

İEE YTU CAS MARİNE ÇK

YÜKSEK SEVİYELİ YAZILIM ÖDEV 4

AD SOYAD: Tolga AVCI

ÜNİVERSİTE: Yıldız Teknik Üniversitesi

BİRİM: İEEE CAS Yüksek Seviyeli Yazılım

ÖDEV 4

SORU 1:

-UI(User Interface) bir kullanıcının herhangi bir uygulama, yazılım veya cihazla iletişim kurmasına olanak sağlayan temel bileşenlerin genel adıdır. Sadece görsellikten ibaret olmamak üzere tuş takımları, sesli komutlar, kod blokları, menüler, butonlar gibi hem görsel, hem de metinsel öğeleri kapsamaktadır. Amaç kullanıcı ile makine arasında anlaşılır ve hızlı bir iletişimi sağlamaktır.

-UI genel bir kavram olmak üzere kullanıcı ve sistem arasındaki tüm iletişim ve arayüzleri kapsar. GUI ise UI bir alt türü olarak karşımıza çıkar. GUI, grafiksel öğeler ile iletişimi sağlar ve kod satırları yazmayı gerektirmez. Örneğin Windows masaüstü, Android uygulamaları GUI örnekleridir. Kısaca UI genel bir kavramdır, GUI ise bu genel kavramın bir alt kümesidir.

-UI, alt türü olan CLI ile örnek olarak sistem yöneticiliği, yazılım geliştirme ve veri analizi gibi alanlarda kullanılmaktadır. Sesli arayüz örnekleri olarak akıllı ev sistemleri gibi örnekler, dokunmatik arayüz olarak telefon ve tabletler gibi örnekler gösterilebilir.

-UI veya GUI geliştirmede kullanılan bazı diller ve kütüphaneleri şunlardır:

Python dilinde:

Tkinter, Python'un yerleşik GUI kütüphanesi, basit uygulamalarda kullanılır.

PyQt5, profesyonel ve modern arayüzler yapmak için güçlü bir kütüphane.

C++:

Qt, hem masaüstü hem de gömülü sistemler için gelişmiş arayüzler tasarlamak içindir.

Java:

Swing, Java'nın eski ama hala kullanılan GUI kütüphanesi.

JavaFX, daha modern ve grafiksel açıdan güçlü bir GUI kütüphanesi.

-GUI Ar-Ge alanında önemli faydalar sağlar. Simülasyon ve görselleştirmeler sayesinde karmaşık verilerin anlaşılmasını sağlar. Yeni ürünlerin prototiplerinde GUI kullanılır. İnsan makine etkileşimi anlamada kritik rol oynar. Veri görselleştirme örnek MATLAB arayüzleri, Python'da Matplotlib ile GUI destekli araçlar.

SORU 2:

-Hough dönüşümü, görüntüde doğru, daire veya belirli şekilleri bulmak için kullanılan bir yöntemdir. Noktalardan şekil bulunduğundan görüntü gürültülü olsa bile çalışabilir.

-Temel prensipleri işe şöyle ilerler. Öncelikle şekilde kenarlar Canny vb. algoritmalarla tespit edilir. Görüntüdeki her kenar noktası, şekil denkleminin parametrelerine dönüştürülür.Örneğin bir doğru için denklem $x\cos\theta + y\sin\theta = \rho$ dir. x, y görüntüdeki kenar noktası, ρ doğrunun orijine uzaklığı, θ ise doğrunun eğim açısına karşılık gelir. Yani her nokta (x, y) tek bir doğruya değil, birçok (ρ, θ) çiftine karşılık gelir. Ondan sonra da oylama mantığı gerçekleşir. Bütün noktalar için hesaplanan (ρ, θ) değerleri “accumulator” tablosuna kaydedilir. Eğer birden fazla değer için bir doğrunun değerleri sağlanıyor ise orada bir doğru vardır. Farklı

şekiller için de aynı prensipler denklem şekle uygun hale getirelerek uygulanabilir.

SORU 3:

-Hough dönüşümü görüntü işleme görevlerinde kenar tespitinden sonraki adımdır. Kenar tespiti pikselleri dağınık bir şekilde bulur, Hough dönüşümü dağınık piksellerden geometrik şekil çıkarımı yapar. Nesne tespitinin en temel hali geometrik şekilleri algılatmaktır. Hough dönüşümü bu geometrik şekilleri tespit ederek karmaşık nesnelerin tespitinde temel atmaya yarar. Hough dönüşümünün bazı avantajları ve dezavantajları mevcuttur. Eksik ve gürültülü kenarlarla çalışabilir, basit bir mantığı vardır ve bu sebeple sonucu nettir kararlıdır. Bunlara karşıt olarak karmaşık nesneleri tespit edemez, hesaplama maliyeti yüksektir; özellikle elips vb. çok parametresi olan denklemlerde işlem yükü artar. Ayrıca parametre seçimine de duyarlıdır. Yanlış aralık seçiminde çok fazla yanlış ve doğru bulunamayan sonuçlarla karşılaşılması muhtemeldir.

SORU 4:

- Kontur tespiti, nesnelerin sınırlarını belirleyerek görüntüden anlamlı bilgiler çıkarılmasını sağlayan temel yöntemlerden biridir. Gerçek zamanlı görüntü işleme sistemlerinde kontur tespiti, nesne bölütleme, şekil tanıma, boyut ölçme, hareket takibi ve endüstriyel kalite kontrol gibi birçok uygulamada önemli bir rol oynamaktadır. Bu yönüyle kontur analizi, düşük maliyetli ve görece hızlı bir çözüm sunduğu için özellikle gömülü sistemler, robotik uygulamalar ve güvenlik sistemlerinde yaygın olarak kullanılabilmektedir.

- Gerçek zamanlı uygulamalarda her kare üzerinde kenar tespiti, eşikleme ve kontur çıkarma işlemlerinin yapılması yüksek hesaplama maliyeti doğurur. Yüksek çözünürlüklü görüntülerde işlem yoğunluğu arttığından, sınırlı işlem gücüne sahip cihazlarda kare hızında düşüş yaşanabilmektedir. Bu nedenle çözünürlüğün düşürülmesi, ROI kullanımı, paralel işlem teknikleri veya GPU hızlandırma yöntemleri performans sorunlarını azaltmak için tercih edilmektedir.

- Kontur tespitinin doğruluğu, görüntüdeki çevresel faktörlerden doğrudan etkilenmektedir. Aydınlatma değişimleri, gölgeler veya yansıma gibi koşullar konturların eksik ya da hatalı çıkmasına neden olabilir. Ayrıca gürültü ve hareket bulanıklığı çok sayıda sahte kontur üretebilir. Karmaşık arka planlarda nesne ile zemin arasındaki kontrastın düşük olması da sınırların doğru şekilde yakalanmasını zorlaştırmaktadır. Bunun yanında kenar tespiti ve thresholding adımlarında kullanılan parametrelerin hassas olması, küçük değişikliklerde bile sonuçların kararsız hale gelmesine yol açabilmektedir.

- Genel olarak kontur tespiti, gerçek zamanlı görüntü işleme sistemlerinde uygulanabilir ve birçok senaryoda başarılı sonuçlar verebilen bir yöntemdir. Ancak performans ve doğruluk açısından yaşanan zorluklar göz önünde bulundurulmalı bulanıklaştırma, morfolojik işlemler ve donanım optimizasyonlarıyla desteklenmelidir. Daha karmaşık nesne tanıma ve takip problemlerinde ise kontur tabanlı yaklaşımlar, derin öğrenme yöntemleriyle birlikte hibrit şekilde kullanıldığında daha güvenilir ve ölçeklenebilir çözümler sunabilmektedir.

SORU 5:

-Bir nesnenin konturu çıkarıldıktan sonra, bu kontur üzerinde çeşitli matematiksel ölçümler yapılarak nesnelerin boyutları, şekilleri, konumları, yönelimleri hakkında bilgi edinilebilir. Kontur üzerinden elde edilebilecek geometrik özellikler ise şunlardır:

-Alan, kontur içindeki piksel sayısını verir. `cv.contourArea(cnt)`

-Çevre, nesnenin sınır uzunluğunu ölçer. `cv.arcLength(cnt, True)`

-Sınır dikdörtgeni (Bounding Box), konturu kapsayan en küçük dikdörtgen. `Cv.boundingRect(cnt)`

-Çevreleyen daire (Minimum Enclosing Circle), konturu tamamen içine alan en küçük daireyi bulur. `cv.minEnclosingCircle(cnt)`

-Dışbükey kapsama alanı (Convex Hull), konturu çevreleyen en küçük dışbükey şekli bulur.

-Momentler, konturun ağırlık merkezi, eğim ve dağılım gibi özelliklerini verir. `cv.moments(cnt)`

-Bu geometrik özellikler konturlar üzerinden elde edebildiklerimizden bazılarıdır.

SORU 6:

-Convex hull , bir nesneyi çevreleyen en küçük dışbükey şeklin elde edilmesini sağlayan geometrik bir yöntemdir. Görüntü işleme ve bilgisayarla görme alanlarında kontur analizinin önemli bir parçası olarak kullanılan convex hull, nesnelerin dış sınırlarının düzenlenmesi ve şekil özelliklerinin çıkarılması için yaygın biçimde uygulanmaktadır.

-Uygulama örnekleri arasında en bilinenlerden biri el ve parmak tespiti. El konturunun convex hull'u çıkarılarak, hull ile kontur arasındaki boşluklar incelenir ve bu sayede parmak sayısı belirlenebilir. Bu yöntem jest tanıma ve insan-bilgisayar etkileşiminde sıkça kullanılmaktadır. Bir diğer uygulama alanı ise endüstriyel kalite kontroldür. Parçaların konturları convex hull ile karşılaştırılarak yüzeyde kırık veya bozukluk olup olmadığı tespit edilebilir. Ayrıca çarpışma algılama problemlerinde nesneler convex hull ile basitleştirilmiş şekilde temsil edilerek daha hızlı ve güvenilir çarpışma kontrolü yapılabilmektedir.

Convex hull, optik karakter tanıma (OCR) süreçlerinde de metin bloklarının düzgünleştirilmesinde kullanılmakta, böylece yamuk veya dağınık yazıların sınırları daha net hale getirilmektedir. Bunun yanı sıra nesne takibi uygulamalarında hull kullanılarak nesnenin hareketi sırasında kaybolan ya da bozulan konturların yerine daha güvenilir bir kapsama alanı elde edilir. Biyomedikal görüntülemelerde ise hücre kümeleri ve mikroskop altında incelenen yapıların dış sınırları convex hull ile belirlenerek daha doğru sayım ve analiz yapılabilmektedir.

Sonuç olarak convex hull; el-parmak tespiti, kalite kontrol, çarpışma algılama, OCR, nesne takibi ve biyomedikal analiz gibi pek çok alanda farklı problemleri çözmekte kullanılmaktadır. Bu yönüyle hem basitleştirilmiş nesne temsili sağlamakta hem de kontur tabanlı analizlerde doğruluk ve güvenilirliği artırmaktadır.

SORU 7:

- Kontur momentleri, bir nesnenin geometrik ve mekansal özelliklerini sayısal olarak ifade etmeye yarayan ölçümlerdir. Momentler sayesinde bir konturun alanı, ağırlık merkezi, yönelimi

ve simetrisi gibi bilgiler elde edilebilir. Bu nedenle kontur momenti, şekil analizinde nesnenin daha soyut ve matematiksel bir biçimde temsil edilmesini sağlar.

- Bir konturun ağırlık merkezi, momentlerden türetilerek hesaplanır. Bu merkez, nesnenin şekli nasıl olursa olsun, alanın kütle dağılımı eşit kabul edilerek hesaplanan ortalama konumdur. Formül ayrı ayrı x ve y koordinatları için ayrı ayrı tüm x ve y koordinatlarının toplanıp toplam alana bölünmesiyle elde edilir. OpenCV’de bu hesaplama `cv.moments(cnt)` fonksiyonu ile yapılabilir.

-Merkezin nesne takibindeki kullanımında, merkez her bir nesnenin karedeki konum noktası olarak kullanılır. Merkez hesaplanıp kaydedildiğinde, nesnenin hareket yolu(trajectory) kolayca çıkarılabilir. Örneğin bir topun sahadaki gidiş yönü, fabrikada bantta ilerleyen ürünlerin takibi yapılabilir. Burada merkez tüm nesneyi tespit etmektense bir nokta üzerinden takip edilmesiyle hızlı ve hafif bir çözüm sağlar.

- - Merkez tek başına sınıflandırma için yeterli değildir ama momentlerle birlikte güçlü bir araçtır. Nesnelerin merkezinin yeri, kontur etrafındaki dağılımı, momentlerden türetilen özellikler farklı şekilleri birbirinden ayırmada kullanılır. Örneğin bir dairenin ağırlık merkezi daima geometrik ortadadır, ama üçgende merkez farklı dağılır. Normalleştirilmiş momentler sayesinde nesne döndürülse de tanınabilir. OCR sistemlerinde harflerin farklı açılardan görüntülerinin doğru sınıflandırılması bu yöntemle yapılır.

KAYNAKÇA:

<https://heycoach.in/blog/convex-hull-applications>

https://docs.opencv.org/3.4/d9/db0/tutorial_hough_lines.html

https://www.youtube.com/watch?v=K5UBpC_BiXk

https://tr.wikipedia.org/wiki/Hough_dönüşümü

<https://chatgpt.com>