

# MOBILE SENSOR FUSION PROCESSORS: FROM ALGORITHMS TO DEDICATED HARDWARE

eingereichtes  
HAUPTSEMINAR  
von

Lukas Lischke

geb. am 01.06.1993  
wohnhaft in:  
Kazmaistr. 32  
80339 München

Lehrstuhl für  
STEUERUNGS- und REGELUNGSTECHNIK  
Technische Universität München

Prof. J. Conradt

Betreuer: Cristian Axenie  
Beginn: 21.04.2015  
Abgabe: 07.07.2015



April 21, 2015

A D V A N C E D   S E M I N A R  
for  
Lukas Lischke, Mat.-Nr. 03628242

## **Mobile Sensor Fusion Processors: From Algorithms to Dedicated Hardware**

### Problem description:

As sensor-based applications and use cases become more prevalent in smartphones and wearable devices, the need for fast and reliable sensor fusion is increasing. However, as battery life increasingly becomes the largest complaint of device users, OEMs are forced to evaluate their power budgets and seek alternative methods of implementing functions at lower power. In this context, functionality switches from algorithmic software support to dedicated hardware, which spawns important new developments in mobile products. The quest for efficient sensor fusion mechanisms is still on, such that for many applications, working out the subtleties of sensor integration, signal enhancement, calibration, magnetic interference, sensor drift and power consumption has proven to be more difficult than first thought. Although this initiative was supported by advances in VLSI technology, the core processing schemes are still in development. Ranging from purely mathematical approaches, to efficient heuristics or intelligent algorithms based on cognitive modelling and neuroscience, various sensor fusion processors were developed. From devices fusing voice and motion to create natural user experiences, to motion processors for sports, and always-on context awareness processors, state-of-the-art fusion processors technology provides an excitement and challenging research area. The goal of this advanced seminar is to provide an overview over such technologies in order to extract the main development trends and user requirements.

### Tasks:

- Research typical applications for sensor fusion in mobile devices.
- Study the algorithms typically used in mobile sensor fusion processors.
- Investigate the use of intelligent (learning) algorithms in sensor fusion processors.
- Give an overview of dedicated sensor fusion processors on the market.
- Provide a comparison of the various technologies with respect to: complexity, precision, robustness, flexibility and efficiency.

Supervisor: Cristian Axenie

(J. Conradt)  
Professor



## **Abstract**

In current mobile devices multiple sensors are built in, gathering data. Every sensor alone gives an insight of the measured surroundings. The data from all sensors combined in a sensor fusion algorithm lead to drift and noise compensated and self calibrated measurements, which can extract a detailed model of the users world and allow the creation of context aware devices.

This advanced seminar provides an overview and comparison on currently used data fusion approaches and chips on the market.

A trend to low power FPGAs and data fusion coprocessors implementing Kalman filters and neural networks is noticeable.

## **Zusammenfassung**

Heutzutage werden vermehrt Sensoren in mobilen Geräten verbaut. Bereits ein Sensor ermöglicht einige Rückschlüsse auf dessen Umfeld. Die Daten aller Sensoren eines mobilen Gerätes in einem Datenfusionsalgorithmus zusammengefasst ermöglichen drift- und rauschkompensiertes und selbst kalibriertes Messen, sowie das Erstellen eines detaillierten Umgebungsmodelles, das zustandsbezogene Applikationen ermöglicht.

Dieses Hauptseminar bietet eine Übersicht und einen Vergleich zu den aktuell genutzten Datenfusionsalgorithmen und den dazugehörigen elektrischen Bausteinen auf dem Markt.

Es ist ein Trend zu niedrigenergie FPGAs und Datenfusionscoprozessoren, sowie zu Fusionsalgorithmen wie dem Kalman Filter und neuronalen Netzen erkennbar.



# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>Typical Applications for Sensor Fusion in Mobile Devices</b>	<b>7</b>
2.1	Motion Recognition . . . . .	7
2.2	Pattern recognition . . . . .	8
2.3	Healthcare . . . . .	8
2.4	Indoor/GPSless Positioning (Pedestrian Navigation) . . . . .	8
2.5	Object recognition . . . . .	9
2.6	Voice Control . . . . .	9
2.7	Contextual Awareness . . . . .	10
<b>3</b>	<b>Fusion Processors on the Market</b>	<b>11</b>
3.1	Development platforms and tools for sensor fusion . . . . .	12
3.2	The main sensor fusion offers on the market . . . . .	12
3.3	Examples of non mainstream approaches . . . . .	13
<b>4</b>	<b>Typically used Algorithms</b>	<b>15</b>
4.1	Standard Algorithms . . . . .	15
4.2	Learning Algorithms . . . . .	16
<b>5</b>	<b>Comparison</b>	<b>17</b>
5.1	Approach comparison . . . . .	17
5.2	Processor comparison . . . . .	18
<b>6</b>	<b>Conclusion</b>	<b>19</b>
	<b>Bibliography</b>	<b>23</b>





# 1 Introduction

The idea of sensor fusion and its implementations fascinates scientists since many decades. That data from multiple sensors and even from different sensor types evaluated together reveals a lot more information about surroundings than one single sensor, is intuitive. Every smart existence uses multiple biological sensors combined on a situation dependent weighted base to get an impression of the world around itself. A lot of neuroscientists and bioengineers tried to adapt the computational power of our brain, applying sensor fusion on all our senses, to the computer driven technical world, to mimic these capabilities and get a sort of context awareness via fusion of the available sensors to the electronic machines. An early stage and the basics are well described in [STL<sup>+</sup>99].

Sensors and chips are getting smaller and smaller over time and gain therefore more and more mobility. For chips Moore's Law is a good approximation of their shrinkage at constant processing power. One projecting force behind the sensor miniaturization is the development in the MEMs technology [Inc11, Sca12] featuring mechanical sensors manufactured directly on silicon chips. Both trends together lead to smaller System on Chip (SoC) solutions including often both, a MCU and sensors in one package. This is a further driver of miniaturization in mobile devices.

The research goals of this advanced seminar were targeted in the following manner: The current sensor fusion processors on the market are the first point of interest. Distributors and teardown videos/manuals revealed fusion processors used in smart phones / wearables. They lead to their typical applications, Application Notes and Technical Reports. These documents include information of the used regular and intelligent/learning algorithms. In the end the listed processors are compared in the manner of complexity, precision, robustness, flexibility and efficiency.



## 2 Typical Applications for Sensor Fusion in Mobile Devices

A way to classify sensor fusion processes due to their complexity is the JDL (Joint Directors of Laboratories) data fusion process model as introduced in [HM04, p. 37]. The model includes five Levels where the first three levels apply to the complexity of the fusion process and the last two levels describe optimizations and refinements on the algorithms. Level one describes a basic data fusion, level three a complex system able to predict the impact of the measurements and perceive multiple states. The following applications are characterized in the manner of the required sensors, their complexity and possible/mostly implemented realization.

### 2.1 Motion Recognition

The most spread and established sensor fusion application in mobile devices and wearables is the motion recognition. Mostly three types of sensors are used, accelerometer, gyroscope and magnetometer. Often the marketing advertises their product with nine axis motion detection functionalities, i.e. are nine degrees of freedom belonging to the three different types of three axis sensors. The sensor data is normally fused with an 'Extended Kalman Filter' (EKF) to get (full 3D) motion information of the device. In the JDL model the motion recognition is a Level 1 processing (object refinement), and allows a relatively easy and unique implementation in both software and hardware. Standard approaches use a Cortex M0 or M4 microcontroller [FS15] and run the sensor fusion and pattern recognition software on the Cortex core using a lot of instructions and therefore time and power. To reduce both, the next step is to use a Digital Signal Processor (DSP) for the sensor fusion algorithm. This is a flexible method compared by the power consumption. The Cortex M4 offers an inbuilt DSP unit. To further reduce the power consumption, a coprocessor [AD15, MT15, PS15, Aud14b, Aud15], implementing the whole sensor fusion part in dedicated hardware, can be used. This system leads to the lowest power consumption but reduces flexibility and increases costs, because an extra chip and its development is needed. This effect can be neglected when the lower power consumption allows the usage of a cheaper battery. If a low power and always-on implementation is required, but because of the missing flexibility a dedicated ASIC is not practicable, a low power FPGA [LS15] might be the solution. The disadvantage is the higher power consumption in idle mode and the higher development

effort for the FPGA configuration. Motion recognition is used for years in image stabilization algorithms in cameras or orientation detection (portrait/landscape) of devices.

## 2.2 Pattern recognition

The JDL Level 2 situation refinement process, pattern recognition, applied on the (3D) motion data extends the possible output information to recognizable states and events. If the correct motion pattern or gesture is detected it triggers predefined action in the running application. A typically used gesture is the double tab on the smart phone screen to wake up the device, without the necessity of a constantly activated touchscreen, to save power. Another typical gesture is the flipping of the device to mute the played music or the ring tone. An in hardware and software implementations easily detectable pattern is a step, as usually used by fitness trackers and pedometers. Other common uses of detected patterns in motion data are free fall detection to save hard-disk drive data via controlled shutdown and shock recording. An overview of pattern recognition procedures is described in [MD15].

## 2.3 Healthcare

Wearables tracking our step count, our heart rate, the fat content of our body, the burned calories, and so on are spreading, especially in the sportive market. They include a motion recognition module, body electrodes for the body fat measurement and a light reflectivity sensor for the heart rate measurement over changes in the dermal structure. In medical devices more data is gathered over chemical sensors measuring blood values, i.e. insuline level etc. Together the gathered data allows predictions of the walked distance, the burned calories and the humans state of health. In case of a detected injury or abnormal heart rate a wearable connected to a smart phone is able to apply a rescue call. This is especially helpful for elderly. Most of the functions used by health care devices especially for sports use Level 1 processing. In rare cases, for example medical state surveillance the complexity reaches up to JDL Level 3 processing (thread refinement/impact assessment).

## 2.4 Indoor/GPSless Positioning (Pedestrian Navigation)

Indoor positioning is a further processed motion recognition using Level 2 processing (situation refinement). It requires an initial position and in best case a map of the building. From the initial position and the fused motion information the new position can be calculated, this process is also called 'dead reckoning' [Pat14]. The longer the system has no clear position reference like GPS the less exact the actual

calculated position is. But together with re-referencing the actual position via indoor installed WLAN or Bluetooth beacons or acoustic sources with known positions, the device position can be estimated with a high resolution and correctness.

## 2.5 Object recognition

A standard use case for camera object recognition in mobile devices are the smart-stay, -scroll and -pause functions implemented in Samsung smart screen, where the front camera is used to track the eyes of the user and trigger actions dependent on where he is looking on the screen. Another example are QR code scanners using the cameras video stream and algorithms to analyze the picture. It is again a Level 2 processing. These functions run on the application processor and require a lot of computational power and are therefore applied with a low repetition rate with only a few times per second. The current usage of object recognition do not require always-on functionality and because of the complexity of image processing algorithms and the required flexibility it has not yet been implemented in dedicated hardware. This could change in the future because of intelligent data glasses like the 'Google Glass', where an always-on image processing can make sense to show augmented reality elements. At the moment the 'Google Glass' for example uses an OMAP4430 Applications Processor manufactured by Texas Instruments, which offers nearly all the components a mobile device needs in just one package. This is necessary to implement these functionalities in such a condensed space as offered by glasses. There are a lot of FPGA implementations [JGB04] for image processing tasks beside the mobile devices segment, which are the first step to an dedicated ASIC for image processing tasks.

## 2.6 Voice Control

Voice control is a JDL Model Level 2 process. It normally requires an always on Key Word (pattern) recognition based on continuous microphone data, because targeted devices like smart phone models normally do not have an extra activation key for voice recognition, which makes it really power consuming running on a standard microcontroller or application processor. The power saving approach is to have an activation keyword recognition built up in dedicated hardware, waking up the processor or coprocessor, who is processing the rest of the verbal message and gets to sleep mode again afterwards [Aud14a, Aud15]. Mostly more than one microphone is used to recognize and suppress surrounding noise, or to estimate where the noise source or speaker is localized relative to the device.

## 2.7 Contextual Awareness

The most interesting application for app developers is the contextual awareness. The main issue of this fusion application is the state recognition of the device, for example is it in use, lying at the table or in the pocket. Furthermore information about the owner are available like if he or she is lying, sitting, walking, running, using an elevator, driving by car, bus or train, etc. This gives the application developers the possibility to react on the actual situation the phone or owner is in. This feature is only implementable with an JDL Model Level 3 process (thread refinement/impact assessment) and requires all the available sensors in current mobile devices such as the ones used for motion recognition, plus light sensor and audio background microphone. For this application an interesting chip is the [Aud15] because it implements the motion and audio processing in one single chip using computational intelligent systems. Another approach are sensor fusion hubs, like the [Qui15] supporting up to 12 sensors, which are programmable to nearly every sensor fusion task at a minimized power consumption, or complex low power FPGA implementations [LS15, STH<sup>+</sup>10]. An implementation of context awareness in a personal navigation system is described in [SMES14].

## 3 Fusion Processors on the Market

To get an impression of the market supply with sensor fusion chips, the main distributor homepages like

- DigiKey (<http://www.digikey.com/>)
- EBV Electronics (<http://www.ebv.com>)
- Farnell (<http://www.farnell.com/>)
- Mouser (<http://www.mouser.com>)

were researched. This lead at a first glance to the following manufacturers and products:

### **(Low power) Microcontollers:**

- Freescale (MK64FN1M0CAJ12R, Kinetis K64 Family) [FS14]
- NXP Semiconductors (LPC54102) [Sem15]
- Texas Instruments (MSP430F5xx, MSP432P401x) [MTR12, TI15]
- Toshiba (TZ1001MBG) [Tos15]

### **Coprocessors/System on Chip (SoC)/System in Package (SiP):**

- Audience (MQ100, eS800, N100) [Aud14b, Aud14a, Aud15]
- Freescale (MMA955xL) [FS15]
- Microchip (MM7150, SSC7102) [MT15]
- PNI Sensor Corporation (Sentral Motion Coprocessor) [PS15]
- Sensoplex (SP-10C, SP-M310) [Sen14b, Sen14a]
- ST (LSM6DS3) [STM15]

### **Sensorhub/FPGA:**

- Lattice (low power FPGA)[LS14, LS15]

- QuickLogic (ArcticLink 3 S2) [Qui15]

#### **PCB Module / System on Platform (SoP) / System on Board (SoB):**

- AnalogDevices (ADIS16480) [AD15]
- ST (iNemo) [STM13]

#### **Software Library:**

- Kionix (Sensor Fusion Library for Android)[KIO15]

This list covers the different technical realizations and gives a overview of each approach supplied on the market. Most of the listed products target the motion fusion. The audience products also apply acoustic sensor fusion. And the sensorhub/FPGA can be configured to the developers needs.

### **3.1 Development platforms and tools for sensor fusion**

Because of the boom in the wearable market many manufacturers like TI, Freescale, ST, NXP and others provide a sensor fusion library for their (low power) MCUs. A few Manufacturers go even further and supply a wearables development kit for sensor fusion like Toshiba [Dro15, Tos15] or Freescale (Xtrinsic Sensors).

Also the manufacturers of the main operation systems for mobile devices, Android and Windows Phone, supply software libraries for sensor fusion [Mic15, And15] then called virtual sensors [Pen12] using the power consuming application processor for the fusion calculations<sup>1</sup>.

### **3.2 The main sensor fusion offers on the market**

The main sensor fusion chips offered on the market are motion recognition platforms, mostly low power microcontrollers with DSP capabilities (Cortex M4, or Cortex M0). The basic sensor data preprocessing (low- or high pass filtering) and most parts of the fusion algorithm (in the motion recognition case mostly the EKF) are calculated on the DSP. The rest is running on the main core. Because of the used high performance cores the fusion algorithm can be calculated very fast and the chip can rest most of the time in sleep mode between two sample time stamps, saving up on energy. Most manufacturers offer a sensor fusion library to use with their core. At the moment motion coprocessors, with integrated MEMs sensors in the same chip and always-on functionality, are continually growing on the market, because

<sup>1</sup>For Android there are multiple sensor fusion test applications available in the Google Play Store (search for: 'sensor fusion site:play.google.com'), for Windows Phone the google search: 'sensor fusion site:windowsphone.com/de-de/store' resulted in just one match (04 July 2015)



the developer does not have to care about the sensor fusion difficulties and can save space on the PCB, as he has one chip including most sensors and sensor fusion functionality, offering continuous fused sensor data or perceived events to the application processor.

### 3.3 Examples of non mainstream approaches

The recent developments for always-on functionality with even lower power consumption are leading to more functionality embedded directly in the silicon. Processors are built together with specialized modules to precompute for example the rotational matrix from the sensor data. Voice recognition setups have direct keyword recognition built in silicon. These approaches often use human inspired neural networks to detect the keyword, so they can be trained to different speakers and voice patterns. On the one hand this can be done by special chips, designed to support neural networks, with a lot of parallel adders and multipliers for each virtual neuron. On the other hand it can be realized with the DSP functionality, for example multiplying and adding in one instruction, of the Cortex M4 core or a dedicated DSP, who also gives the possibility for more advanced efficient calculations, like neural network implementations 4.2. Especially free configurable sensorhubs often rely on an FPGAs, which work highly parallel and are free configurable, only limited by the amount of logic cells. There are many papers about implementing neural networks or Kalman filter logic in an FPGA[OR06], because an FPGA implementation is normally the first development stage of the final ASIC.



## 4 Typically used Algorithms

Sensor fusion relies nearly always on the Bayes' rule and is mostly implemented via probabilistic grids. The details would exceed the extent of this paper and can be looked up in [SO08, 25.1]. Most data fusion filters rely on these basics.

The newest trends on data fusion are yearly discussed and presented at the Fusion Conference organised by the International Society of Information Fusion<sup>1</sup>.

### 4.1 Standard Algorithms

The Kalman filter and its variants are a common method to combine multiple sensor inputs on a weighted base.

All Kalman filters rely on two stages, prediction and update. In the prediction phase the Kalman filter calculates the next state of the system based on the last state and the model that describes the change of the system. For example from the last position and last known speed and acceleration the new position is calculated. In the update phase the fresh sampled values of the sensors are compared to the predicted state. The closer the sensor data matches the prediction the lower the error probability of the sensor is estimated. From the error probability the Kalman gain is calculated and multiplied with this gain, the sensor value is added to the new state. For example, the GPS Sensor predicts a position way off what the physical model of movement with the speed and acceleration would predict. Then the gain for the GPS is very low and has only a weak influence on the new position.

The Kalman filter is used for linear sensor data input with a Gaussian distributed noise and measurement predictions.

The extended Kalman filter enhances the capabilities of the Kalman filter to unlinear systems by applying a local linearization to the nonlinear model and is the most used filter in products on the market from section 3.

The Information filter has the same prediction-update structure as the Kalman filter. Rather than generating state estimates and covariances it uses information state variables and information matrices.

Particle filters are useful, when you do not have any reference, starting point or a non-linear system of higher order, because they take a distributed field of weighted measurements and combine them statistically, but this requires more computational effort than the Kalman filters. They are also called Monte Carlo filters. Monte Carlo

---

<sup>1</sup>The publications can be found at <http://www.isif.org/content/conference-proceedings> (last called 07.07.2015)

(MC) filter methods describe probability distributions as a set of weighted samples (short: particles) of an underlying state space. MC filtering then uses these samples to simulate probabilistic inference usually through Bayes' rule. Many samples or simulations are performed. By studying the statistics of these samples as they progress through the inference process, a probabilistic picture of the process being simulated can be built up. Monte Carlo (MC) methods are well suited to problems, where state transition models and observation models are highly non-linear[SO08, 25.1].

The Unscented Kalman filter is a compromise between the low computational effort of the Kalman filter and high non-linearity capabilities of the particle filter.

A comparison of the different approaches for the implementation of a Kalman filter regarding the computational complexity can be found at [HA13].

## 4.2 Learning Algorithms

Machine learning supports multiple learning procedures and algorithmic attempts, such as decision tree learning, association rule learning, Bayesian networks, clustering, support vector machines, genetic algorithms, artificial neural networks and others.

Advanced fusion applications like contextual awareness, or speech recognition have to adapt to changes in the surroundings such as a new user with a different voice, or different behavior in his daily schedule and a new environment. This behavior can be realized with neural networks, if the correct model of the setting is known. The rarely used fuzzy logic can be practicable for the state driven categorization, like walking, standing or riding, because it estimates which state is matching to a percentage, and does not only set states in comparison to the real state as true or false.

A neural network consists of virtual neurons with many weighted inputs coming from other neurons or sensors and one graduated output. These inputs are added after the multiplication with the input weights, the result is displayed on the output. This operation can also be computed with a single instruction by some DSPs. The weights change due to the learning process during the multiple iterations. The output then is connected to other neurons or actuators.

Together the two methods, artificial neural network and fuzzy logic, create algorithms processing data nearly humanly.

There are many research approaches with FPGAs addressing neural network implementations [OR06, WW15], because of the high grade of parallelism in those structures ideally suited for FPGAs.

## 5 Comparison

### 5.1 Approach comparison

Depending on the sensor fusion application, how often and continuous the feature is used, how advanced the data fusion, pattern or state recognition is, it makes sense to implement the whole fusion functionality in hardware, in flexible FPGA logic, or to use a DSP or even a normal processor core to run the fusion algorithms.

A typical candidate to be implemented directly in silicon is a linear resolvable fusion algorithm with the requirement of an always-on functionality, as motion recognition. In contrast camera based object recognition is only needed in rare situations, is highly task dependent, complex and therefore normally implemented with algorithms on the application processor.

A user is not able to distinguish between a functional implementation in silicon or a software application on a processor or split between a processor and a coprocessor. But the user could perceive the side effects, for example the different power consumption of the various approaches, or small differences in the responsiveness and reaction time and precision.

For the developers and the manufacturers every approach has specific advantage and disadvantage in chip and development cost, flexibility, reusability, performance and precision.

If a chip manufacturer starts a new development project for a smart phone application processor with a dedicated sensor fusion processor or an sensor fusion coprocessor, it is necessary to support most of the sensors on the market. A significant risk for the finished chip is the potential incompatibility with new developments in MEMs technology leading to better sensors with other main characteristics.

In contrast software implementation and algorithms can be amend by firmware updates with a high flexibility and are less expensive in the initial development. But in comparison software implementation cannot achieve the same efficiency, low power standards and data throughput as dedicated hardware or FPGA implementation.

These days the chip manufacturers often provides software libraries and supports inbuilt preprocessors and peripherals to enable a short time to market for the device developers.

This is a notable advantage for the developer, who can use the sensor fusion library of a MCU, or the API of a coprocessor. But for the manufacturers it takes a lot effort to customize the firmware for a FPGA to wanted fusion tasks or to implement own fusion filter libraries.

## 5.2 Processor comparison

Because of the missing offers for more sophisticated sensor fusion functionalities in dedicated hardware, the comparison focuses on the motion recognition implementations in hardware and in software.

The following tabular 5.2 gives an overview on the different power consumption of the manufacturers.

Manufacturer	Part	Type	V <sub>DD</sub>	I <sub>sleep</sub>	I <sub>active</sub>
TI Freescale	MSP432P401x	MCU	3.0 V	25 nA	4.3 mA@48 MHz
	MK64FN1M0-CAJ12R	MCU	3.0 V	0.84 $\mu$ A	46 mA@120 MHz
Toshiba	TZ1001MBG	MCU	3.3 V	2.9 $\mu$ A	3.1 mA@48 MHz
Microchip	SSC7150	CoPr	3.3 V	70 $\mu$ A	7.65 mA
PNI Sensor Corp.	SENtral Motion	CoPr	1.8 V	7 $\mu$ A	300 $\mu$ A
Audience	MQ100 Motion	CoPr	-	200 $\mu$ W	4.5 mW
QuickLogic	ArcticLink 3 S2	FPGA	1.1 V	-	68 $\mu$ A
Lattice	iCE40 Ultra	FPGA	1.2 V	-	71 $\mu$ A
ST	LSM6DS3	SoC	1.8 V	6 $\mu$ A	1.25 mA
Analog Devices	ADIS16480	PCB	3.3 V	45 $\mu$ A	254 mA

Table 5.1: Power comparison at room temperature (25 °)

It is visible that dedicated motion coprocessors have a low power consumption in active state and the low power FPGA has a appreciable performance, but the power consumption of the regular MCUs in sleep mode is lower than the ones of the coprocessors.

Therefore coprocessors make sense if they are used in an always-on mode. If a sample every few millisecond is needed it might require less power to use a regular MCU, which is most of the time in sleep mode.

An FPGA is a practicable alternative to the coprocessor, because it can be reconfigured to different sensor fusion tasks depending on the current need of the running application on the application processor or to adapt newly developed or improved sensors in the fusion algorithms.

Low power FPGAs have a even lower powerconsumption than the coprocessors. But the reconfiguration and initialization at bootup drains several mA of current.

## 6 Conclusion

For low level sensor fusion applications like motion recognition, dedicated hardware implementations and coprocessors get a increasing relevance in the mobile devices market, because of their greater performance and always-on capability, accompanied with the reduced power consumption and simple usage for developers. Also the small space requirements of chips including most of the sensors and the fusion logic in one package are supporting the trend in mobile devices.

More sophisticated or personalized fusion functionality is still implemented on microcontrollers or application processors, sometimes accompanied by a DSP, because the high initial costs of the development of an dedicated ASFI would not be worthwhile for a rarely used and very specialized function. A algorithm implementation allowing great sleep mode times for the MCU is the best power saving possibility here.

Still FPGA based implementations are the most flexible approach, ranging also in the power budget level of the coprocessors, allowing always-on implementations, but require more development work to create and maintain the firmware defining the hardware functionality and software running on a FPGA embedded processing core. Nevertheless low power FPGAs were implemented in many wearables like the Pebble Time Smartwatch for advanced sensor fusion tasks.

Approaches regarding learning algorithms and neural networks are mainly applied in changing environments and highly sophisticated tasks as voice recognition and contextual awareness. Their implementation ranges between FPGA, Sensorhubs and MCUs with DSPs, because of the still upcoming improvements preventing a profitable, not immediately outdated, dedicated hardware implementation.

The object recognition implementation manner may change from software algorithms to low power FPGAs or even to dedicated chips, if the data glasses with their augmented reality applications get to the main consumer market. Augmented reality in general will provide changes, regarding the computational complexity, flexibility, efficiency and power consumption of mobile devices, if it will be implemented.





## List of Abbreviations

API	Application Programming Interface
ASIC	Application-Specific Integrated Circuit
DSP	Digital Signal Processor
EKF	Extended Kalman Filter
FPGA	Field Programmable Gate Array
GPS	Global Positioning System
JDL	Joint Directors of Laboratories
MCU	Microcontroller Unit
MEMS	Micro-Electro-Mechanical System
PCB	Printed Circuit Board
QR code	Quick Response Code
SoB	System on Board
SoC	System on Chip
SoP	System on Platform
SiP	System in Package



## Bibliography

- [AD15] Inc. Analog Devices. Ten degrees of freedom inertial sensor with dynamic orientation outputs. Web: <http://www.analog.com/media/en/technical-documentation/data-sheets/ADIS16480.pdf> last called 28.06.2015, 2015. Rev. E; More: <http://www.analog.com/en/products/sensors/isensor-mems-inertial-measurement-units/adis16480.html#product-overview> last called 28.06.2015.
- [And15] Inc. Android. Sensors overview. Web: [http://developer.android.com/guide/topics/sensors/sensors\\_overview.html](http://developer.android.com/guide/topics/sensors/sensors_overview.html) last called 03.07.2015, 2015.
- [Aud14a] Inc. Audience. Audience es800 series – industry leading advanced voice & smart audio codecs designed for challenging environments and new devices. Web: [http://audience.com/images/AUD\\_eS800\\_ProdBrief\\_110414.pdf](http://audience.com/images/AUD_eS800_ProdBrief_110414.pdf) last called 20.06.2015, 2014. More: <http://audience.com/multisensory-processors> last called 20.06.2015.
- [Aud14b] Inc. Audience. Mq100 – motion processor, motionq – featuring low-power, always-on sensing. Web: [http://audience.com/images/Audience\\_MQ100\\_ProdBrief.pdf](http://audience.com/images/Audience_MQ100_ProdBrief.pdf) last called 20.06.2015, February 2014. Website: <http://audience.com/multisensory-processors/mq-100> last called 20.06.2015.
- [Aud15] Inc. Audience. The n100 multisensory processor – fusing voice and motion to create natural user experiences inspired by neuroscience. Web: [http://audience.com/images/AUD\\_N100\\_ProdBrief\\_022515.pdf](http://audience.com/images/AUD_N100_ProdBrief_022515.pdf) last called 20.06.2015, 2015. More: <http://audience.com/multisensory-processors> last called 20.06.2015.
- [Dro15] Stefan Drouzas. Referenz-kit fuer sportuhren. *Elektronik*, (10):26–28, may 2015.
- [FS14] Inc Freescale Semiconductor. Kinetis k64 sub-family data sheet with 1 mb flash. Web: [http://cache.freescale.com/files/microcontrollers/doc/data\\_sheet/K64P142M120SF5.pdf](http://cache.freescale.com/files/microcontrollers/doc/data_sheet/K64P142M120SF5.pdf) last called 04.07.2015, September 2014. Rev. 5 More: [http://www.freescale.com/webapp/sps/site/prod\\_summary.jsp?code=K64\\_120](http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=K64_120) last called 04.07.2015.

- [FS15] Inc Freescale Semiconductor. Mma955xl intelligent motion-sensing platform. Web: [http://cache.freescale.com/files/sensors/doc/data\\_sheet/MMA955xL.pdf](http://cache.freescale.com/files/sensors/doc/data_sheet/MMA955xL.pdf) last called 27.06.2015, May 2015. Rev. 3.1 More: <http://www.freescale.com/webapp/sps/site/overview.jsp?code=XTRSICSNSTLBOX> last called 27.06.2015.
- [HA13] Sayed Amir Hoseini and Mohammad Reza Ashraf. Computational complexity comparison of multi-sensor single target data fusion methods by matlab. *International Journal of Chaos, Control, Modelling and Simulation*, 2(2):8, June 2013. IJCCMS.
- [HM04] David L. Hall and Sonya A. H. McMullen. *Mathematical Techniques in Multisensor Data Fusion*. ARTECH HOUSE, INC, 685 Canton Street, Norwood, MA 02062, second edition, 2004.
- [Inc11] IHS Inc. Mems sensor fusion to spawn important new developments in centralized processing. Web: <https://technology.ihs.com/394883/mems-sensor-fusion-to-spawn-important-new-developments\discretionary{-}{-}{-}in-centralized-processing> last called 20.06.2015, November 2011.
- [JGB04] CT Johnston, KT Gribbon, and DG Bailey. Implementing image processing algorithms on fpgas. In *Proceedings of the Eleventh Electronics New Zealand Conference, ENZCon'04*, pages 118–123, 2004. Web: [https://www.researchgate.net/publication/228914467\\_Implementing\\_image\\_processing\\_algorithms\\_on\\_FPGAs](https://www.researchgate.net/publication/228914467_Implementing_image_processing_algorithms_on_FPGAs) last called 06.07.2015.
- [KIO15] Inc. KIONIX. Sensor fusion. Web: <http://www.kionix.com/sensor-fusion> last called 03.07.2015, 2015.
- [LS14] Corp. Lattice Semiconductor. ice40lm family data sheet introduction. Web: <http://www.latticesemi.com/~media/LatticeSemi/Documents/DataSheets/iCE/DS1045.pdf> last called 03.07.2015, January 2014. DS1045 Version 1.5; More: <http://www.latticesemi.com/iCE40> last called 03.07.2015.
- [LS15] Corp. Lattice Semiconductor. ice40 ultra family data sheet. Web: <http://www.latticesemi.com/~media/LatticeSemi/Documents/DataSheets/iCE/iCE40UltraFamilyDataSheet.pdf> last called 03.07.2015, June 2015. DS1048 Version 1.8; More: <http://www.latticesemi.com/iCE40Ultra> last called 03.07.2015.
- [MD15] M Narasimha Murty and V Susheela Devi. Introduction to pattern recognition and machine learning. 5, 2015.

- [Mic15] Inc. Microsoft. Sensor fusion implementation details. Web: [https://dev.windowsphone.com/en-US/OEM/docs/Driver\\_Components/Sensor\\_fusion\\_implementation\\_details](https://dev.windowsphone.com/en-US/OEM/docs/Driver_Components/Sensor_fusion_implementation_details) last called 03.07.2015, May 2015.
- [MT15] Inc. Microchip Technology. Motion coprocessor. Web: <http://ww1.microchip.com/downloads/en/DeviceDoc/00001885A.pdf> last called 27.06.2015, January 2015. DS00001885A; More: <http://www.microchip.com/wwwProducts/Devices.aspx?product=MM7150> last called 27.06.2015.
- [MTR12] Erick Macias, Daniel Torres, and Sourabh Ravindran. *Nine-Axis Sensor Fusion Using the Direction Cosine Matrix Algorithm on the MSP430F5xx Family*. Texas Instruments Incorporated, slaa518a edition, February 2012. Web: <http://www.ti.com/lit/an/slaa518a/slaa518a.pdf> last called 04.07.2015.
- [OR06] Amos R. Omondi and Jagath Chandana Rajapakse. *FPGA implementations of neural networks*, volume 365. Springer, 2006.
- [Pat14] Saumil Patel. Multi-sensor fusion for pedestrian localization. In *SEMINAR ON TOPICS IN SIGNAL PROCESSING*, page 65, 2014.
- [Pen12] James Steele Penton. Understanding virtual sensors: From sensor fusion to context-aware applications. Web: <http://electronicdesign.com/ios/understanding-virtual-sensors-sensor-fusion\discretionary{-}{-}{-}context-aware-applications> last called 03.07.2015, July 2012.
- [PS15] Corporation PNI Sensor. For unparalleled sensor fusion that consumes only 1% as much power as other processors. Web: <http://www.pnicorp.com/download/458/422/SENtralMotionCoprocesor.pdf> last called 20.06.2015, 2015. Website: <http://www.pnicorp.com/products/sentral> last called 20.06.2015.
- [Qui15] Corp. QuickLogic. Quicklogic arcticlink 3 s2 solution platform brief. Web: <http://www.quicklogic.com/assets/pdf/solution-platform-briefs/ArcticLink-3-S2-SPB.pdf> last called 20.06.2015, February 2015. Website: <http://www.quicklogic.com/platforms/sensor-hub/al3s2/> last called 20.06.2015.
- [Sca12] Bob Scannell. Mems enable medical innovation. Technical Article MS-2393, Analog Devices, Inc., <http://www.analog.com/media/en/technical-documentation/technical-articles/MEMS-Enable-Medical-Innovation-MS-2393.pdf>, November 2012. TA11147-0-11/12.

- [Sem15] NXP Semiconductors. An11703; lpc5410x sensor processing - motion solution. Web: [http://www.nxp.com/documents/application\\_note/AN11703.pdf](http://www.nxp.com/documents/application_note/AN11703.pdf) last called 04.07.2015, June 2015. Rev. 1.0; More: <http://www.nxp.com/demoboard/OM13078.html> last called 04.07.2015.
- [Sen14a] Inc. Sensoplex. 10 - axis sensor module with bluetooth le and optional ant+ link. Web: [http://www.sensoplex.com/wp-content/uploads/2013/10/SP-M310-Brief-Rev\\_a.pdf](http://www.sensoplex.com/wp-content/uploads/2013/10/SP-M310-Brief-Rev_a.pdf) last called 04.07.2015, 2014. More: <http://www.sensoplex.com>.
- [Sen14b] Inc. Sensoplex. 10 - axis sensor module with bluetooth le link. Web: <http://www.sensoplex.com/SP-10C.pdf> last called 04.07.2015, 2014. Rev F 20140311; More: <http://www.sensoplex.com>.
- [SMES14] Sara Saeedi, Adel Moussa, and Naser El-Sheimy. Context-aware personal navigation using embedded sensor fusion in smartphones. *Sensors*, 14(4):5742–5767, 2014.
- [SO08] Bruno Siciliano and (Eds.) Khatib Oussama. *Springer Handbook of Robotics*. Springer-Verlag Berlin Heidelberg, 2008. Part C/25 [http://www.springer.com/cda/content/document/cda\\_downloaddocument/9783540239574-c2.pdf](http://www.springer.com/cda/content/document/cda_downloaddocument/9783540239574-c2.pdf).
- [STH<sup>+</sup>10] Filippo Sironi, Marco Triverio, Henry Hoffmann, Martina Maggio, and Marco D Santambrogio. Self-aware adaptation in fpga-based systems. In *Field Programmable Logic and Applications (FPL), 2010 International Conference on*, pages 187–192. IEEE, 2010.
- [STL<sup>+</sup>99] Adrian Stoican, Tyson Thomas, Wei-Te Li, Taher Daud, and James Fabunmi. Extended logic processing system for sensor fusion. ISIF VP Communications, 1999. More Conference Papers at [www.isif.org](http://www.isif.org). This Conferencepaper: <http://www.isif.org/fusion/proceedings/fusion99CD/C-163.pdf> last called 20.06.2015, Fuzzy Logic and Neural Networks for Sensor Fusion.
- [STM13] Inc. STMicroelectronics. Inemo-m1; inemo system-on-board. Web: <http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/DM00056715.pdf> last called 04.07.2015, October 2013. DocID023268 Rev 1.
- [STM15] Inc. STMicroelectronics. Lsm6ds3; inemo inertial module: always-on 3d accelerometer and 3d gyroscope. Web: <http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/DM00133076.pdf> last called 04.07.2015, May 2015. DocID026899 Rev 5; More: <http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/DM00133076.pdf>

- [//www.st.com/web/en/catalog/sense\\_power/FM89/SC1448/PF261181](http://www.st.com/web/en/catalog/sense_power/FM89/SC1448/PF261181)  
last called 04.07.2015.
- [TI15] Incorporated Texas Instruments. Msp432p401x mixed-signal microcontrollers. Web: <http://www.ti.com/lit/ds/slas826a/slas826a.pdf> last called 05.07.2015, March 2015. SLAS826A; Website: [http://www.ti.com/llds/ti/microcontrollers\\_16-bit\\_32-bit/msp/low\\_power\\_performance/msp432p4x/overview.page](http://www.ti.com/llds/ti/microcontrollers_16-bit_32-bit/msp/low_power_performance/msp432p4x/overview.page) last called 05.07.2015.
- [Tos15] Corporation Toshiba. Application processor lite app lite tz1001mbg. Web: <http://toshiba.semicon-storage.com/info/docget.jsp?did=29868&prodName=TZ1001MBG> last called 04.07.2015, June 2015. Rev. 1.3.
- [WW15] Tong WANG and Lianming WANG. A modularization hardware implementation approach for artificial neural network. *2nd International Conference on Electrical, Computer Engineering and Electronics*, 2015. ICECEE 2015.





## License

This work is licensed under the Creative Commons Attribution 3.0 Germany License. To view a copy of this license, visit <http://creativecommons.org> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.