

영상코드화 기법을 이용한 PIV 알고리즘에 대한 연구

김 성 균*
(1999년 2월 1일 접수)

A Research on the PIV Algorithm Using Image Coding

Sung Kyun Kim

Key Words: Particle Image Velocimetry(영상유속계), Image Coding(영상코드화), Synthetic Image(가상영상), New Three Step Search(새삼단계추적법), Projection Image Matching(정사영상기법)

Abstract

A Particle Image Velocimetry(PIV) algorithm is developed to analyze whole flow field both qualitatively and quantitatively. The practical use of PIV requires the use of fast, reliable, computer-based methods for tracking numerous particles suspended in a flow field. The TSS, NTSS, FFT-Hybrid, which are developed in the area of image compression and coding, are introduced to develop fast vector search algorithm. The numerical solution of the lid-driven cavity flow by the ADI algorithm with the Wachspress Formula is introduced to produce synthetic data for the validation of the tracking algorithms. The algorithms are applied to image data of real flow experiments. The comparisons in CPU time and mean error show, with a small loss of accuracy, CPU time for tracking is reduced considerably.

1. 서 론

비정상 유동을 포함한 전유동장 해석기법으로서의 PIV의 활용이 최근 증가하고 있으며, 상용화된 소프트웨어의 등장 및 관련 하드웨어시스템의 발전에 힘입어, PIV의 정밀도와 정확도에 있어서 레이저유속계(LDV)의 수준에 접근하고 있다. PIV 유동해석기법은 중립미세입자를 유동장에 첨가한 후, 실린더 렌즈로 편광처리된 레이저면으로 조명된 연속 영상들을 CCD카메라와 프레임그래버를 통하여 디지털화한 영상데이터들을 얻은 다음, 진처리, 패턴매칭에 의한 속도벡터추적, 오류벡터검증, 가시화 등 후처리 단계의 알고리즘들을 거쳐 완성된다. 이중 벡터추적과정은 많은 계산량을 요하며, 전체 전산처리시간의 대부분을 점하고 있다.

고화질 TV, 위성회의 등의 통신분야에서도 대용량의 데이터를 초당 30프레임 이상 전송하기 위해 차영상(difference image)을 이용한 영상데이터 압축기법으로 PIV에 있어서 벡터추적과 동일한 방법을 사용하고 있다. PIV 분야에서는 상호상관계수를 FFT를 이용해서 계산하는 정도의 연산회수의 감축 노력이 있어 왔으나, 영상코드화 및 압축분야에서는 많은 기법들이 개발되어 사용되고 있다(MPEG, MPEG II, H. 261 등). 이 기법들은 대부분 정확도보다 처리속도, 하드웨어 구현에 더 큰 비중을 두고 있으며, 최대 이동 픽셀 수를 7로 제한한 것들로서 PIV해석에 적용되기 위해서는 검증과 개선이 필요하다.

본 논문에서는 삼단계추적법(Three Step Search: TSS),⁽¹⁾ 새삼단계추적법(New Three Step Search: NTSS),⁽²⁾ 정사영상기법(Hierarchical Projection Matching: HPM)⁽³⁾ 등을 PIV 해석에 도

* 회원, 건국대학교 기계공학과
E-mail : skkim@kkucc.konkuk.ac.kr
TEL : (02)450-3472 FAX : (02)3436-0540

입하여 계산시간과 정확도를 비교함으로써 도입 가능성 여부를 타진하고자 한다.

위 방법들을 도입한 PIV해석 결과를 기존의 직접 상호상관계수 계산에 의한 전영역추적법(full search)과 상호상관계수 계산에 있어 FFT를 이용하여 정수픽셀이동거리를 구한 후 서브픽셀 계산은 직접계산하는 방법(HYB)과 비교하였다. 속도벡터 추적과정을 제외한 모든 과정은 동일한 알고리즘을 이용하였으며, 전처리과정을 생략하였고, 가우스 3점 서브픽셀보간,⁽⁴⁾ 9점국소 중간치 필터⁽⁵⁾를 이용한 검정, 9점저대역필터에 의한 속도벡터의 조성 등 단순화된 해석을 가상영상데이터(synthetic image data)에 적용하여 CPU와 평균오차를 구하여 요약하고 분석하였다. 노이즈가 없는 영상데이터의 경우, TSS, NTSS는 정확도의 큰 손상없이 계산량을 크게 감소시킬 수 있었으며, 노이즈가 포함된 경우, HPM은 정확도가 떨어지는 대신 빠른 추적이 가능하며, HYB는 정확도와 계산시간에 있어서 효율적이었다. HPM은 10 픽셀 이상 움직일 경우에는 현재로는 많은 개선이 필요하다.

앞으로의 PIV 기법은 실시간 유동 계측을 위한 빠른 알고리즘과 난류유동 등을 위한 고정밀도 알고리즘의 개발로 양극화될 전망이며, 본 논문에서는 전처리과정의 생략, 훌쩍 영상분리, 빠른 벡터 추적, 국소중간필터(local median filter, LMF)에 의한 오류벡터의 검정과 추정의 온라인화를 통하여 실시간 측정으로의 접근을 모색하였다.

2. 가상영상 데이터

다양한 이동 거리와 속도구배를 갖는 영상을 구하기 위해, Re수 1000의 사각공동유동(driven cavity flow)의 수치해를 이용하여 가상영상데이터를 생성하였다. 81×81 균일 격자점에 대해 Wachspress 공식을 이용한 ADI 알고리즘⁽⁶⁾을 이용하여 수치해석해를 구한 후, 임의의 위치에 다음의 가우스 밝기 분포를 갖는 입자들을 생성하였다.

$$I(i, j) = I_0 \exp \left[-\frac{(r - r_0)^2}{2\sigma^2} \right] \quad (1)$$

여기서 I_0 는 255, σ 는 0.5와 1, r_0 는 입자중심의 화상좌표이며, 입자의 직경은 3픽셀, 600×600 영상데이터에 7000개의 입자를 생성하였다.

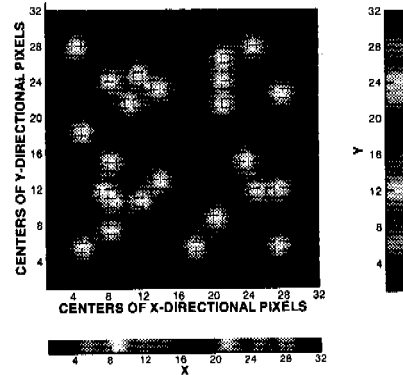


Fig. 1 Synthetic data and its projections of horizontal and vertical direction

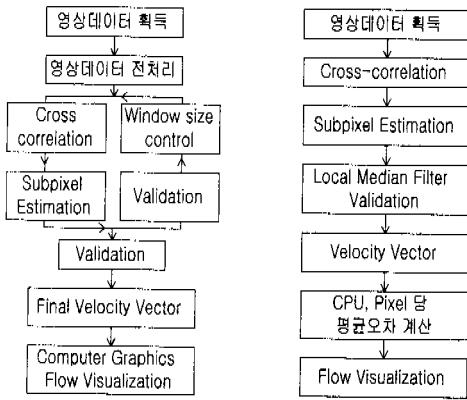
생성된 입자위치에서의 속도벡터를 2차원 선형보간으로 구한 후, Runge - Kutta 방법에 의해 시간간격 Δt 후의 2번째 영상 데이터를 생성하였다. Δt 는 0.015, 0.022로 하였고, 이는 각각 최대 이동거리 7, 10픽셀에 해당한다. 실제의 영상 노이즈를 묘사하기 위해 평균밝기 75 표준편차 30(콘트라스트 약 70)의 가우스 노이즈를 첨가한 영상틀도 생성하였다. 생성된 영상의 부분 예는 Fig. 1과 같다. 아래와 우측의 선형데이터는 HPM에 쓰이는 정사영상의 예이다.

3. PIV 유동해석

본 논문에서 채택한 2프레임 PIV기법은 짧은 시간간격으로 획득된 2개의 영상데이터를 화상처리하여 노이즈를 제거한 후, 상호상관계수(cross correlation coefficient)를 이용하여 속도벡터를 동시에 결정한 다음, 오류벡터를 제거하고 보간에 의해 최종 벡터를 구하고, 컴퓨터그래픽을 이용하여 유동을 가시화하는 과정들로 이루어졌다. (Fig. 2(a) 참조) 본 논문에서는 화상데이터의 전(前)처리과정과 윈도우 크기조절 과정은 생략하고 256 그레이레벨 데이터를 직접 이용한 단순화된 과정(b)을 채택하였다.

3.1 벡터추적

벡터추적과정은 MPEG 등 화상압축 및 코드화, 애니메이션, AI Vision분야 등에서 많이 연구되고



(a) General PIV Flow Chart (b) Simplified PIV Flow Chart
Fig. 2 Flow chart of PIV measurement

있으며,⁽⁷⁾ 프레임간의 입자분포 형태가 짧은 시간 안에서는 거의 변하지 않는다는 가정 하에, 프레임 1에서의 입자 분포 패턴과 가장 유사한 패턴을 프레임 2에서 찾는 과정이다.

먼저 첫 프레임에서 일정 크기의 상관영역을 설정한 후, 둘째 프레임의 동일한 입자분포패턴을 갖는 상관영역을 찾아서 두 영역중심간의 거리를 구하여, 시간간격으로 나누면 속도를 계산할 수 있다. 식 (2-1), 식 (2-2)에 나타낸, 두 영역간의 상호상관계수(cross-correlation coefficient)가 최대인 점이 동일패턴을 갖는 상관영역의 중심이 되며, 벡터의 종점이 된다. 여기서, $I(m,n,t)$ 는 첫 프레임에서의 그레이레벨값이며, $\bar{I}(t)$ 는 상관영역내의 평균그레이레벨 값이다. 식 (2-2)는 식 (2-1)의 개선된 형태이며, FFT방법에는 사용될 수 없다.

유동장을 일정한 개수의 격자로 나누어, 상관영역의 중심을 격자점으로 하여 패턴 추적을 실시한다. 이 방법의 장점은 벡터 추적의 결과가 일정한 격자점에 걸쳐서 생성되기 때문에 보간이나 미·분값 계산 및 유동가시화에 편리하다.

기존의 전영역 추적법은 예상 최대이동거리 내에서 전 영역에 걸쳐 검색하므로 계산량이 방대하다.

본 연구에서는 방대한 계산량을 줄이기 위해 운동보상방식(motion compensation)을 이용한 영상코드화 기법들 중에서 FFT, TSS, NTSS, HPM을 도입하였다. HYB는 식 (2-1)에서 픽셀(pixel)단위 이동거리는 FFT를 이용하여 계산하고, 서브픽셀수정은 직접계산하는 방법이며, TSS와 NTSS는 프레임간의 최대 이동거리를 3단계(4-2-1)로 나눈 후, 각 단계별로 상관영역의 중심점과 외곽 8개 지점중 최대 상관계수 값을 가진 격자점을 다음 단계 상관 영역의 중심으로 잡아서 최종적으로 추적된 최대 상관계수 값을 가진 영역을 벡터 생성에 이용한다. 프레임간 최대 이동 픽셀수가 N (≤ 7 pixel)이라 할 때, 한 격자점에 대해 식 (2-1)와 식 (2-2)를 기존의 격자점 추적법이 $(2N + 1)^2$ 번을 계산하는 반면, TSS와 NTSS는 약 25회 정도 계산하므로 최대 픽셀이동거리가 늘어날수록 계산효율이 현저히 증가된다.

Fig. 3은 TSS기법의 벡터추적과정을 보여주고 있는데, 상관계수 값은 실제 움직인 위치 부근에 높은 값이 밀집해 있음을 볼 수 있고, 1단계씩 거처감에 따라서 상관 계수가 높은 쪽으로 수렴되어 가는 것을 볼 수 있다. 또한, 식 (2)연산에서 1픽셀 건너 뛰어 계산하면(sub-sampling),⁽⁸⁾ 계산량을 1/4가량 줄일 수 있다. HPM은 Fig. 1의 경우, 우측 아랫부분에서 보듯이, 이차원 영상을 수직·수평 2개의 정사된 1차원 데이터로 변환시켜 1차원 상관계수를 구하여 속도벡터를 구하는 방법이며, 최소의 계산시간을 요한다.

상호상관계수에 의한 속도벡터계산의 공간정밀도는 픽셀(pixel)단위로 제한된다. 상호상관계수값의 분포를 식 (3)의 가우스(Gauss)분포로 가정하 3점근사를 도입하면, 서브픽셀(sub-pixel)의 높은 정확도를 얻을 수 있다.⁽⁴⁾

$$C_{ij} = \frac{\sum_{m=1}^M \sum_{n=1}^N I(m,n,t) \cdot I(m+i,n+j,t+\tau)}{\left[\sum_{m=1}^M \sum_{n=1}^N I^2(m,n,t) \right]^{1/2} \cdot \left[\sum_{m=1}^M \sum_{n=1}^N I^2(m+i,n+j,t+\tau) \right]^{1/2}} \quad (2-1)$$

$$C_{ij} = \frac{\sum_{m=1}^M \sum_{n=1}^N (I(m,n,t) - \bar{I}(t)) \cdot (I(m+i,n+j,t+\tau) - \bar{I}(t+\tau))}{\left[\sum_{m=1}^M \sum_{n=1}^N (I(m,n,t) - \bar{I}(t))^2 \right]^{1/2} \left[\sum_{m=1}^M \sum_{n=1}^N (I(m+i,n+j,t+\tau) - \bar{I}(t+\tau))^2 \right]^{1/2}} \quad (2-2)$$

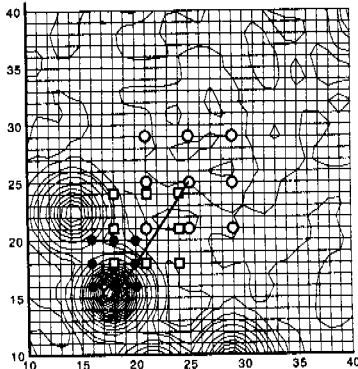
$$C.C(x) = C.C_{max} \cdot \exp\left(\frac{-(x-x_0)^2}{2\sigma^2}\right) \quad (3)$$

C.C는 상관계수값을 의미하고, x_0 는 최고상관 계수값을 갖는 점의 x 좌표이다. 식 (3)을 이용하여, 최대상관계수를 갖는 픽셀점 주위의 3픽셀에서의 C.C값을 이용하여 $C.C_{max}$, x_0 , σ 를 계산한다.

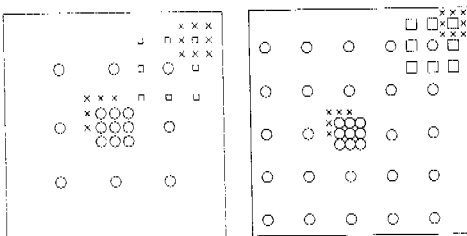
3.2 평가 알고리즘 및 후(後)처리 과정

실험 기법의 개선과 동일 비중의 입자 사용 등으로 개선될 수 있으나, 오류 벡터의 발생은 피할 수 없으며, 발생한 오류벡터들은 변형도나 와도, 또는 유선계산에 큰 악영향을 미친다. 평가의 목적은 프로그래머의 개입 없이 오류벡터들을 제거하고, 정확한 값으로 대체하는데 있다.

여러 종류의 평가방법을 도입하여 가상화상 데이터를 이용하여 성능을 비교한 결과, 국소중간치필터(local median filter)⁽⁵⁾를 이용한 방법이 가장 적절하였다. 크기와 방향에 있어서, 격자 주위의 최대 8개 벡터의 중간치와 비교하여 각각 50% 이상의 차이를 보인 오류벡터들을 국소중간치로 대체하였다. 계산된 속도벡터는 저대역필터



(a) Three step search(TSS)



Max. distance = 7 pixel Max. distance = 11 pixel
(b) New three step search(NTSS)

Fig. 3 Vector tracking by TSS and NTSS

에 의해 조정되었다.

PIV해석의 최종단계는 구해진 격자점에서의 속도 벡터값을 이용하여 유선, 등와도선, 등온선, 등용력선, 난류강도 등을 도시하는 후처리 과정이다.

4. PIV 유동가시화 실험

폭 100mm, 높이 60mm, 계단높이 20mm의 후향 계단유동($Re=8000$)에 대해 NTSS 기법을 이용한 PIV 알고리즘을 적용하였다. 주요 재원은 Fig. 4에서와 같이, 6W 아르곤이온레이저를 광원으로 하고, 흑백 CCD 카메라(SONY XC-711), 프레임 그래버(Matrox, 640×480), PC(Pentium II 266MHz CPU)로 이루어져 있다.

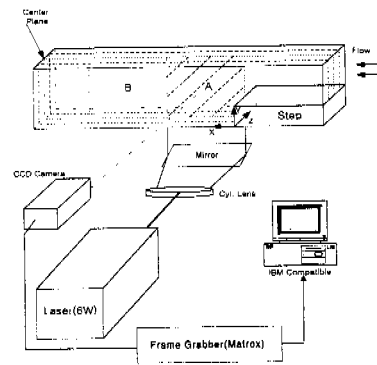
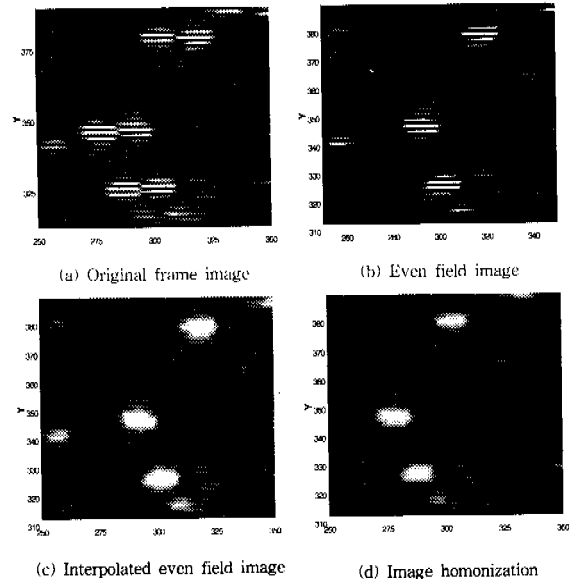


Fig. 4 Schematic view of experimental setup



(a) Original frame image (b) Even field image
(c) Interpolated even field image (d) Image homonization

Fig. 5 Image separation

Re수 8000의 난류유동을 형성하기 위해 계단입구의 평균유속은 20cm/sec로서 60mm×80mm의 영상을 30프레임/초로 획득할 경우 최대이동거리는 36pixel에 이르고, 입자영상은 레적이 길다.

본 실험에서는 Fig. 5와 같이, 1프레임의 영상을 짝·홀수 필더(Even and Odd Field)영상으로 분해하여 시간간격을 1/60초로 반감시키고, CCD카메라 내장 셔터스피드를 1/125(even, odd field 각각에 대해)초로 하여 개선된 영상을 이용하였다. 그림에서 입자는 실제지경 3픽셀 정도이나, 예시를 위해 흐린화면(blurred image)으로 과장하였다.

5. 결과 및 고찰

5.1 이미지코드화 기법들의 성능 비교 결과

레이놀즈수 1000인 사각공동유동에 대한 수치해와 이를 이용한 가상영상 데이터와 PIV 해석과정 및 생성된 유선을 Fig. 6에 나타내었다. 서브픽셀교정의 효과로 저속 2차 와동의 정확도가 증가함을 볼 수 있다. 도입된 TSS, NTSS, HPM, HYB를 이용한 PIV 해석에서 소요된 CPU(Pentium II 260MHz) 연산시간과 평균픽셀오차를 Table 1, 2에 요약하였으며, CPU는 속도벡터추적, 서브픽셀교정, LMF검정 저대역필터에 의한 벡터순화과정에 사용된 시간을 컴퓨터 내부 클락을 사용하여 측정하였다. Table에서 HYB1은 2단계 (32×32, 16×16) FFT에 의한 정수 픽셀 추적후 직접 서브픽셀을 교정하는 개선 알고리즘을 말한다. Table 2는 식 (2-2)를 사용하여 개선된 결과를 보여주며 FFT를 이용한 HYB등에는 사용될 수 없다. NTSS나 TSS는 이 경우 계산효율 및 정확도 면에서 HYB에 비해 좋은 결과를 보였다. 또한 공동유동의 정확도를 비교하기 위한 주와동, 2·3차 와동의 중심좌표와 유동함수값을 Table 3에 요약하였다.

노이즈 없는 최대이동거리 7 pixel의 결과를 보면, NTSS, HPM방식은 FULL에 비해 0.1 이내 픽셀오차 증가를 대가로 계산시간을 크게 단축시킬 수 있었으며, HYB 및 NTSS방식은 오차변화가 거의 없이 계산시간을 1/10 이내로 감소시켰다.

콘트라스트 70의 비교적 심한 노이즈가 포함된 화상의 경우, FFT를 이용한 HYB와 정상영상을 이용한 HPM이 상대적으로 정확하여, 이들 방법이 영상데이터에 대한 저대역필터(low pass filter) 기능을 잠재적으로 가지고 있음을 볼 수 있었다.

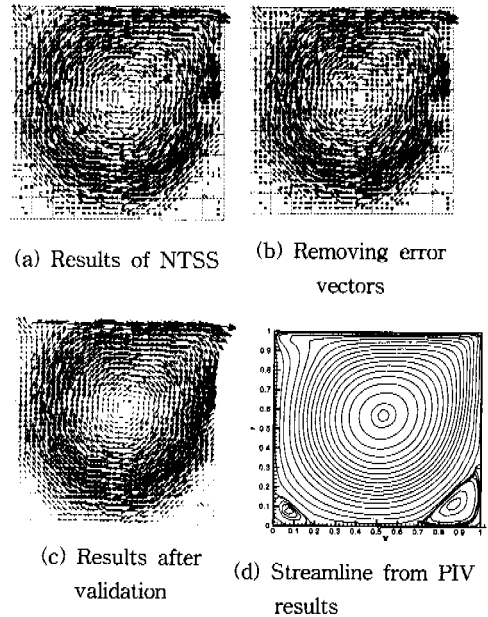


Fig. 6 PIV results of synthetic data

큰 이동거리의 경우 (10 pixel), TSS, NTSS, HPM 등 이미지코드화 기법들은 결과속도벡터들에 대한 저대역필터링에 의한 조정이 없으면 일반적으로 큰 오차를 나타내었으나, 저대역필터에 의한 조정과정과 식 (2-2)의 사용으로 결과는 개선되어 FULL기법뿐만 아니라 HYB에 비해서도 나은 결과를 보였다(Table 2 참조).

위 결과들로부터 새로 도입된 기법들을 평가해보면, 프레임당 간격을 조절하여, 최대 이동거리를 7픽셀 이내로 하면, NTSS는 정확하고도 빠른 PIV 알고리즘으로 사용될 수 있으며, 최대이동거리 10픽셀에서도 NTSS 및 TSS는 식 (2-2)와 속도벡터의 저대역필터링의 도입을 통하여 정확하고 빠른 PIV 해석이 가능하였다. 단, Table의 결과에서 노이즈가 없는 경우, 식 (1)의 σ 를 0.5와 1.0으로 한 화상에 대한 해석결과는 차이가 있었다. σ 를 0.5로 할 경우의 영상데이터는 입자지경 1 픽셀과 유사하게되어 Table 1의 TSS*의 경우 오차가 노이즈가 있는 경우보다 크게 나왔으나, σ 를 1.0으로 하면(Table 2) 정상적인 결과를 보였다. 그러나 FFT, FULL, HPM은 거의 무관하거나 아주 미세하게 반대의 경향을 보였다.

5.2 실제유동에 적용한 결과

후향계단유동의 경우, 레이놀즈수 8000의 난류 유동에 대하여, NTSS 기법을 이용하였다. 영상데이터는 초당 30 프레임으로 99프레임을 획득하여 98개의 벡터장을 통계처리하여 그 결과를 Fig. 7,8에 나타내었다.

Kelvin-Helmoltz 와류이탈 주기는 약 2.5Hz로 나타났으며, 중간 단면에서 평균 재부착 길이는 계단높이의 6배 정도이다. 비슷한 기하학적 조건과 Re수를 갖는 DNS결과⁽⁹⁾ 및 renormalized $\kappa - \epsilon$ 모델에 의한 수치해석 결과를 비교하여 Fig. 9에 나타내었다. 평균유속의 경우 비교적 정확하나, 난류장도(Reynolds Stress)의 경우 더 많은 프레임의 데이터와 공간정밀도가 필요하다고 본다.

6. 결 론

영상코드화 및 압축분야에서 개발된 NTSS, TSS, HPM 등을 도입하여 PIV해석의 빠른 벡터추적 알고리즘을 만들어 가상영상, 실험영상에 적용하였다. HPM은 가장 빠른 알고리즘이었으나 오차가 가장 크며, 10픽셀이상의 이동거리에서는 많은 개선을 요하였다. NTSS 및 TSS는 수정된 상관계수공식과 결과벡터들에 대한 저대역 필터링을 장착할 경우 FULL, HYB 등 기존의 PIV해석 알고리즘 보다 계산 효율면에서 개선된 결과를 보였다.

흐락 영상분리, 전처리과정생략, NTSS 등 빠른 벡터추적, 국소중간필터 및 9점 저대역필터의 도입은

Table 1 CPU time mean error : Max. movement = 7 pixels

Method	CPU(sec.)	Mean Error (pixel)	
		Without noisy($\sigma=0.5$)	With noisy($\sigma=1.0$)
FULL	119.84	0.119806	0.181148
FULL'	33.23	0.133672	0.190137
HYB	5.11	0.170201	0.179420
HYB1	9.39	0.128517	0.146426
TSS	12.03	0.163133	0.213580
TSS'	3.68	0.367900	0.220273
NTSS	12.64	0.123035	0.195907
NTSS'	3.63	0.154722	0.207645
HPM	1.78	0.194254	0.216993

Table 2 CPU time mean error : Max. movement = 10 pixels

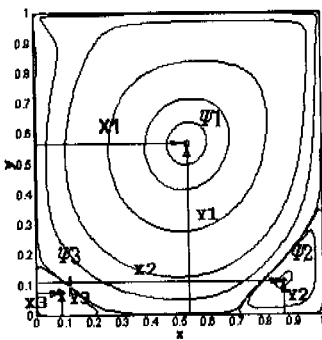
Method	식(2-1)			식(2-2)		
	CPU(sec.)	Mean Error (pixel)		CPU(sec.)	Mean Error (pixel)	
		Without noisy ($\sigma=1.0$)	With noisy ($\sigma=1.0$)		Without noisy ($\sigma=1.0$)	With noisy ($\sigma=1.0$)
FULL	197.29	0.174014	0.203476			
FULL'	55.09	0.182148	0.211345	60.31	0.171470	0.173093
HYB	5.49	0.226442	0.224044			
HYB1	9.93	0.182484	0.178355			
TSS	16.26	0.177232	0.214750	18.12	0.169386	0.175660
TSS'	4.61	0.187137	0.250288	5.17	0.177737	0.213058
NTSS	20.04	0.160965	0.177365	22.47	0.1630308	0.1635876
NTSS'	5.06	0.171179	0.202612	6.59	0.1771459	0.1899406
HPM	2.42	0.286797	0.274698			

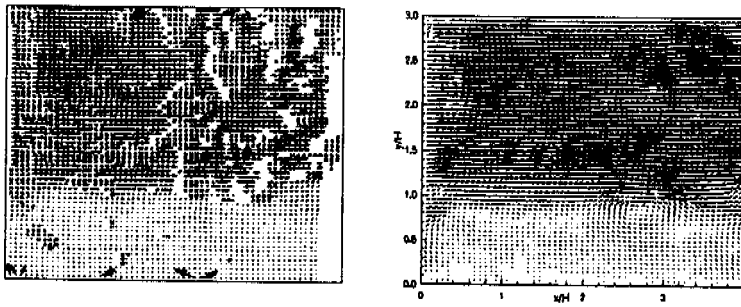
*는 Subsampling 한 경우

Table 3 Comparison of center of vortices and its stream function value

	ψ_1	X_1	Y_1	ψ_2	X_2	Y_2	ψ_3	X_3	Y_3
FULL*	0.116863	0.525	0.575	7.46098E-4	0.899	0.125	-1.17782E-4	0.050	0.100
FULL	0.116448	0.525	0.575	-1.06133E-3	0.875	0.125	-8.09536E-5	0.075	0.075
TSS*	0.107750	0.525	0.575	-9.71873E-4	0.875	0.125			
TSS	0.110190	0.530	0.550	-1.32100E-3	0.875	0.125			
NTSS*	0.113043	0.525	0.575	-8.44159E-4	0.899	0.125	-1.31317E-4	0.050	0.125
NTSS	0.112910	0.525	0.575	-1.24630E-3	0.875	0.125	-3.29998E-5	0.075	0.050
HPM*	0.114592	0.525	0.550	-5.93334E-4	0.899	0.125	-3.69143E-4	0.100	0.075
HYB	0.116006	0.525	0.575	-8.99324E-4	0.875	0.125	-8.98499E-5	0.075	0.075
HYB1	0.116362	0.525	0.575	-1.03811E-3	0.875	0.125	-1.01286E-4	0.075	0.075
NUM.	0.118051	0.525	0.563	-1.75187E-3	0.8625	0.1125	-2.1388E-4	0.0875	0.075

*는 Subsampling 한 경우

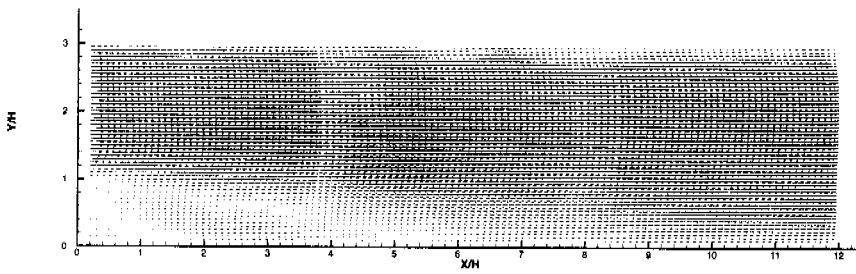




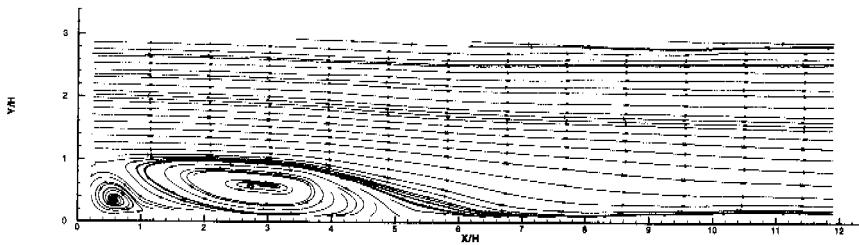
(a) NTSS result

(b) Final result after validation

Fig. 7 PIV results of backward step flow : Instantaneous velocities

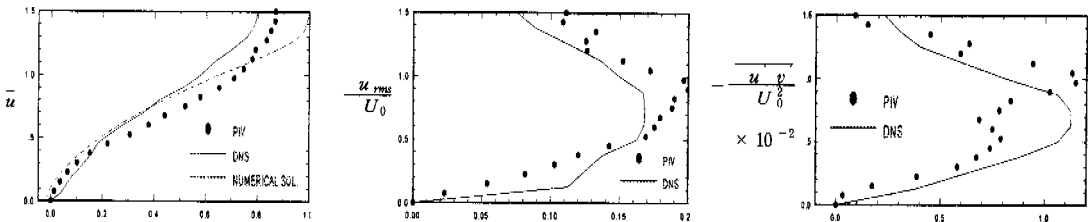


(a) Mean velocity vector distribution



(b) Mean streamline

Fig. 8 PIV results of backward step flow : Mean quantity



(a) Average velocity

(b) Turbulent intensity

(c) Reynolds shear stress

Fig. 9 Comparison of velocity and turbulent intensity with DNS and numerical results

저가의 실험 장비로 최대이동거리 36픽셀의 고속 유동해석도 가능케하였다.

MPEG II 등 새로운 코드화 패키지에서는 새로운 알고리즘들이 계속 제안되고 있으며, 본 논문에서 다루어지지 않은 하드웨어 구현성을 포함하여 초고속 실시간 PIV보드 개발 등을 고려하면 영상코드화 분야에서 개발된 기법들의 도입 및 개선은 유용하다고 보여진다.

후 기

본 연구는 후기 교육부의 1998년도 기계공학 학술 연구 조성비 지원(과제번호 : 1998-018-E00034)에 의해 수행되었으며, 이에 대해 감사드립니다.

참고문헌

- (1) Jain, A. K., 1981, "Image Compression : A Review," *Proc. IEEE*, Vol. 69, pp. 349~389.
- (2) Li, R. et al., 1994, "A New Three-Step Search Algorithm for Block Motion Estimation," *IEEE Trans, Circuits, Systems for video Tech.*, Vol. 4, No. 4, pp. 438~442.
- (3) Ken Sauer and Brian Schwartz, 1996, "Efficient Block Motion Estimation Using Integral Projections," *IEEE Trans. Circuits, Systems for video Tech.*, Vol. 6, No. 5, pp. 513~518.
- (4) Willert. C. E. and Gharib. M., 1991, "Digital Particle Image Velocimetry," *Exp. Fluids*, 10, pp. 181~193.
- (5) Westerweel et al., 1996, "Measureent of Fully Developed Turbulent Pipe Flow with Digital Particle Velocimetry," *Exp. Fluids*, 20, pp. 165~177.
- (6) 김성균, 1996, "다수의 주상체들의 저진폭동위상 진동에 의한 2차 정상유동 해석," 대한기계학회 (B), 제20권 제2호, pp. 649~658.
- (7) Tokumaru. P. T. and Dimotakis. P. E., 1995, "Image Correlation Velocimetry," *Exp. Fluids*, 19, pp. 1~15.
- (8) Chan, Y. and Siu, W. 1996, "New Adaptive Pixel Decimation for Block Motion Vector Estimation," *IEEE Trans. Circuits, Systems for Video Tech.*, Vol. 6, No. 4, pp. 113~122.
- (9) Kasagi. N. and Matsunaga. A., 1995, "Three-dimensional Particle Tracking Velocimetry Measurement of Turbulence Statistics and Energy Budget in a Backward-facing Step Flow," *Int. J. Heat and Fluid Flow*, Vol. 16, pp. 477~485.