

FPGA TABANLI GÖRSEL ATALET ODOMETRİ ALGORİTMASI İÇİN SENSÖR SİSTEMİ TASARIMI

Mustafa Ensar Işkın¹, Ramazan Yeniçeri²
İstanbul Teknik Üniversitesi, İstanbul

ÖZET

Otonom araçların sahip olması gereken en önemli bilgilerden birisi kendilerinin konum bilgileridir. Farklı navigasyon teknikleri kullanılarak otonom araçlar konum bilgilerini elde edebilir. Görsel Atalet Odometri (Visual Inertial Odometry), görsel ve atalet verilerini birlikte kullanarak konum tahmini yapabilmektedir. Bu projede Görsel Atalet Odometri algoritması için Zybo Z7-20 SoC kartı kullanılarak gerekli olan görsel ve ivme verilerinin toplanması amaçlanmıştır. Görsel veriyi elde etmek için PCAM 5C adlı kamera modülü kullanılmıştır. İvme verisi ise üzerinde 3 eksen ivmeölçer, 3 eksen gyroscope ve 3 eksen manyetometre bulunan Pmod NAV sensörü kullanılmıştır. İvmeölçer verisini iyileştirmek adına ayrıca gyroscope verisi de kaydedilmiştir. Gyroscope verisini kullanarak ivme verisinden yer çekimi etkisi çıkartılarak daha doğru sonuçlar elde edilebilir. Elde edilen IMU sensör verisi SD karta kaydedilmiş olup, görüntü verisi ise Zybo Z7-20 kartı üzerindeki HDMI çıkışından iletilmektedir.

GİRİŞ

Bildiğimiz üzere insansız araçlar için navigasyon çok önemlidir. Ancak yalnızca GPS kullanarak konum bilgisinin elde edilmesinin sebep olduğu bazı sıkıntılar vardır. Öncelikle GPS, Amerika Birleşik Devletleri Savunma Bakanlığına bağlı bir teknolojidir ve GPS alıcılarının uyması gereken bazı sınırlamalar vardır ["SPS performance standard – GPS", 2020]. Buna ek olarak GPS sinyallerindeki bozulmalar ya da GPS sinyalinin kaybı konum verisinin elde edilmesini imkânsız hale getirebilmektedir. Bu sebeple navigasyon problemi için farklı çözümler gerekmektedir. Navigasyon bilgisini elde etmek için GPS kullanmaktan farklı olan başka bir alternatif odometridir. Odometri tabanlı birden çok farklı yaklaşım mevcuttur. Bunlardan bazıları:

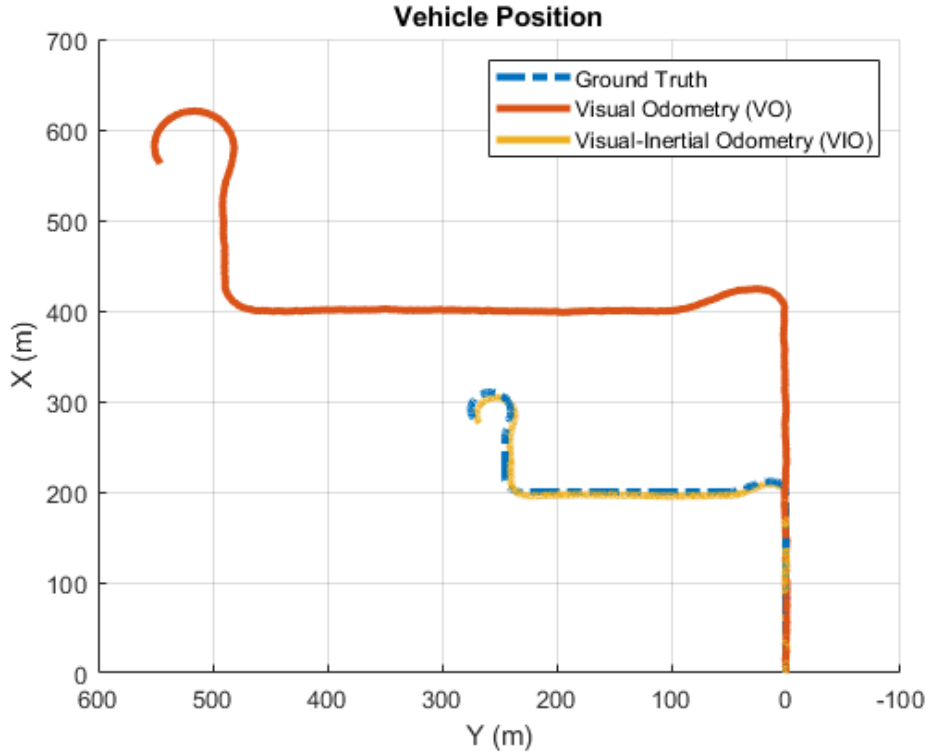
- Atalet odometri
- Görsel odometri
- Görsel atalet odometri
- Lazer odometri yaklaşımlarıdır [Sherif, 2019]

Atalet odometri pozisyon tahminini bir ivme sensöründen elde edilmiş olan ivme verileri ile yapmaya çalışır. Fakat ivme sensöründeki ölçüm hatalarından dolayı meydana gelen ve ivmeden konum verisi elde ederken ortaya çıkan integrasyon hatası bu yöntemin doğruluğunu sınırlamaktadır. Görsel odometri peş peşe iki görüntü karesi arasındaki farkları saptayarak

¹ Lisans Öğrencisi, Elektronik ve Haberleşme Müh. Böl. E-posta: iskin17@itu.edu.tr

² Dr. Öğretim Üyesi, Uçak Müh. Böl, E-posta: yenicerir@itu.edu.tr

pozisyon tahmini yapar. Fakat derinliğin algılanmadığı durumlarda ve uzun mesafelerde hatalar meydana gelmektedir [Sherif, 2019] [Jiang, 2010]. Görsel atalet odometri bu iki yaklaşımı birleştirerek çok daha doğru tahminler yapılmasına olanak sağlar. Şekil 1’de MATLAB’in resmi internet sitesinde paylaşılmış olan görsel odometri ve görsel atalet odometri sonucunda elde edilen sonuçlar gösterilmiştir [“Visual-Inertial Odometry Using Synthetic Data”, nd]. Görüldüğü üzere görsel atalet odometrisi gerçeğe çok daha yakın sonuçlar vermektedir.



Şekil 1: Görsel Atalet Odometri ve Görsel Odometri karşılaştırması

Bu çalışmada da amacımız görsel atalet odometri için bir sensör sistemi tasarlamaktır. Bu sensör sistemini Zybo Z7-20 SoC kartındaki hem PL hem PS kısmını kullanarak tasarlamayı amaçladık. SoC çipin PL kısmında görüntüyü elde edebilmek için video alıcı tasarımı yapılmıştır. Buna ek olarak PL kısmında SPI modülü yerleştirilmiş ve PMOD NAV verisi okunmuş ve FIFO'lara yazılmıştır. PS kısmında ise görüntü alıcısını başlatan ve FIFO'lara kaydedilmiş olan ivme ve gyroscope verisini SD karta yazan bir C kodu yazılmıştır.

SİSTEM TASARIMI

Geliştirilmiş olan sensör sistemi görüntü ve ivme verisini elde etmek amacıyla tasarlanmıştır. Görüntü verisini SD karta yazmak için gerekli olan yazma hızına erişilemediği için direkt olarak HDMI çıkışı üzerinden yollanmaktadır. Bunun yanı sıra IMU sensör verisi ise SD karta kaydedilmektedir. Bu kısımda ilk olarak kullandığımız donanımlar tanıtılacaktır.

Geliştirme Kartı

Geliştirme kartı olarak Zybo Z7-20 ARM&FPGA SoC Geliştirme Kartı kullanılmıştır. Bu kart üzerinde bulunan Zynq SoC hem donanım hem de yazılım tasarımı yapabilmemizi sağlamaktadır. Bu sayede kamera okuyucu tasarlanabilmiş ve SPI haberleşmesi ise yazılıma ek yük bindirmemiştir. Buna ek olarak Zybo Z7-20 üzerinde bulunan MIPI CSI-2 uyumlu konektör sayesinde ek donanımlar ile uğraşmadan direkt geliştirme kartına kamera takmamız mümkün olmuştur.

Kamera Modülü

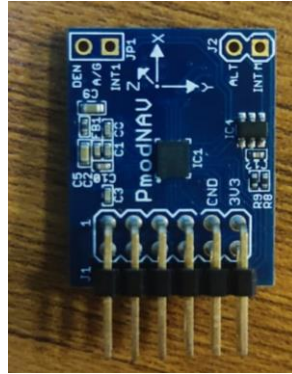
Kamera modülü olarak PCAM 5C adlı modül kullanılmıştır. Bu modülü kullanmamızdaki sebep Zybo Z7-20 üzerindeki MIPI CSI-2 konnektörü ile uyumlu olması ve bu iki donanımın birlikte sıkıntısız bir biçimde çalıştığını gösteren örnekler olmasıdır. Buna ek olarak PCAM 5C kamera modülü bizim elde etmek istediğimiz çözünürlük ve kare hızlarına çıkabiliyor. Şekil 2’de PCAM 5C kamera modülü gösterilmiştir.



Şekil 2: PCAM 5C Kamera Modülü

IMU Sensör

IMU sensör tercihimizi de hali hazırda elimizde bulunmasından ve Zybo Z7-20 üzerindeki Pmod konnektörleri ile uyumlu olan Pmod NAV yönünde kullandık. Pmod NAV saniyede 952 örnek almamıza olanak sağlıyor. Şekil 3’te Pmod Nav IMU sensör modülü gösterilmiştir.



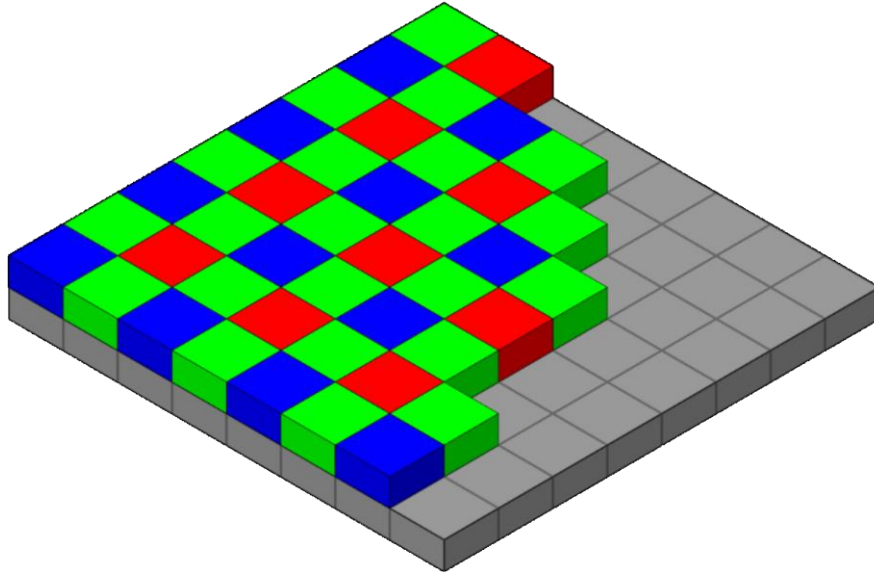
Şekil 3: Pmod NAV IMU Sensörü

Kamera Okuyucu Donanım Tasarımı

Pcam 5C MIPI CSI-2 arayüzü ile görüntü verisi aktarımını gerçekleştirmektedir. Fakat MIPI CSI-2 alıcısını sıfırdan tasarlamak çok uzun soluklu ve zor bir iştir bu sebeple kamera alıcı kısmı için hazır bir IP kullanımına karar verilmiştir. MIPI CSI-2 alıcı IP’si Digilent ve Xilinx tarafından ücretsiz olarak sunulmaktadır. Bu iki IP’nin amacı ve çalışma mantığı aynı olsa da aralarında belirgin performans ve kullanım farklılıkları vardır. Digilent tarafından paylaşılmış olan IP bazı zamanlama problemlerine sahip olmasını yanı sıra tasarımcıya çok az konfigürasyon şansı tanımaktadır. Xilinx tarafından paylaşılmış olan IP zamanlamada herhangi bir problem yaşamamaktadır ve istenilen sistemin tasarlanması için çok daha fazla seçenek sunmaktadır. Digilent ve Xilinx tarafından paylaşılmış olan MIPI CSI-2 alıcı IP’lerinin tasarımcıya sunduğu konfigürasyon seçenekleri sırayla Şekil 4 ve Şekil 5’te gösterilmiştir.

Kontrolcü kısmı: Kontrolcü kısmı Zynq içindeki A9 işlemci ve saat kontrolcülerinden oluşmaktadır. Yapılan tasarımda Zynq7 içindeki A9 işlemcilerden biri kullanılmakta olup baremetal bir uygulama gerçekleştirilmiştir. Kontrolcü kısmı, tasarımdaki diğer bileşenlere AXI ara yüzü ile bağlanmaktadır. Kontrolcünün işlevi diğer alt sistemlerin kontrolü ve bu alt sistemlerin saat ayarını yapmaktır. Ayrıca kontrolcü kısmı kullanıcı girişlerine göre alt sistemleri yeniden konfigüre etme yeteneğine sahiptir.

Görüntü alma kısmı: Görüntü alma kısmı aslında MIPI CSI-2 alıcıdan oluşmaktadır. Pcam 5C iki kanal MIPI hattına sahip olduğu için MIPI CSI-2 alıcısı 2 kanal olarak konfigüre edilmiştir. 2 kanalın anlamı kameradan gelen veri iki bağımsız kanaldan gelmektedir. 2 kanal kullanılarak daha yüksek FPS ve daha yüksek çözünürlükler çıkmak mümkün olmaktadır. Bu kısmın sonunda elde edilen görüntü RAW10 formatındadır. Şekil 7'de elde edilen görüntünün formatı gösterilmektedir ["Bayer Filter", 2021].



Şekil 7: Bayer (RAW) Formatı

Sinyal işleme kısmı: Görüntü alma kısmının çıkışı RAW10 formatındadır fakat çıkış aygıtımız olan monitörler RGB formatında görüntüler ile çalışmaktadır. RGB formatında her piksel kırmızı, yeşil ve mavi değerlerine sahiptir. RAW formatındaki veriyi RGB formatına dönüştürmek için Sensor Demosaic adlı IP kullanılmıştır. Bu IP çıkışındaki görüntü RGB formatındadır. Bu IP ardından görüntünün gama değerini ayarlamak için kontrolcü tarafından da kontrol edilebilen Gamma Correction adlı IP kullanılmıştır.

Veri depolama kısmı: Sinyal işleme kısmından sonra elde edilen görüntüler daha sonradan kontrolcü tarafından erişilebilmek üzere DDR RAM'e yazılmaktadır. Yazma işlemi VDMA modülü kullanılarak gerçekleştirilmiştir. VDMA modülü sayesinde kontrolcüye yük bindirmeden elde edilen görüntü DDR RAM'e yazılabilmektedir.

Video çıkış kısmı: Video çıkış kısmı DDR RAM'e yazılan kareleri sırayla okur ardından kendi zamanlayıcısını kullanarak görüntüyü HDMI çıkışına iletir.

IMU Sensör Okuyucu ve SD Karta Veri Kaydı

Bu sistemde kullanılan IMU sensörü olan Pmod Nav sensörü üzerindeki ivmeölçer ve gyroscope saniyede üç eksende 952 örnek verebilmektedir. MATLAB'te yapılan görsel atalet odometri simülasyonunda tahminin doğruluğunun ivme verisinin frekansı ile doğru orantılı olduğu gözlemlenmiştir. Bu sebeple IMU sensörü maksimum çalışma frekansında çalıştırılmıştır. Sensörden veri elde işlem SoC çipin PL kısmındaki SPI-master modülü ile gerçekleştirilmiştir. İlk olarak SPI protokolünü tanımakta fayda vardır. SPI protokolünde kullanılacak olan modülün CS girişi lojik 0'a çekilir ve ardından SCLK girişinden belli bir frekansta saat işareti yollar. Genellikle MOSI pininden yollanan ilk bit yazma mı yoksa okuma mı yapılacağını belirtir. Bu bitten sonra gelen 7 bit ise modülün register adresini belirtir. Yazma yapılacak ise yine MOSI pininden yazılacak olan değerler yollar, eğer okuma yapılacaksa modül veriyi MISO pininden bize iletir.

Pmod Nav'dan elde edilen veriler SPI-master modülünden sonra FIFO'lara yazılmaktadır. Ardından Zynq içindeki ARM A9 çekirdek AXI ara yüzü kullanarak FIFO içindeki verileri alarak SD karta paketler halinde yazılmaktadır. Yapılan testler sonucu SD karta yazılan paket boyutunun yazma hızını doğrudan etkilediği gözlemlenmiştir. Farklı paket boyutları ile ulaşılan yazma hızları Tablo-1'de verilmiştir. Bu veriler kullanılan SD kart ile değişebilmektedir.

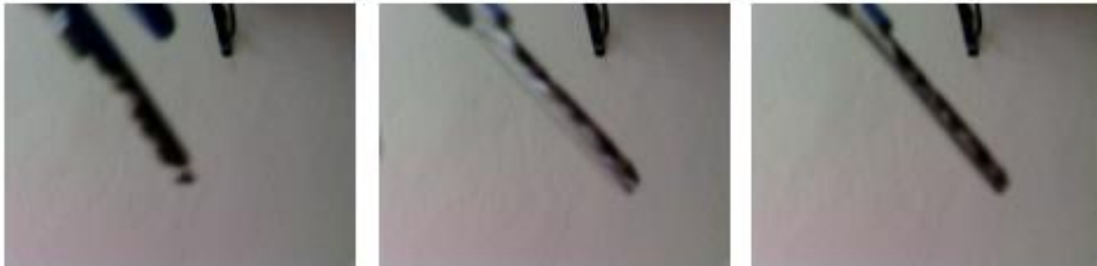
Paket Boyutu	Yazma Hızı
1KB	0.12MB/s
10KB	0.55MB/s
100KB	1.65MB/s
1MB	2.25MB/s
5MB	2.36MB/s
10MB	2.48MB/s

Tablo 1: SD Kart Paket Boyutları & Yazma Hızı

Üçer ekseninde elde edilen ivme ve gyroscope verilerinin her biri 2 Byte boyutundadır. Ayrıca her pakette 4 Byte boyutunda zaman bilgisi de bulunmaktadır. Ayrıca veriyi SD karta kaydederken hexadecimal formatında kaydettiğimiz için her Byte için 2 Byte boyutunda yazma yapmamız gerekmektedir. Buna ek olarak her veri arasında da bir adet boşluk bıraktığımız için her pakette ekstra 6 Byte yazmamız gerekmektedir ve satır sonuna da yeni satır yazmamız gerekmektedir. Sonuç olarak paket boyutu $2x(2x6 + 4) + 7 = 39$ byte olmaktadır. Saniyede 952 paket yazmayı hedeflediğimiz için ulaşmamız gereken yazma hızı $\frac{952x39}{1024x1024} = 0.0354$ MB/s olmaktadır. Tablo 1'de de görüldüğü gibi bu hıza erişmek 1KB'lık veri ile kolayca aşılmaktadır. Bu sebeple donanım tarafına koyduğumuz FIFO boyutu 1KB boyutundadır.

UYGULAMALAR VE DEĞERLENDİRME

Projenin son kısmında kamera alıcısı ve PMOD Nav okuyucu birleştirilmiş ve sistemin çalıştığı teyit edilmiştir. Daha önceden bahsettiğim gibi erişilen yazma hızları ile görüntüyü sd karta yazmak mümkün değildir. Düşük yazma hızının sebebinin kullanılan kütüphane olduğu ve baremetal kod yazarken başka bir alternatifin olmadığı araştırmalar sonucu elde edilen bilgiler arsındadır. Görüntü verisini SD karta yazmanın en kolay yolu ise baremetal çalışmak yerine Zybo Z7-20 SoC üzerinde Petalinux çalıştırmak ve Linux kernelinde bulunan V4L2 sürücülerini ile görüntüyü elde etmektir. Şekil 8'de bu yöntem ile kaydettiğimiz birkaç kare gösterilmiştir.



Şekil 8: Pcam 5C ile Kaydedilmiş Anahtar Fotoğrafı

Bu yöntem kullanılırken 60 FPS HD, VGA ve QVGA görüntüler sorunsuz bir şekilde SD kaydedilmiştir fakat görüntü alma işlemini durdurup FIFO'daki IMU verilerini SD karta yazdıktan sonra tekrar sürücüyü başlatmak ulaşılabilen kare hızını düşürmektedir. Bu sebeple ilk önermemiz olan görüntünün HDMI çıkışından iletilip IMU verisinin ise SD karta kaydedilmesi final tasarımıdır.

SONUÇ

Çalışmada kameradan farklı çözünürlüklerde ve farklı FPS değerlerinde görüntüler başarıyla alınmıştır. Aynı zamanda IMU sensöründen de veriler alınarak SD karta kaydedilmiştir. Elde edilen veriler ile görsel atalet odometri için kullanılabilir. SD kart yazma hızı yetmediği için görüntü verisi HDMI çıkışından verilmiştir. Alternatif olarak 60 FPS görüntüyü SD karta başarıyla yazdık fakat aynı anda IMU verisini de yazmaya çalıştığımızda kare hızımız düşmüştür. Karşılaşmış olduğumuz kare hızı sorunu bir sonraki çalışmanın konusu olabilir. Buna ek olarak ilk görüntü verisinin oluşumu ile ilk IMU sensör verisinin okunması arasında küçük bir zaman farkı vardır. Bu iki veri arasında sağlanacak senkronizasyon ile görsel atalet odometrisi daha doğru sonuçlar verecektir.

Kaynaklar

2020 SPS performance standard – GPS (2020, April 5). Retrieved from <https://www.gps.gov/technical/ps/2020-SPS-performance-standard.pdf>

Bayer Filter (2021, May 21). Retrieved from https://en.wikipedia.org/wiki/Bayer_filter

R. Jiang, R. Klette and S. Wang, "Modeling of Unbounded Long-Range Drift in Visual Odometry," 2010 Fourth Pacific-Rim Symposium on Image and Video Technology, 2010, pp. 121-126, doi: 10.1109/PSIVT.2010.27.

Sherif A. S. Mohamed, M. -H. Haghbayan, T. Westerlund, J. Heikkonen, H. Tenhunen and J. Plosila, "A Survey on Odometry for Autonomous Navigation Systems," in IEEE Access, vol. 7, pp. 97466-97486, 2019, doi: 10.1109/ACCESS.2019.2929133.

Visual-Inertial Odometry Using Synthetic Data (nd). Retrieved from <https://www.mathworks.com/help/driving/ug/visual-inertial-odometry-using-synthetic-data.html>