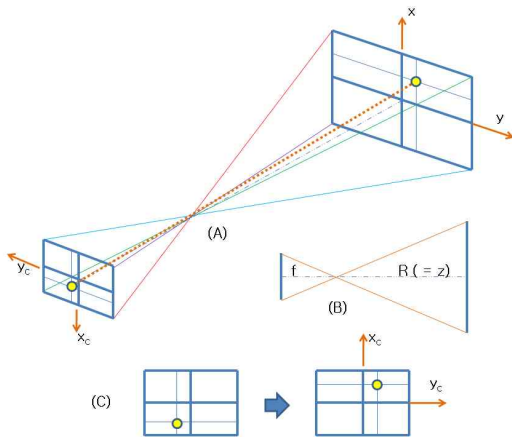


Fig. 3. Camera frame geometry

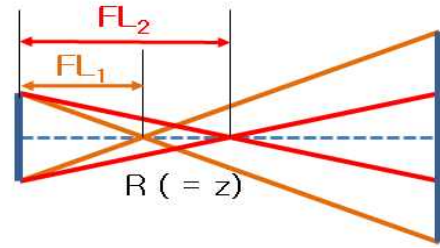


Fig. 4. Camera geometry for Zoom in/out

$$\begin{aligned} x_c &= \tan\left(\frac{FOV}{2}\right) \times \frac{x_c f}{b_x} \\ y_c &= \tan\left(\frac{FOV}{2}\right) \times \frac{y_c f}{b_y} \end{aligned} \quad (5)$$

여기서 시야각(FOV)는 Fig. 4와 같이 카메라 확대/축소에 의한 기하학적 관계에 의해 정해지며 관계식은 수식 6과 같다.

$$FOV = 2 \times \tan^{-1}\left(\frac{\text{Film Length}}{2 \times FL}\right) \times \frac{180}{\pi} \quad (6)$$

수식 2와 수식 3을 결합하여 수식 7을 얻을 수 있다.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_B^I = \begin{bmatrix} x \\ y \\ z \end{bmatrix}_i^I + T_{I/i} T_{i/t} \begin{bmatrix} \frac{R_i}{f} x_{t_i} \\ \frac{R_i}{f} y_{t_i} \\ R_i \end{bmatrix} \quad (7)$$

여기서 미지수는  $x_B^I, y_B^I, z_B^I, R_i$ 이며 조건식은 3개이므로 underdetermined system이다. 이는 시선벡터의 방향만으로 풍선의 위치를 결정할 수 없다는 것이다. 따라서 풍선의 크기를 이용하지 않는다면 카메라가 2대 이상 필요하다. 카메라를 추가할 경우 조건식은 3개가 늘어나고 미지수는 거리  $R$ 만 추가되므로 overdetermined system이 된다. 이것은 측정 잡음 등을 고려할 때 최소승방법 등을 사용하여 추정치의 정확도를 향상시킬 수 있음을 의미한다. 수식 8은 2대의 카메라를 사용하는 경우의 풍선의 위치 및 상대 거리 추정식이다.  $I_{n \times n}$ 은  $n$ 차원의 단위행렬이다.

$$\begin{bmatrix} I_{3 \times 3} & -T_{I/1} T_{1/t} \begin{bmatrix} x_{c1}/f \\ y_{c1}/f \\ 1 \end{bmatrix} & 0 \\ I_{3 \times 3} & 0 & -T_{I/2} T_{2/t} \begin{bmatrix} x_{c2}/f \\ y_{c2}/f \\ 1 \end{bmatrix} \end{bmatrix} \begin{bmatrix} x_B^I \\ y_B^I \\ z_B^I \\ R_1 \\ R_2 \end{bmatrix} = \begin{bmatrix} x_1^I \\ y_1^I \\ z_1^I \\ x_2^I \\ y_2^I \\ z_2^I \end{bmatrix} \quad (8)$$

### 2.1.1.2 영상처리 정보를 이용한 풍속추정 알고리즘

영상처리 정보만을 이용하여 풍속을 추정할 경우 수식 2로부터 수식 9를 유도할 수 있다.

$$\begin{aligned} \vec{R}_B^I = & \vec{R}_i^I + T_{I/i} \dot{T}_{i/t} \vec{r}_i^i \\ & + T_{I/i} \dot{T}_{i/t} \vec{r}_i^i + T_{I/i} T_{i/t} \dot{r}_i^i \end{aligned} \quad (9)$$

카메라 위치( $\vec{R}_i^I$ )가 고정되어 있고 수식 9에 수식 4를 적용하여 수식 10를 얻을 수 있다.

$$\begin{aligned} \vec{R}_B^I = & T_{I/i} \dot{T}_{i/t} \vec{r}_i^i + T_{I/i} T_{i/t} \dot{r}_i^i \\ = & T_{I/i} (-\Omega_{i/t}^j T_{i/t}) \begin{bmatrix} \dot{R}_i \\ f x_{i_i} x_{i_i} \\ \dot{R}_i \\ f y_{i_i} y_{i_i} \\ R_i \end{bmatrix} + T_{I/i} T_{i/t} \begin{bmatrix} \dot{R}_i \\ f x_{i_i} + \frac{R_i}{f} x_{i_i} \\ \dot{R}_i \\ f y_{i_i} + \frac{R_i}{f} y_{i_i} \\ \dot{R}_i \end{bmatrix} \end{aligned} \quad (10)$$

수식 10를 카메라 2대에 관한 component 형태로 쓰면 수식 11과 같다.

$$\begin{aligned} & \begin{bmatrix} I_{3 \times 3} & -T_{I/1} T_{1/t} \begin{bmatrix} x_{c_1}/f \\ y_{c_1}/f \\ 1 \end{bmatrix} & 0 \\ I_{3 \times 3} & 0 & -T_{I/2} T_{2/t} \begin{bmatrix} x_{c_2}/f \\ y_{c_2}/f \\ 1 \end{bmatrix} \end{bmatrix} \begin{bmatrix} \dot{x}_B^j \\ \dot{y}_B^j \\ z_B^j \\ \dot{R}_1 \\ \dot{R}_2 \end{bmatrix} \\ = & \begin{bmatrix} T_{I/1} T_{1/t} \begin{bmatrix} \dot{x}_{c_1} \\ y_{c_1} \\ 0 \\ \dot{x}_{c_2} \\ y_{c_2} \\ 0 \end{bmatrix} \frac{R_1}{f} \\ T_{I/2} T_{2/t} \begin{bmatrix} \dot{x}_{c_2} \\ y_{c_2} \\ 0 \end{bmatrix} \frac{R_2}{f} \end{bmatrix} + \begin{bmatrix} T_{I/1} (-\Omega_{1/t}) \begin{bmatrix} x_{c_1}/f \\ y_{c_1} \\ 1 \end{bmatrix} \\ T_{I/2} (-\Omega_{2/t}) \begin{bmatrix} x_{c_2}/f \\ y_{c_2} \\ 1 \end{bmatrix} \end{bmatrix} \end{aligned} \quad (11)$$

여기서  $\Omega_{i/t}^j$ 는 카메라 좌표계에서의 각속도 변환 행렬이다[4].

### 2.1.1.3 칼만 필터와 영상처리 정보를 이용한 풍속추정 알고리즘

칼만 필터(Kalman Filter)는 잡음이 포함된 선형 및 비선형의 상태를 추적하는 재귀 필터로서 과거, 현재, 미래의 상태를 예측하거나 추정하는 모델시스템에 활용되고 있다[5].

칼만 필터를 이용한 프로세스 추정 계산은 예측단계와 추정단계로 이루어진다. 예측단계에서는 직전 추정 값( $\hat{x}_{k-1}$ )과 오차 공분산( $P_{k-1}$ )을 입력으로 받아 예측 값( $\hat{x}_k, P_k$ )을 계산한다. 추

정단계에서는 예측단계의 예측값과 측정값( $z_k$ )으로 추정 값( $\hat{x}_k$ )과 오차 공분산( $P_k$ )을 계산한다.

본 연구에서는 영상처리 정보를 통해 풍선의 위치만을 얻을 수 있으므로 상태변수를 수식 12로 설정하고, 이에 대한 시스템 모델을 수식 13과 같이 설정하였다.

$$x = \begin{bmatrix} \text{위치} \\ \text{속도} \end{bmatrix} \quad (12)$$

$$\begin{aligned} x_{k+1} = & Ax_k + w_k \\ = & \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0 \\ 1000 \end{bmatrix} \end{aligned} \quad (13)$$

$$\begin{aligned} z_k = & Hx_k + v_k \\ = & [1 \ 0] x_k + 5 \end{aligned}$$

### 2.1.2 영상처리 알고리즘

풍속을 추정하기 위하여 풍선의 움직임을 카메라로부터 영상처리를 통해 추적해야 한다. 본 연구에서는 색 정보를 이용한 Centroid 기법을 적용하였다.

영상의 색상 정보를 이용한 검색 방법은 컬러 영상을 사용한 이후 영상처리에서 가장 쉽게 접근할 수 있는 방법이다. 추출하고자 하는 물체가 주변에 비해 구분이 뚜렷한 색상정보를 가지고 있을 경우 영상 내 조명 변화나 관측 위치, 크기 변화 등에 어느 정도 무관하게 적용 가능하다는 장점이 있다[6]. 따라서 풍선을 주변 배경(초록색, 파랑색)으로부터 구분이 쉬운 색(빨간색)으로 선정하였다. 그리고 획득된 영상의 RGB 색 정보를 기반으로 한 이진화를 수행하였다.

이진화 정보로부터 풍선의 위치를 검출하기 위해 Centroid 기법을 사용하였다. 획득된 이진화 정보(픽셀 좌표)를  $x$ 축과  $y$ 축 각각 평균값을 내어 이를 풍선의 중심위치로 결정하였다.

## 2.2 검증 및 시뮬레이션

### 2.2.1 시뮬레이션 시스템 구성

풍속 추정 알고리즘을 검증하기 위해 시뮬레이션 시스템을 구성한다. Fig. 5는 검증 및 시뮬레이션 시스템 구성도이다.

실제의 시스템을 시뮬레이션으로 구성하기 위해 풍선을 모델링하여 풍속에 의한 풍선의 위치를 구한 후 OpenGL을 사용하여 카메라 영상을 생성한다. Fig. 6이 검증을 위한 가시화 프로그램이다. 이와 같이 생성된 영상으로부터 영상처리 알고리즘과 풍속 추정 알고리즘을 통해 풍선의

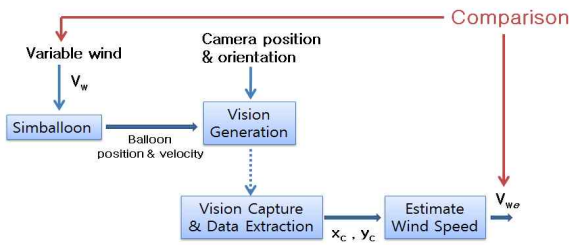


Fig. 5. Simulation and Verification System

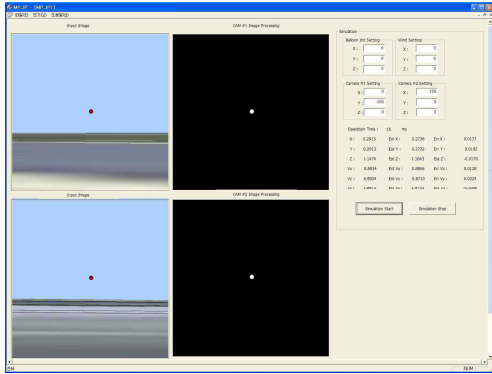


Fig. 6. Visualization Program

위치와 속도를 추정한다. 그리고 입력된 풍속과 추정된 풍속을 비교하여 알고리즘을 검증한다.

### 2.2.2 풍선 모델링

풍선의 움직임을 통해 풍속을 추정하므로 검증을 위해 풍선 모델링이 필요하다. Fig. 7은 관성좌표계에서의 풍선의 자유물체도이다.

여기서  $W$ 는 풍선의 자중,  $m_R$ 은 빈 풍선의 질량,  $\rho_G$ 는 헬륨가스의 밀도,  $\rho_A$ 는 공기의 밀도이다. 공기 1몰에 4g의 헬륨이 있으며 분자량은 28.8이다.  $V$ 는 풍선의 부피,  $g$ 는 중력가속도,  $R$ 은 풍선의 반지름이다.

$$\begin{aligned} \vec{W} &= -(m_R + \rho_G V)g\vec{z} \\ \rho_G &= \rho_A \times 4/28.8 \\ V &= \pi R^3 \end{aligned} \quad (14)$$

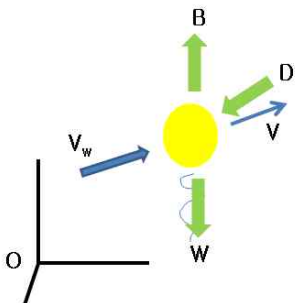


Fig. 7. Free Body Diagram of Balloon

$\vec{B}$ 는 풍선이 받는 부력이고 관계식은 수식 15이다.  $\vec{v}$ 는 풍선의 상대 속도이고  $\vec{v}_b$ 는 풍선의 속도,  $\vec{v}_w$ 는 바람의 속도이며 관계식은 수식 16이다.  $\vec{D}$ 는 풍선의 항력이고,  $C_D$ 는 항력계수이며 관계식은 수식 17이다.

$$\vec{B} = \rho_A V g \vec{z} \quad (15)$$

$$\vec{v} = \vec{v}_b - \vec{v}_w \quad (16)$$

$$\begin{aligned} \vec{D} &= -\frac{C_D}{2} \rho_A v \vec{v} A \\ A &= \pi R^2 \end{aligned} \quad (17)$$

풍선에 작용하는 힘의 수직성분을 이용하여 풍선의 수직종단속력  $v_T$ 를 수식 18과 같이 구한다.

$$v_T = \sqrt{\frac{2(\rho_A V g - m_B g)}{C_D \rho_A A}} \quad (18)$$

### 2.2.3 검증을 위한 영상획득 알고리즘

풍속 추정 알고리즘을 정확히 검증하기 위해서 카메라 영상 처리 정보를 가상으로 생성하였다. 수식 7로부터 카메라 영상 정보를 수식 19와 같이 얻을 수 있다.

$$\begin{bmatrix} \frac{R_i}{f} x_{t_i} \\ R_i \\ \frac{R_i}{f} y_{t_i} \\ R_i \end{bmatrix} = T_{i/I} T_{I/t} \left( \begin{bmatrix} x \\ y \\ z \end{bmatrix}_B - \begin{bmatrix} x \\ y \\ z \end{bmatrix}_i \right) \quad (19)$$

수식 10으로부터 카메라 영상 속도 정보를 수식 20과 같이 얻을 수 있다.

$$\begin{aligned} \frac{R_i}{f} \begin{bmatrix} \dot{x}_{t_i} \\ \dot{y}_{t_i} \\ 0 \end{bmatrix} &= T_{t/i} T_{i/I} \dot{R}_B^I \\ &- T_{t/i} (-\omega_{i/t}^i T_{i/t}) \begin{bmatrix} x_{t_i} \dot{x}_{t_i} / f \\ y_{t_i} \dot{y}_{t_i} / f \\ 1 \end{bmatrix} R_i - \begin{bmatrix} x_{t_i} / f \\ y_{t_i} / f \\ 1 \end{bmatrix} \dot{R}_i \end{aligned} \quad (20)$$

### 2.2.4 단위 테스트

#### 2.2.4.1 풍선 모델링 검증

Table 1은 풍선 모델링 검증을 위한 시뮬레이션 조건이고 Fig. 8은 결과이다. 풍선의 속도가 풍선의 수직 속도와 바람 속도의 합임을 확인하였다.

Table 1. Initial Condition of Balloon Simulation

$R$	$\rho_A$	$m_R$
12cm	1.21kg/m <sup>3</sup>	1.3g
$\vec{v}_{w_x}$	$\vec{v}_{w_y}$	$\vec{v}_{w_z}$
10	5	0

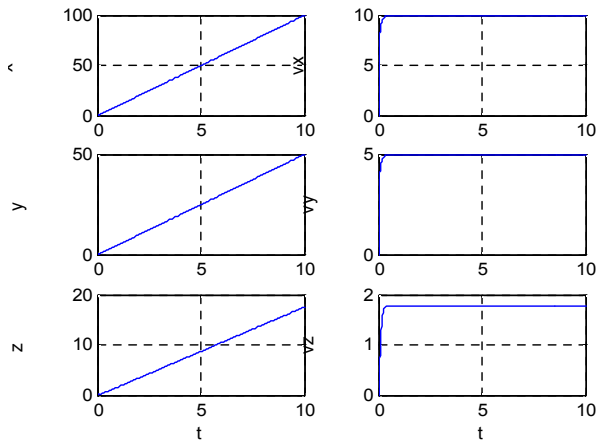


Fig. 8. Simulation Result of Balloon Model

2.2.4.2 영상처리 정보만을 이용한 풍속 추정 알고리즘 검증

앞서 기술한 풍선 모델링과 검증을 위한 영상 획득 알고리즘을 사용하여 영상처리 정보만을 이용한 풍속 추정 알고리즘을 검증하였다.

Table 2는 시뮬레이션 조건이며 Fig. 9는 수행 결과이다. 풍선 속도 에러가  $10^{-13} \sim 10^{-15}(m/s)$ 으로 풍선이 이동한 위치에 비해 매우 작으므로 본 풍속추정 알고리즘의 타당성을 확인할 수 있다.

Table 2. Initial Condition of Simulation for Estimation of Wind velocity

Simulation Time	100 sec
Period	0.04 sec
Camera Number	2
$f$	0.05 m
Camera Position	[50, -50, 0] [50, 50, 0]
Camera Orientation	[-90, -90, 45] [-90, -90, 135]
Wind Velocity	$[\sin(0.1t), 10 \times \sin(0.5t), 0.1] m/s$

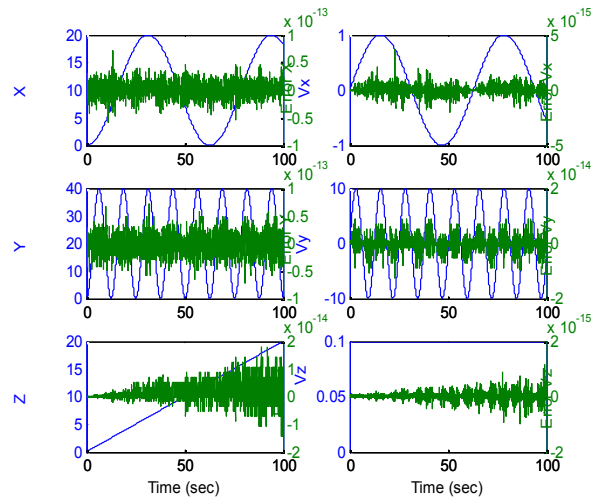


Fig. 9. Simulation Result of Estimation Wind Velocity Simulation using Image Processing

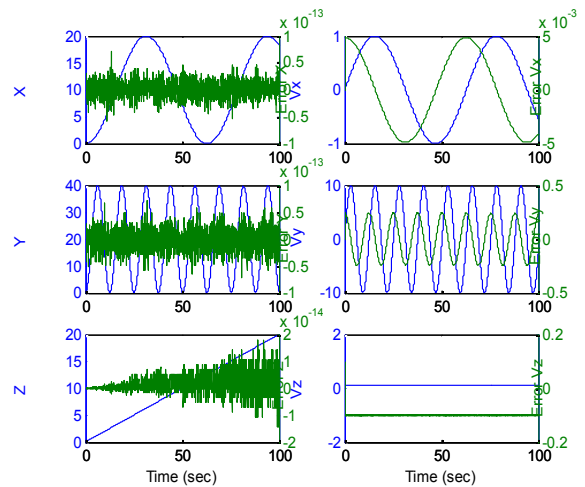


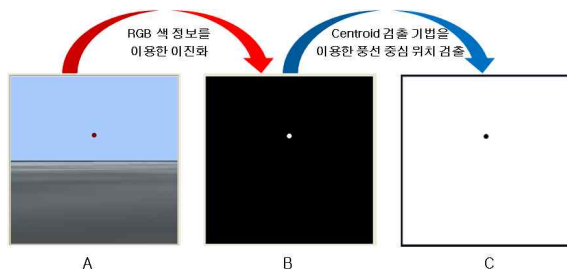
Fig. 10. Simulation Result of Estimation Wind velocity Simulation using Kalman Filter

2.2.4.3 칼만 필터와 영상처리 정보를 이용한 풍속 추정 알고리즘 검증

풍선 모델링과 검증을 위한 영상 획득 알고리즘을 사용하여 영상처리 정보를 얻었다. 그리고 이로부터 풍선의 위치를 구하고 칼만 필터를 사용하여 풍속 추정을 수행하였다. 시뮬레이션 조건은 동일하게 Table 2와 같이 설정하였고 결과는 Fig. 10과 같다. 위치정보만으로 속도를 추정하므로 영상처리 정보만을 이용하였을 때보다 오차가 증가하였다. 그러나 10m/s의 풍속이 불 때 0.25m/s의 오차가 발생하므로 오차율이 최대 2.5%로 작음을 알 수 있다.

**Table 3. Wind Velocity Estimation Algorithm Results**

	Maximum Error	Maximum Error rate
Using Image Processing	$10^{-13}(\text{m/s})$	$10^{-11}(\%)$
Using Image Processing and Kalman Filter	0.25(m/s)	2.5(%)



**Fig. 11. Image Processing Result**

**2.2.4.4 영상처리 알고리즘 구현**

RGB 색 정보를 이용한 Centroid 기법으로 영상처리를 수행하였다. Fig. 11의 A는 가상의 생성 영상이다. B는 영상 A로부터 색 정보를 이용하여 이진화하였다. C는 이진화 된 정보를 사용하여 풍선의 중심 위치를 검출하는 Centroid 기법을 적용한 결과이다. 이 기법은 색의 분리에 있어서 강인하며 연산 속도가 0.016초로 본 연구에서 제안하는 알고리즘의 sampling time인 0.04 초보다 작아 본 시스템에 적용 가능함을 확인하였다.

**2.2.5 통합 테스트**

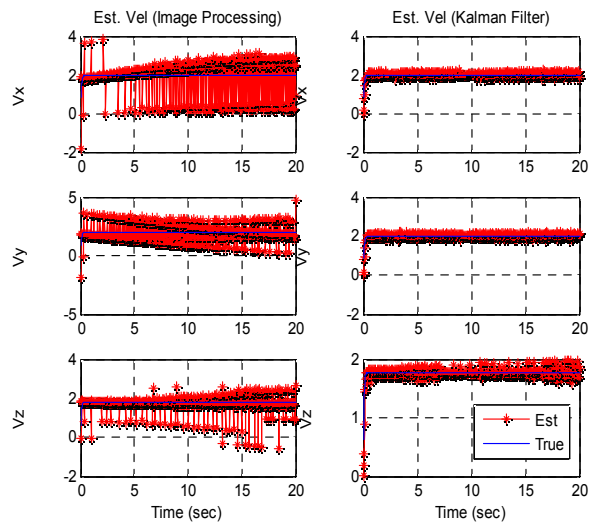
앞서 검증한 알고리즘을 모두 통합하여 테스트를 수행하였으며 정속바람일 경우와 변속바람일 경우를 테스트 하였다. 그리고 영상 확대/축소와 카메라 수 증가에 의한 영향을 확인하였다.

**2.2.5.1 정속바람**

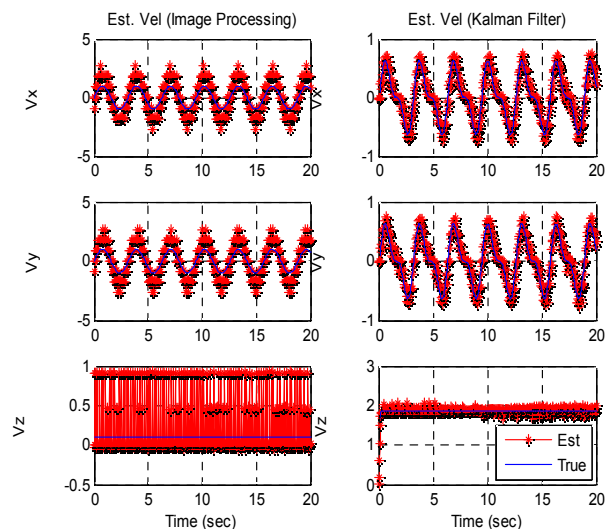
Figure 12는 Table 4와 같은 동일한 조건에서 알고리즘만 다르게 시뮬레이션을 수행한 결과이다. 좌측의 그래프는 영상처리 정보만을 사용하여 속도를 예측한 결과이고, 우측의 그래프는 칼만 필터와 영상처리 정보를 사용한 결과이다. 영상처리만 사용한 풍속추정은 영상처리 픽셀 오차로 인해 속도 추정 오차가 크게 발생하였다. 칼만 필터를 사용한 풍속추정은 오차가 상대적으로 적게 발생한 것을 확인할 수 있다.

**Table 4. Initial Condition(Steady Wind)**

Simulation Time (sec)	20 sec
Period (sec)	0.04 sec
Number of Cameras	2
Focal Length (m)	0.1374 m
Camera Position (m)	[0, -100, 0] [100, 0, 0]
Camera Orientation (deg)	[-90, -90, 0] [0, -90, 0]
Wind Velocity (m/s)	[2.0, 2.0, 0.0]



**Fig. 12. Simulation Result(Steady wind)**



**Fig. 13. Simulation Result(Variable wind)**



2.2.5.2 변속바람

Figure 13은 Table 4와 동일한 조건에서 풍속을 변속바람  $[\sin(2t), \sin(2t), 0.1]$ 로 주었을 때의 시뮬레이션 결과이다.

변속바람에서도 영상 픽셀 오차로 인해 영상 처리만 수행한 풍속추정 알고리즘은 속도 오차가 크게 발생하였고, 칼만 필터를 추가한 알고리즘은 속도를 잘 추정하는 것을 확인할 수 있다.

2.2.5.3 영상 확대 축소에 의한 영향

칼만 필터와 영상처리정보를 사용하여 풍속을 추정하는 알고리즘은 풍선의 위치정보를 기반으로 구해지고, 풍선의 위치정보는 영상처리 정보를 기반으로 얻어지므로 영상처리의 오차를 줄이는 것이 중요하다. 영상처리정보만을 사용하여 풍속을 추정하는 알고리즘은 풍선의 위치 뿐 아니라 풍속을 구할 때도 영상처리정보를 사용하므로 영상처리 오차를 줄이는 것이 더욱 중요하다. 따라서 한 픽셀당 의미하는 실제 위치 오차를 줄이기 위해 영상 확대에 대한 영향을 확인

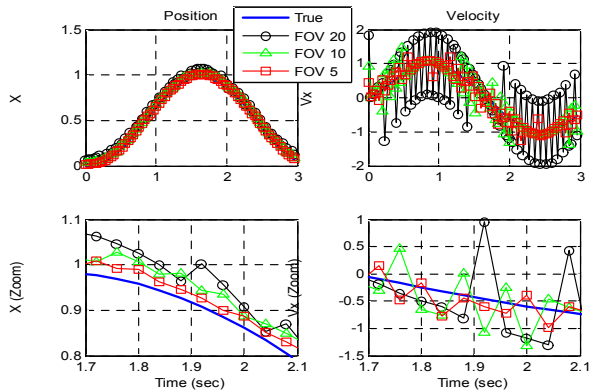


Fig. 14. Simulation Result w.r.t. Zoom in/out (Image Processing)

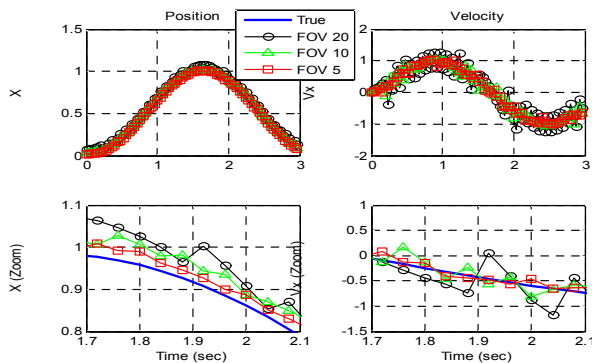


Fig. 15. Simulation Result w.r.t. Zoom in/out (Kalman Filter)

해 보았다.

Figure 14의 위의 그래프는 영상처리 정보만을 사용하여 풍속을 추정한 결과이고 아래는 위의 그래프를 확대한 것이다. Fig. 14의 좌측 그래프는 위치에 대한 정보이다. FOV가 20, 10, 5도로 점점 확대가 될수록 풍선의 위치 오차가 반으로 줄어들었음을 확인할 수 있다. 마찬가지로 우측 그래프는 풍속에 대한 정보이며 FOV가 작아져 영상이 확대될수록 오차가 반으로 줄어들었음을 확인할 수 있다.

Figure 15는 영상처리 정보와 칼만 필터를 사용하여 풍속을 추정한 결과 그래프이다. 마찬가지로 아래 그래프는 위의 그래프를 확대한 것이다. Fig. 15도 마찬가지로 FOV가 작아짐에 따라 풍선의 위치오차가 줄어들었음에 따라 위치 정보로 추정한 풍속의 오차도 줄어들었음을 확인할 수 있었다.

2.2.5.4 카메라 수에 의한 영향

본 추정 알고리즘은 카메라가 2대 이상 필요하다. 카메라 수를 증가시켜 오차에 대한 보정을 확인하였다. Fig. 16의 영상처리 정보만을 사용하여 풍속을 구한 알고리즘이고, Fig. 17은 영상처리 정보와 칼만 필터를 사용한 알고리즘의 결과이다. 두 그래프 모두 그래프는 카메라 수 증가에 의한 풍선 위치 정보이다. 우측 그래프는 카메라 수 증가에 의한 풍속 추정 그래프이다. 카메라 수 증가로 인해 오차의 폭이 감소하였으나 카메라 수 증가에 비해 오차감소 양이 적은 것을 확인할 수 있다.

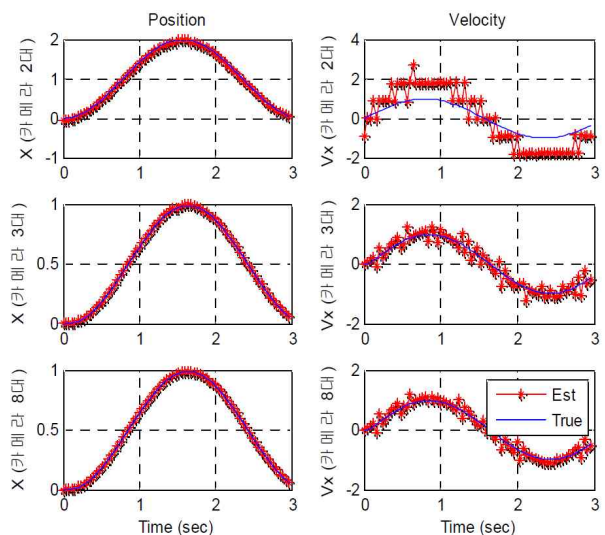


Fig. 16. Simulation Result w.r.t. number of camera (Image Processing)



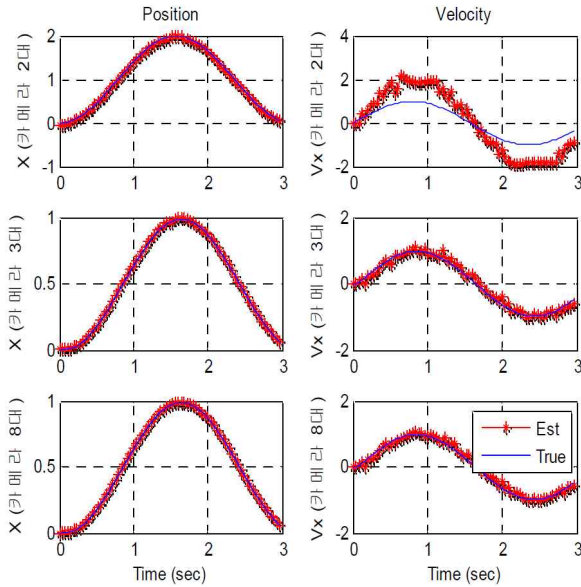


Fig. 17. Simulation Result w.r.t. number of camera (Kalman Filter)

### III. 결 론

본 논문에서는 풍선을 모델링하여 검증하고 검증된 풍선 모델을 공중에 띄우는 시뮬레이션을 하였다. OpenGL로 영상을 생성하고 생성된 영상을 영상처리를 통해 카메라 추적을 수행하였다. 그리고 영상처리 정보를 사용하여 풍선의 위치를 추정하고 이를 통하여 풍속을 추정하는 두 가지 알고리즘을 제안하였다. 첫 번째 방법은 영상처리 정보만을 통해 풍속을 추정하였고, 두 번째 방법은 영상처리 정보로 추정된 풍선의 위치로부터 칼만 필터를 사용하여 풍속을 추정하였다. 시뮬레이션 결과 영상처리정보만을 사용한 풍속추정은 변속바람에서 큰 오차를 가지고 추정을 수행하지 못하였으나, 칼만 필터를 사용한 풍속추정은 정속바람이나 변속바람 모두 수용 가능한 오차를 가지고 풍속을 추종하였다. 두 가지의 알고리즘을 단위테스트 하였을 때는 Table 3에서 보는 바와 같이 영상처리정보만을 사용한 알고리즘의 오차가 훨씬 적었으나, 영상처리를 포함한 통합 시뮬레이션에서는 오히려 칼만 필터의 풍속 추정 알고리즘이 다양한 환경에서 더 강인함을 확인할 수 있었다. 카메라의 FOV를 변경하여 시뮬레이션을 수행한 결과 FOV가 작아질수록(영상

확대) 풍속 추정 결과가 좋았으며, 카메라 수를 증가하여 시뮬레이션을 수행한 결과 카메라 수를 크게 증가하여야만 결과가 나아짐을 확인하였다.

본 논문에서는 제안한 풍속추정 알고리즘 및 시스템을 통해 풍속 추정이 가능하다는 것을 확인하였다. 향후 제안한 알고리즘을 시뮬레이션이 아닌 실제 환경에서 시험하기 위해 카메라 비동기화 등의 발생 가능한 문제를 고려한 시뮬레이션을 수행하고, 검증된 다양한 바람모델을 넣어 시뮬레이션을 수행해야한다.

### 후 기

본 연구는 인하대학교의 교내학술연구비 지원을 받아 진행되었으며, 이에 감사드립니다.

### References

- 1) Park, N. R., Cheong, S. J., and Yoon, H. R., "Wind Direction, Wind Speed and Baro-GPS Height calculation using GPS and ADC," *KSAS Journal*, Dec, 1996, pp132-138.
- 2) Kim, Y. R., Lee, S. C., Ko, S. H., and Kang, J. Y., "Development of the Program for the Unmanned Aerial Vehicle(UAV) Flight Test Planning through Atmospheric Considerations," *KSAS Spring Conference*, Apr. 2011, pp.797~800.
- 3) Ji, Y. M., Kim, H. G., Chung, C. H., Han, K. S., and Park, H. C., "Wind speed measurement using SODAR and LIDAR," *KSNRE Fall Conference*, Nov. 2009, pp.453~455.
- 4) Oh, H. J., "Estimation of Wind Speed Using Motion Tracking of a Balloon," Master's Thesis, In-ha University.
- 5) G. Welch and G. Bishop, "An Introduction to the Kalman Filter," UNC\_Chapel Hill TR 95-041, July 2006.
- 6) Bang, S. B., Nam, J. Y., and Choi, J. G., "A Centrid-based Image Retrieval Scheme Using Centroid Situation Vector," *KOSBE Journal*, Vol. 7, Jun. 2002, pp.126-135.