

물체추적을 위한 FPGA 구현에 대한 연구

양 찬 우, 김 동 훈, 신 윤 수, 고 광 철
한양대학교 전자통신전파공학과
전화 : 02-2290-0348 / 핸드폰 : 016-384-7779

A Study about Implementation of Object Tracking on FPGA

Chan Woo Yang, Donghun Kim, Yun Soo Shin, Kwang Cheol Ko,
Div. of Electrical & Computer Engineering, Hanyang University
E-mail : ychanwoo@ahpe.hanyang.ac.kr

Abstract

This study describes an implementation of object tracking algorithm on FPGA. The global system detect the zone there is more motion in, attending to the generated optical flow, and centers its attention to it to improve the details
In this case, To obtain image in Camera, Image aquisition board make use of SAA7113 Video Input processor and algorithm is applied to motion estimation and difference picture. Also, This work can be applied kalman filter to reliability of tracking.

I. 서론

영상처리 분야에서 가장 흥미로운 분야 중 하나는 영상 내 존재하는 물체의 움직임을 감지하고 이것들의 움직임을 실시간으로 추적하는 시스템이다. 이러한 시스템은 특정장소에서 인간의 역할을 대신하기도 하고 또는 화자의 움직임을 판단해 이를 추적하므로 대상물의 안정된 형상을 취득해 정보를 전달해 주기도 한다. 하지만 많은 계산량과 주변 장치와의 인터페이스 문제, 또 시스템 성능 확장 시 확장비용의 문제로 인해 아직까지 시스템의 효율성이 충분히 보장 되지 못하고 있다.

따라서 본 논문에서는 기존의 소프트웨어 기반으로 구현된 영상인식 및 물체추적 알고리즘을 FPGA에 구현하므로 추적알고리즘을 모듈화해 시스템 확장 시 또는 알고리즘 수정 시 적용할 수 있도록 제안한다.

한편 일반적으로 영상의 추적방법은 카메라 움직임 계산, 특징 추출 및 추적, 움직임 검지 및 필터링, Pan/Tilt 속도제어 등으로 구성되는데, 이 중에서도 특히 고려되어야 할 사항이 추적에 있어서는 카메라 움직임과의 실시간 동기화를 맞추는 것이다. 그 후 한번의 감지와 추적에 요구되는 시간은 이전프레임과 현재프레임을 비교하여 카메라에 명령을 보내는 시간과 카메라가 그 명령을 받아 수행하고 응답하는 시간이 가능한 짧을 수록 신뢰 할 수 있는 추적을 할 수 있게 된다. 그러므로 FPGA 상에서 추적알고리즘 구현 시 카메라와의 동기를 맞추는 것도 고려하도록 한다.

II. 시스템 구성

2.1 전체 시스템의 개요

Video 영상에서 움직이는 object를 검출하는 것은 이미지의 쌍을 정렬시키는 것과 difference image를 생성하는 것으로 이루어진다. 움직이는 물체는 clustering과 같은 알고리즘을 사용하는 difference image에서 발

견할 수 있는데, 불행하게도 이미지의 배열은 모든 화면상의 점에 대해 일반적으로 정확하지 않아서(즉 difference image 가 noisy해서), 혼란을 야기하게 된다. 또 이미지 정렬과 subtraction이 가지는 문제점으로는 실행시간이다.(모든 이미지 point에 대해 동시에 registration, alignment, subtraction, clustering을 시험하는 실행 시간.) 그러므로 Real time system이 필요하다면 특별한 하드웨어가 사용되어야 한다. 본 논문에서는 카메라로부터 영상을 획득하고 그 영상을 추적하는 알고리즘을 FPGA에 구현하는 것을 목표로 하였다. 그래서 기존의 영상을 획득하는 방법과 비슷하게 카메라로부터 입력 받아 그 영상을 메모리에 저장하고 저장된 영상에 특정 object를 FPGA 상에 구현된 추적 알고리즘을 이용해 추적하게 된다. 물체를 추적하는 알고리즘은 VHDL을 이용해 구현하고 ALTERA의 MAX+PLUSII로 컴파일 해 FPGA에 다운로드 하게 된다. 아래 그림 1은 전체 시스템의 개요도를 나타낸다.

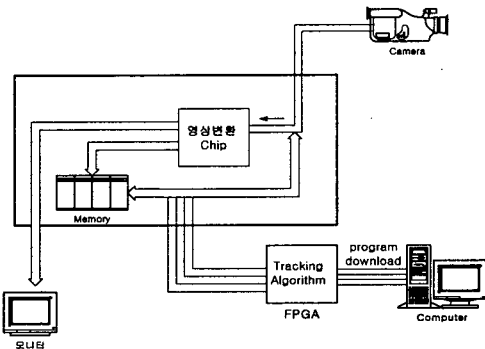


그림1. 시스템의 개요도

2.2 Image Aquisition Board와 FPGA

FPGA상에 추적 알고리즘을 구현하기 전에 성능을 테스트하기 위해 카메라로부터 들어온 영상을 처리해주는 Image Aquisition Board를 구현하고 이를 테스트하기 위해 카메라로부터의 영상의 입력을 받아 처리하는 비디오 입력 프로세서 칩을 이용하여 그림2와 같은 보드를 구성 하였다.

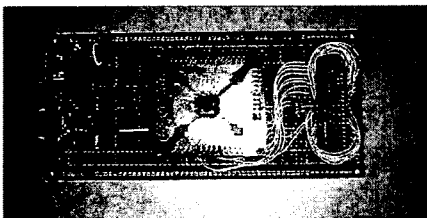


그림2. 영상처리보드

이 보드는 SAA7113칩을 이용해 카메라로부터 영상을 획득해 그 영상을 A/D 변환해서 YUV신호로 영상을 출력하게 된다. 한편 UP-1X보드의 FPGA에서는 출력 신호와 수직동기, 수평동기, 그리고 내부 클럭을 사용하여 카메라와의 동작을 제어한다.

그림3은 영상처리보드와 FLEX10K70RC240-4 라는 FPGA를 탑재한 ALTERA의 UP1X보드와의 연결 모습이다. FPGA는 LOOK-UP TABLE(LUT)을 사용하여 디지털 논리 회로를 구현하는 FLEX를 기용하며, 이 디바이스 계열은 SRAM을 Reconfigurable Element로써 사용하기 때문에 프로그래밍 할 수 있는 횟수가 제한되지 않는 장점이 있으나 프로그램의 결과를 전원이 OFF되었을 경우에는 유지할 수 없다는 단점을 가지고 있다.

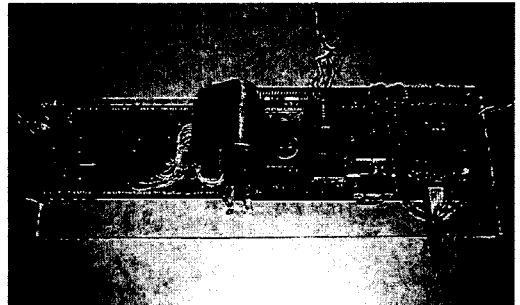


그림3. FPGA와의 연결

2.3 FLEX 10k 디바이스 구조

FLEX 10k 디바이스 구조는 아래 그림과 같이 Logic Array, FastTrack Interconnect, I/O핀, Embedded Array등으로 구성 되어 있다. Logic Array는 일반적인 논리 회로를 구성하는데 사용되며 Embedded Array는 메모리와 특별한 논리 회로를 구성하는데 사용된다.

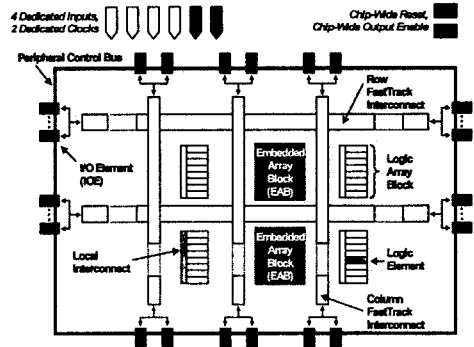


그림4. FLEX 10K 디바이스 구조

III. 움직임 추정과 검출

1. 움직임 추정

영상에서는 시간적으로 이웃한 화면들을 이용하여 예측부호화의 성능을 높이는 방법이 있는데 현재 화면의 화소의 예측치를 이전화면으로부터 구해서 예측오차를 부호화하는 화면 간 예측 부호화 방법과 예측치를 구하기 위해 화면간의 움직임을 구하는 움직임 추정, 그리고 그 추정된 움직임으로부터 예측치를 구해 예측 오차를 구하는 과정 등이 있다.

그리고 현재 화소가 이전화면의 어느 위치에 대응되는지 구할 수 있다면, 좋은 예측치를 구할 수 있는데 이에 따라 움직임 보상의 표현은 아래와 같이 나타낼 수 있다. 시각을 t , 위치 (x, y) 에서 밝기를 $X(x, y, t)$ 라 하고, 이전 화면의 재생영상을 $\hat{X}(x, y, t - \Delta t)$ 라 하면, $X(x, y, t)$ 의 예측치 $\tilde{X}(x, y, t)$ 는

$$\tilde{X}(x, y, t) = \hat{X}(x - \Delta x, y - \Delta y, t - \Delta t) \quad (1)$$

이고, 추정된 움직임은 $\Delta x, \Delta y$ 이고, 움직임 벡터는 $D = (\Delta x, \Delta y)^T$ 로 표현 된다.

2. 움직임 검출

2.1 Difference Pictures

두 프레임 사이에 움직임이나 화면의 변화를 알아내는 가장 쉽고도 널리 쓰이는 방법은 두 프레임에서 동일한 화소를 비교하는 것이다.

Difference Picture는 임의의 문턱치에 대하여 참조 영상과 비교 영상을 비교하여 이진화 된 영상을 만드는 과정이다. 따라서 Difference Picture는 두 개의 연속된 프레임에 대하여 변화된 영상을 구하는 방법이다. 프레임 $F(x, y, j)$ 와 $F(x, y, k)$ 간의 Difference Picture $DP_{jk}(x, y)$ 는 다음과 같이 얻어진다.

$$DP_{jk}(x, y) = \begin{cases} 1 & \text{if } |F(x, y, j) - F(x, y, k)| > \tau \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

여기서 τ 는 문턱치 값이다. 매우 느린 속도로 움직이는 물체의 추적이나, 천천히 변하는 조도의 차이를 감지하기 위해서는 τ 의 값을 적절히 결정하는 것이 중요하다.

2.2 Likelihood Ratio

Difference Picture를 이용한 화면의 변화감지의 신뢰

도를 높이기 위해서는 단순한 화소간의 비교뿐만 아니라, 화소 주위의 지역적인 명도의 강도 분포에 대한 고려가 필요하다. likelihood ratio를 적용하기 위해서는 마스크를 설정할 필요가 있는데, 여기에는 superideal을 이용하여 임의의 $m \times n$ 영역으로 구분하는 방법과 local mask를 이용하여 화소와 화소 주위의 8 이웃 화소들을 포함한 mask를 만드는 방법이 있다. 본 연구에서는 후자의 방법을 사용하였다.

Difference Pictures는 likelihood ratio 기법을 적용하면 다시 다음과 같이 정의된다.

$$DP_{jk}(x, y) = \begin{cases} 1 & \text{if } \lambda > \tau \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$$\lambda = \frac{\left[\frac{\sigma_1 + \sigma_2}{2} + \left(\frac{\mu_1 - \mu_2}{2} \right)^2 \right]^2}{\sigma_1 * \sigma_2} \quad (4)$$

여기서 μ 와 σ 는 평균 명암도와 편차를 나타낸다.

2.3 목표 물체의 선택

앞의 이론들은 화면의 명암도의 변화에만 관련이 있으므로 물체의 크기나 색상에 좌우되지만, 물체의 속도에 관련된 요소는 없다. 속도 벡터를 목표 물체 선택에 이용하면 물체의 이동도를 목표물 선택에 적용할 수 있다.

프레임 $F(x, y, j)$ 안의 $P_1(x_1, y_1)$ 에 있는 화소가 프레임 $F(x, y, k)$ 의 $P_2(x_2, y_2)$ 로 움직였을 때의 속도 벡터는 다음과 같이 구해진다.

$$\begin{aligned} \frac{x_2 - x_1}{k - j} &= v_x \\ \frac{y_2 - y_1}{k - j} &= v_y \end{aligned} \quad (5)$$

따라서 목표물의 결정은 Difference Pictures에서 size filter를 적용한 화소군들의 크기와 속도 벡터를 곱한 값에 의해 결정된다. 따라서 크기와 속도 벡터의 절대값의 곱이 가장 큰 화소군의 중심점을 움직임이 일어난 추적해야 할 목표물로 정의한다.

IV 움직임 추적

Tracking에서 중요한 부분은 이미지 전체의 정보에서 무엇을 추적하고 물체의 어떤 특징에 일치 할 것인지를 결정하는 것이다. 즉, 측정하는 sequence로부터 물체의 전 위치에 대한 것을 어떻게 추론할 것인가 하는 것이다.

1. 칼만 필터

칼만 필터는 시스템잡음과 측정 잡음이 존재하는 경우에 상태추정오차의 기대 값이 최소가 되도록 상태변수를 추정하므로, 잡음이 존재하는 환경에서도 시스템 상태를 추정할 수 있게 된다. 다시 말해 각 상태의 Update estimate가 새로 들어온 입력이나, 앞선 추정으로부터 계산 되어 진다는 의미이다. 그림5는 칼만 필터의 전체 구조를 나타낸 것이다.

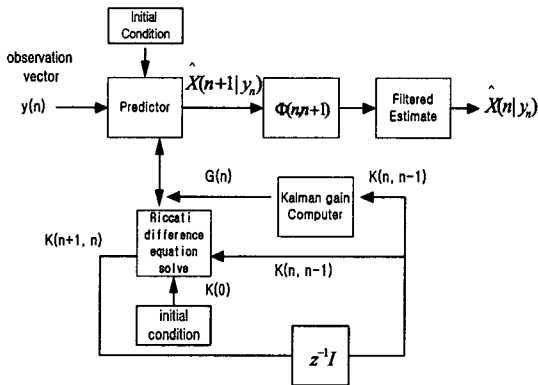


그림5. 칼만 필터의 전체 구조도

매 시간 반복적으로 dynamic system의 상태를 추정하는 칼만 필터는 (LMMSE: linear minimum mean square error sense)에서 이러한 상태 변수와 선형적으로 연관된 Scalar Observation 또는 벡터의 time series로 정해진다. 만약 상태 변수와 noise가 연관성이 없거나 가우시안 병렬 처리로 이루어진다면, 칼만 필터는 제곱의 평균 에러 추정량(mean square error estimator)가 최소가 된다. 다음은 칼만 필터의 방정식을 나타낸 것이다.

* State-Prediction equation

$$\hat{z}_b(k) = \Phi(k, k-1)\hat{z}_a(k-1) \quad (6)$$

*Covariance-prediction equation

$$P_b(k) = \Phi(k, k-1)P_b(k-1)\Phi(k, k-1)^T + Q(k) \quad (7)$$

* Kalman-gain equation

$$K(k) = P_b(k)H^T(k)(H(k)P_b(k)H^T(k) + R(k))^{-1} \quad (8)$$

* State-update equation

$$\hat{z}_a(k) = \hat{z}_b(k) + H(k)[y(k) - H(k)\hat{z}_b(k)] \quad (9)$$

* Covariance-update equation

$$P_a(k) = P_b(k) - K(k)H(k)P_b(k) \quad (10)$$

V. 결론

본 논문은 Difference Picture, 속도 벡터를 이용하여 물체의 움직임을 검출하고 여기에 잡음이 존재하는 환경에서도 상태 추정에 좋은 성능을 발휘하는 칼만필터를 적용해 물체추적을 실행하는 알고리즘을 FPGA상에 제안하였다. FPGA는 FLEX10K70을 사용하고 영상 획득하여 저장하는 보드는 SAA7113 영상 입력 처리기와 A/D 컨버터, 메모리를 이용하여 구현하였다. 아직은 추적 알고리즘을 하드웨어로 모듈화해서 영상 추적 장치에 응용하는 연구가 지속적으로 요구된다.

Reference

[1] A.E. Nelson "Implementation of Image Processing Algorithms on FPGA Hardware", 2000
 [2] A. Murat Tekalp, "Digital Video Process", Prentice Hall PTR, 1995
 [3] Ramesh Jain, "Machine Vision", McGraw Hill, 1995
 [4] Jie Chen, Ut-va Koc, K.J.Ray Liu, "Design of Digital Video Coding Systems", Marcel Dekker, Inc, 2002
 [5] B. Draper, W. Najjar, W. B ohm, J. Hammes, R. Rinker, C. Ross, M. Chawathe, and J. Bins. "Compiling and optimizing image processing algorithms for FPGA's", In IEEE Workshop on Computer Architectures for Machine Perception (CAMP), Oct. 2000.
 [6] S.K. Jung, K.Y. Wohn, "3-D tracking and motion estimation using hierarchical Kalman filter", IEE Proc.-Vis. Image Signal Process, Vol. 144, No.5, October 1997