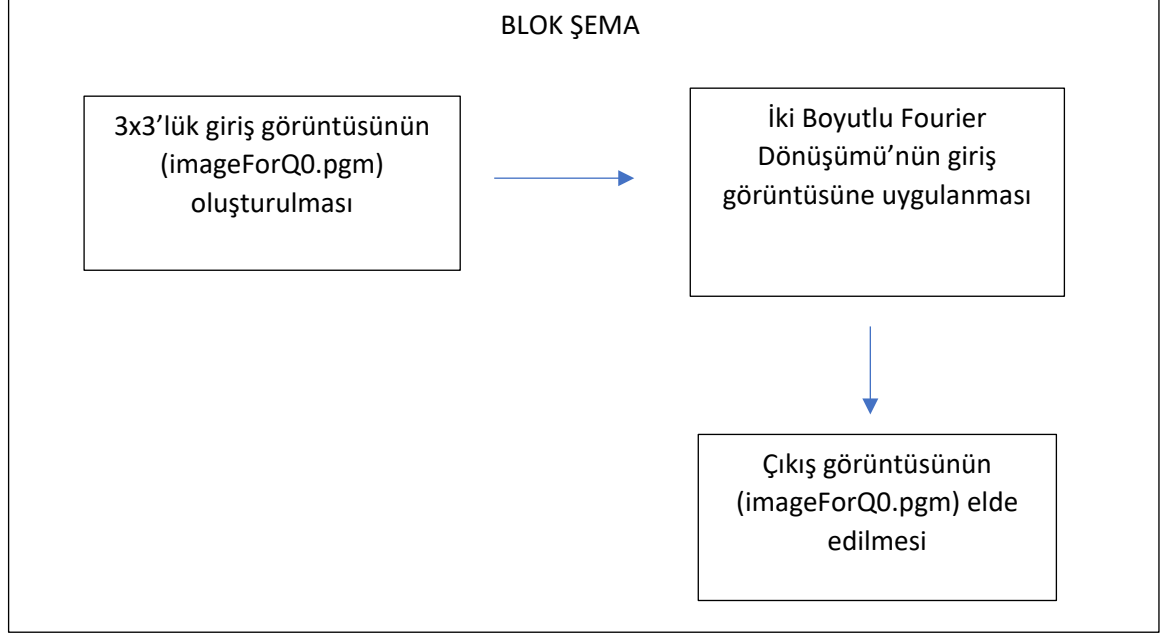


LAB3 RAPORU

1. Soru 0 :

Bu kısımda yapılan işlemler ve ilgili işlemler sonucunda elde edilen çıktıların verildiği blok şema Şekil 1’de görüldüğü gibidir.



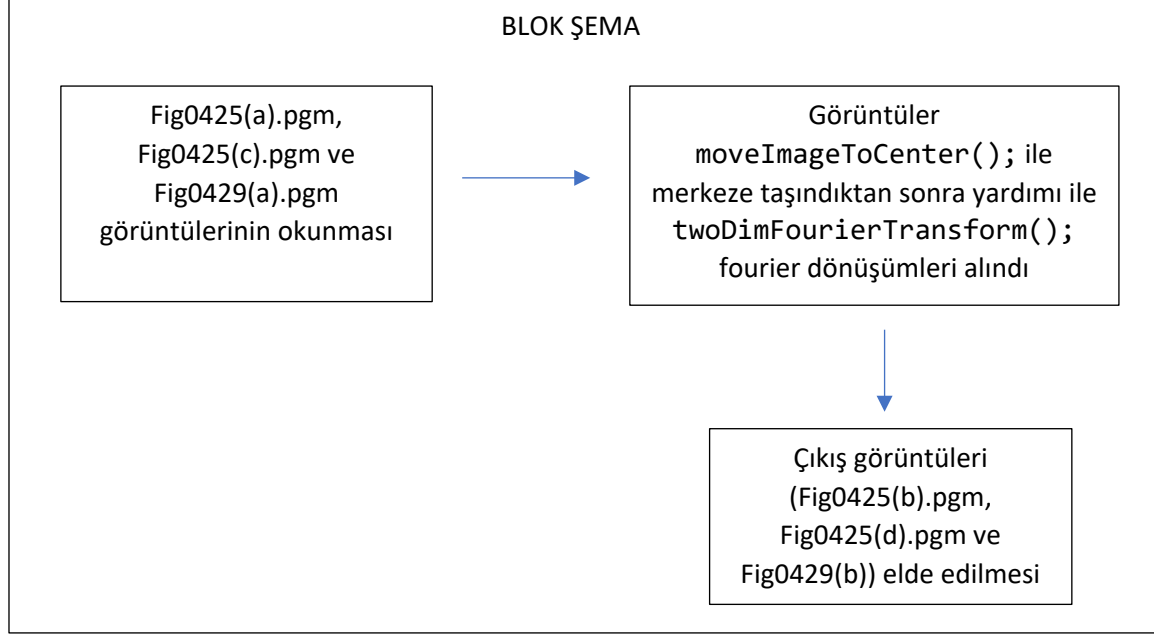
Şekil 1. Blok Şema

İlk olarak, belirtilen 3x3'lük görüntünün oluşturulması işlemi yapılmıştır. Bunun için `createImageforlab3q0()`; fonksiyonu kullanılmıştır. Daha sonra, elde edilen giriş görüntüsüne `twoDimFourierTransform()`; fonksiyonu ile İki Boyutlu Fourier Dönüşümü uygulanmıştır.

Oluşturulan görüntü incelendiğinde beklendiği gibi en sol üstten başlayıp en sağ alta kadar görüntünün koyulaştığı görülebilir. (Bunun için .exe çalıştırıldığında oluşan `imageForQ0.pgm` görüntüsüne zoom yapılmalıdır.) İki Boyutlu Fourier Dönüşümü uygulanmış görüntü ise .exe çalıştırıldığında oluşan `imageForQ0_out.pgm` görüntüsüdür. (Aynı şekilde görebilmek için oluşan görüntüye zoom yapılmalıdır.) Bu sayede görüntünün frekans uzayında da incelenme imkanı elde edilmiş olur. 2D Fourier Dönüşümü, ilk olarak satırlara sonra da sütunlara Fourier Dönüşümü'nü ayrı ayrı uygulayarak görüntüyü frekans uzayında incelemeyi sağlar. Ayrı ayrı dönüşümler alındıktan sonra elde edilen gerçek ve sanal kısımların kareleri alınıp toplanarak elde edilen sonucun karekökü alındığında bu sonuç bir görüntüye atanır. Son adım ise görüntünün uygun boyutlarda çalışması için ölçekleme işleminin yapılmasıdır. 8 bitlik görüntüye uygun ölçekleme yapıldıktan sonra görüntünün son hali elde edilir. Kod da bu mantığa uygun olarak yazılmıştır. Sonuçlar temel anlamda beklenildiği gibi elde edildi. Dönüşüm yapıldığında görüntünün zaman uzayındaki halinden çok daha farklı bir formatta ortaya çıktığı görüldü. Bu durumun gerekli anlarda görüntüyü incelemeyi ve analiz etmeyi kolaylaştırdığı düşünülebilir.

2. Soru 1 :

Bu kısımda yapılan işlemler ve ilgili işlemler sonucunda elde edilen çıktıların verildiği blok şema Şekil 2’de görüldüğü gibidir.



Şekil 2. Blok Şema

Bu soruda yapılan ilk işlem, `moveImageToCenter()`; ile köşelerdeki görüntü bileşenlerini merkeze taşıma fonksiyonunun ilgili görüntüye uygulanmasıdır. Bu fonksiyonun uygulanma sebebi, DC'ye merkeze alabilmek ve görüntüyü daha doğru yorumlamayı sağlamaktır.

Görüntünün DC bileşeni merkeze alındıktan sonra, `twoDimFourierTransform()`; fonksiyonu ile görüntüye 2D Fourier Dönüşümü uygulanabilir. Bir önceki adımda detaylı bir şekilde açıklandığı üzere, ilk olarak satırlara sonra sütunlara uygulanan dönüşüm ile görüntü frekans uzayında incelenebilir hale gelecektir.

Görüntünün FD'si alındıktan sonra karşılaşılan stackdump sorunun çözümü, oldukça uğraştırıcı ve zaman alıcı bir sorun haline gelmiştir. Sorunun çözümü için birçok farklı yöntem denenmiştir. (Görüntülerin array olarak tutulması, `createImage` ile direkt olarak oluşturulması vs.) En sonunda, `createImage` fonksiyonu ile oluşturulan görüntüye, Fourier Dönüşümü alımının sonunda bir ölçekleme uygulanarak istenilen sonuca bir adım daha yaklaşılabildiği görülmüştür.

Net görüntüye ulaşmak ve görüntüyü daha açık şekilde inceleyebilmek için görüntüye ilk olarak logaritma transformu, sonrasında parlaklığını artırabilmek için bir gama transformu uygulanmıştır. Elde edilen sonuçlar incelendiğinde aslında istenilen görüntünün tam tersi, yani negatifine ulaşıldığı anlaşılmıştır. Bu yüzden son aşamada görüntünün negatif transformu ile negatifi alınmıştır ve nihayet beklenen sonuca ulaşılmıştır. Bir görüntü için Fourier Dönüşümü almanın C++ dilinde zahmetli, yorucu ve birden çok fonksiyonu bir arada kullanmayı gerektirebilecek bir işlem olduğu yorumu da ayrıca yapılabilir.

Referanslar:

1. Ders notları
2. Digital Image Processing by Rafael C. Gonzalez, Richard E. Woods