



Real-time Implementation of Synthetic Aperture Vector Flow Imaging on a Consumer-level Tablet

di Ianni, Tommaso; Kjeldsen, Thomas Kim; Villagómez Hoyos, Carlos Armando; Mosegaard, Jesper; Jensen, Jørgen Arendt

Published in:

Proceedings of 2017 IEEE International Ultrasonics Symposium (IUS)

Link to article, DOI:

[10.1109/ULTSYM.2017.8092359](https://doi.org/10.1109/ULTSYM.2017.8092359)

Publication date:

2017

Document Version

Peer reviewed version

[Link back to DTU Orbit](#)

Citation (APA):

di Ianni, T., Kjeldsen, T. K., Villagómez Hoyos, C. A., Mosegaard, J., & Jensen, J. A. (2017). Real-time Implementation of Synthetic Aperture Vector Flow Imaging on a Consumer-level Tablet. In Proceedings of 2017 IEEE International Ultrasonics Symposium (IUS) IEEE. <https://doi.org/10.1109/ULTSYM.2017.8092359>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Real-time Implementation of Synthetic Aperture Vector Flow Imaging on a Consumer-level Tablet

Tommaso Di Ianni¹, Thomas Kim Kjeldsen², Carlos Armando Villagómez Hoyos¹,
Jesper Mosegaard², Jørgen Arendt Jensen¹

¹Center for Fast Ultrasound Imaging, Department of Electrical Engineering,
Technical University of Denmark, DK-2800 Lyngby, Denmark

²Visual Computing Lab, Alexandra Institute, DK-8200 Aarhus N, Denmark

Abstract—In this work, a 2-D vector flow imaging (VFI) method based on synthetic aperture sequential beamforming (SASB) and directional transverse oscillation is implemented on a commercially available tablet. The SASB technique divides the beamforming process in two parts, whereby the required data rate between the probe and back-end can be reduced by a factor of 64 compared to conventional delay-and-sum focusing. The lowered data rate enables real-time wireless transfer for both B-mode and VFI data. In the present setup, element data were acquired from a straight vessel with the SARUS research scanner and processed by a first-stage beamformer in a fixed focus. The data were subsequently transferred to an HTC Nexus 9 tablet through an ASUS RT-AC68U Wi-Fi router to simulate a wireless probe. The second-stage beamforming of the B-mode and flow data and the velocity estimation were implemented on the tablet's built-in GPU (Nvidia Tegra K1) through the OpenGL ES 3.1 API. Real-time performance was achieved with rates up to 26 VFI frames per second (38ms/frame) for concurrent processing and Wi-Fi transmission.

I. INTRODUCTION

This paper presents an implementation of 2-D synthetic aperture (SA) vector flow imaging (VFI) on a commercially available tablet. The objective is the integration of quantitative blood flow imaging in portable ultrasound systems.

Pocket-size devices have the potential to open new possibilities for the healthcare system bringing ultrasound out of the radiology department and closer to the point of care [1]. An attempt of integrating a flow imaging modality in a general-purpose mobile device has been previously made by Hemmsen *et al.* [2] based on the method presented in [3]. However, the implementation has two major limitations. First, the velocity estimation is limited to the axial component and a prior knowledge of the flow angle is required. Second, it relies on a cross-correlation estimator that necessitates significant computational resources for the calculation of the correlation functions, which makes the method not optimal for the strict requirements of a mobile device. Other portable ultrasound systems with flow capabilities have been previously presented in literature [4], [5] or are available on the market. However, these systems only perform 1-D flow estimation and are affected by flow angle limitations.

VFI methods estimate the velocity in 2-D and do not require any manual angle adjustments providing quantitative velocity measurements [6]–[8]. For these reasons, the integration of

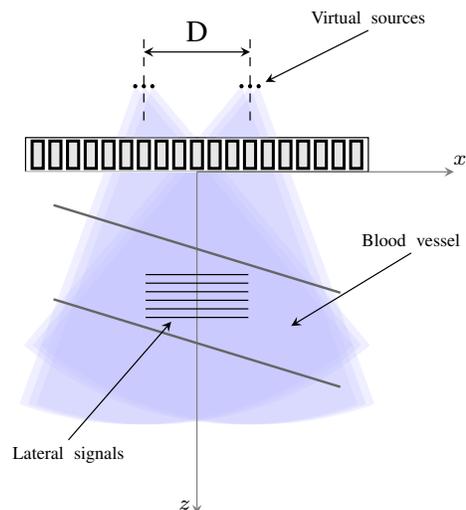


Fig. 1. Imaging setup with a linear array transducer and relative position of the VSs.

VFI in hand-held devices has the potential to improve the clinical workflow. A 2-D SA VFI method for portable ultrasound has been recently proposed in [9] combining SA sequential beamforming (SASB) [10] and directional transverse oscillation (TO) [11]. The method enables the wireless transmission of the ultrasound data and uses a relatively inexpensive 2-D phase-shift approach for the velocity estimation. The present paper demonstrates a real-time implementation on a consumer-level mobile device using processing in the built-in GPU concurrent with the transmission of the data over Wi-Fi.

II. BACKGROUND AND THEORY

The ultrasound data received by L transducer elements are initially beamformed by a first-stage fixed-focus beamformer. The transmit/receive focal point is referred to as virtual source (VS). This is located behind the array, therefore a diverging wave is emitted to insonify the whole image area at each emission. The fixed-focus beamformer can be integrated in the front-end of the ultrasound probe [12], therefore one single beamformed line, referred to as low-resolution line (LRL), is obtained for each emission, with a reduction of the data

TABLE I
MEASUREMENT SETUP AND PROCESSING PARAMETERS

Parameter	Flow	B-mode	Unit
<i>Transducer</i>			
Array type	Linear		-
Element pitch	0.2		mm
Element height	6		mm
Number of elements	192		-
Elevation focus	38		mm
Center frequency	4.1		MHz
<i>Imaging setup</i>			
Excitation	4 cycles	2 cycles	-
Center frequency - f_0	4.1		MHz
Tx/Rx apodization	Tukey ($\alpha = 0.6$)		-
VS axial position	-15	-30	mm
Active elements - L	64	96	-
f-number - $f\#$	-1.17	-1.56	-
Number of VSs	$K = 6$	64	-
Distance between VAs - D	48	-	elements
Pulse repetition freq. - PRF :	9		kHz
<i>Processing</i>			
Lateral sampling interval	0.1		mm
Axial sampling frequency	20		MHz
Lateral signal length - M	32	-	-
Number of HRIs - N	16	-	-

throughput by a factor L . The LRLs are then sent to the mobile device, where a second beamforming stage takes place.

A high-resolution image (HRI) is created in the second stage after K emissions. The second beamformer is based on the VS assumption [13], and its computational complexity equals that of a monostatic SA beamformer. The K VSs are divided in two virtual apertures (VAs) that are laterally separated by a distance D as shown in Fig. 1. This separation gives rise to a TO in the HRIs, which enables the velocity estimation in both the lateral and axial directions.

A moving-average-subtraction filter is used on the LRLs to remove the signal from the tissue. The beamformed HRIs are Hilbert transformed in the lateral direction, and the lateral frequency is estimated through a Fourier transform [9]. For each velocity point, M lateral samples are selected, and the velocity is estimated from N HRIs using an autocorrelation approach [11]. The method provides velocity estimates everywhere in the image, and continuous data acquisition is obtained.

III. METHODS

A. Measurement setup

An experimental setup was constructed to assess the data transmission and processing performance on a consumer-level tablet. A linear array was connected to the SARUS scanner [14] for the acquisition of the element data, and a duplex sequence was designed with the parameters in Table I. The VAs were steered toward the center of the image with an angle of $\pm 2.5^\circ$. No apodization was used on the VAs. The B-mode VSs were regularly distributed between ± 15 mm in the lateral direction. B-mode and flow emissions were interleaved to achieve continuous data acquisition, i.e. one B-mode pulse

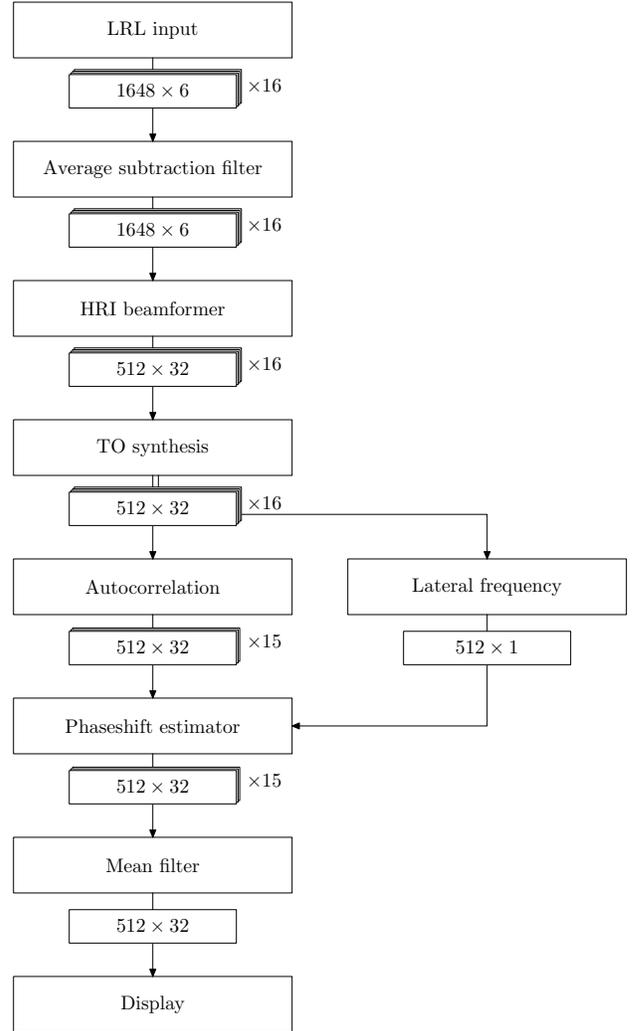


Fig. 2. Schematic overview of the VFI processing sequence on the GPU.

was transmitted after every $K = 6$ flow emissions, with an effective pulse repetition frequency $PRF_{eff} = PRF/(K+1)$.

Measurements were performed on a flow rig system consisting of a centrifugal pump (Cole-Parmer, Vernon-Hills, IL, USA) circulating a blood mimicking fluid. A laminar flow profile was created in a vessel with a radius of 6 mm placed at a depth of approximately 20 mm with a beam-to-flow angle of 90° . The volume flow was measured for reference by a magnetic flow meter (MAG3000, Danfoss, Nordborg, Denmark) and was set to obtain a peak velocity of approximately 0.2 m s^{-1} .

For each emission, the channel data received by $L = 64$ transducer elements were beamformed in Matlab (The MathWorks, Inc., Natick, MA, USA) using a fixed-focus to create the LRLs. These were sent to a Nexus 9 tablet (HTC Corp., Taoyuan, Taiwan) through a Wi-Fi link to simulate a wireless probe. The connection was established through an ASUS RT-AC68U (ASUS, Taipei, Taiwan) router with a maximum theoretical link speed of 867 Mbit/s. B-mode and VFI processing were performed on the tablet's built-in GPU.

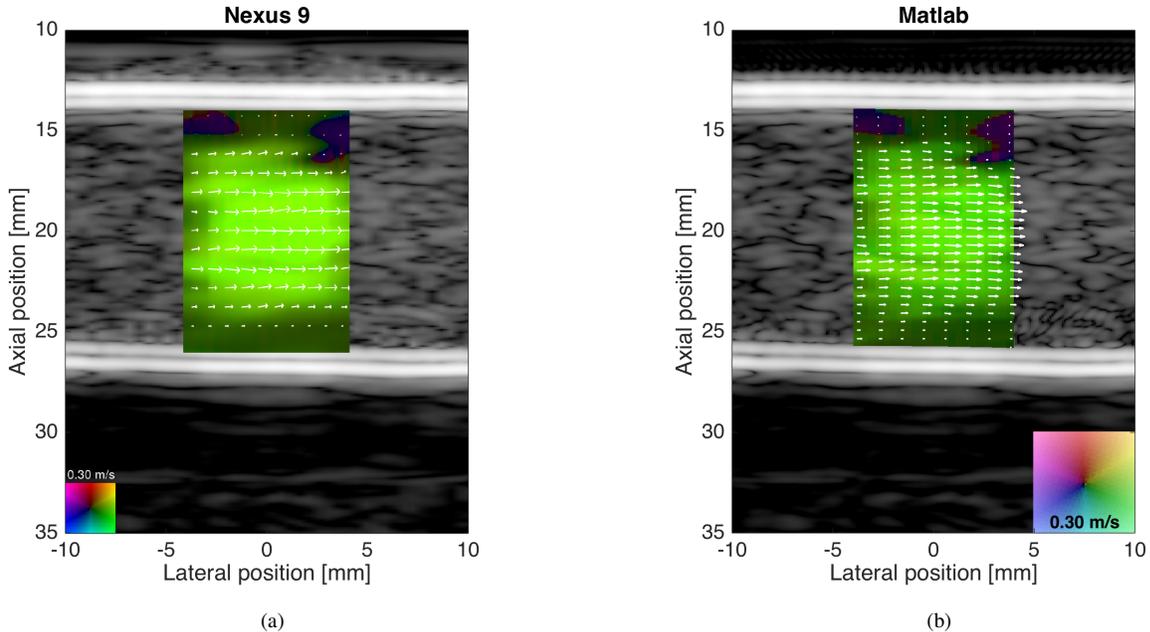


Fig. 3. Comparison of the 2-D VFI frames processed in the tablet (a) and in Matlab (b). The VFI and B-mode images are overlapped. The arrows show the velocity vectors, and the angle is shown encoded as in the color wheel at the bottom of the images.

B. GPU implementation

To achieve real-time performance, all the VFI computations outlined in Sec. II and described in [9] must be implemented on the tablet's GPU (Nvidia Tegra K1). Fig. 2 shows the GPU processing graph for each VFI frame in addition to the data flow arranged in 2-D texture arrays. The GPU was instructed to perform general purpose computations through the OpenGL ES 3.1 API, and each of the tasks in Fig. 2 was implemented as compute or fragment shaders. The LRLs were buffered in the tablet's memory and copied to the GPU texture memory for further processing. The second-stage beamformer refocused the LRLs creating high-resolution lateral signals of length $M = 32$ samples, and 512 samples were used for each velocity line in the axial direction. The velocities were estimated from $N = 16$ HRIs.

The GPU is perfectly suited for most of the steps, such as individual per-pixel operations, which are straightforwardly executed in parallel. However, some of the tasks require more effort to run efficiently, in particular concerning the TO synthesis, where spatial quadrature signals are created by discrete Hilbert transformations. These are generated by making the frequency spectrum one-sided using forward and inverse fast Fourier transforms (FFTs) [15]. In the present work, radix-2 FFTs were implemented in OpenGL ES 3.1 compute shaders with use of fast inter-thread shared memory for efficient parallel computation.

Finally, each VFI frame was interleaved with a B-mode frame, which was computed on the GPU by refocusing, envelope detection, and scan conversion as described in detail in [16], [17].

The software implementation was written in C++ and GLSL and compiled either to a desktop application or cross-compiled to Android with the Android Native Development Kit. The performance can, thus, be evaluated easily on a range of discrete and mobile GPUs and operating systems.

IV. RESULTS

The results of the 2-D VFI implementation are shown in Fig. 3, where frames processed in the tablet (3a) and in Matlab (3b) are shown for comparison. The VFI and B-mode images are overlapped. The arrows show the velocity vectors, and the angle is shown encoded as in the color wheel in the bottom of the images. The 2-D velocities were calculated over 32 lines.

The mean velocity profile \pm one standard deviation (SD) is shown in Fig. 4 calculated from 50 velocity profiles at the center of the VFI image processed in the tablet. The average bias was 2.91% in the lateral direction and 0.93% in the axial direction. The SD was 10.13% and 3.51% for the two velocity components.

The combined VFI and B-mode peak frame rate was 26 frames per second, which is sufficient for real-time performance. The corresponding data rate for the transmission of LRLs over the Wi-Fi link was 13 MB/s for 1648 16-bit samples per LRL. This was lower than the average throughput of 30 MB/s measured between the tablet and the Wi-Fi router. Hence, the operational rate of the entire system is limited by the GPU processing speed rather than the data bandwidth.

Previously, it was identified that the particular tablet model suffers from heating issues when the GPU and network card operate concurrently close to the maximum load [2]. To test

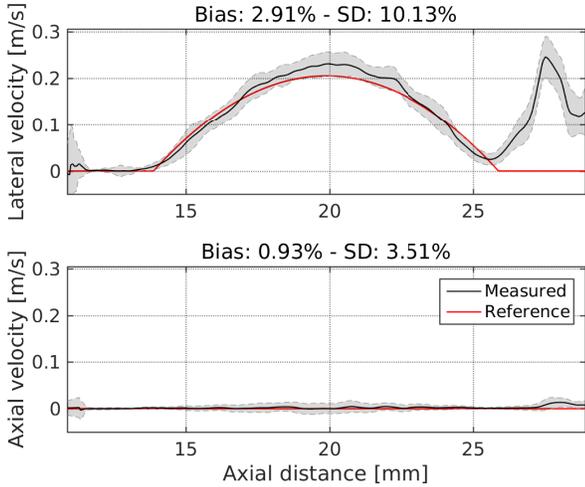


Fig. 4. Mean lateral (top) and axial (bottom) velocity profiles \pm one standard deviation calculated from 50 velocity profiles along the central axial line of the VFI image.

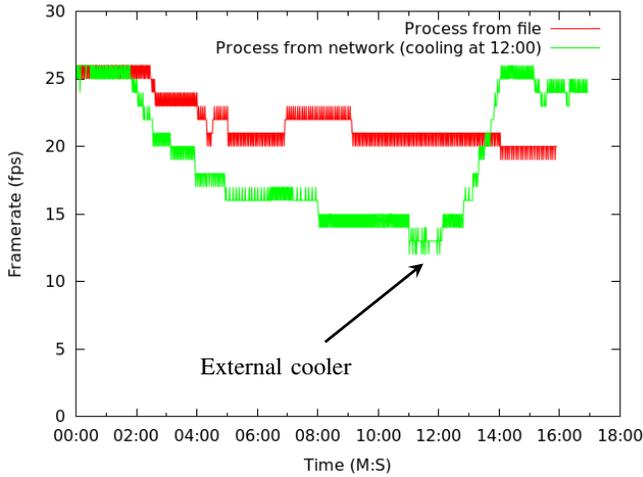


Fig. 5. Processing framerates for continuous operation with data input from a local file (red) and from a Wi-Fi network (green). An external cooler was activated after 12 minutes for the Wi-Fi case.

whether this is the case for the present implementation, continuous scanning simulations were carried out for a duration of several minutes. The benchmark covered data input from a) a pre-recorded dataset stored on the device and b) the Wi-Fi link. In Fig. 5, the processing frame rate is shown as a function of the time since the start of the session. It can be noticed that the frame rate was lowered significantly after a few minutes of combined GPU and network load. This issue is indeed related to thermal throttling because the initial peak frame rate was restored when an external cooler was activated. Note that the thermal issues are much less severe if the data are read from the memory and the network card is idle.

V. CONCLUSION

In conclusion, SA VFI is feasible in a commercial tablet with real-time performance. The computational tasks were

efficiently implemented on the GPU, and the data reduction of the sequential beamforming enables real-time wireless transmission. For continuous scanning sessions of several minutes duration, however, the performance degraded due to heating of the chipset of the Nexus 9 tablet used in the experiment.

ACKNOWLEDGEMENT

This work was supported by grant 82-2012-4 from the Danish National Advanced Technology Foundation and by BK Ultrasound.

REFERENCES

- [1] C. Prinz and J. U. Voigt, "Diagnostic accuracy of a hand-held ultrasound scanner in routine patients referred for echocardiography," *J. Am. Soc. Echocardiog.*, vol. 24, no. 2, pp. 111–116, 2011.
- [2] M. C. Hemmsen, L. Lassen, T. Kjeldsen, J. Mosegaard, and J. A. Jensen, "Implementation of real-time duplex synthetic aperture ultrasonography," in *Proc. IEEE Ultrason. Symp.*, 2015, pp. 1–4.
- [3] Y. Li and J. A. Jensen, "Synthetic aperture flow imaging using dual stage beamforming: Simulations and experiments," *J. Acoust. Soc. Am.*, vol. 133, no. 4, pp. 2014–2024, 2013.
- [4] C. Huang, P. Lee, P. Chen, and T. Liu, "Design and implementation of a smartphone-based portable ultrasound pulsed-wave Doppler device for blood flow measurement," *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, vol. 59, no. 1, pp. 182–188, 2012.
- [5] E. Jeong, S. Bae, M. Park, W. Jung, J. Kang, and T. Song, "Color Doppler imaging on a smartphone-based portable US system: Preliminary study," in *Proc. IEEE Ultrason. Symp.*, 2015, pp. 1–4.
- [6] B. Dunmire, K. W. Beach, K.-H. Labs., M. Plett, and D. E. Strandness, "Cross-beam vector Doppler ultrasound for angle independent velocity measurements," *Ultrasound Med. Biol.*, vol. 26, pp. 1213–1235, 2000.
- [7] G. E. Trahey, J. W. Allison, and O. T. von Ramm, "Angle independent ultrasonic detection of blood flow," *IEEE Trans. Biomed. Eng.*, vol. BME-34, no. 12, pp. 965–967, 1987.
- [8] J. A. Jensen and P. Munk, "A new method for estimation of velocity vectors," *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, vol. 45, no. 3, pp. 837–851, 1998.
- [9] T. Di Ianni, C. Villagomez-Hoyos, C. Ewertsen, T. Kjeldsen, J. Mosegaard, and J. A. Jensen, "A vector flow imaging method for portable ultrasound using synthetic aperture sequential beamforming," *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, 2017.
- [10] J. Kortbek, J. A. Jensen, and K. L. Gammelmark, "Sequential beamforming for synthetic aperture imaging," *Ultrasonics*, vol. 53, no. 1, pp. 1–16, 2013.
- [11] J. A. Jensen, "Directional transverse oscillation vector flow estimation," *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, vol. 64, no. 8, pp. 1194–1204, 2017.
- [12] T. Di Ianni, M. C. Hemmsen, P. L. Muntal, I. H. Jørgensen, and J. A. Jensen, "System-level design of an integrated receiver front end for a wireless ultrasound probe," *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, vol. 63, no. 11, pp. 1935–1946, 2016.
- [13] M. H. Bae and M. K. Jeong, "A study of synthetic-aperture imaging with virtual source elements in B-mode ultrasound imaging systems," in *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, vol. 47, 2000, pp. 1510–1519.
- [14] J. A. Jensen, H. Holten-Lund, R. T. Nilsson, M. Hansen, U. D. Larsen, R. P. Domsten, B. G. Tomov, M. B. Stuart, S. I. Nikolov, M. J. Pihl, Y. Du, J. H. Rasmussen, and M. F. Rasmussen, "Sarus: A synthetic aperture real-time ultrasound system," *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, vol. 60, no. 9, pp. 1838–1852, September 2013.
- [15] S. Marple, "Computing the discrete-time 'analytic' signal via FFT," *IEEE Trans. Sig. Proc.*, vol. 47, no. 9, pp. 2600–2603, 1999.
- [16] M. C. Hemmsen, T. Kjeldsen, L. Lassen, C. Kjær, B. Tomov, J. Mosegaard, and J. A. Jensen, "Implementation of synthetic aperture imaging on a hand-held device," in *Proc. IEEE Ultrason. Symp.*, 2014, pp. 2177–2180.
- [17] T. Kjeldsen, L. Lassen, M. Hemmsen, C. Kjær, B. Tomov, J. Mosegaard, and J. Jensen, "Synthetic aperture sequential beamforming implemented on multi-core platforms," in *Proc. IEEE Ultrason. Symp.*, 2014, pp. 2181–2184.